DIAGONALLY NON-COMPUTABLE FUNCTIONS AND BI-IMMUNITY

CARL G. JOCKUSCH, JR. AND ANDREW E.M. LEWIS

Abstract. We prove that every diagonally nonrecursive function computes a set A which is bi-immune, meaning that neither A nor its complement has an infinite computably enumerable subset.

1. Introduction

A function $f : \omega \to \omega$ is called *diagonally nonrecursive* (or DNR, for short), if, for all e, $f(e) \neq \varphi_e(e)$, where $\{\varphi_e\}$ is the standard enumeration of the partial computable functions. Two basic methods of producing noncomputable functions are diagonalization (producing the DNR functions) and coin-tossing (producing, with probability 1, random sets, in various senses of "random"). It is natural to compare the computational power required to produce functions using these methods. The answer turns out to depend on whether we consider arbitrary functions in DNR, or only those taking values in $\{0, 1\}$. Specifically, every $\{0, 1\}$-valued DNR function computes a 1-random set (by [?], Theorem 8.4), and every 1-random set computes a DNR function ([?], Remark 10.2). Furthermore, these results are strict in the sense that there is a 1-random set which computes no $\{0, 1\}$-valued DNR function ([?], Theorem 10.4), and there is a DNR function which computes no 1-random set ([?], Theorem 10.4).

It would be remarkable if this intertwining of diagonalization and randomness could be extended by showing that **all** DNR functions compute sets satisfying some particular weak randomness property. A natural property to try first is weak 1-randomness, which is also known as Kurtz-randomness. Here we obtain a negative result by analyzing a proof from [?]:

Theorem 1.1. *There is a computably bounded DNR function which does not compute any Kurtz-random set.*

To obtain a positive result, we consider a yet weaker property, bi-immunity. A set A is *bi-immune* if neither A nor its complement $\overline{A}$ contains an infinite c.e. subset. We finally obtain a positive result, which is the main result of this paper:

Theorem 1.2. *Every DNR function computes a bi-immune set. This holds uniformly in the sense that there is a Turing functional Ψ such that Ψ^f is bi-immune for all DNR functions f.*

2010 *Mathematics Subject Classification.* 03D28.

Lewis was supported by a Royal Society University Research Fellowship. This research was partially done whilst Lewis was a visiting fellow at the Isaac Newton Institute for the Mathematical Sciences in the programme 'Semantics & Syntax'. This work was partially supported by Bulgarian National Science Fund under contract D002-258/18.12.08.

Corollary 1.3. *Every DNR function is Turing equivalent to a bi-immune set.*

The corollary follows at once from Theorem **??** and the upward closure of the degrees of bi-immune sets [**?**]. However, the latter result was proved in a highly nonuniform fashion, so we don't know whether the corollary holds in any uniform sense.

Theorem **??** is not the first positive result on the computational power of DNR functions. The Arslanov completeness criterion (see [**?**], Theorem V.5.1) implies that every c.e. set which can compute a DNR function has degree $\mathbf{0}'$. This result is extended in [**?**], Theorem 5.1, to show that every n-CEA set A which computes a DNR function has degree $\geq \mathbf{0}'$. Also, it was shown by A. Kučera ([**?**], Theorem VII.1.10), that every Δ^0_2 DNR function computes a non-computable c.e. set. These results were actually stated in terms of fixed-point free functions (those satisfying $(\forall e)[W_e \neq W_{f(e)}]$), rather than DNR functions, but it is easily seen ([**?**], Lemma 4.1) that every DNR function computes a fixed-point free function, and vice-versa.

In the other direction, it is known that DNR functions can be computationally weak. It follows from the low basis theorem that there are $\{0,1\}$-valued DNR functions of low degree. Also, it was shown by M. Kumabe and the second author [**?**] that there are computably bounded DNR functions of minimal degree.

Our results fit in naturally with a number of previous results on the relative complexity of various classes of functions related to diagonalization, randomness and bi-immunity. This complexity is best discussed in terms of strong (Medvedev) reducibility and weak (Muchnik) reducibility. Recall that if P and Q are subsets of Baire space ω^ω, we say that P is *weakly* (or Muchnik) reducible to Q (written $P \leq_w Q$) if for every function $f \in Q$ there is a function $g \in P$ such that g is Turing reducible to f. If this holds uniformly, i.e. there is a fixed Turing functional Ψ such that $\Psi^f \in P$ for all $f \in Q$, we say that P is *strongly* (or Medvedev) reducible to Q, written $P \leq_s Q$. For example, our main result states that the class of bi-immune sets is strongly reducible to DNR. See [**?**], for example, for further information on weak and strong reducibilities and [**?**] for further information on 1-randomness and Kurtz-randomness.

