The Journal of Symbolic Logic
Volume 84, Number 4, December 2019

A WEAKLY 2-GENERIC WHICH BOUNDS A MINIMAL DEGREE

RODNEY G. DOWNEY AND SATYADEV NANDAKUMAR

Abstract. Jockusch showed that 2-generic degrees are downward dense below a 2-generic degree. That is, if $\mathbf{a}$ is 2-generic, and $\mathbf{0} < \mathbf{b} < \mathbf{a}$, then there is a 2-generic $\mathbf{g}$ with $\mathbf{0} < \mathbf{g} < \mathbf{b}$. 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. While there have been full approximation constructions of Δ^0_3 minimal degrees before, our proof is rather novel since it is a computable full approximation construction where both the generic *and* the minimal degrees are $\Delta^0_3 - \Delta^0_2$.

§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[1] 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, no Cohen generic degree has minimal degree. But there are restricted forms of genericity 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 1.1. 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) (*A meets S*) or $\exists\tau \prec A \forall\sigma \in S(\tau \not\preceq \sigma)$. (*A avoids S*)
2. A set of strings 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 if it contains a (weakly) n-generic set.

Received January 18, 2016.
2010 *Mathematics Subject Classification.* Primary 03D28, Secondary 03D30, 03D55.
Key words and phrases. Turing degree, minimal degree, weakly 2-generic.

[1] Here, and henceforth, "string" is meant to mean finite string.

0022-4812/19/8404-0002
DOI:10.1017/jsl.2018.68

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, see Kurtz [14] or Downey and Hirschfeldt [9], for a more readily available reference.) How do n-genericity and minimality relate? First, it is easy to see that no 1-generic degree can be minimal, but Kurtz showed that every hyperimmune degree is weakly 1-generic and hence there can be minimal degrees containing weakly 1-generic sets. Jockusch [12] was the first person to give a detailed analysis of notions of (weak) n-genericity and their relationship with Turing reducibility. In particular, Jockusch 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 [4] 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 [11] 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 [13] and Chong and Downey [3] 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 [3] 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 [5], they used this local condition to construct a minimal degree below $\mathbf{0}'$ *not* computable from a 1-generic, and Downey and Hirschfeldt [9] (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 [14]. Finally, Downey and Yu [7] 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, but a 1-generic degree can bound a minimal degree. In this paper, we give an affirmative answer to 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 1.2. *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. It is true that both weak 2-genericity and minimality constructions are easily done by using finite extension and perfect set forcing, respectively. Minimality can also be achieved using forcing with partial computable trees. The difficulty is the construction of the reduction[2] $\Gamma^G = M$. Thus, to use two forcing-type constructions to construct G and M, you would somehow need to specify Γ *in advance*, and hence likely a truth table construction, or find a local condition like that of Chong and

[2] In this paper we will always view a reduction as a partial computable map Γ from strings to strings, such that if $\Gamma^\sigma\downarrow$ and $\Gamma^\tau\downarrow$, $\sigma \prec \tau$, then $\Gamma^\sigma \preceq \Gamma^\tau$, and $\lim_{\sigma \prec G \wedge \Gamma^\sigma\downarrow} \Gamma^\sigma = M$. Occasionally, to emphasise this view, we might write $\Gamma(\sigma)$ in place of Γ^σ.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

Downey (but more complex), and then run a second construction like they did. While we acknowledge one of these might be possible, they both seem extremely difficult. The simplest thing seems to be to construct Γ along with the construction, and since Γ needs to be computable, this will entail the construction being computable.

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 [8], 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 binary strings is denoted by $2^{<\omega}$ and the set of infinite binary sequences by 2^{ω}. We will also use strings from $\omega^{<\omega}$, finite sequences of natural numbers. We point out that, up to Turing degree, (weak) n-genericity in $\omega^{<\omega}$ and $2^{<\omega}$ are identical. If σ is a finite string, then $[\sigma]$ denotes the *cylinder* determined by σ, *i.e.,* the set of infinite binary sequences with prefix σ. If S is a set of 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.

We remind the reader of our view that procedures/reductions/functionals $\Theta^Z = Y$ are partial computable maps from strings to strings such that if $\sigma \prec \tau$ and $\Theta^\sigma\downarrow, \Theta^\tau\downarrow$, then $\Theta^\sigma \preceq \Theta^\tau$, and $\lim_{\sigma \preceq Z} \Theta^\sigma = Y$.

§**3. Minimal pair.** In this section we prove the following easy result, surely known to anyone who thought about it.

Proposition 3.1. *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 $S = \{\sigma \mid [\exists s_0 \forall s > s_0(\Phi^\sigma\downarrow[s] \not\prec X_s) \vee (\exists n \forall \tau \forall s)(\sigma \preceq \tau \to \Phi^\tau(n)\uparrow[s])]\}$.

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. ⊣

§**4. The Proof of Theorem 1.2.** We build a weakly 2-generic G and a set M of minimal degree and a procedure Γ with $\Gamma^G = M$. Proposition 3.1 imposes some restrictions on the constructions of both G and M. Typically in computable constructions of sets X and Y, with functionals Θ being built in the constructions,

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

we ensure that from some point onwards $\Theta^\sigma = \tau$ for some $\sigma \prec X$ and $\tau \prec Y$, and for all stages s beyond some point $\sigma \prec X_s$ and $\tau \prec Y_s$. This is impossible here as it would make X and Y both Δ^0_2, since initial segments have come to limits, by Proposition 3.1.

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 denotes the e-th Turing procedure, and we will let $S_0, S_1, \dots$ 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.

4.1. 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, suppose that σ appears in S_i at stage s, with witness s_0. By this statement we mean 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 $t \leq s+1$.

4.2. Further conventions. When we write $\tau \in S_{i,s}$ we mean that τ appears to be in $S_{i,s}$ in the sense above. *Additionally*, if τ appears to be in $S_{i,s}$ with witness s_0, then we will ask that $s_0 > |\tau|$. That is, we ask that long strings τ must have large witnesses s_0. This additional convention helps when it comes to choosing strings appearing to be in $S_{i,s}$ during the priority construction. These conventions are more or less standard.

The requirements we must meet are the following.

$$\mathcal{R}_e : S_e \text{ dense } \Rightarrow G \text{ meets } S_e \qquad\qquad \text{[weak 2-genericity]}$$
$$\mathcal{N}_e : \Phi^M_e \text{ total } \Rightarrow (\Phi^M_e \equiv_T \emptyset) \vee (M \leq_T \Phi^M_e). \qquad \text{[minimality]}$$

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 $\mathcal{R}_e$.

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

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

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

discussion, we assume that $\rho(e,s)$ is a true initial segment of G, and moreover $\Gamma^{\rho(e,s)} \downarrow [s]$. In particular, in the real construction, we will also have that $\Gamma^{\rho(e,s)}$ lies in a tree $T_{e,s}$ where we are building the minimal degree and this image is in a good "e-state", a concept we will discuss in the next section where we are discussing the minimal degree construction. The only relevance for us here is that we are assuming that the minimality machinery won't initialize this string.

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

The most important of these cones for this discussion are $[\rho(e,s)^\frown i^\frown 0]$ for $i \in \{0,1\}$. This is because we will simplify things and pretend that the left hand side (building G) will be built in the same e-state as that of $\rho(e,s)$. By doing this we will avoid the effects of other requirements. All of the other $[\rho(e,s)^\frown i^\frown j]$ for $j \geq 1$ play a role in forcing this simplification to be true, or we will gain some higher priority progress, as we later see.[3]

So concentrating on these two strings, we work as follows. It will be convenient in the construction to also make sure that $\Gamma^{\rho(e,s)^\frown 0^\frown 0} \downarrow [s]$ and $\Gamma^{\rho(e,s)^\frown 1^\frown 0} \downarrow [s]$ are incompatible extensions of $\Gamma^{\rho(e,s)}$. The cone $[\rho(e,s)^\frown 0^\frown 0]$ (and in fact $[\rho^\frown 0]$) is devoted to $\mathcal{R}_e$'s Π^0_2 strategy and $[\rho^\frown 1]$ the Σ^0_2 strategy. We will want the images of these cones to be incompatible, because, as we will later see actions on the *right*, where M is constructed, have consequences on the left where G is constructed. So if we have defined $\Gamma^\sigma = \tau[s]$ where are forced to obey that forever. It is hard to reconcile that fact if we had distinct cones on the left but comparable images on the right. In fact, one of the goals of the construction is to have alignment of the leftmost paths of the priority tree and those of the constructions. As we see, this incompatibility feature will necessitate certain complexities in the construction, but these will be discussed later.

The strategy is the obvious one. If we see some $\tau(e,s) \succ \rho(e,s)^\frown 1^\frown 0$ 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 $\rho(e,s)^\frown 0^\frown 0$. That is, until we see such a $\tau \in S_{e,s}$, we have $\rho(e,s)^\frown 0^\frown 0 \prec G_t$. We regard this as outcome ∞.[4]

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 will be to re-route G_t through $\rho(e,s)^\frown 0^\frown 0$ again. When we move back to $\rho(e,s)^\frown 0^\frown 0$, we will play outcome ∞, for at least one stage. At stage $t+1$ we will again seek a $\tau(e,t) \in S_{e,t}$ extending $\rho(e,s)^\frown 1^\frown 0$.

