A weakly-2-generic which bounds a minimal degree

Rodney G. Downey Satyadev Nandakumar

December 27, 2015

Abstract

Jockusch showed that 2-generic degrees are downward dense below a 2-generic degree, but in the case of 1-generic degrees Kumabe, and independently Chong and Downey constructed a minimal degree computable from a 1-generic degree. We explore the tightness of these results.

We solve a question of Barmpalias and Lewis-Pye by constructing a minimal degree computable from a weakly 2-generic one. The proof is rather novel since it is a computable full approximation construction where both the generic and the minimal degrees are $\Delta_3^0 - \Delta_2^0$.

1 Introduction

Two of the fundamental construction techniques in set theory and computability theory are Cohen and Sacks/Spector forcing. The first uses finite strings as conditions and the second perfect trees. Computability theory allows us to look at fine grained restricted versions of these notions. Cohen forcing gives us various forms of genericity and Sacks/Spector allows for various forms of minimality and computable domination.

This paper follows a tradition asking "How can these two notions interact?". In their unrestricted forms the notions are incompatible, but in their restricted forms sometimes that can interact via Turing reducibility.

The reader should recall the following definitions (which are really theorems due to Jockusch and Posner, but have become standard in the literature as definitions.).

Definition 1. Let $n \geq 1$.

1. A set A is called *n-generic* iff A meets or avoids all Σ^0_n sets of strings. That is, if S is a Σ^0_n set of strings, then either $\exists\sigma \in S(\sigma \prec A)$ (σ is an initial segment of A) or $\exists\tau \prec A \forall\sigma \in S(\tau \not\preceq \sigma)$.

2. A set B is called *dense* if for all $\nu \in 2^{<\omega}$, there is a $\rho \in B$ such that $\nu \preceq \rho$. We say that a set C is *weakly n-generic* iff for all dense Σ^0_n sets of strings S, C meets S.

3. We say a degree $\mathbf{a}$ is (weakly) n-generic is it contains a (weakly) n-generic set.

The natural relationship is that weak $n+1$-genericity is implied by $n+1$-genericity and implies n-genericity, and these implications cannot be reversed. (For example, Kurtz [12].) Jockusch [10] was the first person to give a detailed analysis of notions of (weak) n-genericity and their relationship with Turing reducibility. In particular, showed that if $\mathbf{a}$ is a nonzero degree below a 2-generic degree, then $\mathbf{a}$ bounds a 2-generic degree. As a consequence, no 2-generic degree can bound a minimal degree.

This result was extended by Chong and Jockusch [5] who proved that if $\mathbf{g}$ is 1-generic and $\mathbf{0} < \mathbf{a} < \mathbf{g} < \mathbf{0}'$ then $\mathbf{a}$ bounds a 1-generic degree. Later Haught [9] extended this result to prove the very attractive result that if $\mathbf{g}$ is 1-generic and $\mathbf{0} < \mathbf{a} < \mathbf{g} < \mathbf{0}'$ then in fact $\mathbf{a}$ is 1-generic.

At the time, it seemed reasonable to conjecture that the restriction that $\mathbf{g} < \mathbf{0}'$ could be removed. Independently, Kumabe [11] and Chong and Downey [4] proved that this restriction cannot be removed, both papers constructing a 1-generic degree $\mathbf{g} < \mathbf{0}''$ bounding a minimal degree $\mathbf{m} < \mathbf{0}'$. Indeed, Chong and Downey [4] gave a local iff condition (now called "having no tight cover") which characterized when a set B could be computed from a 1-generic set. In [3], they used this to construct a minimal degree below $\mathbf{0}'$ *not* computable from a 1-generic, and Downey and Hirschfeldt [8] (page 387) also used this characterization to show that almost every set is not computable from a 1-generic, although this was known earlier by the work of Kurtz [12]. Finally, Downey and Yu [6] used this characterization to construct a hyperimmune-free (minimal) degree computable from a 1-generic, this being of interest since the construction of a hyperimmune-free degree is a much "purer" form of perfect set forcing than is the construction of a minimal degree which can use various approximation techniques.

Thus, we know no 2-generic degree can bound a minimal degree, and a 1-generic degree can bound a minimal degree. In this paper, we give an affirmative answer the natural question of Barmpalias and Lewis-Pye [2] (see also [1]) who asked whether a weakly 2-generic degree can bound a minimal degree.

Theorem 2. *There exist $M <_T G <_T \emptyset''$ with M of minimal Turing degree and G weakly 2-generic.*

On general grounds, we point out that this theorem is unlikely to be proven by forcing, and hence some kind of limit/approximation construction will be needed. Moreover, as we first prove, if G is weakly 2-generic then the degree of G forms a minimal pair with $\mathbf{0}'$ (something that might have been already known, but we could not find in the literature). Thus we will need a computable construction to construct both G and M, *neither* of which is Δ^0_2 and hence at no stage will initial segments come to limits. Full approximation constructions of Δ^0_3 sets have occurred in the literature such as Downey [7], but they are rare and complex. Moreover, no full approximation construction of a weakly 2-generic has previously occurred. Thus the proof here is also of some technical interest as it involves techniques which may have wider applications.

The proof consists of two interacting full approximation arguments one of a weakly 2-generic and the other of a minimal degree, where the interactions are controlled by a priority tree of strategies.

2 Notation

The set of finite binary strings is denoted by $2^{<\omega}$ and the set of infinite binary sequences by 2^ω. If σ is a finite string, then $[\sigma]$ denotes the *cylinder* σ, *i.e.* the set of infinite binary sequences with prefix σ. If S is a set of finite strings, then $[S]$ is the set of all infinite sequences with some prefix in S. We say that $\sigma \preceq \tau$ if the finite string σ is a prefix of the finite string or infinite sequence τ. We also use the relation $<_L$ to denote the lexicographic ordering of strings.

3 Minimal Pair

In this section we prove the following easy result.

Theorem 3. *Suppose that $X \leq_T G, \emptyset'$ and G is weakly 2-generic. Then X is computable.*

Proof. Suppose that $\Phi^G = X$ with $X \leq_T \emptyset'$, $X = \lim_s X_s$, and G weakly 2-generic.