Let DNR_k be the class of DNR functions taking values in $\{0, 1, \ldots, k-1\}$, and let $\mathrm{DNR}_{\mathrm{REC}}$ be the class of DNR functions f such that there is a computable function g with $g(n) \geq f(n)$ for all n, i.e. the class of DNR functions which are *computably dominated.* Let BI be the class of bi-immune sets, let 1R be the class of all 1-random sets, and let KR be the class of all Kurtz-random sets, which are also known as weakly 1-random sets. Our results, together with previously known results, enable us to understand how weak and strong reducibility behave on the classes which have just been defined.

We carry this out first for weak reducibility. It is shown in [**?**], Theorem 5, that DNR_k is weakly equivalent to DNR_2 for all $k \geq 2$, so we need not consider DNR_k for $k > 2$. We then have the following strict chain which includes all classes under consideration except for KR:

$$\mathrm{DNR}_2 >_w \mathrm{1R} >_w \mathrm{DNR}_{\mathrm{REC}} >_w \mathrm{DNR} >_w \mathrm{BI}$$

See Theorem 10.4 of [**?**] for references to the proofs of the first three inequalities above. In particular, the work of Ambos-Spies, Kjos-Hanssen, Lempp, and Slaman [**?**] plays a major role here. For the final inequality DNR $>_w$ BI, of course our main result, Theorem **??**, implies that BI $\leq_w$ DNR. To see that DNR $\not\leq_w$ BI, consider a c.e. degree $\mathbf{a}$ such that $\mathbf{0} < \mathbf{a} < \mathbf{0}'$. Then $\mathbf{a}$ contains a bi-immune set A by [?], Theorem 5.2, but there is no A-computable DNR function by the Arslanov completeness criterion.

We continue to consider weak reducibility and now bring KR, the class of Kurtz-random sets, into the picture. We have a strict chain:

$$1\mathrm{R} >_w \mathrm{KR} >_w \mathrm{BI}$$

Here the reductions are obvious (using the identity functional), since $1\mathrm{R} \subseteq \mathrm{KR} \subseteq \mathrm{BI}$. To show that the first inequality is strict, it suffices, since $\mathrm{DNR} \leq_w 1\mathrm{R}$, to show that $\mathrm{DNR} \not\leq_w \mathrm{KR}$. To prove this, again let $\mathbf{a}$ be a c.e. degree such that $\mathbf{0} < \mathbf{a} < \mathbf{0}'$. Then $\mathbf{a}$ is hyperimmune by Dekker's Theorem (see Theorem V.2.5 of [?]) and hence contains a Kurtz-random set A by a result of Kurtz (see Corollary 8.11.8 of [?]). Again by the Arslanov completeness criterion, there is no A-computable DNR function since $\mathbf{a}$ is c.e. and $\mathbf{a} < \mathbf{0}'$. This shows that $\mathrm{DNR} \not\leq_w \mathrm{KR}$ and hence $1\mathrm{R} \not\leq_w \mathrm{KR}$. To see that the second inequality is strict, we need to show the existence of a bi-immune set which does not compute any Kurtz-random set. This follows from known results. First, S. Simpson ([?], Theorem 25) showed that there is a minimal, hyperimmune-free degree $\mathbf{a}$ which contains a bi-immune set A. (His proof used forcing with coinfinite computable conditions, and the corresponding generic sets have minimal degree by a theorem of Lachlan [?]. Alternatively, this result of Simpson follows immediately from Theorem **??** and Theorem **??**, although this approach seems less straightforward than the original.) Second, it was shown by Nies, Stephan, and Terwijn (see Theorem [?], Theorem 8.11.11) that every Kurtz-random set of hyperimmune-free degree is 1-random. Finally, no 1-random set is computable or of minimal degree by a theorem of Kurtz (see [?], Corollary 6.9.5). Hence, if $B \leq_T A$ is Kurtz-random, then B has hyperimmune-free degree, so B is a 1-random set which is computable or of minimal degree, a contradiction. Thus, A is a bi-immune set which computes no Kurtz-random set, and it follows that $\mathrm{KR} \not\leq_w \mathrm{BI}$.

To complete the picture for $\leq_w$, we show that KR is incomparable under $\leq_w$ with both $\mathrm{DNR}_{\mathrm{REC}}$ and DNR. We have already remarked that $\mathrm{DNR} \not\leq_w \mathrm{KR}$, and it follows that $\mathrm{DNR}_{\mathrm{REC}} \not\leq_w \mathrm{KR}$. By Theorem **??**, $\mathrm{KR} \not\leq_w \mathrm{DNR}_{\mathrm{REC}}$, and hence $\mathrm{KR} \not\leq_w \mathrm{DNR}$.