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

[3] In some sense, this shows the length 2 extensions of ρ have two roles. One is to meet $\mathcal{R}_e$ and the other will be to reveal information about the behavior of splitting for $\mathcal{N}_e$.

[4] Originally, we had a separate waiting outcome w, but have chosen to simplify the combinatorics of the construction to have only two outcomes $\infty <_L f$, where ∞ will either mean waiting for S_e to provide a string forever, or infinitely many stages occur where candidate strings τ leave S_e, as we see below.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

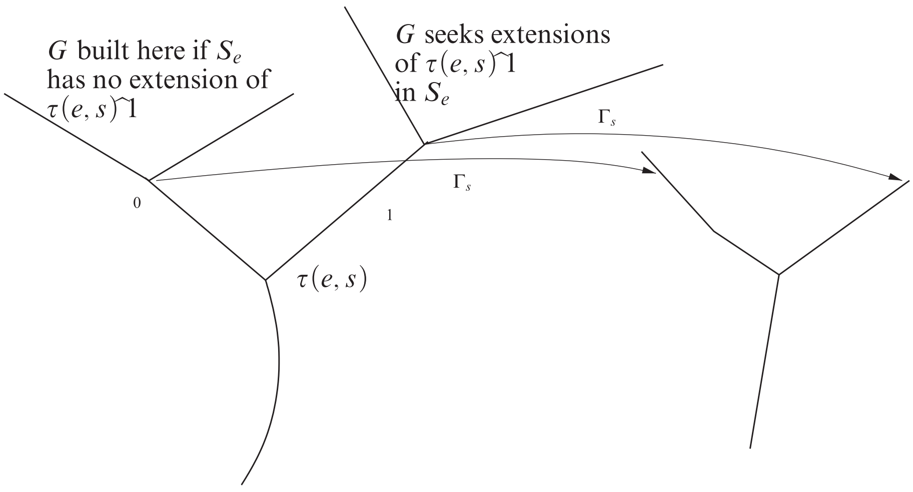
G built here if S_e has no extension of $\tau(e,s)\frown 1$
G seeks extensions of $\tau(e,s)\frown 1$ in S_e
Γ_s
Γ_s
0
1
$\tau(e,s)$

FIGURE 1. Basic Module for $\mathcal{R}_e$

Π_2/Σ_2 arguments, we 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.[5]

Note that if S_e is really dense, eventually we will find $\tau = \lim_s \tau(e,s)$ on extending $\rho(e,s)\frown 1\frown 0$, to get stuck on. This is the Σ_2^0 outcome f. If no such τ is found, then we will either switch to $\rho\frown 0\frown 0$ infinitely often (outcome ∞, the Π_2^0 outcome) or get stuck extending $\rho\frown 0\frown 0$ from some point on, also outcome ∞. On the priority tree, we have $\infty <_L f$, as mentioned above.

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 model in mind.

Note also, in the background, we will also be mapping $\Gamma_s^G \to M_s$ in conjunction with the above. We point out that $\mathcal{R}_e$ has no actual desire to make Γ total. For example, in the basic $\mathcal{R}_e$ module, we would naturally map $\Gamma^{\tau(e,s)} = \Gamma^{\rho(e,s)\frown 1\frown 0}$ and potentially Γ maps all extensions of $\rho(e,s)\frown 0\frown 0$ to $\Gamma^{\rho(e,s)\frown 0\frown 0}$. Plainly there are problems with this idea since we need to make Γ total. Problems are revealed when we consider the strategy in combination with others. See Figure 1 below.

REMARK 5.1. Remember, $\mathcal{R}_e$ does not care about the totality of Γ for its satisfaction. As we will see, it is the definition of Γ itself which causes difficulties with the satisfaction of $\mathcal{R}_e$ if we are careless. The point is that if we decide to move to some $\tau \in S_{e,s}$ and Γ^τ is already defined, firstly it needs to be the case that Γ^τ extends $\Gamma^{\rho(e,s)\frown 1\frown 0}$. Secondly, it must not be that this action causes us to injure higher priority minimality requirements by forcing us off the "e-splitting" part of the relevant tree, something we glossed over in the discussion above and something we now discuss.

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

We mention these points in passing, for the reader to keep in mind when we discuss the requirements below.[6]

§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. That is, a $\mathbf{0}''$ oracle is used with perfect trees as conditions. For the forcing, extension means subtree so we construct a sequence of computable perfect trees $T_{-1} \supseteq T_0 \supseteq \cdots$. At step e, we either put all paths on an "e-splitting tree", or there is some σ on T_e such that if we take T_{e+1} as the full subtree of T_e above σ, then either we force divergence or force computability. (Precise definitions are given below.)

Less well known are full approximation constructions, 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. We write $\sigma \in T$. The set of paths in T are denoted by $[T]$, where $P \in 2^\omega$ is a path iff for all $\sigma \preceq P$, there exists $\sigma' \succ \sigma$ with σ' on T and $\sigma' \prec P$. A set M is said to be a *on* T if infinitely many prefixes of M are on T. Recall the following standard definition.

Definition 6.1. A string σ on a function tree T is said to e-*split* if there are incompatible extensions τ and ρ of σ on T, and an input n such that $\Phi_e^\tau(n) \downarrow \neq \Phi_e^\rho(n) \downarrow$. It follows that a string σ on T is *non-e-splittable* if for every pair of extensions τ, ρ of σ on T 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.

Finally, a tree T is called e-splitting iff for all v, $T(v0)$ and $T(v1)$ e-split.

Henceforth, we drop "function" where it is obvious. The notion of e-splitting trees is useful for the construction of sets of minimal degrees because of the following fundamental property.

Lemma 6.2 (Essentially, Spector [18]). *Let T be e-splitting and $M \in [T]$. If Φ_e^M is total, then $M \leq_T \Phi_e^M$.*

As mentioned earlier, in the classical Spector construction,[7] construct a nested sequence of computable trees $2^{<\omega} = T_{-1} \supseteq T_0 \supseteq T_1 \supseteq \cdots$, and at step e, see if we can find a full subtree of T_e which is not e-splittable (in which case Φ^M will be computable if it is total), or construct T_{e+1}, an e-*splitting subtree* of T_e. We recall that this is done inductively. We begin at λ, find $T_e(\sigma_0) = v_0$ left of $T_0(\sigma_1) = v_1$ and on T_e which e-split. Then, we define $T_{e+1}(\langle i \rangle) = v_i$ for $i = 0, 1$, and then find

[6] More specifically, as we discuss later, we cannot allow $\mathcal{R}_e$ to move us off a higher priority "e-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 $M_{s+1} \succeq \sigma$ if $G_{s+1} \succeq \tau$. This will generate the key tension in the construction.

[7] Or at least as re-formulated by Shoenfield [16].

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

e-splittings above each of these nodes on T_e, creating a perfect subtree T_{e+1} of $T - e$ which e-splits.

Remark 6.3. We will try to describe the full approximation construction, concentrating on the devices we introduce to make it work. One of the difficulties is that many things are interacting, both dynamically, and simultaneously, so looking at things in isolation (as one can in an oracle construction) is a bit misleading.

In the full approximation construction used here, the first difference is that we focus on some approximation to M, M_s for our attention. So, for example, if we never see $\Phi_e^{M_s}(n) \downarrow$ for some n, then we will conclude $\Phi_e^M(n) \uparrow$ even though there might be strings σ on the relevant tree T_e where $\Phi_e^\sigma(n) \downarrow$.

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 (i.e., $T_{-1,s}(\xi)$ for some ξ of length s although this is really not important) path on $T_{0,s}$. Initially, $T_{0,0} = T_{-1,0}$, so that $T_{0,0}(v) = v$. At each stage s, we will associate with each node v in $\mathrm{dom}(T_0)$ a 0-*state* which is one of ∞ or f. Abusing notation, we also will regard $T_{0,s}(v)$ as having the 0-state of v on $T_{0,s}$. This 0-state will indicate whether we think that $T_{0,s}(v)$ 0-splits or not. f means that we don't think v 0-splits, and ∞ means we do. Anticipating things somewhat, we will use e-states which will be strings of length $e+1$ from $\{\infty, f\}^{e+1}$ where $\infty <_L f$. "Raising" and "lowering", "higher" and "lower", states refer to this lexicographic ordering. The interpretation of a node v having a 2-state $\infty f \infty$ is that the node with this 2-state is on $T_{2,s}$, it has two extensions on this tree which are both 0 and 2-splitting, but also thinks it is is part of $T_{1,s}$ and also $T_{2,s}$, as we see, where we believe that we won't again see a 1-split. The notion of e-state goes back to Friedberg's maximal set construction [10]. Their use in full approximation minimal degree constructions goes back to the original papers of Yates [19] and of Cooper [6].

This is done in a somewhat obvious inductive way. We will begin with $v = \lambda$, the empty string. Initially we have no computations. We give $T_{0,s}(\lambda)$ the 0-state f.[8] As the construction proceeds, we monitor $\Phi_0^{M_s}$. At the first stage s, if any we see $\Phi^{M_s}(0) \downarrow [s]$ we will issue a description of $\Phi_0^M(0)$, and argue that this is correct. Hence Φ_0^M is computable. Notice that this has no effect on $T_{0,s}$.