Let $S = \{\sigma \mid [\exists s_0 \forall s > s_0 (\Phi^\sigma \downarrow [s] \not\prec X_s) \vee (\forall \tau \forall s)(\sigma \preceq \tau \rightarrow \Phi^\tau \uparrow [s])]\}$.

If S is dense then G meets S which is a contradiction. Thus S is not dense.

Therefore there is some σ_0 such that for all $\sigma \in S$, $\sigma_0 \not\preceq \sigma$.

Then for all σ extending σ_0 there is some τ, $\sigma \preceq \tau$ and $\Phi^\tau \downarrow$. But also for such a τ, $\Phi^\tau \prec X$, so that X is computable. $\square$

4 The Proof of Theorem 2

We build a weakly 2-generic G and a set M of minimal degree and a procedure Γ with $\Gamma^G = M$. The reader should think of Γ as a partial computable function from strings to strings with the usual continuity conditions for a Turing procedure. Namely, if $\nu \preceq \tau$ and $\Gamma^\nu \downarrow, \Gamma^\tau \downarrow$, then $\Gamma^\nu \preceq \Gamma^\tau$. As remarked earlier Theorem 3 imposes some restrictions on the constructions of both G and M. While the initial segments of both G and M do not come to limits in the construction, we will be able to read them off the true path of the construction and the construction will ensure that there are arbitrarily long initial segments $\rho \prec G, \sigma \prec M$ with $\Gamma^\rho \downarrow = \sigma$.

It is most convenient to build M in Cantor Space and G in Baire space. We will think of G as being the "left" construction and M the "right" construction with Γ the partial computable mapping of strings in the left construction to strings in the right construction.

As usual, Φ_e denoted the e-th Turing procedure, and we will let $S_0, S_1, \ldots$ be a standard enumeration of the Σ^0_2 sets of strings in Baire space. For example, if Q_i denotes the i-th partial computable binary relation, we can let $\sigma \in S_i$ iff $\exists s \forall t Q_i(\sigma, s, t)$. As is well known, we can choose Q_i here to be the i-th primitive recursive 3-place relation, so not worry about halting considerations.

Hat convention It is most convenient to use certain conventions about the approximation to S_i. We will adopt a kind of "hat" convention. That is, if σ appears in S_i at stage s, with witness s_0, meaning that

- $Q_i(\sigma, s_0, t)$ holds for all $t \leq s$.
- s_0 is least with this property.

Then if $Q_i(\sigma, s_0, s+1)$ fails to hold, we will regard σ to *not* appear to be in S_i at stage $s+1$, even if there is some s_1 with $Q_i(\sigma, s_1, t)$ for all $t \leq s+1$.

Conventions When we write $\tau \in S_{i,s}$ we mean that τ appears to be in $S_{i,s}$ in the sense above. Moreover, if τ app rears to be in $S_{i,s}$ with witness s_0, then we will ask that $s_0 > |\tau|$. So long strings must have large witnesses. This last convention helps when it comes to choosing strings appearing to be in $S_{i,s}$ during the priority construction.

These conventions is more or less standard.

The requirements we must meet are the following.

$$\begin{array}{lll} \mathcal{R}_e : S_e \text{ dense} & \Rightarrow G \text{ meets } S_e & \text{[Weak-2-Genericity]} \\ \mathcal{N}_e : \Phi_e^M \text{ total} & \Rightarrow (\Phi_e^M \equiv_T \emptyset) \ \vee (M \leq_T \Phi_e^M) & \text{[minimality]} \end{array}$$

Additionally, we will need to make M noncomputable. This could be added as an explicit feature of the construction, but in fact, noncomputability of M will be a consequence of the construction method and the Recursion Theorem, in a way we will later discuss.

We will discuss the meeting of the requirements in isolation and then later analyze the interactions of the requirements. We begin with R_e.

5 Weakly-2 generic construction - Basic module for $\mathcal{R}_e$

Now, in isolation the idea is the following. We will assume $\mathcal{R}_e$ has at its disposal an initial segment $\rho(e, s)$ of G. Of course. in the real construction, there will be several versions of such ρ which depend upon what seems correct at the current stage. However, for the present discussion, we assume that is a true initial segment of G, and moreover $\Gamma^{\rho(e,s)} \downarrow [s]$.

Now, the idea is to set aside $\rho(e,s)^\frown 1$ as the part of $\omega^{<\omega}$ where we try to meet $\mathcal{R}_e$, should S_e be dense, and $\rho(e,s)^\frown 0$ is where we will meet $\mathcal{R}_e$ if we are in the lucky case that S_e is not dense.

It will be convenient in the construction to also make sure that $\Gamma^{\rho(e,s)^\frown 0}\downarrow [s]$ and $\Gamma^{\rho(e,s)^\frown 1}\downarrow [s]$ are incompatible extensions of $\Gamma^{\rho(e,s)}$. As we see, this will necessitate certain complexities in the construction, but will be discussed later.

The strategy is pretty simple. If we see some $\tau(e,s) \succ \rho(e,s)^\frown 1$ and $\tau(e,s) \in S_{e,s}$, then we would like to route $G_{s+1} \succ \tau(e,s)$. Should it be the case that $\tau(e,s) \in S_{e,t}$ for all $t \geq s$, we will be done as now G meets S_e. This is outcome f on the priority tree.

While we are waiting for such a $\tau(e,s)$ to occur, we route G_t through $\rho(e,s)^\frown 0$. That is, until we see such a τ, we have $\rho(e,s)^\frown 0 \prec G_t$. We regard this as outcome ∞.

Now should we think we have found $\tau(e,s)$ and the $\tau(e,s) \notin S_{e,t}$ at $t \geq s+1$, our action would be to re-route G_t through $\rho(e,s)^\frown 0$ again. At stage $t+1$ we would again seek a $\tau(e,t) \in S_{e,t}$ extending $\rho(e,s)^\frown 1$.