We now consider strong reducibility on the same classes. The picture is generally similar. Some of the reductions we mentioned above in discussing weak reducibility are actually strong reductions, as mentioned there. Negative results for weak reducibility carry over immediately to strong reducibility, since strong reducibility implies weak reducibility. However, there are some differences between strong and weak reducibility on the classes we are studying. The main one is that $\mathrm{DNR}_k \not\leq_s \mathrm{DNR}_{k+1}$ for all $k \geq 2$, as shown in [?], Theorem 6. Our main theorem (Theorem **??**) gives that $\mathrm{BI} \leq_s \mathrm{DNR}$. We thus have an infinite strict chain:

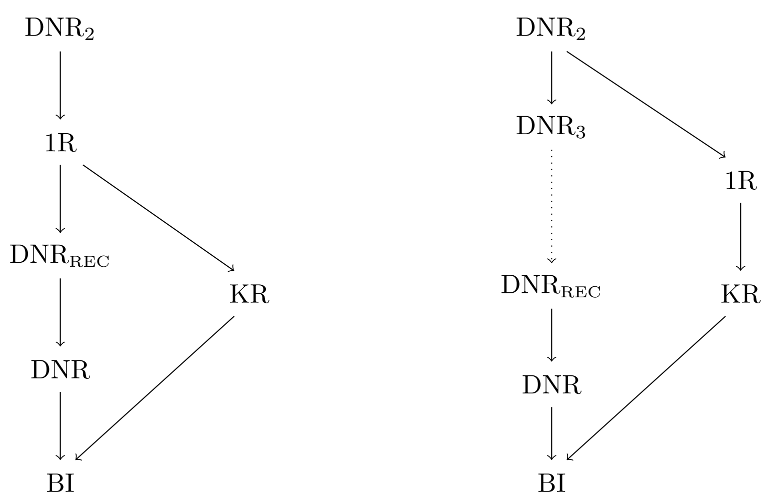
$$\mathrm{DNR}_2 >_s \mathrm{DNR}_3 >_s \cdots >_s \mathrm{DNR}_{\mathrm{REC}} >_s \mathrm{DNR} >_s \mathrm{BI}$$

By [?], Corollary 8.4, and the effective universality of DNR_2 for nonempty Π^0_1 classes $P \subseteq 2^\omega$, we have $1\mathrm{R} \leq_s \mathrm{DNR}_2$. Then by previous remarks we have another strict chain:

$$\mathrm{DNR}_2 >_s 1\mathrm{R} >_s \mathrm{KR} >_s \mathrm{BI}$$

Except for the top and bottom elements (which coincide) all of the elements of the first chain are incomparable with all of the elements of the second chain. To see this, it suffices to show that KR $\not\leq_s$ DNR_3 and DNR $\not\leq_s$ 1R. The former result is Theorem 5.4 of [**?**], which is proved from a Ramseyan result on edge-labeled ternary trees. It is an elementary exercise that DNR $\not\leq_s$ 1R using that 1R is topologically dense and DNR has no computable element. We omit the proof.

The above discussion gives a complete description of weak and strong reducibility on the classes DNR_k, $\mathrm{DNR}_{\mathrm{REC}}$, DNR, 1-random, KR, and BI. This information is summarized in the following two diagrams:



Weak reducibility Strong reducibility

FIGURE 1. Strong and weak reducibility

2. NOTATION AND TERMINOLOGY

We use the variables e, i, j, k, n, m, x to range over ω; the variables f and g to range over functions $\omega \to \omega$; we use h and T to range over functions $\omega^{<\omega} \to \omega^{<\omega}$; $\mathcal{T}$ to range over subsets of $\omega^{<\omega}$; $\alpha, \beta, \gamma, \sigma, \tau$ to range over $\omega^{<\omega}$. We use the variables Ψ and Φ to range over Turing functionals. Also, $|\sigma|$ denotes the length of σ. We write $\sigma^\frown\tau$ to denote the concatenation of σ and τ, and for $i \in \omega$ we often identify i with τ of length 1 such that $\tau(0) = i$. Thus we may write $\sigma^\frown i$ to denote $\sigma^\frown\tau$ such that $|\tau| = 1$ and $\tau(0) = i$. A string σ is *DNR* if $\sigma(e) \neq \varphi_e(e)$ for all e in the domain of σ.

We let φ_e be the eth partial computable function $\omega \to \omega$ according to a fixed effective listing of all such functions, and let W_e denote the domain of φ_e. We assume that if $x \in W_n[s]$ then $x < s$. We write 0^i to denote the sequence of i many zeros, and we let λ denote the empty string.