More generally, suppose that we have issued descriptions of $\Phi_0^M(m)$ for $m < n$, and we are dealing with some $v \prec M_s$ of length n. We'd await a stage where $\Phi_e^{M_t} \restriction n \downarrow$, and issue a description of $\Phi_0^M(n)$.

The only time we are wrong is that we see some n where $\Phi_0^{v_0}(n)$ and $\Phi_0^{v_2}(n)$ 0-split for some v_0, v_1 on $T_{-1,s}$. If, at some stage, we observe this, then, supposing wlog $v_0 <_L v_1$, we raise the 0-state of $\lambda = T_{0,s}(\lambda)$ to ∞, refining the tree $T_{0,s+1}$ so that we define, for all $\xi \in 2^{<\omega}$, $T_{0,s+1}(0\widehat{\ }\xi) = T_{-1,s}(\eta_0\widehat{\ }\xi)$ where $T_{-1,s}(\eta_0)$ is the use of $\Phi_0^{v_0}(n)$ on $T_{-1,s}$, and $T_{0,s+1}(1\widehat{\ }\xi) = T_{-1,s}(\eta_1\widehat{\ }\xi)$, where $T_{0,s}(\eta_1)$ is the use of $\Phi_0^{v_1}(n)$. In the $T_{0,s}$ construction, the actual use (here regarded as the whole string up to the largest number used in the computation) will be on $T_{-1,s}$ as it is initially the

[8] It is possible to separate the *three* states, non-halting, all extensions giving the same answer on all arguments (i.e., Φ_0^M computable), or 0-splitting, but using three 0-states the construction even more elaborate, so we choose to combine the first two possibilities as f.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

identity tree. In the inductive strategies, we will use the shortest string extending the use actually on the tree.

In the full construction, we implement the strategy outlined above with a parameter we call *Test*, or more precisely, *Test*(α, s), where α will be a guess from the priority tree, and s is a stage number. Here with one minimality requirement, we will only need Test(∞, s). In the above, initially Test(∞, s) is set to be the empty setting λ, which is being tested to see if it has a 0-split above it. Should a 0-split be found, one of the extensions of the split will be the next Test(∞, s). For example, we will choose Test$(\infty, s+1) = T_{0,s+1}(0)$, if no other requirements are around, and have $M_{s+1} \succ$ Test$(\infty, s+1)$. In the full construction, it might be that genericity requirements ask that Test$(\infty, s+1) = T_{0,s+1}(1)$, because we think that we might be able to have the f outcome for $\mathcal{R}_0$, and $\Gamma^{-1}(\text{Test}(\infty, s+1)) = T_{0,s+1}(1)$ might be sympathetic to this cause, as we see below (Fig. 2).

In the real construction, the test parameter α is not for a single procedure 0, but will be an e-state and this is testing for a split of some kind in matching $e-1$-state on $T_{e-1,s}$, as well as other issues relating to higher priority $\mathcal{R}_j$'s. Test locations can be moved by the interactions of the requirements, but the reader should keep the following guiding principle. If we have a test location for some e-split at some string v we are *pressing* Φ_e to *prove* that it e-splits above v on T_{e-1}. If no splits are to be found, then this gives a *global win* on Φ_e since we have that Φ_e^M is computable or Φ_e^M is not total, should we keep the construction within $[v]$ in T_e. The play-offs as to when and how we pursue this pressing strategy is one of the key tensions in the proof.

The construction is seeking to put M on a 0-splitting partial computable subtree of T_0. At stage s, this corresponds to part of the tree $T_{0,s}$ containing M as one of its paths, where the 0-state of the initial segments of M on $T_0 = \lim_s T_{0,s}$ is ∞, in the limit. Should we hit some place v on $T_{0,s}$ which is a fixed initial segment of M where we can't raise the 0-state of v, then we will have $M \in [v]$ in T_0, and hence either Φ_0^M is partial or it is computable.

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

Figure 1 gives a general position of the construction in the tree $T_{0,s}$.

Remark 6.4. We point out that this discussion cannot be completely correct as it would make M Δ_2^0 which is impossible. The reason is that potentially each T_e could limitwise pick some cone for M to be built in. But it is a good "image" for the reader to keep in mind.

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 approximation 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.

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

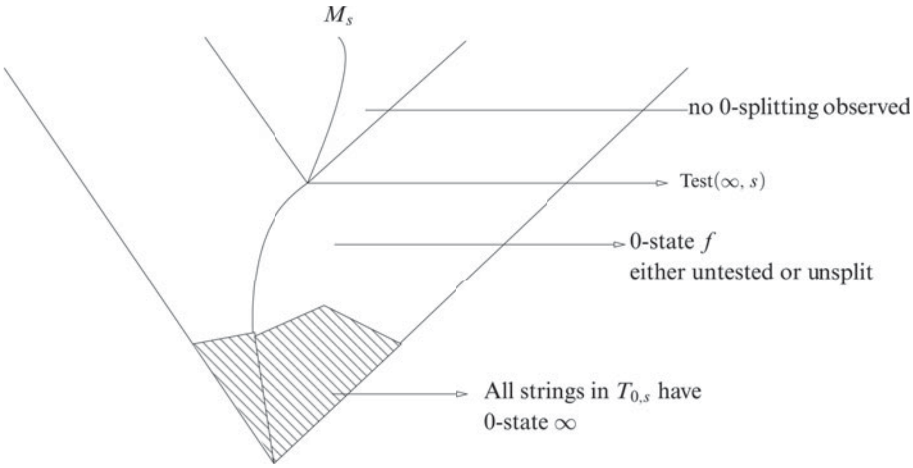

FIGURE 2. Basic module for $\mathcal{N}_0$.

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.[9]

At each stage we will associate with a string ρ on $T_{e,s}(\sigma)$ an e-state (as we do with σ). These are changed as above according to whether the construction observes $T_{e,s}(\sigma)$ e-splits on $T_{e,s}$ (i.e., the splitting nodes must be on $T_{e,s}$). That is, attention was focused on $\rho \prec M_s$ by a test, and we saw an e-split of ρ on $T_{e-1,s}$ with the same $e-1$-state as that of ρ. We then raise e-states by replacing the last symbol f by ∞ if splits are observed and refining the tree $T_{e,s}$. (In the construction, this is reflected as follows: if a string v on the priority tree represents $\mathcal{N}_e$ and we see a new e-split, as described above, at a stage where v looks correct, we will say that the stage is a $v\widehat{\ }\infty$ stage, else a $v\widehat{\ }f$ stage.)

e-states have the nice property that the highest one is the one beginning with ∞ in the first place. Thus, maximizing them means that we are placing M on a 0-splitting tree. If we can do this using the tests above, we will. Thus, the action of $\mathcal{N}_0$ in refining $T_{0,s}$ has implications on $T_{e,s}$ for $e > 0$. To wit: 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 $\hat{\alpha}\widehat{\ }f$ for some $\hat{\alpha} <_L \alpha$ where $\infty <_L f$, meaning that some tree $T_{\hat{e},t}$ becomes refined $(\hat{e} < e)$.

We can visualize this using the notion of "boundaries" on the various trees.[10] On tree $T_{0,s}$, there is a boundary below which every string σ is 0-splittable in the sense here described, 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 in the 0-splitting subtree of $T_{0,s}$. Above that, is a layer of strings which lie in the 0-splitting part of $T_{0,s}$, but not the 1-splitting part of $T_{1,s}$ which is also in

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

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

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

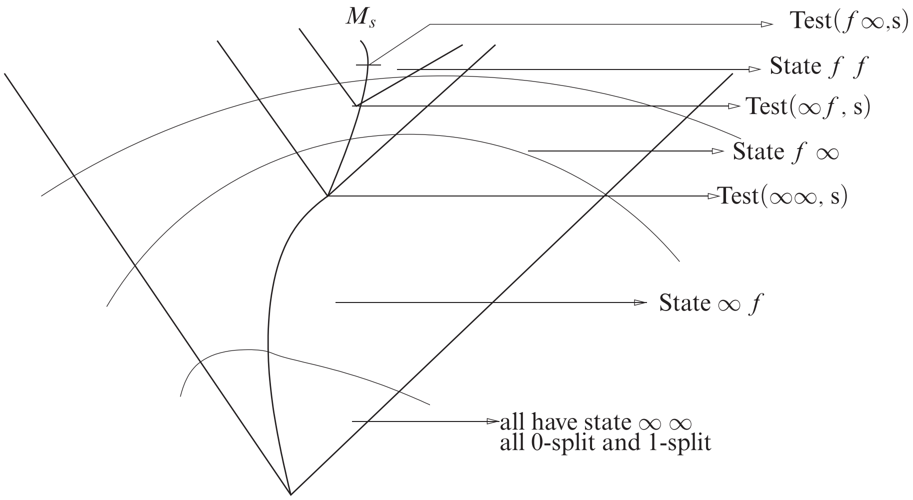

FIGURE 3. Boundaries for splitting trees.

the 0-splitting part of $T_{0,s}$. The third layer from the bottom consists of strings in the non-0-splitting part of $T_{0,s}$ but in this section have 1-splits in $T_{1,s}$. The topmost layer consists of nodes which are neither 0-splittable nor 1-splittable. The reader should refer to Figure 3.