Consider a stage $u \geq t+1$. Now the question is "*Which* $\tau(e,u)$ to pick?", since there could be many possible choices of strings appearing in $S_{e,u}$. As with most Σ_2 arguments, we would pick the $\tau(e,u)$ which has been there the longest time. That is, if we think $\tau_i \in S_{e,u}$ with witnesses s_i for $i \in \{1,2\}$, then choose the one with the least s_i, and then if both have the same s_i, choose the lexicographically least one[1].

Note that if S_e is really dense, eventually we would find $\tau = \lim_s \tau(e,s)$ to get stuck on extending $\rho(e,s)^\frown 1$. This is the Σ^0_2 outcome f. If no such τ is found, then we would either switch to $\rho^\frown 0$ infinitely often (outcome ∞, the Π^0_2 outcome) or get stuck from some point on, outcome w. On the priority tree, we have $\infty <_L f <_L w$.

Of course, as mentioned earlier, the above is a simplification for the *Basic Module*, as there will be several versions of ρ on the guesses as per the behaviour of higher priority requirements, but the reader should keep this in mind.

Note also, in the background, we will also be mapping $\Gamma^G_s \to M_s$ in con-

[1]The reader here should pay attention to the second convention concerning S_e, in that long strings cannot have small witnesses.

junction with the above. For example, in the basic module, we would naturally map $\Gamma^{\tau(e,s)} = \Gamma^{\rho(e,s)\frown 1}$ and potentially Γ maps extensions of $\rho(e,s)\frown 0$ to $\Gamma^{\rho(e,s)\frown 0}$. Plainly there are problems with this since we need to make Γ total. Problems are revealed when we consider the strategy in combination with others.

Remark 4. Moreover, as we discuss later, we cannot allow $\mathcal{R}_e$ to move us off a higher priority "state" for M. The point is, if at some stage we define $\Gamma^{\eta} = \sigma$ and we see some $\tau \in S_{e,s}$ with $\tau \preceq \eta$, then we would be forced to make $M_{s+1} \preceq \sigma$ if $G_{s+1} \preceq \tau$. This will generate the key tension in the construction.

6 Minimal degree construction: Basic Module for $\mathcal{N}_e$

The standard minimal degree construction using e-splitting trees and full trees is well-known to computability theorists. Less well known is the full approximation construction, and this is particularly true in the setting where $M \not\leq_T \emptyset'$. Thus we will take the liberty of describing in detail how this will work.

The reader should recall that a function $T : 2^{<\omega} \to 2^{<\omega}$ is called a *(function) tree* if for every finite binary string σ, $T(\sigma 0)$ and $T(\sigma 1)$ are incompatible extensions of $T(\sigma)$. A string σ is said to be *on* T if it is an element of the range of T. The set of paths in T are denoted by $[T]$. A set M is said to be *on* T if infinitely many prefixes of M are on T. Recall the following standard definition.

Definition 5. A string σ on a function tree T is said to *e-split* if there are incompatible extensions τ and ρ of σ, and an input n such that $\Phi_e^{\tau}(n)\downarrow \neq \Phi_e^{\rho}(n)\downarrow$. A string σ on T is said to be *non-e-splittable* if for every pair of extensions τ, ρ of σ and every $n \in \mathbb{N}$, if both $\Phi_e^{\tau}(n)\downarrow$ and $\Phi_e^{\rho}(n)\downarrow$, then $\Phi_e^{\tau}(n) = \Phi_e^{\rho}(n)$.

A set M is said to be *e-splittable* on T if every prefix of M on the tree T, is e-splittable. M is said to be *non-e-splittable* on T if M has a non-e-splittable prefix on T.

The notion of e-splitting trees is useful for the construction of sets of minimal degrees because of the following fundamental property.

Lemma 6 (Essentially, Spector [13]). *Let T be e-splitting and $M \in [T]$. If Φ_e^M is total, then $M \leq_T \Phi_e^M$.*

The basic module for $\mathcal{N}_0$ is to build a tree $T_{0,s}$ as follows. For any stage s, we set $T_{-1,s} = 2^{<\omega}$. At stage s, M_s will be a length s path on $T_{0,s}$. Initially, $T_{0,0} = T_{-1,0}$, so that $T_{0,0}(\nu) = \nu$. At each stage s, we will associate with each node ν a 0-*state* which is one of ∞ or f. This will indicate whether we think that $T_{0,s}(\nu)$ 0-splits or not.

This is done in a somewhat obvious way. We will begin with $\nu = \lambda$, the empty string. Now initially we have no computations. We give $T_{0,s}(\lambda)$ the 0-state f. As the construction proceeds, we monitor $\Phi^\nu(n)$ for all n such that for all $m < n$, for all $\nu \in T_{0,s}$ of length $\leq s$. At the first stage s, if any we see $\Phi^{M_s}(n) \downarrow [s]$ we would like to issue a description of $\Phi_0^M(n)$, and argue that this is correct, so that Φ_0^M is computable.

The only time we would be wrong would be that we saw some n where $\Phi_0^{\nu_0}(n)$ and $\Phi_0^{\nu_2}(n)$ 0-split for some ν_0, ν_1 on $T_{0,s}$[2]. If at some stage we observe this, then, supposing wlog $\nu_0 <_L \nu_1$, we would raise the 0-state of $\lambda = T_{0,s}(\lambda)$ to ∞, refining the tree $T_{0,s+1}$ so that we define, for all $\xi \in 2^{<\omega}$, $T_{0,s+1}(1\widehat{\ }\xi) = T_{0,s}(\eta_0\widehat{\ }\xi)$ where $T_{0,s}(\eta_0)$ is the use of $\Phi_0^{\nu_0}(n)$ on $T_{0,s}$, and $T_{0,s+1}(0\widehat{\ }\xi) = T_{0,s}(\eta_1\widehat{\ }\xi)$ $T_{0,s}(\eta_1)$ is the use of $\Phi_0^{\nu_1}(n)$.

Clearly, the strategies for $T_{0,s+1}$ will now inductively try to raise the 0-state of the nodes $T_{0,s+1}(0)$ and $T_{0,s+1}(1)$ independently, refining the tree when splits are found, as above, and working on each of the successors of these nodes if splits are found.