3. DNR functions and Kurtz randoms

In order to show that there is a DNR function f which does not compute any Kurtz-random set, it suffices to observe that the DNR minimal degree constructed in [?] is automatically hyperimmune-free. The fact that it is hyperimmune-free follows from an analysis of the trees that the function f is constructed to lie on.

Theorem 3.1. *There is a DNR function f such that the degree of f is both minimal and hyperimmune-free. Hence, there is a computably bounded DNR function which does not compute any Kurtz-random set.*

Proof. We first observe that the second statement follows from the first. Let f be as in the first statement. Then f is computably bounded since its degree is hyperimmune-free. The argument given in the introduction to this paper that KR $\not\leq_w$ BI actually shows that no function of hyperimmune-free minimal degree computes a Kurtz-random set, so f computes no Kurtz-random set.

We now show that the DNR minimal degree constructed in [?] is hyperimmune-free. By a function-tree we mean a partial function $T : \omega^{<\omega} \to \omega^{<\omega}$ such that for any $\sigma \in \omega^{<\omega}$ and $i \in \omega$, if $T(\sigma^\frown i)\downarrow$ then:

(i) $T(\sigma)\downarrow$ and $T(\sigma) \subset T(\sigma^\frown i)$;
(ii) for all $i' < i$, $T(\sigma^\frown i')\downarrow$ and $T(\sigma^\frown i')$ is incompatible with $T(\sigma^\frown i)$;
(iii) there exists i' such that $T(\sigma^\frown i')\uparrow$.

We write $\tau \in T$ when τ is in the range of T and we write $f \in [T]$ when there exist an infinite number of initial segments of f in (the range of) T. We say that τ is of level n in T if $\tau = T(\sigma)$ for σ of length n. The strings τ and τ' are Ψ-*splitting* if Ψ^τ is incompatible with $\Psi^{\tau'}$.

Definition 3.2. *We say that a function-tree T is* delayed Ψ-splitting *if whenever $\tau_0, \tau_1 \in T$ are incompatible, any $\tau_2, \tau_3 \in T$ properly extending τ_0 and τ_1 respectively are Ψ-splitting.*

The DNR function f constructed in [?] satisfies the property that for every $\{0,1\}$-valued functional Ψ such that Ψ^f is total and non-computable, $f \in [T]$ for some function-tree T which is delayed Ψ-splitting and partial computable, with computable domain.

Now suppose that $\Phi^f \in \omega^\omega$ (so that Φ^f is total but not necessarily $\{0,1\}$-valued). We have to show that Φ^f is computably dominated. It is reasonable to assume that if $\Phi^\sigma(n)[s]\downarrow$ then $n < s$ and $\Phi^\sigma(n')[s]\downarrow$ for all $n' < n$, so that Φ^σ is a finite string. We define Ψ which is $\{0,1\}$-valued and which codes Φ in a natural way. For $\sigma \in \omega^{<\omega}$ we define $h(\sigma)$ by induction on $|\sigma|$. For the empty string λ, we define $h(\lambda) = \lambda$. Given $h(\sigma)$, we define $h(\sigma^\frown i) = h(\sigma)^\frown 0^i 1$. Then if $\Phi^\sigma = \tau$, we define $\Psi^\sigma = h(\tau)$. The definition of Ψ is consistent since if σ_1 extends σ_0, then $\Phi^{\sigma_1} \supseteq \Phi^{\sigma_0}$ and hence $h(\Phi^{\sigma_1}) \supseteq h(\Phi^{\sigma_0})$. Now suppose that $f \in [T]$, where T is delayed Ψ-splitting and partial computable with computable domain. Then T is also delayed Φ-splitting, since any Ψ-split pair of strings is also Φ-split because h preserves compatibility of strings. Suppose that $T(\sigma^\frown i) = \tau_0 \subset f$ and that, for some $j \neq i$, $T(\sigma^\frown j) = \tau_1$ has proper extensions in T. Let $T(\sigma) = \tau$ and put $\ell = |\Phi^\tau|$. Let τ_0' be the initial segment of f of level $|\sigma| + 2$ in T. Then any proper extension τ_1' of τ_1 in T must Φ-split with τ_0', and hence $\Phi^{\tau_0'}(\ell)\downarrow$. We may conclude that, for any $g \in [T]$ which is not an isolated path through T (i.e. such that every initial segment of g has more than one infinite extension in $[T]$), Φ^g is total. However, any isolated path through

T is computable, since it is an isolated element of the computably bounded Π^0_1 class $[T]$. Since no DNR function is computable, we conclude that Φ^g is total for every DNR function $g \in [T]$.