As with all full approximation constructions, the details are very messy but the idea is straightforward.

REMARK 6.5. We remark in passing that the above is not quite correct when the inductive strategies are considered, in the sense that there might be play-offs between the priorities of the actions. For instance, consider the situation that we have a requirement $\mathcal{N}_e$ of *lower* priority than $\mathcal{R}_j$. 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 $T_{e-1,s}$. This image string cannot be removed with priority $\mathcal{R}_j$. So it is unreasonable for $\mathcal{N}_e$ to be allowed to remove it as we think we are currently meeting $\mathcal{R}_j$ with it. This is implemented by where the relevant test string is, at any stage. The point is that we make e-states finite strings, 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. In this example, to $|\sigma| > |\Gamma^{\rho(1,s)}|$. More on this later.

§7. The inductive strategies.

We will now discuss the inductive strategies, which Soare [17] refers to as the "α-modules". Certain modifications, some of which we have already foreshadowed, are needed to make the requirements live with each other.

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 $\mathcal{R}_0$.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

The driver for $\mathcal{N}_0$ is to build M in a high 0-state tree T_0. It is natural for $\mathcal{R}_0$ to guess the eventual state of $\mathcal{N}_0$, and hence there will be two versions of $\mathcal{R}_0$, namely $\mathcal{R}_0^f$ and $\mathcal{R}_0^\infty$. Initially, $\mathcal{R}_0$ must guess state f, and $\mathcal{R}_0^f$ would have erected a genericity location $\rho = \rho_0$. (For $\mathcal{R}_0$, ρ_0 would be λ on the left hand tree.) As mentioned earlier, satisfaction is pursued on $\rho^\frown 0^\frown n$ and $\rho^\frown 1^\frown n$ for $n \in \omega$.

Now at any one time only *four* of these nodes are in action. $\rho^\frown 0^\frown 0$ and $\rho^\frown 1^\frown 0$ are *never* initialized, and these two are the possible locations we use to meet $\mathcal{R}_0^\infty$, that is, $\mathcal{R}_0$ should it turn out that the true outcome of $\mathcal{N}_0$ is ∞ so that M lies on a 0-splitting subtree of T_0.

The construction will ensure that for every *string η in either $[\rho^\frown 0^\frown 0]$ and $[\rho^\frown 1^\frown 0]$ if $\Gamma^\eta\downarrow$, then Γ^η has 0-state ∞ in T_0.*

Also, at each stage s, there will be two other uncancelled strings of the form $\rho^\frown 0^\frown n$ and $\rho^\frown 1^\frown n$ with $n \neq 0$, *one* which will be currently serving the role of ρ in the case that f is the final state of M in T_0. The idea is that one of the form $\rho^\frown 0^\frown n$ $(n \neq 0)$ will correspond to a place where we are trying to meet $\mathcal{R}_0^f$. That is, should $\rho^\frown 0^\frown n$ be the current string of interest on the left hand side, we will currently be stuck testing for the next 0-split above some v extending $\Gamma(\rho^\frown 0^\frown 0)$ in the right hand side (which will be the current Test(∞, s)). Because of this, precisely as with the basic module, $\rho^\frown 0^\frown n$ will need two extensions, $\rho^\frown 0^\frown n^\frown 0$ will be where we will build G should,

- This be the final proof that $\mathcal{R}_0^f$ is the correct version, in that no further 0-splits are found above $\Gamma(\rho^\frown 0^\frown 0)$ in the right hand side, as proven by Test(∞, s).
- Given the first point, there is no permanent extension of S_0 in the cone $[\rho^\frown 0^\frown n^\frown 1]$.

By the same token, $\rho^\frown 0^\frown n^\frown 1$ will be the place we are pursuing the Σ_2^0 outcome of $\mathcal{R}_0^f$ provided that we never play the infinite outcome of $\mathcal{N}_0$ again. Thus, were this last condition correct, we will either flip infinitely often to $[\rho^\frown 0^\frown n^\frown 0]$ or get permanently stuck in one of $[\rho^\frown 0^\frown n^\frown 0]$ or $[\rho^\frown 0^\frown n^\frown 1]$, the latter meaning that we have found a permanent witness for S_0 there.

Similar comments pertain to the strings of the form $\rho^\frown 1^\frown n^\frown 0$ and $\rho^\frown 1^\frown n^\frown 1$ $(n \neq 0)$. These two correspond to the construction testing for 0-splits above $\rho^\frown 1^\frown 0$, and guessing that we have reached a stage with the final proof that $\mathcal{R}_0^f$ is the correct version of $\mathcal{R}$ since this test never returns. (This is not quite correct, as we need to have slightly more testing of Φ_0 to prove it is 0-splitting, because of the combinatorics of the construction and because we want to make the driver "*every* string η in either $[\rho^\frown 0^\frown 0]$ and $[\rho^\frown 1^\frown 0]$ if $\Gamma^\eta\downarrow$, then Γ^η has 0-state ∞ in T_0" correct. This will be discussed in more detail soon.)

The current string $\rho^\frown i^\frown n$ $(n \neq 0)$ being used will be denoted by $\rho_{i,f,s}$ for $i \in \{0, 1\}$. The subscript "i" indicates we are currently testing Φ_0 above $\Gamma(\rho^\frown i^\frown 0)$, and the "$f$" indicates that we are guessing this test won't return a 0-split on the right. (In the full construction, this f will be a longer string correlating to a guess.) As indicated, they have two length 1 extensions, $\rho_{i,f,s}{}^\frown 0$ and $\rho_{i,f,s}{}^\frown 1$, each mapped to incomparable strings in $T_{0,s}$ extending Test(∞, s).[11] These attempt to meet $\mathcal{R}_0$, *on the assumption that $\Gamma^{\rho_{i,f,s}}$ is now stuck in the low 0-state and ∞ never again looks correct for $\mathcal{N}_0$.* That is, while this assumption looks correct, we will play $\rho_{i,f,s}{}^\frown 0$ when $\mathcal{R}_0$ looks

[11] Strictly speaking this is not quite what is done, as the initialization machinery to later be described, will mean that only one of these might be used.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

like it has the ∞ outcome, and $\rho_{i,f,s}{}^\frown 1$ will be played at stage when we believe that we have a $\tau \in S_{0,s}$ extending $\rho_{i,f,s}{}^\frown 1$. We remark that each time the hypothesis that $\rho_{i,f,s}$ is being built upon proves false (i.e., Test(∞, s) reveals another 0-split), $\rho_{i,f,s}$ is cancelled forever, and a new $\rho_{i,f,s+1}$ is picked. Finally, which string of the $\rho^\frown i^\frown n$ will be chosen as the $\rho_{f,j,s}$ is determined by a series of events in the construction, and the behaviour of the both $\mathcal{R}_0$ and $\mathcal{N}_0$ as described in detail below.

In more detail, how this all works is as follows. Initially, $\rho_{i,f,s} = \rho^\frown i^\frown 1$. We will route the construction through $\rho_{0,f,s}$ and Γ-map its two extensions $\rho_{i,f,s}{}^\frown j$ for $j \in \{0, 1\}$ to incompatible extensions $\eta_0|\eta_1$[12] in $T_{0,s}$. Since this is the first action, we could simply pick $\langle 0\rangle, \langle 1\rangle$ as the two η_j.

Recall that the version of $\mathcal{R}_0$ guessing ∞ is denoted by $\mathcal{R}_0^\infty$. While waiting for the $\mathcal{R}_0^\infty$ strategy to act, we will work on the assumption that it won't, and we will pursue the basic $\mathcal{R}_0$-strategy exactly as we discussed it in Section 5, as discussed above, with $\rho_{0,f,s}$ taking the role of ρ there. This is called the $\mathcal{R}_0^f$ strategy. That is, whilst we don't see a 0-split, we would either extend $\rho_{0,f,s+1}{}^\frown 0$ infinitely often, where $\mathcal{R}_0^f$ has the ∞ outcome, or from some point onwards we extend some τ extending $\rho_{0,f,s+1}{}^\frown 1$; this all assumes that this is the true version with guess f about $\mathcal{N}_0$. We remark that in the second case, we will also protect τ-while it appears in $S_{0,t}$-from removal from the left hand tree, by keeping its image in *all* the right hand trees (modulo priority) as discussed below in more detail in Remark 7.1 below.

REMARK 7.1. In the construction, we will have defined $\Gamma(\tau) = \kappa$ for some κ on $T_{0,t}$, or possibly $\Gamma(\upsilon) = \hat{\kappa}$ for some $\upsilon \succeq \tau$ and $\hat{\kappa} \succeq \kappa$. For the discussion below we regard κ and $\hat{\kappa}$ as the same. Anticipating things somewhat (to aid in the meeting of $\mathcal{R}_0$) whilst τ remains good, we would not like lower priority $\mathcal{N}_q$ removing κ from any of the trees T_q. The $\mathcal{N}_q$ requirements do this removal via the q-state machinery.[13] So we ensure that all such trees contain this κ and only work to raise the q-states for extensions of κ. We would do this by redefining their Tests, described below, to extend κ. In the construction, we will do this by initializing all the relevant parts of the trees $T_{q,\upsilon}$ each time we play $\mathcal{R}_0$ with a new τ.