In the actual construction, the $\mathcal{N}_0$ strategy asks that M be on the inductive tree with the high 0-state, ∞, if possible. In the construction above, when we raise the state of λ we would ask that $T_{0,s}(\eta_i) \prec M_{s+1}$ for some $i \in \{0, 1\}$.

As far as the Basic Module is concerned, this will mean that for each s, M_s is a length s path on $T_{0,s}$ in the sense that at each stage s we will have a shortest σ where $T_{0,s}(\sigma) \prec M_s$ and $T_{0,s}(\sigma)$ has 0-state f.

More generally, at each stage s, we now build a sequence of total computable function trees with the following property : for any stage s and any e, we have a total computable tree $T_{e,s}$ which represents the s-stage approx-

[2] And, in particular, $\Phi_0^{M_{s_1}}(n)$ and $\Phi_0^{M_{s_2}}(n)$ 0-split $\Phi_0^{\nu_0}(n)$ and $\Phi_0^{\nu_1}(n)$ 0-split, after we issued a description of $\Phi_0^{M_{s_1}}$, say.

imation to a tree T_e. Further, we will ensure that (paths in) the trees form a nested sequence as follows.

$$[T_{-1,s}] \supseteq [T_{0,s}] \supseteq \cdots \supseteq [T_{s,s}].$$

For any index e, we will consider the following tree constructed in the limit.

$$T_e = \lim_{s\to\infty} T_{e,s},$$

where the limit is defined pointwise — *i.e.*, for every string σ, $T_e(\sigma) = \lim_{s\to\infty} T_{e,s}(\sigma)$. This has the consequence that the limit tree T_e may not be computable. [3]

At each stage we will associate with nodes $T_{e,s}(\sigma)$ a certain e-state which will be a string consisting of symbols from $\{\infty, f\}$ of length $e+1$. These are changed as above according to observations about whether $T_{e,s}(\sigma)$ e-splits on $T_{e,s}$ (i.e. the splitting nodes must be on $T_{e,s}$). We then raise e-states by replacing the last symbol f by ∞ if splits are observed and refining the tree $T_{e,s}$.

On the priority tree, if a node ν represents $\mathcal{N}_e$ and we see a new split at a stage where ν looks correct, we would say that the stage is a $\nu\widehat{\ }\infty$ stage, else a $\nu\widehat{\ }f$ stage.

The reader should note that in the construction, we might see that $T_{e,s}(\sigma)$ raises its state to $\alpha\widehat{\ }\infty$ as we see a split, but later it might be that this split is removed from the tree $T_{e,t}$ $(t > s)$. If this happens then it will be the case that the state increases to $\alpha'\widehat{\ }f$ for some $\alpha' <_L \alpha$ where $\infty <_L f$, meaning that some tree $T_{e',t}$ becomes refined.

We can visualize this using the notion of "boundaries" on the various trees. [4] On tree $T_{0,s}$, there is a boundary below which every string σ is 0-splittable, and above which $T_{0,s}$ is the full tree. For the tree $T_{1,s}$, there are four boundaries. The nodes below the bottom-most boundary consists of nodes which have 1-splits above 0-splits. Above that, is a layer of nodes which have 0-splits, but no 1-splits. The third layer from the bottom consists of nodes which have no 0-splits but have 1-splits. The topmost layer consists of nodes which are neither 0-splittable nor 1-splittable.

As with any of these constructions, the details are very messy but the idea is straightforward.

[3] However, we will argue that T_e will contain a *partial computable* function tree T_e^* satisfying $\mathcal{N}_e$.

[4] Note: Here we refer to the nodes in the domain of the trees.

Remark 7. The above is not quite correct when the inductive strategies are considered, in the situation that we have a requirement $\mathcal{N}_e$ of *lower* priority than a $\mathcal{R}_j$, and the latter might force certain nodes to remain on $T_{e-1,s}$ for the sake of keeping a witness $\rho(j,s)$ (for instance) on the left tree because $\Gamma^{\rho(j,s)}$ has an image in $T_{j,s}$ and hence $T_{e-1,s}$ which cannot be removed with priority j. This feature will be implemented by the "indicator nodes" where we are allowed to raise e states on $T_{e-1,s}$. (The point is that we make e-states a finite string, and only initially raise e states on T_{e-1} for nodes $T_{e-1,s}(\sigma)$ with $|\sigma| > e$. Higher priority strategies might lengthen the places we are allowed to raise e-states. More on this later.

7 α-module and the inductive strategies

We now examine the interactions of the various requirements amongst themselves and describe the inductive strategy in the construction.

First, consider how a single $\mathcal{R}_e$ requirement copes with a single $\mathcal{N}_j$ of higher priority. We begin by looking at $\mathcal{N}_0$ being of highest overall priority and consider $\mathcal{R}_0$.

The driver for $\mathcal{N}_0$ is to build M in a high 0-state tree. It is natural for $\mathcal{R}_0$ to guess the eventual state of $\mathcal{N}_0$. Initially, $\mathcal{R}_0$ must guess state f, and $\mathcal{R}_0$ would have erected a genericity location ρ_f with two extensions $\rho_f\widehat{\ }0$ and $\rho_f\widehat{\ }1$.

These would have been mapped by Γ two incompatible strings in $T_{0,s}$ as the construction has proceeded. That is, $\Gamma(\rho_f) = \sigma$, (actually, here $\sigma = \lambda$ as this is the first split) $\Gamma(\rho_f\widehat{\ }i) = \sigma_i$ with $\sigma_1|\sigma_0$.

Now, what the version of $\mathcal{R}_0$ guessing ∞ is waiting for is to see some 0-split in $T_{0,s}$ before defining Γ. If we see an n where $\Phi_0^{\nu_0}(n)$ and $\Phi_0^{\nu_1}(n)$ 0-split for some ν_1, ν_2 o n $T_{0,s}$, as in the discussion of the basic $\mathcal{N}_0$ module, we would now be free to define $\Gamma(\rho_\infty i) = \nu_i$, where ρ_∞ is a new node chosen in $\omega^{<\omega}$. ρ_f and all of its extensions are now abandoned (forever), as the guess they were predicated on (we will see no more 0-splits) is wrong.