Now we simply apply compactness together with the fact that T has computable domain. For each n, there is a length ℓ such that all strings $\tau \in T$ of level l either satisfy $\Phi^\tau(n)\downarrow$ or else τ is not DNR. Such a length can be found uniformly in n using a computable search since the family of non-DNR strings is c.e. This allows us to bound $\Phi^f(n)$ and so computably dominate Φ^f. □

4. The intuition behind the proof of Theorem ??

We construct a $\{0,1\}$-valued Turing functional Ψ so that Ψ^f is bi-immune for all DNR functions f. The requirements are as follows:

R_{2n} : If W_n is infinite, then for all DNR f, $\Psi^f \cap W_n \neq \emptyset$;
R_{2n+1} : If W_n is infinite, then for all DNR f, $\overline{\Psi^f} \cap W_n \neq \emptyset$.

In this section we give the basic idea behind the construction, by showing how to simultaneously satisfy two requirements. There are then further challenges to be met as one looks to satisfy all requirements, and in Section **??** we formally define the construction which suffices to achieve this.

First of all, let us consider how we might satisfy a single requirement, say R_0. The strategy in this case is very simple. At each stage s with $W_{0,s} = \emptyset$, for all σ of length $s+1$, we define $\Psi^\sigma(s) = 1$. At the first stage s_0 (if any) at which some number x is enumerated in W_0, the strategy stops acting. In this case, we have $x < s_0$ by convention, so $\Psi^f(x) = 1$, and hence $\Psi^f \cap W_0 \neq \emptyset$, for all f, by the action of the strategy at stage x. Of course, such a stage s_0 must exist if W_0 is infinite.

Now let us see how one might go about satisfying another requirement as well as R_0, R_3 say (we consider R_3 instead of R_1 for the sake of greater generality, since R_1 also corresponds to W_0). By the recursion theorem, we may assume we are given a number n such that we may define the value $\varphi_n(n)$ at some point during the construction. We call such an n a "diagonalisation point". As we work to satisfy R_3, we use a fixed diagonalisation point n to ensure that our action does not injure R_0.

We divide R_3 up into an infinite number of subrequirements. The first of these looks to satisfy R_3 for all f such that $f(n) = 0$, the second for those f with $f(n) = 1$, and so on. The strategy for the first subrequirement becomes active at stage $s_0 = n$. Once it is active (until it *finishes*), at each stage s, for all σ of length $s+1$ such that $\sigma(n) = 0$, we define $\Psi^\sigma(s) = 0$. For all other σ of length $s+1$ we define $\Psi^\sigma(s) = 1$, if R_0 so requests. The strategy waits until a number $x \geq s_0$ enters W_1, say at stage s_1. Since $s_0 \leq x < s_1$, the action of the strategy has ensured that R_3 is satisfied for all f with $f(n) = 0$, via its action at stage x. If no such number ever appears in W_1, then clearly W_1 is finite and the entire requirement R_3 (not just this subrequirement) is satisfied. When such a number appears at stage s_1, and hence we have satisfied R_3 for all f with $f(n) = 0$, and we say that the strategy for the first subrequirement *finishes*. Then the strategy for the next subrequirement becomes active at the next stage $s_1 + 1$. While this subrequirement is active, at each stage s, and for all σ of length $s+1$ such that $\sigma(n) = 1$, we define $\Psi^\sigma(s) = 0$. For all other σ of length $s+1$ we define $\Psi^\sigma(s) = 1$ if so requested by R_0. The strategy for the second subrequirement, then waits until a number $\geq s_1 + 1$ enters W_1. If this happens then the strategy for the second subrequirement finishes, and the strategy for the third subrequirement begins at the next stage, and so on. Thus, either one of the subrequirements becomes active and never finishes and hence R_3 is satisfied, or each eventually becomes active and eventually finishes. In the latter case, all subrequirements of R_3 are met, and hence R_3 is met for all functions f, whatever the value of $f(0)$. Thus R_3 is met in all cases. However, how do we know that R_0 is met? (Note that we have defined Ψ according to the wishes of R_3 rather than R_0 when they request opposite values. This may seem strange, but if we always followed the wishes of R_0, the construction would obviously fail if $W_0 = \emptyset$.) As we have already remarked, R_0 is obviously met if W_0 is empty, so assume $W_0 \neq \emptyset$. At the first stage when a number x enters W_0, we have to be sure that the action we have taken for R_3 does not prevent R_0 from being satisfied. When x is enumerated into W_0, we look to see which of the subrequirements for R_3 (if any) was active at stage x. There will be precisely one of these if $x \geq n$, and this will be the only subrequirement which defines $\Psi^\sigma(x)$ for any string σ . If this was the subrequirement which looks to satisfy R_3 for all f with $f(n) = i$, then we define $\varphi_n(n) = i$. The effect of this is that no string σ with $\sigma(n) = i$ is DNR. Hence if f extends σ for which we have defined $\Psi^\sigma(x) = 0$, then f is not DNR, and R_0 is satisfied. Above we assumed that $x \geq n$. If $x < n$, then no subrequirement of R_3 is active at stage x, but this is no problem since then we win by the basic R_0 strategy without defining $\varphi_n(n)$.