Back to the construction, we consider the version of $\mathcal{R}_0$ guessing ∞ for $\mathcal{N}_0$, $\mathcal{R}_0^\infty$. We define the parameter Test(∞, s) to be λ, the empty string, in $T_{0,s}$. Now, what the version of $\mathcal{R}_0$ guessing ∞ is waiting for,

- is to see some 0-split of in $T_{0,s}$ *before defining* Γ.

This happens in two steps as we now discuss in detail.

First, we see an n where $\Phi_0^{\nu_0}(n)$ and $\Phi_0^{\nu_1}(n)$ 0-split λ for some ν_0, ν_1. In this case, we refine the $T_{0,s}$-tree to make $T_{0,s+1}$, with $T_{0,s+1}(j) = \nu_j$, and otherwise leave $T_{0,s}$ unchanged. That is, nothing happens, except we re-define $T_{0,s+1}(j\xi) = \nu_j\xi$. (This formula works because $T_{0,s}$ is initially $2^{<\omega}$. If we wrote this with an eye

[12] In the construction many such immediate extension strings are labelled ξ_i for $i \in \{0, 1\}$ and we will always be meaning that ξ_0 is left of ξ_1. We will adopt this convention so as not to clutter the construction. Here $\eta_0 <_L \eta_1$. The idea is that the leftmost path (visited infinitely often) on the right construction will be mapped by Γ to the leftmost path in the right; and this correlates to the leftmost path of the priority tree, i.e., the true path, TP.

[13] It is *only* the $\mathcal{N}$-requirements which will remove strings from the range of Γ, and consequently the pre-images from the right hand side.

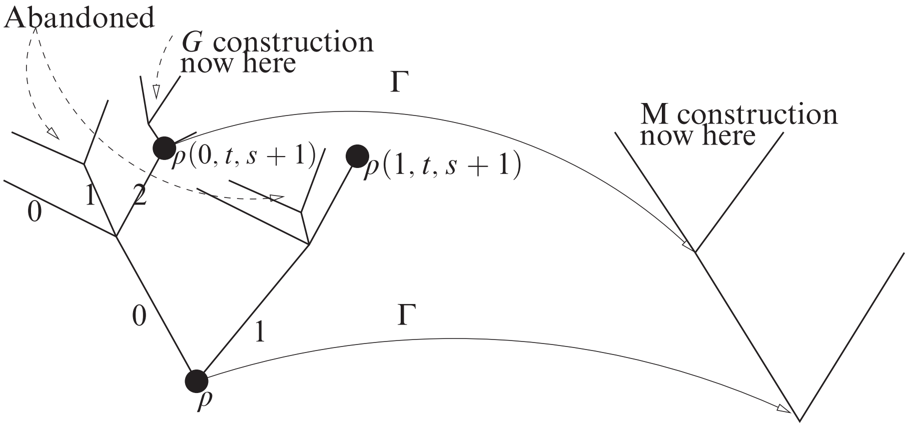

FIGURE 4. Abandonment of paths.

towards the inductive strategies, the formula would be $T_{0,s+1}(j\xi) = T_{0,s}(\zeta_j\xi)$ where $T_{0,s}(\zeta_j) = \nu_j$.)

At this stage, we will *not yet play* $\mathcal{R}_0^\infty$, *as our* fundamental guiding principle *is that we only allow* Γ *to be mapped by this strategy to strings in the high state in* $T_{0,s}$ *and we don't yet have* proof *that either of the* $T_0(j)$ *are in the high state.* Thus, our only actions are to

- Define $\Gamma(\rho, s+1) = \lambda$ (as we know λ now has the high state).
- Initialize $\rho_{i,f,s}$ and define $\rho_{i,f,s+1} = \rho\widehat{\ }0\widehat{\ }\langle 2\rangle$. (If this was a general step of the construction, this formula would read as $\rho_{i,f,s+1} = \rho\widehat{\ }i\widehat{\ }\langle n+1\rangle$ where $\rho_{i,f,s} = \rho\widehat{\ }i\widehat{\ }\langle n\rangle$.)
- Set $\text{Test}(\infty, s+1) = \langle 0\rangle$. (Now we are testing to see if $\langle 0\rangle$ (i.e., $T_0(\langle 0\rangle)$) 0-splits in T_0.)
- Give $\rho_{i,f,s+1}$ for $i \in \{0,1\}$, two length 1 extensions, $\rho_{i,f,s+1}\widehat{\ }0$ and $\rho_{i,f,s+1}\widehat{\ }1$, each mapped to incomparable strings in $T_{0,s}$ extending ν_i. (Hence, in particular, the extensions of $\rho_{0,f,s+1}\widehat{\ }0$ extend $\text{Test}(\infty, s+1)$.)

Note that $[\rho_{i,f,s}]$ for $i = 0, 1$ are now both abandoned forever. In particular, neither $\mathcal{R}_0^\infty$ nor $\mathcal{R}_0^f$ will ever again seek witnesses there. (See Figure 4.)

The second step in the strategy is similar, but in this step we will really only deal with $[\rho\widehat{\ }0]$, until it is resolved. While we await a further ∞ "confirmation" by $\mathcal{N}_0$, using the new $\text{Test}(\infty, s+1)$, we will continue our construction in $[\rho_{0,f,s+1}]$, as in the basic module of Section 5 and as above with the $\mathcal{R}_0^f$ strategy.

We pursue the $\mathcal{R}_0^f$ strategy with base $\rho_{0,f,s+1}$, until we see a stage where we find in $T_{0,t}$ a 0-split of $\text{Test}(\infty, s+1)$ at some stage $t \geq s+1$. Should no 0-split of $\text{Test}(\infty, s+1)$ occur, then the left construction will be carried out in the cone $[\rho_{0,f,s+1}]$, and right construction will be carried out in the cone $[\text{Test}(\infty, s+1)]$; that is, in $[\nu_0]$. In this case, again we have globally met $\mathcal{N}_0$ as we have proof that Φ_0^M is not total or Φ_0^M is not computable.

Finally, should a 0-split ζ_0, ζ_1 of $\text{Test}(\infty, s+1)$ be found in $[\text{Test}(\infty, s+1)]$ at some state t, we

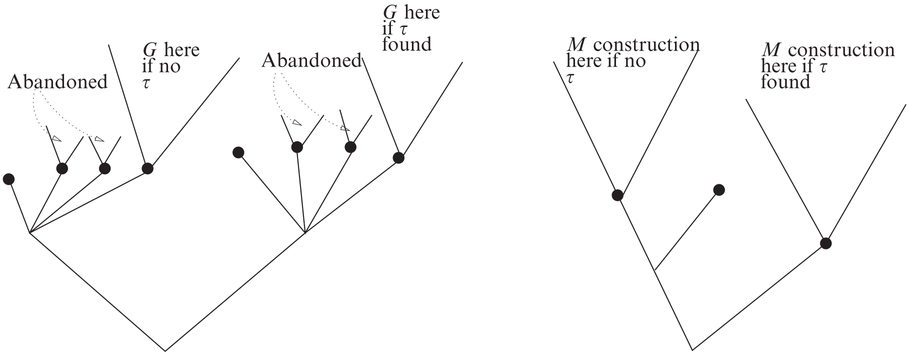

FIGURE 5. Redefining Test $(\infty, t+1)$.

- Refine $T_{0,t+1}$ using this 0-split above v_0. (i.e., $T_{0,t+1}(0i) = \zeta_0$, for $i \in \{0,1\}$, etc.)
- Define $\Gamma(\rho\widehat{\ }0\widehat{\ }0, t+1) = v_0$ (as we know v_0 now has the high state).
- For $i \in \{0,1\}$, initialize $\rho_{i,f,t}$ and define $\rho_{i,f,t+1} = \rho\widehat{\ }0\widehat{\ }\langle 3\rangle$.[14] (If this was a general step of the construction, this formula would read as $\rho_{i,f,t+1} = \rho\widehat{\ }i\widehat{\ }\langle n+1\rangle$ where $\rho_{i,f,t} = \rho\widehat{\ }i\widehat{\ }\langle n\rangle$.)
- Get ready to redefine Test$(\infty, t+1)$. This is slightly more complex than the first case and is described below.
- Give $\rho_{i,f,t+1}$ for $i \in \{0,1\}$, two length 1 extensions, $\rho_{i,f,t+1}\widehat{\ }0$ and $\rho_{i,f,t+1}\widehat{\ }1$, each mapped to incomparable strings in $T_{0,t+1}$ extending v_i, (e.g., ζ_0, ζ_1 for v_0).

7.1. Redefining Test$(\infty, t+1)$**.** The re-definition of Test$(\infty, t+1)$ is slightly more complex. We first look to see if we should switch to trying to meet $\mathcal{R}_0$ via some $\tau \in S_{0,t}$ extending $\rho\widehat{\ }1\widehat{\ }0$ on the left hand side.

CASE 1. If there is no such τ, there is no reason to leave our current location, so we simply set Test$(\infty, t+1) = \zeta_0$, and repeat the above inductively.

See Figure 5 below.

CASE 2. The other possibility is that we see some such candidate string τ. We would *like* to use this string to try to meet $\mathcal{R}_0^\infty$, but this necessitates a proof that τ can or will be mapped to something on the right hand side in the high 0-state. (*By the fundamental principle.*) Before we can put τ into the domain of Γ above $\rho\widehat{\ }1\widehat{\ }0$, we force this to happen as above on a potential image on the right hand side.