To make this ρ_∞ location stable, we will now ask that we would not improve the 1-states of ν_i, and hence ρ_∞ would be the location where we would meet the genericity requirements should ∞ be the correct 0-state.

Now ∞ being the correct 0-state will only be revealed slowly. So whilst we wait for this we would begin a new strategy for $\mathcal{R}_0$ on the assumption it would be imprudent for us to define Γ on extensions of (particularly) $\rho_\infty\widehat{\ }1$ based on whether $\tau \in S_{0,s}$ extends $\rho_\infty\widehat{\ }1$ in such a way that $\Gamma(\tau)$ does *not* have state ∞.

Carelessness would allow that this occurs. The way we avoid this is as follows. In the α-module where we have interactions, we will have additional extensions of $\rho_\infty\widehat{\ }i$, namely $\rho_\infty\widehat{\ }i\widehat{\ }0$ and $\rho_\infty\widehat{\ }i\widehat{\ }1$ (to begin) for $i \in \{0,1\}$.

The two nodes $\rho_\infty\widehat{\ }i\widehat{\ }0$ put in the construction of G the explicit guess that there are infinitely many ∞-stages for extensions of ν_i. (i.e. where the outcome ∞ looks correct for $\mathcal{N}_0$.)

We would begin at ν_0. First we would route the construction through $\rho_\infty\widehat{\ }0\widehat{\ }1 = \tau_{f,0,s}$, indicating it is a "testing" or "pressing" node for the guess ∞ in $T_{0,s}$ for the node ν_0. It has guess f, as indicated.

The idea would be that we route the construction in the cone $[\tau_{f,0,s}]$ (on the left construction) and consequentially in the cone $[\nu_0]$ in the right construction, pressing Φ_0 to *prove* that it is 0-splitting in this cone. Note that if there is no stage $s_1 > s$ such that we see a 0-split above ν_0 in $T_{0,s}$ we have forced $\mathcal{N}_0$ to have the f outcome.

Whilst we are awaiting this, we would continue a construction in $[\tau_{f,0,s}]$ all nodes having the f guess for the outcome of $\mathcal{N}_0$. In particular, a new version of $\mathcal{R}_0$ will be spawned in this cone. For example, we would take the two extensions of $\tau(f,0,s)$ as the new $\rho_f\widehat{\ }i$, taking $\rho_f = \tau(f,0,s)$.

Now either this is correct or at some stage $s_1 > s$ we might see a 0-split of ν_0 in T_{0,s_1}, say ξ_0, ξ_1. The construction would refine the tree T_{0,s_1} by redefining $T_{0,s_1+1}(00) = \xi_0$, $T_{0,s_1+1}(00) = \xi_1$, and we would raise the 0-state of ν_0 to be ∞ in T_{0,s_1+1}.

At this stage we would allow ourselves *one* step in the construction routing G_{s_1+1} through $\rho_\infty\widehat{\ }0\widehat{\ }0$. We would initialize the construction that went through $\tau(f,0,s)$ by cancelling it *forever* and pick a new $\tau(f,0,s_1+1)$ to the right of $\tau(f,0,s)$ in $\omega^{<\omega} \cap \rho_\infty\widehat{\ }0$.

At this stage we would look to see if there is some ν appearing to be in S_{0,s_1} extending $\rho_\infty\widehat{\ }1\widehat{\ }0$. If there is such a ν, then we would *prefer* to route the construction through $\rho_\infty\widehat{\ }1\widehat{\ }0$. However, we won't do this unless we can do so in the high 0-state.

Thus what we do at stage $s_1 + 2$ is to route the construction through

$\rho_\infty\widehat{\ }1\widehat{\ }1 = \tau(f, 1, s_2 + 2)$ The game is similar. We seek a 0-split of ν_1 in $T_{0,t}$ for $t \geq s_2 + 2$.

Now whilst we are waiting for $\mathcal{N}_0$ to produce extensions of ν_1 in the high 0-state, we will be making the construction in the cone $[\tau(f, 1, s_2+2)]$, guessing the low 0-state f for $\mathcal{N}_0$ for extensions of ν_1 in $T_{0,t}$ Again we can spawn a version of $\mathcal{R}_0$ in this cone.

Now again we are pressing $\mathcal{N}_0$. We can remain in this pressing situation until we see a 0-split in $T_{0,t}$ above ν_1.

Should this happen, we would initialize $\tau(f, 1, t + 1)]$ picking a new extension of $\rho_\infty\widehat{\ }1$ to the right of $\rho_\infty\widehat{\ }1\widehat{\ }1$. Now we would be free to have a single stage in the cone $[\rho_\infty\widehat{\ }1\widehat{\ }\widehat{\ }0]$ provided that ν has remained in $S_{e,u}$ for all stages u between $s_2 + 2$ and t.

Now, if this is not true for ν then we would go back to trying to play $\rho_\infty\widehat{\ }0$ by moving into the new cone $[\tau(f, 0, s_1 + 1)]$ picked earlier.

We would repeat the sequence above for the new $\tau(f, 0, s_1 + 1)$ until we had another 0-split(s) as determined by the construction. (For example, if ν_0 had two extensions $\xi_0 <_L \xi_1$, and they were corresponding to some genericity location for $\mathcal{R}_1$, say, then we would ask that both were "confirmed" to have 0-split extensions before we believed $\rho_\infty\widehat{\ }0\widehat{\ }0$ again.)

The cycle would then repeat as above.

Now, this is not quite correct as we now see. Let's consider the situation where $\mathcal{R}_1$ has a genericity location ρ with two extensions mapped by Γ to ξ_i for $i \in \{0, 1\}$.

In point of fact we would guess that since $\mathcal{R}_1$ has lower priority than $\mathcal{N}_1$, we would need to look also at the 1-state of $\rho = \rho_\infty\widehat{\ }0\widehat{\ }0$. So this ρ should be ρ_α where $\alpha \in \{f, \infty\}^2$.