Note that in the above, there is no diagonalization witness associated with R_0 because there is no higher priority requirement than R_0, while there is one diagonalization witness associated with R_3 because there is one requirement of higher priority than R_3 (namely R_0) and that requirement finishes at most once. This theme is amplified in the next section.

5. The construction

In the previous section, the single requirement R_0 gave rise to infinitely many subrequirements of the next requirement R_3. We now iterate this idea, so that each subrequirement of R_n gives rise to infinitely many subrequirements of R_{n+1}. Hence, the construction is carried out on the infinitely branching tree $\mathcal{T} = \omega^{<\omega}$. Note, however, that our tree of strategies will not be used in the conventional fashion, in the sense that there will not be a special path defined at each stage. At any given stage many incompatible nodes of the tree may act.

To deal with all requirements simultaneously, we need an infinite computable set D of diagonalization points. The existence of such a set D follows from a slight generalization of the recursion theorem ([**?**], Proposition II.3.4) which asserts that every computable function has an infinite computable set of fixed points. Thus, for each $n \in D$, we are allowed to define $\varphi_n(n)$ during the construction.

Each node $\alpha \in \mathcal{T}$ of length n is devoted to a subrequirement R_α of R_n. More specifically, we define for each $\alpha \in \mathcal{T}$ a partial function θ_α with finite domain, and then R_α asserts that R_n holds for all functions f extending θ_α. We also assign to each $\alpha \in \mathcal{T}$ a finite set E_α of diagonalization points. The domain of θ_α will be the union of all sets E_β for $\beta \subseteq \alpha$.

5.1. **Defining E_α and θ_α.** We start by defining the sets E_α recursively. For the empty string λ, let E_λ be empty. If E_α is defined, where α has length n, let $E^+_\alpha \subseteq D$ be a set of $n+1$ diagonalization points, and set $E_{\alpha^\frown i} = E^+_\alpha$ for all $i \in \omega$. Further, arrange that E_β and E_γ are disjoint if β and γ are distinct and are not siblings, i.e.

are not immediate successors of the same node. We use $n+1$ new diagonalization points, since there are $n+1$ strings β extended by α, and each R_β will finish at most once. Each diagonalization point can be used to prevent $R_{\alpha^\frown i}$ from interfering with a particular R_β, for $\beta \subseteq \alpha$.

Next we define the partial functions θ_α recursively. Let θ_λ be the empty partial function. If θ_α is defined, let E^+_α be as above, and effectively enumerate the extensions of θ_α to $\cup_{\beta\subseteq\alpha} E_\beta \cup E^+_\alpha$ as $\theta_0, \theta_1, \cdots$, in such a way that each extension appears precisely once in the list. Let $\theta_{\alpha^\frown i} = \theta_i$ for all $i \in \omega$. We show in the verification by induction on n that every function $f \in \omega^\omega$ extends θ_α for exactly one α of length n. Thus, to meet R_n it suffices to meet R_α for all α of length n.

5.2. **The instructions for R_α.** Let W_α be the c.e. set associated with R_α. Let $i(\alpha) = 1$ if α has even length and otherwise let $i(\alpha) = 0$. We say that a string or function has *type* α if it extends θ_α.

At each stage at which it is *active* the strategy for R_α proceeds as follows. Let s_α be the first stage at which it was active, as defined in the construction below.

(1) The strategy requests that $\Psi^\sigma(s) = i(s)$ for all σ of length $s+1$ of type α.
(2) If $s \geq s_\alpha$ is minimal such that $W_\alpha[s]$ has an element $x \geq s_\alpha$, then we declare that R_α *finishes* at stage s. In this case we say that R_α finishes *via* the least such x (and will not be active at future stages). For each $\beta \supseteq \alpha$ we now use the diagonalization points in E_β to ensure that values of $\Phi^\sigma(x)$ defined by R_β do not prevent R_α from being satisfied. For each $\beta \supset \alpha$ we proceed as follows. If at stage x the subrequirement R_β was active, let $e \in E_\beta$ be minimal such that $\varphi_e(e)$ is not yet defined, and set $\varphi_e(e) = \theta_\beta(e)$. We will see in the verification by a trivial counting argument that such an e always exists.