To wit: In Case 2, we will define Test$(\infty, t+1) = v_1$. The construction will then proceed in the cone $[\rho_{1,f,t+1}] = [\rho_{1,f,s+1}]$ on the left hand side (guessing that no 0-split of v_1 is found). If a split α_0, α_1, is found at stage v, akin to the above, we will

[14] Actually, there is no reason in this step to initialize $\rho_{1,f,t}$. That is because we will either stay in $[\rho\widehat{\ }0]$, as we see below, or move to $[\rho\widehat{\ }1]$ for the first time; and we have not yet proven that v_1 has a 0-split in T_0. However, in the construction, we can regard each time we verify that the ∞ outcome for $\mathcal{N}_0$ looks correct, all strategies $\mathcal{R}_e$ guessing that it has the f-outcome will be initialized. This make the presentation of the construction smoother. All we have to say is "initialize strategies right of TP_s" to indicate this action.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

- Refine $T_{0,v+1}$ to have this 0-split above ν_1.
- Define $\Gamma(\rho^\frown 1^\frown 0) = \nu_1$.
- Initialize $\rho_{1,f,v}$ and make define $\rho_{1,f,v+1}$ the next string right, as before for 0, and map two length 1 extensions to direct extensions α_i of ν_1.

Now, we need to make a decision. Can we still work to win $\mathcal{R}_0$ here? Certainly, we need to see that τ has remained in $S_{0,v}$ since for all stages between $t+1$ and the current one v. If the answer is yes, then in this initial attack, we simply ask that the construction now be carried out in the cone $[\tau]$ on the left hand side. In this case we will define $\Gamma(\tau) = \nu_1$. And in this case we'd have $\text{Test}(\infty, v+1) = \alpha_0$.

We remark that in subsequent attacks, later in the construction, we might already have a definition of $\Gamma(\hat{\tau}) = \kappa$ for some longest $\hat{\tau} \preceq \tau$, or $\Gamma(\alpha)$ already defined for some α extending τ.

In the former case, the construction will have ensured that κ already has 0-state ∞ in $T_{0,v}$. Thus, we will also be safe to map $\Gamma(\tau) = \kappa$.

In the latter case $\Gamma(\alpha)$ will already have 0-state ∞ in $T_{0,v}$, and there is nothing to do.

Finally, if τ is no longer in $S_{0,v}$ or has entered and left, we move the left construction back to the cone $[\rho_{0,f,t+1}]$, and now make $\text{Test}(\infty, v+1) = \zeta_0$.

In the case that we found τ the construction on the right hand side $\mathcal{R}_0^\infty$ will continue to either

(i) work in $[\tau]$ each time the $\text{Test}(\infty, p)$ returns a new relevant 0-split, or
(ii) will eventually get stuck on some cone $[\rho_{1,f,p}]$ mapping to strings in some cone $[\text{Test}(\infty, p)]$ on the right hand side in T_0 (which is a subcone of $[\nu_1]$), or
(iii) discover that $\tau \notin S_{0,p}$ for some larger p.

In Case (i), $\mathcal{R}_0^\infty$ has outcome f and wishes to remain in $[\tau]$. However, the rules of engagement are that in this cone, *only* strings ξ with 0-state ∞ in T_0 can be of the form $\Gamma(\alpha) = \xi$. The construction will, of course, be making an infinite extension of τ, so part of the construction is to wait for more and more such ξ to occur. While we wait for such ξ, while τ remains good, so that (iii) is not invoked, we will either be in $[\tau]$ at a ∞-stage for $\mathcal{N}_0$, or we will be working in some $[\rho_{1,f,u}]$ which is cancelled each time we move above $[\tau]$. We then move back to $[\rho_{1,f,u+1}]$ after playing above τ with a new $\text{Test}(\infty, u+1)$ above ν_1.

In the Case (iii), as above we will move the left construction back to the cone $[\rho_{0,f,p+1}]$, and now make $\text{Test}(\infty, p+1) = \zeta_0$.

See Figure 5 for the situation at this point, where we are about to consider our alternatives for $\rho_\infty{}^\frown 1$.

The above is a two-step process on each side. That is because we will initially have to verify the base λ on $T_{0,s}$. Once this is done the verification process-that the true outcome of $\mathcal{N}_0$ is ∞-will only need one step; verifying that $\text{Test}(\infty, s)$ 0-splits on $T_{-1,s}$.

7.2. Summary. First we might get stuck on some $\mathcal{R}_0^f$ strategy, and this will only happen if $\mathcal{N}_0$ has the f-outcome, and we are stuck on the left hand side in some cone $[\rho_{i,f,s}]$ from some point onwards. In this case, $G \succ \rho_{i,f,s}{}^\frown j$ for some $j \in \{0, 1\}$. In the case $j = 1$ there is some $\tau \in S_e$ with $\rho_{i,f,s}{}^\frown 1 \preceq \tau \prec G$, and Γ^τ has 0-state f in T_0. In the case that $j = 0$, there is no $\tau \in S_e$ extending $\rho_{i,f,s}{}^\frown 1$, so S_e is not dense.

If we don't get stuck on some $\mathcal{R}_0^f$ strategy, then the $\mathcal{R}_0^\infty$ strategy is correct. In this case, the first possibility is that we only play to try to extend $\rho^\frown 1$ finitely often. The first possibility for this case is that from some point onwards, there is some fixed $\tau \in S_{e,s}$ extending $\rho^\frown 1^\frown 0$, and $G \succ \tau$. Then we will infinitely often alternate between working above $\rho^\frown 1^\frown 0$ in $[\tau]$ and working above $[\rho_{1,f,u}]$. The $\{\rho_{1,f,u} \mid u \in \omega\}$ have no limit.

The other possibility for this case is that no stable τ is found. Thus Case (iii) is invoked infinitely often. In this case, $G \succ \rho^\frown 0^\frown 0$, and there is no $\tau \in S_e$ extending $\rho^\frown 1^\frown 0$.

The remaining case is that we play in $[\rho^\frown 0]$ infinitely often, and in this case there is no $\tau \in S_e$ extending $\rho^\frown 1^\frown 0$.

7.3. More requirements. The only remaining details we need for more strategies is the discussion of how we allow for the inclusion of more trees etc.

$\mathcal{R}_0$ decides how we work with T_1. If the true version of $\mathcal{R}_0$ is $\mathcal{R}_0^f$ then for some i, we will eventually get stuck on some $\rho_{i,f,s}$ forever, for $s \geq s_0$. Then Test(∞, s_0) never gets 0-state ∞ on $T_{0,s}$, and Test$(\infty, s_0) =$ Test(∞, s), $s \geq s_0$. If the true outcome of $\mathcal{R}_0^f$ is ∞, so that we return to $\rho_{i,f,s_0}{}^\frown 0$ infinitely often (including from some point onwards), then we would be free to try to meet $\mathcal{R}_1$, by declaring its version of ρ, ρ^1 as $\rho_{i,f,s_0}{}^\frown 0$. This version of $\mathcal{R}_1$ has guess f about $\mathcal{N}_0$. But it should also have a guess about $\mathcal{N}_1$. What we will do is to define Test$(f\infty, s_0)$ =Test(∞, s_0). This $\mathcal{N}_1$ strategy is attempting to refine $T_{1,u}$ for $u \geq s_0$ to state $f\infty$ by looking for 1-splits in T_0 above Test$(f\infty, s_0)$. This strategy will refine the T_1-tree within the T_0-tree. Assuming that $\rho_{i,f,s_0}{}^\frown 0$ is the final location for the $\mathcal{R}_1$-strategy, the strategy works in exactly the same way as we did for $\mathcal{R}_0$, within this cone.

Test$(f\infty, s)$ might change infinitely often, but it will always extend Test(∞, s_0). The $\mathcal{R}_1$-strategy guessing this follows the same naming pattern and is called the $\mathcal{R}_1^{f\infty}$-strategy, meaning that it is guessing f for $\mathcal{N}_0$ and ∞ for $\mathcal{R}_0$.

The other possibility in the case that Test$(f\infty, s_0)$ has reached its limit at some τ extending $\rho_{i,f,s_0}{}^\frown 1$ and the true outcome of $\mathcal{R}_0^f$ is f. In this case all of the above is the same, except that we would try to meet $\mathcal{R}_1$, by declaring its version of ρ, ρ^1 as τ (which extends $\rho_{1,f,s_0}{}^\frown 1$), and declare that Test$(f\infty, s_0) = \Gamma(\tau)$. This latter condition is to make sure that we don't remove τ from the left hand side by raising the 1-state of something on the right. We will call this the $\mathcal{R}_1^{ff}$ strategy, meaning that it is guessing f for $\mathcal{N}_0$ and f for $\mathcal{R}_0$.