This will entail there being several versions of $\mathcal{R}_1$ genericity locations on the left side.

One version is only spawned when ∞ looks correct for $\mathcal{N}_0$. This version would have guess state $\infty\infty$ and there would indeed be two versions according to the outcome of $\mathcal{R}_0$ guessing ∞ for $\mathcal{N}_0$.

Thus, for instance, each time ∞ looks correct for $\mathcal{N}_0$ we would route the construction through one of $\rho_\infty\widehat{\ }i\widehat{\ }j$ for $i, j \in \{0, 1\}$ as above. So, for instance, suppose that we believe that we should route G through $\rho_\infty\widehat{\ }0\widehat{\ }0$

because we think that the correct 0-state is ∞ and we think that there is no apparent $\mathcal{R}_0$ witness (the ∞-outcome) extending $\rho_\infty\widehat{\ }1\widehat{\ }0$. We will have mapped $\rho_\infty\widehat{\ }0\widehat{\ }0$ by Γ to some string ν_0, say, in M_s, on $T_{0,s}$.

Then this ν_0 will be tested not just for 0-states, but additionally for 1-states. So at the least we would need extensions $\rho(\infty\infty\infty, s)\widehat{\ }i\widehat{\ }j$ and $\rho(\infty\infty f, s)\widehat{\ }i\widehat{\ }j$, of $\rho_\infty\widehat{\ }0\widehat{\ }0$ for the possible behaviour of the 1 state of extensions of ν_0.

The plan is that weaker guesses will correspond to strings right and be initialized each time things look incorrect. For example, we might for a long time believe that $\infty\infty f$ looks correct but then see some new 1-split on the 0-splitting tree (i.e. an $\infty\infty$-split) and hence we would have g_{s+1} extend one of the strings $\rho(\infty\infty\infty, s)\widehat{\ }i\widehat{\ }j$ depending on the behaviour of $S_{1,s}$.

The idea now extends to any length of e-state.

Construction

The construction proceeds in substeps where we generate a string $TP_s \in \{\infty, f\}$ the apparent true path at stage s, which gradually gets longer with s.

In the construction associated with guesses $\alpha \in \{\infty, f\}^{<\omega}$ there (may) be "tests" which are defined as Test(α, s) and these will be strings in $2^{<\omega}$. If at any stage we move left of α in the construction this test is initialized to $\emptyset$.

At stage 0 assign ρ_∞ and $\rho_{f,s}$ as two strings in Baire space of length 2 with $\rho_\infty <_L \rho_{f,s}$. Also take two extensions $\xi_0 < \xi_1$ of $\rho_{f,0}$. Declare the *current test string* for $\mathcal{N}_0$ Test $(\infty, 0) = \lambda$ (in $T_{0,0}$). Let $G_0 = \xi_0, M_0 = \lambda$

Substage 0 At stage $s+1$, we begin at λ. Let Test$(\infty, s) = \nu$. See if there is a 0-split $\nu_0 <_L \nu_1$ extending ν in $T_{0,s}$.

Case 1. If there is, refine $T_{0,s}$ to $T_{0,s+1}$ by taking this 0-split, changing the current state of ν so that the leading element is ∞.

Subcase 1.1. If Test$(\infty, s) = \lambda$, then this is the first time we have believed that M might lie on a 0-splitting tree. We would extend ρ_∞ by two nodes, say $\rho_\infty 0, \rho_\infty 1$, and in turn each of these by two extensions, $\rho_\infty\widehat{\ }0\widehat{\ }0, \rho_\infty\widehat{\ }1\widehat{\ }0$ which will be the "high state" locations. We would set Test$(\infty, s+1) = \nu_0$. We would define $\Gamma(\rho_\infty\widehat{\ }0\widehat{\ }0) = \nu_0$ and $\Gamma(\rho_\infty\widehat{\ }1\widehat{\ }0) = \nu_1$. We would let $G_{s+1} = \rho_\infty\widehat{\ }0\widehat{\ }0$ and $M_{s+1} = \nu_0$. This would end the stage.

Subcase 1.2 If $\text{Test}(\infty, s) = \nu \neq \lambda$, then we will have already previously defined $\rho_\infty \preceq G_t$ for $t \leq s$.

Now we need to determine what to do at ρ_∞. Since ∞ looks correct for $\mathcal{N}_0$, we will have G_{s+1} extend one of $\rho_\infty \frown i \frown 0$ for $i \in \{0, 1\}$.

Subcase 1.2.1 If G_t has not properly extended $\rho_\infty \frown 0 \frown 0$ (so that this is the first time), then we will let $G_{s+1} \succ \rho_\infty \frown 0 \frown 0$, and move to the next substage. Declare that $s + 1$ is an $\infty\infty$-stage.

Subcase 1.2.2 If we have never had G_t extend $\rho_\infty \frown 1 \frown 0$ and there appears to be some $\tau \in S_{0,s}$ extending $\rho_\infty \frown 1 \frown 0$, choose the one that has been there the longest, (as per the convention before the construction) let $G_{s+1} = \tau$, and let $\Gamma(\tau) = \Gamma(\rho_\infty \frown 1 \frown 0 (= \nu_1)$. Declare that $s+1$ is an ∞f-stage. $\text{Test}(\infty, s+1) = \nu_1$.

Subcase 1.2.3 If we have never had G_t extend $\rho_\infty \frown 1 \frown 0$ and there appears to be no $\tau \in S_{0,s}$, then we ask thatG_{s+1} extends $\rho_\infty \frown 0 \frown 0$. and move to the next substage. We declare that $s + 1$ is an $\infty\infty$-stage.

Subcase 1.2.4 If we have previously had G_t extend $\rho_\infty \frown 1 \frown 0$, see if there is some $\tau \in S_{e,s}$ extending $\rho_\infty \frown 1 \frown 0$. If, the last time we visited ρ_∞ we extended $\rho_\infty \frown 1 \frown 0$, via τ we will insist on the hat convention that τ has appeared to be in $S_{e,t}$ for all intervening stages. If this is not true, then we will ask that G_{s+1} extend $\rho_\infty \frown 0 \frown 0$. We declare that $s + 1$ is an $\infty\infty$-stage and move to the next substage.