Note that R_α finishes at most once. If R_α becomes active but never finishes then W_α is finite, so R_n, where $n = |\alpha|$, (not just R_α) is met in this case. Now consider the effect of our use of the diagonalization points when R_α finishes via x. Suppose that $\beta \supset \alpha$ was active at stage x and that we set $\varphi_e(e) = \theta_\beta(e)$ for some $e \in E_\beta$. The effect of this definition is that no function of type β is DNR, since θ_β is not DNR.

5.3. **The construction.** It is convenient to assume that W_0 is empty. R_λ is active at all stages. For all α, $R_{\alpha^\frown 0}$ becomes active at the first stage $s \geq \max E_{\alpha^\frown 0}$. For $i \geq 0$, if $R_{\alpha^\frown i}$ finishes at stage s, then it is not active at stages $s' > s$, and $R_{\alpha^\frown (i+1)}$ becomes active at stage $s+1$.

At each stage s, take the R_α which are active at stage s in lexicographical order, and perform their instructions. Then for all σ of length $s+1$, define $\Psi^\sigma(s) = i(\alpha)$, where α is the *longest* string such that R_α is active at stage s and σ has type α.

5.4. **The verification.** The basic idea is that when R_α finishes, our use of diagonalization points means that it is permanently satisfied. If we consider a single n, we can see that R_α is met for all α of length n, and hence R_n is met. Namely, if the strategy for some R_α starts and fails to finish, then R_n is met as remarked above. Otherwise, R_α finishes for every α of length n, meeting R_α permanently. Now let us see this in more detail.

First we show by induction on n, that for every function f there is a unique α of length n such that f extends θ_α. This is obvious for $n = 0$. Now suppose that it is true for n. First we show existence. Let f be given. Choose α of length n such that f extends θ_α. Then the restriction of f to $\cup_{\beta\subseteq\alpha}E_\beta \cup E^+_\alpha$ is an extension of θ_α to $\cup_{\beta\subseteq\alpha}E_\beta \cup E^+_\alpha$, and so is equal to $\theta_{\alpha^\frown i}$ for some i. In order to show uniqueness, first note that α is unique by the induction hypothesis. If $f \supseteq \theta_{\alpha^\frown j}$, then $\theta_{\alpha^\frown j}$ and $\theta_{\alpha^\frown i}$ are compatible, meaning that $i = j$.

Next we observe that, when $\beta \supset \alpha$, and α finishes via x, there is a least $e \in E_\beta$ such that $\varphi_e(e)$ is not yet defined, and for which we can define $\varphi_e(e) = \theta_\beta(e)$ if β was active at stage x. This follows since $|E_\beta|$ is the same as the number of proper initial segments of β, and each of these can finish at most once. When $\alpha \subset \beta$, and α finishes via x, there is at most one sibling β' of β, which is active at stage x, meaning that for $e \in E_\beta = E_{\beta'}$ with $\varphi_e(e)$ as yet undefined, we are free to define $\varphi_e(e) = \theta_{\beta'}(e)$ for this β'. (Here we consider β to be a sibling of itself.)

In order that the instructions should be well defined, we also have to check that if $|\sigma| = s + 1$ then there is a longest α such that R_α is active at stage s and σ has type α. Since R_λ is always active, there exists at least one α such that R_α is active at stage s and σ is of type α. The fact that there is a longest such, then follows from the fact that θ_α and θ_γ are incompatible when α and γ are incompatible. If both α and γ are active at stage s then any σ of length $s + 1$ which is of type α and type β extends θ_α and θ_β.

We show next that R_{2n} is met (the verification for R_{2n+1} being almost identical). Since we have shown that for every function f there is a unique α of length $2n$ such that f extends θ_α, it suffices to show that R_α is met for all α of length $2n$. Fix such an α, fix f which is DNR and extends θ_α and assume that W_n is infinite. We must show that $\Psi^f \cap W_n$ is nonempty. Since we are assuming that W_0 is empty, we can let β and i be such that $\alpha = \beta^\frown i$. Since W_n is infinite, it is easy to show by induction on j, that each requirement $R_{\beta^\frown j}$ starts acting at some stage and also finishes. Suppose α starts acting at stage s and finishes at stage t, via the enumeration of x into W_n. We have $s \leq x < t$, so at stage x the requirement R_α is active. Letting σ be the initial segment of f of length $x + 1$, we have that σ is of type α since f extends θ_α. If $\Psi^\sigma(x) = 1$, then $x \in \Psi^f \cap W_n$, so we are done. Otherwise, at stage x, some R_γ requests that $\Psi^\sigma(x) = 0$ where $|\gamma| \geq |\sigma|$ and σ is also of type γ. We claim that γ properly extends σ. Since σ extends both θ_γ and θ_α, γ and α are compatible. Since γ and β have lengths of opposite parity and $|\gamma| \geq |\sigma|$, it follows that γ properly extends σ, as claimed. By construction, since γ was active at stage x, for some $e \in E_\gamma$, we set $\varphi_e(e) = \theta_\gamma(e)$, so that θ_γ is not DNR. Now $f \supseteq \sigma \supseteq \theta_\gamma$ so f is not DNR, and hence this case does not arise.