The other strategies for $\mathcal{R}_1$ and $\mathcal{N}_1$ are entirely similar. If they guess the infinite outcome for $\mathcal{N}_0$, then they live in one of the cones provided by the $\mathcal{R}_0^\infty$ strategy. The $\mathcal{R}_1^{\infty\infty}$ strategy will work in $\rho^\frown 0^\frown 0$ and makes ρ^1 equal to that. $\mathcal{N}_0$ tries to raise 1-states inside $[\Gamma(\rho^\frown 0^\frown 0)]$ inside of T_1. It seeks 1-splits which were already in 0-state ∞ in T_0. Test$(\infty\infty, s) \preceq$ Test(∞, s) at every stage s. As above, precisely where this test starts from depends on the outcome of $\mathcal{R}_0^\infty$. The outcome ∞ allows us to use $\Gamma(\rho^\frown 0^\frown 0)$ as Test$(\infty\infty, s)$, whereas the outcome f would again need $\Gamma(\tau)$ as above.

One subtle point is that there is no reason that the same 1-state will appear above $\Gamma(\rho^\frown 0^\frown i)$ for both $i \in \{0, 1\}$. We could have included the outcome of $\mathcal{R}_0$ as part of the 1-states but this adds even more notation.

The rest simply works inductively. We now turn to some details.

7.4. Construction. The construction proceeds in substeps where we generate a string $TP_{s+1} \in \{\infty, f\}$ the apparent true path at stage $s+1$, which gradually gets longer with s.

The construction works more or less precisely as described above. Beginning at λ in the priority tree PT, we will see if $\text{Test}(\infty, s)$ returns a 0-split on $T_{0,s}$. If so, then s is an ∞-stage, and otherwise it is an f-stage. In the first case we invoke strategy $\mathcal{R}_0^\infty$ and otherwise $\mathcal{R}_0^f$, as described above.

More generally, at substep $t \leq s$ we will have generated TP_{s+1}^t which is a string in $\{\infty, f\}^{t+1}$. The even bits will correspond to g-states on trees $T_{g,s}$ for $2g \leq t$. The odd bits will be the current state of the $\mathcal{R}_k^\sigma$-strategy where σ is the initial segment of TP_{s+1}^t of length $2k+1$.

We first suppose that $t = 2e > 0$. Let α be the string of length $e-1$ consisting of the first $e-1$ even bits of TP_{s+1}^t. $\text{Test}(\alpha\infty, s)^t$ (i.e., the version at substage t) will have been determined by the previous substage $t-1$. See if $T_{e,s}$ contains two e-splitting extensions of $e-1$-state α. If so, then refine the tree $T_{e,s+1}$ to have these two splits, and give $\text{Test}(\alpha\infty, s)^t$ state $\alpha\infty$. Then we will say $s+1$ is an $TP_{s+1}^t\infty$-stage. The determination of what $\text{Test}(\alpha\infty, s+1)$ will be will be decided by the next substage.

If there is no such e-split, then we will say $s+1$ is an $TP_{s+1}^t f$-stage. $\text{Test}(\alpha\infty, s+1) = \text{Test}(\alpha\infty, s)$.

Suppose that $t = 2e+1$. Let α be the string of even bits of TP_{s+1}^t of length $e-1$.

Case A. Suppose that substage t says that $s+1$ is a $\beta =_{\text{def}} TP_{s+1}^{t-1} f$-stage. Also we suppose that we have built G_{s+1}^t. We invoke the $\mathcal{R}_e^\beta$ strategy.

(i) If there is no string $\rho_{\alpha f,s,t}$ already defined, then let this be G_{s+1}^t, give this two length 1 extensions $\rho_{\alpha f,s,t}\frown i$ for $i \in \{0,1\}$, and map them to two strings immediately extending $\Gamma(G_{s+1}^t)$ in $T_{e,s}$ which we claim will have e-state αf.

(ii) If $\rho_{\alpha f,s,t}$ is currently defined, we claim that $G_{s+1}^t \preceq \rho_{\alpha f,s,t}$. We will play the $\mathcal{R}_e^\beta$ strategy, as with the basic module. That is, we seek some extension $\tau \in S_{e,s}$ of $\rho_{\alpha f,s,t}\frown 1$.

(iia) If no such τ is found, then we will define $TP_{s+1}^{t+1} = \beta\infty$ and $G_{s+1}^{t+1} = \rho_{\alpha f,s,t}\frown 0$. $\text{Test}(\alpha f\infty, s+1) = \text{Test}(\alpha f\infty, u)$, where u is the most recent $\beta\infty$-stage.

(iib) If τ is found, we will set $G_{s+1}^{t+1} = \tau$, and define $\Gamma(\tau) = \Gamma(\hat{\tau}) = \kappa$ where $\hat{\tau}$ is the longest substring of τ with $\Gamma(\hat{\tau}) \downarrow [s]$, if $\Gamma(\upsilon)$ is not yet defined for some $\upsilon \succeq \tau$. In the latter case we need to nothing for $\Gamma(\tau)$, and use $\hat{\kappa} = \Gamma(\upsilon)$ in the below. We claim that κ (or $\hat{\kappa}$) will have e-state αf on $T_{e,s}$.

Subcase 1. The first subcase is that that $\tau \in S_{e,s}$ is unchanged since the last βf-stage. In this subcase, set $\text{Test}(\alpha f\infty, s+1) = \text{Test}(\alpha f\infty, v)$.

Subcase 2. $\tau \in S_{e,s}$ is new and there has been a previous βf-stage, or there has not been a previous βf-stage.

Set $\text{Test}(\alpha f\infty, s+1) = \kappa$. (or $\hat{\kappa}$.)

In either subcase, declare that $s+1$ is a βf-stage. In Subcase 2, this ends the stage, otherwise we move on to the next substage.

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

Case B. Suppose that substage t says that $s+1$ is a $\delta =_{\text{def}} TP^{t-1}_{s+1}\infty$-stage. Also we suppose that we have built G^t_{s+1}. We invoke the $\mathcal{R}^\delta_e$ strategy. Again, this is entirely analogous to the Basic Module.

Rather than writing out many subcases we will describe the $\mathcal{R}^\delta_e$ strategy.

If there is no string $\rho = \rho_{\alpha\infty,s,t}$ already defined, then let this be G^t_{s+1}, give this four length 2 extensions $\rho^\frown i^\frown k$ for $i,k \in \{0,1\}$.

If this was the first time, we have $\text{Test}(\alpha\infty, s+1) = \Gamma(G^t_{s+1})$. This completes the stage.

Then we test to see whether we can safely define $\Gamma(\rho^\frown 0^\frown 0)$ to a string in the high e-state $\alpha\infty$. Note that at the next v we visit ρ (assuming it is not initialized) it will necessarily be a α-stage.

If this is an αf-stage, then we pursue Case A, using $\rho_{0,\alpha\infty f,v} = \rho^\frown 0^\frown 1$, and defining $\text{Test}(\alpha f\infty, v) = \Gamma(\rho^\frown 0^\frown 1)$.

We pursue Case A each time we visit ρ, until we see a new e-split. We will do this with the two step process, and hence need e-splits until we will be safe to define $\Gamma(\rho_{0,\alpha\infty f,v}) = v_0$ where this is some immediate extension of $\text{Test}(\alpha\infty, s+1)$ in $T_{t,v}$.

If we reach a stage q we finally do this, we need to decide whether to try to move above $\rho^\frown 1^\frown 0$, according to whether we see some $\tau \in S_{e,s}$ extending $\rho^\frown 1^\frown 0$. If we do, we pursue the analogous strategy in the cone $[\rho^\frown 1]$. If not we define $\text{Test}(\alpha\infty\infty, q+1) = \text{Test}(\alpha\infty, q+1) = v_0$. Then $s+1$ would be a $TP^t_{q+1}\infty$-stage.

The remainder of the $\mathcal{R}^\delta_e$ strategy is entirely analogous, and leads to no further insight.

At the end of the stage, initialize all work based on guesses right of TP_s.

7.5. End of construction. Now we verify the construction.

Let TP be the true path of the construction. That is, the leftmost path visited infinitely often.

Lemma 7.2. *TP exists.*

Proof. There is nothing to prove for length 1 since the tree is finitely branching, and one of ∞ or f will be on TP. Inductively, suppose that $\beta \prec TP_s$ is leftmost visited infinitely often. Let α be the set of even positions of β.

The only time we don't construct a length s string for TP_s is when we deal with the $\mathcal{R}^\beta_e$-strategies. Thus, let $\beta \preceq TP$ and let s_0 be a stage after which we are never left of β.

When we next visit β we will erect ρ's for $\mathcal{R}^{\beta\infty}_e$ and $\mathcal{R}^{\beta f}_e$. The ones for $\beta\infty$ are never initialized and the ones for βf are initialized each time we visit $\beta\infty$. One of $\beta\infty$ or βf are therefore visited infinitely often, and in the latter case there will be a final version of $\rho_{i,\beta f,s}$ for one of $i \in \{0,1\}$.

Because the construction only delays extending the relevant node until the e-state of the guess is verified for the image to be defined, and this delay can happen at most 4 times once we have a stable ρ, we conclude that for almost all stages where the appropriate guess looks correct and we visit ρ (in whatever the correct case is), we will play a proper extension of it. Hence, TP is infinite. $\dashv$

The remainder of the verification is more or less along the lines of the discussions of the Basic Modules.

First we note that the construction maps strings in the leftmost part of the right construction to the same in the left one.

Let $\beta \prec TP$, and let α be the string of even positions in β.