Subcase 1.2.5 Else we will have that $s + 1$ is an ∞f stage. Choose the τ that has been there the longest. Now we claim that *if we have previously defined* $\Gamma^{\hat{\tau}}$ *for any* $\hat{\tau} \preceq \tau$ *at some stage* $t \leq s$ *then* $\Gamma^{\hat{\tau}} = \epsilon$ **and** ϵ **will have 0-state** ∞. For the longest such $\hat{\tau}$ (this could be $\rho_\infty \frown 1 \frown 0$), we will define $\Gamma^\tau = \Gamma^{\hat{\tau}} = \epsilon$, say.

Subcase 1.2.5a Now if this was not already defined at the beginning of stage $s+1$, set $G_{s+1} = \epsilon$. Declare that with priority ∞f we will not improve the 1-state of ϵ, set $\text{Test}(\infty, s + 1) = \epsilon$ and go to stage $s + 2$.

Subcase 1.2.5b On the other hand, it could be that τ is already defined as a witness for $\mathcal{N}_0$ and has been so since the last ∞f-stage.

Claim This would necessarily entail $\text{Test}(\infty, s) \succ \Gamma^\tau$.

If this is the case, then we declare that G_{s+1} will extend τ. We move on to the next substage.

Case 2 There is no 0-split of Test(∞, s) in $T_{0,s}$.

Subcase 2.1 $\rho_{f,s}$ is currently undefined, then choose a new string for this right of ρ_∞ and define it as $\rho(f, s+1)$. Put two incomparable extensions of this on the tree $\xi_0 < \xi_1$ of $\rho_{f,s+1}$. Map $\Gamma^{\xi_i} = \nu_i$ for $i \in \{0, 1\}$ where ν_i are incompatible extensions of τ in $T_{0,s}$, and declare that these strings will currently only have a 0-state. This finishes the stage.

Subcase 2.2 If $\rho(f, s)$ is currently defined, move on to substep 1, asking that $G_s \succ \rho(f, s)$ (i.e. $G_{s+1,t} = \rho(f, s+1) = \rho(f, s)$. We declare that $s+1$ is a is a f-stage in either case.

To be consistent, we need to consider which of the two extensions of $\rho(f, s)$ look correct. This is done as in the basic module. If there is a $\tau \succ \xi_1$ in $S_{e,s}$ with all the relevant conventions, then the stage will be a ff-stage and we would ask that G_{s+1} extend ξ. If this is the first time, then that would compete the stage. If it is not the first time, we would move on to the next substep, asking that $G_{s+1} \succ \xi$, and move on to the next stage. We would define $\Gamma^\tau = \Gamma^{\hat{\tau}} = \epsilon$ for the longest $\hat{\tau} \preceq \tau$, and ask that now the 0-state of ϵ in $T_{0,s+1}$ not be improved with priority ff. The default is $\Gamma^\tau = \nu_1$.

If no such τ exists or the current candidate has left $S_{e,s}$ then we would have $s+1$ be a $f\infty$-stage, and move on to the next substage unless this is the first such time. We ask that $G_{s+1} \succ \xi_0$.

Inductively, suppose that $s+1$ is an α-stage and $|\alpha| = 2e$. The we will be associating $\mathcal{N}_e$ with α. We will have a current approximation to G_{s+1} at this substage t. We will have G^t_{s+1} mapped by Γ to some κ in $T_{e-1,s}$. We need to determine whether to believe that this is an $\alpha\infty$ stage or not. If the Test(α, s) is not yet defined, set it to be κ and end the stage.

Else, we **Claim** Test$(\alpha, s) \succ \kappa$.

See if there is a $\alpha\infty$-split on $T_{e-1,s}$ extending Test(α, s). If there is then we will play ∞ and this will be an $\alpha\infty$-stage. We will then either define $\rho(\alpha\infty, s)\widehat{\ }i\widehat{\ }0$ for the first time, and end the stage, or as above these will have already been defined and we will decide which of them to take according to the behaviour of $\mathcal{R}_e$ at this guess.

If the test fails, then we will define a new extension $\rho(\alpha f, s)$ of G^t_{s+1} if necessary, with two splits ξ_j for $j \in \{0, 1\}$ which we use for $\mathcal{R}_e$ for this weak guess exactly as above. The rest follows the same plan.

End of Construction

Now we verify the construction.

Let TP be the true path of the construction. That is the leftmost path. This exists by induction.

Lemma 8. *TP exists.*

Proof. First, we establish that the length 1 prefix of the true path TP is well-defined. Since the priority tree is finitely branching, either there are infinitely many ∞ stages, or all but finitely many stages are f stages. In the first case, suppose that s is an ∞ stage. Then we will discover a 0-split above the current Test string and the construction will properly extend ∞ at that stage. In the second case, let s_0 such that all $s > s_0$ are f stages. Then we will not discover 0-splits above the Test string at stage s. The construction at that stage s will properly extend the priority f.

Now, we show that the length 2 prefix of the true path TP is well-defined. Suppose the length 1 prefix of the true path TP is α. Either there will be infinitely many α stages with length 2 extension $\alpha\infty$ or all but finitely many α stages have extension αf. Suppose at stage s, S_0 appears sparse. Then we ensure that G extends ρ_∞, hence the construction at stage s extends $\alpha\infty$. Otherwise, if S_0 is dense, then at some stage $t > s$, we will find the longest resident of S_0 above the current prefix $\rho_f{}^\frown i$ of G. From stage t onwards, the construction will properly extend αf.

Inductively, let α be a guess of the priority where $|\alpha| = 2e$. For any $\alpha\infty$ stage s, we will seek e-splits of the current test string. When this split is discovered, the construction will extend $\alpha\infty$. Suppose almost all stages have priority αf. Then the construction will eventually extend the priority αf. The case when $|\alpha| = 2e + 1$ is similar. $\square$

Moreover the leftmost path of the left construction will map to the leftmost path of the right one.