6. An open question

We finish by mentioning an open question concerning a possible strengthening of our main result. We first need a definition

Definition 6.1. *A set A is called* effectively immune *if A is infinite and there is a computable function f such that, for all e, if $W_e \subseteq A$, then $|W_e| < f(e)$. Of course, then A is called* effectively bi-immune *if both A and $\overline{A}$ are effectively immune. Let EBI be the class of effectively bi-immune sets.*

Question 1. *The following related questions are open.*

(i) *Does every DNR function compute an effectively bi-immune set? In other words, is it the case that EBI $\leq_w$ DNR?*

(ii) *Is it the case that EBI* $\leq_s$ *DNR?*
(iii) *Is every DNR function Turing equivalent to an effectively bi-immune set?*

Stephen Simpson has kindly brought to our attention that he announced a positive solution to (iii) in [**?**] but that he subsequently retracted the claim, after it was questioned by Bjørn Kjos-Hanssen. Of course, a positive solution to (ii) would strengthen our main result, Theorem **??**, as well as Theorem 7 of [**?**], which essentially asserts that every DNR function computes a bi-immune set. However, the methods of our paper do not seem adequate to prove (iii).

References

[1] K. Ambos-Spies, B. Kjos-Hanssen, S. Lempp, and T. Slaman, Comparing DNR and WWKL, *The Journal of Symbolic Logic* 69 (2004), 1089-1104.
[2] R. Downey, N. Greenberg, C. Jockusch, and K. Milans, Binary subtrees with few labeled paths, *Combinatorica* 31 (2011), 285-303.
[3] R. Downey and D. Hirschfeldt, Algorithmic Randomness and Complexity, Springer, 2010
[4] C. Jockusch, Semirecursive sets and positive reducibility, *Transactions of the American Mathematical Society*, 131 (1968), 420-436.
[5] C. Jockusch, Upward closure of bi-immune degrees, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 18 (1972), 285-287.
[6] C. Jockusch, Degrees of functions with no fixed points, in *Logic, Methodology, and Philosophy of Science VIII*, edited by J. E. Fenstad, I. T. Frolov, and R. Hilpinen, North-Holland, Amsterdam, New York, Oxford, Tokyo, 1989, 191-201.
[7] C. Jockusch, M. Lerman, R. Solovay, and R. Soare, Recursively enumerable sets modulo iterated jumps and extensions of Arslanov's completeness criterion, *The Journal of Symbolic Logic* 54 (1989), 1288-1323.
[8] M. Kumabe and A.E.M. Lewis, A fixed point free minimal degree, *Journal of the London Mathematical Society*, 80 (3), 785-797, 2009.
[9] A. Lachlan, Solution to a problem of Spector, *Canadian Journal of Mathematics*, 23 (1971), 247-256.
[10] S. Simpson, Recursion theoretic aspects of the dual Ramsey theorem, in Recursion Theory Week, Lecture Notes in Mathematics No. 1141, Springer-Verlag, Berlin, Heidelberg, New York, Tokyo, 1985, 357-371.
[11] S. Simpson, Mass problems and randomness, *The Bulletin of Symbolic Logic*, 11 (1), 1-27, 2005.
[12] S. Simpson, An extension of the recursively enumerable Turing degrees, *Journal of the London Mathematical Society*, 75 (2007), 287-297.
[13] R. Soare, Recursively Enumerable Sets and Degrees, Springer-Verlag, Berlin, Heidelberg, New York, London, Paris, Tokyo, 1987.

Carl G. Jockusch, University of Illinois at Urbana-Champaign, 1409 W. Green Street, Urbana, Illinois 61801-2975, USA
E-mail address: jockusch@math.uiuc.edu
URL: http://www.math.uiuc.edu/~jockusch/

Andrew E.M. Lewis, School of Mathematics, University of Leeds, LS2 9JT Leeds, United Kingdom.
E-mail address: andy@aemlewis.co.uk
URL: http://aemlewis.co.uk