If β and has odd length $2e+1$, then $\mathcal{R}_e^\beta$ has one of two outcomes, ∞ or f. Depending on what α is there is a final ρ which is visited infinitely often at α-stages from some point onwards. Then one of either $\rho\widehat{\ }0\widehat{\ }0$ or $\rho\widehat{\ }1\widehat{\ }0$ will meet $\mathcal{R}_e^\beta$ in the base that $\alpha = \alpha\infty$, or $\rho\widehat{\ }0, \rho\widehat{\ }1$ in the other case. Which it is, is determined exactly as in the Basic Module.

Inductively we can conclude that $\mathcal{R}_e^\beta$ meets $\mathcal{R}_e$, and maps extensions of whichever length 1 or 2 extension meets it to strings of state α in T_e. In the case of an f outcome, there will be a fixed string τ extending one of $\rho\widehat{\ }1$, $\rho\widehat{\ }1\widehat{\ }0$ (depending on what α is) and its e-state in T_e is α and is protected from raising its $e+1$-state.

Finally, if β has length $2e$, inductively it will be building a tree within T_{e-1}. First suppose that $\beta = \delta f$. Let ρ denote the final ρ of $\mathcal{R}_{e-1}^\delta$. Without loss of generality, we suppose that $\rho\widehat{\ }1$ is the final string, the case $\rho\widehat{\ }0\widehat{\ }1$ entirely analogous. Let τ be the witness string. The Test(α, s) will have been set to be $\Gamma^\tau = \kappa$ on T_{e-1}, and this will not be removed from T_{e-1}.

Thereafter, no extension of κ can ever be removed from $T_{j,s}$ for any $j \leq e-1$. By induction, there is a partial computable sub-tree of $e-1$-state γ within T_{e-1} where $\gamma q = \beta$ for some $q \in \{f, \infty\}$. If this q was f, since this is inductively the *final* τ, no further e-splits of κ can be found on T_{e-1} and hence T_e and T_{e-1} agree on extensions of κ. The final e-state will be α, and we conclude that Φ_e^M lies in a cone in T_e with no e-splits. Therefore Φ_e^M is either partial or computable. The case that $q = \infty$ is analogously the one where T_e in this cone will be e-splitting, and we can invoke Spector's Lemma.

Finally, the case that $\beta = \delta\infty$ is entirely similar, using extensions with last bit 0.

This concludes the proof.

§8. Summary and open questions.

We have constructed a weakly 2-generic degree computing a minimal one.

Chong and Downey gave a characterization of provides a characterization of when a set is computable from a 1-generic degree.

Definition 8.1 (Chong and Downey[3, 5]).
- We say that a computably enumerable set of strings S is a *proper cover*[15] of a set X iff for all $\sigma \prec X$, there exists $\tau \in S$, such that $\sigma \preceq \tau$, *and* no $\sigma \in S$ is an initial segment of X.
- We say that X has a *tight cover* S if S is a proper cover and for all proper covers $\hat{S}$, $\exists \sigma \in S \exists \tau \in \hat{S}(\sigma \preceq \tau)$.

Theorem 8.2 (Chong and Downey-[3, 5]).
- *A set X is computable from a* 1-*generic set iff X has* no *tight cover.*
- *Moreover, there exists a procedure Φ such that for all sets X, if X has no tight cover, then there is a* 1-*generic $G \leq_T X''$ such that $\Phi^G = X$.*

This characterization had certain consequences.

[15] In the original paper this was called a Σ_1-*dense set of strings.*

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press

The first was the result by Kumabe [13] and independently Chong and Downey [5]: *There is a minimal degree below* $\mathbf{0}'$ *computable from a 1-generic degree below* $\mathbf{0}''$. Other consequences include:

THEOREM 8.3 (Chong and Downey [3]). *There is a minimal degree below* $\mathbf{0}'$ not *computable from a* 1*-generic degree.*

THEOREM 8.4 (Kurtz-thesis). *Almost no degree is computable from a* 1*-generic.*

Clearly Kurtz's result was first obtained by direct methods. Thus, the question arises whether there is a similar *local* characterization of when a set is computable from a weakly 2-generic. We guess that the answer would be something like

$$X \textit{ has no } \Sigma^0_2 \textit{ "tight proper dense cover".}$$

The first problem here is to figure out precisely what the condition "tight proper dense cover" actually means. Having done that the next problem is that the Chong-Downey proof (even when the slight error was corrected in McInerney's Thesis [15]) is already itself full approximation $\mathbf{0}''$ argument, and any analogue would be to extend the approximation argument above, as well. Thus, we guess that any extension would add another quantifier, making it a $\mathbf{0}'''$-full approximation argument; as the things that need approximating are very complex.

The other question which is implicit in our work is whether the theorem can be obtained using forcing techniques. In all likelihood, this would give a reduction $M \leq G$ with a reduction stronger than $\leq_T$, probably truth-table reductions. This leads to the obvious question about whether such strong reductions are possible.

Finally, we believe that the minimal degree $\deg(M)$ of our result can also be made to be of hyperimmune-free degree. This would entail the proof we have given combined with the methods of Downey [8] where degree of Cantor-Bendixson rank one is given. Such methods involve the construction of M traversing the relevant trees from left to right over and over again, each time verifying that computations converge. Probably this is possible, but the argument would be significantly more complex than the present one.

§9. **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. Nandakumar's research was partly supported by DST SERB grant SB/FTP/ETA-0249/2013.

REFERENCES

[1] G. BARMPALIAS, A. R. DAY, and A. E. M. LEWIS-PYE, *The typical Turing degree*. ***Proceedings of the London Mathematical Society. Third Series***, vol. 109 (2014), no. 1, pp. 1–39.

[2] G. BARMPALIAS and A. LEWIS-PYE, *The information content of typical reals*, ***Turing's Revolution*** (G. Summerauga and T. Strahm, editors), Birkhäuser/Springer, Cham, 2015, pp. 207–224.

[3] C. T. CHONG and R. G. DOWNEY, *Minimal degrees recursive in 1-generic degrees*. ***Annals of Pure and Applied Logic***, vol. 48 (1990), pp. 215–225.

[4] C. T. CHONG and C. G. JOCKUSCH, *Minimal degrees and* 1*-generic sets below* $\mathbf{0}'$, ***Computation and Proof Theory (Aachen, 1983)***, Lecture Notes in Mathematics, vol. 1104, Springer, Berlin, 1984, pp. 63–77.

[5] C. T. Chong and R. G. Downey, *Degrees bounding minimal degrees*. ***Mathematical Proceedings of the Cambridge Philosophical Society***, vol. 105 (1989), pp. 211–222.

[6] S. B. Cooper, *Minimal degrees and the jump operator*, this Journal, vol. 38 (1973), pp. 249–271.

[7] R. Downey and L. Yu, *Arithmetical Sacks forcing*. ***Archive for Mathematical Logic***, vol. 45 (2006), no. 6, pp. 715–720.

[8] R. G. Downey, *On* π^0_1 *classes and their ranked points*. ***Notre Dame Journal of Formal Logic***, vol. 32 (1991), pp. 499–512.

[9] R. G. Downey and D. Hirschfeldt, ***Algorithmic Randomness and Complexity***, Springer, New York, 2010.

[10] R. M. Friedberg, *Three theorems on recursive enumeration. I. Decomposition. II. Maximal set. III. Enumeration without duplication*, this Journal, vol. 23 (1958), pp. 309–316.

[11] C. A. Haught, *The degrees below a* 1*-generic degree* $<\mathbf{0}'$, this Journal, vol. 51 (1986), no. 3, pp. 770–777.

[12] C. Jokusch, *Degrees of generic sets*, ***Recursion Theory: Its Generalizations and Applications*** (F. R. Drake and S. S. Wainer, editors), Cambridge University Press, New York, 1980, pp. 110–139.

[13] M. Kumabe, *A 1-generic degree which bounds a minimal degree*, this Journal, vol. 55 (1990), pp. 733–743.

[14] S. A. Kurtz, ***Randomness and genericity in the degrees of unsolvability***, Ph.D. thesis, University of Illinois at Urbana-Champaign, ProQuest LLC, Ann Arbor, MI, 1981.

[15] M. McInerney, ***Topics in algorithmic randomness and computability theory***, Ph.D. thesis, Victoria University Wellington, New Zealand, 2016.

[16] J. R. Shoenfield, ***Degrees of Unsolvability***, New York, 1971, North-Holland Mathematics Studies, No. 2.

[17] R. I. Soare, ***Recursively Enumerable Sets and Degrees***, Springer, Berlin, Heidelberg, 1987.

[18] C. Spector, *On degrees of recursive unsolvability*. ***Annals of Mathematics. Second Series***, vol. 64 (1956), pp. 581–592.

[19] C. E. M. Yates, *Initial segments of the degrees of unsolvability Part II: Minimal degrees*, this Journal, vol. 35 (1970), pp. 243–266.

DEPARTMENT OF MATHEMATICS AND STATISTICS
VICTORIA UNIVERSITY WELLINGTON
P. O. BOX 600, WELLINGTON, NEW ZEALAND
E-mail: rod.downey@vuw.ac.nz

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY KANPUR
KANPUR 208016, UTTAR PRADESH, INDIA
E-mail: satyadev@cse.iitk.ac.in

https://doi.org/10.1017/jsl.2018.68 Published online by Cambridge University Press