That is, define G via this leftmost path.

Lemma 9. *For every $e \in \mathbb{N}$, the $\mathcal{R}_e$ requirements are satisfied by G and the $\mathcal{N}_e$ requirements are satisfied by M. Moreover, Γ is a partial-computable functional with $\Gamma^G = M$.*

Proof. First, we show that $\mathcal{N}_0$ is met in the construction. If $\infty \preceq TP$, then in infinitely many stages s, we will see a 0-split above the test string

Test(∞, s). This will necessarily imply that every prefix μ of M is 0-splitting. Thus, by Lemma 6, $M \leq_T \Phi_0^M$. Otherwise, if $f \prec TP$, then for almost all stages s, the tree $T_{0,s}$ is the full tree above Test(f, s). In this case, M has a non-0-splittable prefix on T_0. This implies that Φ_0^M is computable. Thus, $\mathcal{N}_0$ is met by the construction.

Now, to see that $\mathcal{R}_0$ is met in the construction, let α be the length-1 prefix of TP, and first consider the case when $\alpha\infty \preceq TP$. Then there are infinitely many α-stages where S_0 looks sparse. Hence, regardless of our specific choice, $\mathcal{R}_0$ will be met. If, on the other hand, $\alpha f \preceq TP$, then for all but finitely many stages s, $S_0[s]$ has an element extending $\rho_{f,s}$. There will be a stage $t \geq s$ where the longest resident in S_0 extending $\rho_{f,s}$ is found. In the construction, we also fix $\Gamma^{\rho_{f,t}}$. By the induction hypothesis, since $\mathcal{N}_0$ is met in the construction, we conclude that for almost all stages, we will identify an extension of ρ_f in the set S_0. Hence, the requirement $\mathcal{R}_0$ is met. Moreover, Γ maps ρ_f to a prefix of M.

If $\sigma \prec TP$ and $|\sigma| = 2e$ then σ is devoted to solving $\mathcal{N}_e$. By induction, the $\mathcal{R}_\nu$ for $\nu \prec \sigma$ and $\mathcal{N}_i$ for $i < e$ with outcome f on TP have determined an initial segment ρ of G. Each extension of $\rho \prec G_s$ will have the e-state of its Γ-projection checked. If we get stuck checking one ξ then all the strings in the cone above ξ will have e state with $e+1$-st element f, and $\xi \prec G$. (So there are no e-splits above ξ, and T_e is a full tree above it.) Otherwise every node that is a predecessor of G e-splits. In either case, $\mathcal{N}_e$ will be met, by Lemma 6.

If $|\sigma| = 2e + 1$ then σ solves $\mathcal{R}_e$. Inductively it eventually gets a final prefix ρ and since we are assuming that all the $\mathcal{N}_j$ of higher priority with the low e-state have been dealt with, this ρ is immortal and is visited infinitely often. Here we meet $\mathcal{R}_e$ as there is no reason not to. Suppose $\sigma\widehat{\ }f \preceq TP$. Then eventually we will find a τ in S_e which extends ρ. Note that when we define Γ^τ, we check only the e-state of the image. The reader should note that at earrlier stages, before we choose τ for the last time, Γ^τ might have had a higher state assigned to it, but a feature of the construction is that when τ is chosen for the satisfaction of $\mathcal{R}_e$, we will automatically lower its state to only consider the e-state for it. Hence the e-state machinery will not move it because of the action of e'-state machinery for $\mathcal{N}_{e'}$ for $e' > e$. It follows by induction that Γ^τ will have the correct e-state. This implies that the test string will change its value only finitely often, and further requirements are free to choose their test strings above Γ^τ. Also, if $\sigma\widehat{\ }\infty \preceq TP$, then there is a prefix β of G such that all extensions β' of β

satisfy $\Gamma^{\beta'} \preceq M$. $\square$

Acknowledgments

This research was initiated during a visit to André Nies' research center at Whiritoa. This research was supported by the Marsden Fund through grants to Rod Downey and André Nies enabling Nandakumar's visit to New Zealand.

References

[1] George Barmpalias, A. R. Day, and Andrew E. M. Lewis. The typical turing degree. *Proceedings of the London Mathematical Society*, 109:1–39, 2014.

[2] George Barmpalias and Andy Lewis-Pye. The information content of typical reals. In G. Sommaruga and T. Strahm, editors, *Turing's ideas - their significance and impact*. Basel Birkhauser/Springer Basel, 2014.

[3] C. T. Chong and R. G. Downey. Degrees bounding minimal degrees. *Proceedings of the Cambridge Philosophical Society*, 105:211–222, 1989.

[4] C. T. Chong and R. G. Downey. Minimal degrees recursive in 1-generic degrees. *Annals of Pure and Applied Logic*, 48:215–225, 1990.

[5] C. T. Chong and Carl Jokusch. Minimal degrees and 1-generic degrees below 0'. In *Proceedings of the Aachen Logic Conference*. Springer Berlin, 1985.

[6] R. G. Downey and Liang Yu. Arithmetical sacks forcing. *Archive for Mathematical Logic*, 45, 2006.

[7] Rod Downey. On π_1^0 classes and their ranked points. *Notre Dame Journal of Formal Logic*, 32:499–512, 1991.

[8] Rod Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.

[9] C. A. Haught. The degrees below a 1-generic degree $< 0'$. *Notre Dame Journal of Formal Logic*, 51:770–777, 1986.

[10] Carl Jokusch. Degrees of generic sets. In F. R. Drake and S. S. Wainer, editors, *Recursion Theory: Its Generalizations and Applications*, pages 110–139. Cambridge University Press, 1980.

[11] Masahiro Kumabe. A 1-generic degree which bounds a minimal degree. *Journal of Symbolic Logic*, 56:1075–1081, 1991.

[12] Stuart Kurtz. *Randomness and genericity in the degrees of unsolvability.* PhD thesis, University of Illinois, Urbana-Champaign, 1981.

[13] Clifford Spector. On degrees of recursive unsolvability. *Annals of Mathematics*, 64:581–592, 1956.