Lowness for Demuth Randomness

Rod Downey and Keng Meng Ng*

Victoria University of Wellington,
School of Mathematics, Statistics and Computer Science,
PO Box 600, Wellington, New Zealand

Abstract. We show that every real low for Demuth randomness is of hyperimmune-free degree.

1 Introduction

A fundamental theme in the study of computability theory is the idea of *computational feebleness*, which might be loosely defined as properties exhibited by non-computable sets resembling computability. This is usually described in literature as a notion of lowness, and indicates weakness as an oracle. The classical example of sets exhibiting a property of this sort are the low sets, which are the sets A such that $A' \equiv_T \emptyset'$. Thus in terms of the jump operator, low sets are indistinguishable from the computable ones. There is a plethora of results in the literature which suggest that low sets resemble computable sets, particularly for the computably enumerable (c.e.) sets.

In notions of lowness, one usually considers a certain set operation and says that A satisfies the notion of lowness if it does not give any extra power to the operation. In the above example of low sets, the operation concerned was the Turing jump operator. Slaman and Solovay demonstrated in [26] a relationship between the low sets, and another seemingly unrelated lowness notion from the theory of inductive inference. In particular they showed that every set A which was low for EX learning was also low (and in fact 1-generic below $\emptyset'$). This result says that lowness for various notions of computation can be intertwined.

In a similar vein, Bickford and Mills [5] introduced the concept of a *superlow* set. A truth-table reduction is a Turing reduction which is total on every oracle string, and a set A was defined to be superlow if $A' \equiv_{tt} \emptyset'$, where the equivalence $\equiv_{tt}$ is induced by the pre-ordering of truth-table reducibility. One expects that the superlow sets would resemble the computable sets very strongly. Indeed a standard construction of a low c.e. set by the preservation of jump computations already made the constructed set superlow (as in the low basis theorem). At first blush we might be tempted to think that the low and superlow sets are very similar, or even the same. However the low and superlow sets have turned out to be not even elementarily equivalent. Recent examples have suggested that the dynamic properties of low and superlow c.e. sets are very different. For instance

* Both authors were partially supported by the Marsden Fund of New Zealand.

K. Ambos-Spies, B. Löwe, and W. Merkle (Eds.): CiE 2009, LNCS 5635, pp. 154–166, 2009.

Downey and Ng [12] showed that there is a low c.e. degree which is not the join of any two superlow c.e. degrees.

Recent development in algorithmic randomness have revealed that the theory of low and superlow c.e. sets is much deeper than originally thought. Various lowness notions for Kolmogorov complexity and other operations arising in algorithmic randomness have suggested a deep connection with subclasses of the low sets. Several subclasses of the superlow sets have sprung up, and have been shown to be even better candidates for studying properties resembling computability. A central theme in these classes is the notion of *traceability*.

An order function h is a total computable, non-decreasing and unbounded function. A set A is said to be *jump traceable* with respect to an order h, if there is a computable g, such that for all x, $|W_{g(x)}| \leq h(x)$, and $J^A(x) \in W_{g(x)}$. Here, $J^A(x)$ denotes the value of the universal function $\Phi_x^A(x)$ partial computable in A. Note that the range of J^A is contained in $\mathbb{N}$, and not restricted to binary values. A set A is said to be jump traceable, if there is an order h for which it is jump traceable with respect to h. This notion was introduced by Nies [23].

A jump traceable set A differs from a superlow set in the sense that we are able to effectively enumerate finitely many candidates for each $J^A(x)$. For a superlow set A we are only able to approximate whether $J^A(x)$ converges. Since we are able to code finite information into $J^A(x)$, it might appear that being jump traceable is stronger than being superlow. Recall that a set A is n-c.e. if there is a computable approximation $A_s(x)$ to A, such that the number of changes in $A_s(x)$ is bounded by n at every x. A is ω-c.e. if the number of changes is bounded by a computable function.

Ng [19] showed that for n-c.e. sets, jump traceability and superlowness were the same. Earlier, Nies [23] showed that they coincide on the c.e. sets. However when we consider the next level on the Ershov hierarchy, these two notions separate: it is not hard to see that no jump traceable set can be Martin-Löf random, so a superlow Martin-Löf random set cannot be jump traceable. In the other direction, Nies [23] showed that there was an ω-c.e. jump traceable set which was not superlow. If we consider non-Δ_2^0 sets, the situation becomes even more bizzare. There is a perfect Π_1^0 class of sets which are jump traceable, via an exponential bound. Such a phenomenon highlights an important inherent property of being traceable; we are only able to enumerate possible values of $A\restriction_n$, but beyond that we are given no additional information to suggest which one of the enumerated values is correct. Indeed Kjos-Hanssen and Nies [17] showed that jump traceable sets could even be superhigh.

Traceability plays a very important role in understanding lowness notions arising in algorithmic information theory. If R is a notion of effective randomness, then *low for R* would denote all the sets A for which $R^A = R$ (i.e. every random Z is still random relative to A). The work of Terwijn and Zambella [28], Kjos-Hanssen, Nies and Stephan [18], and Bedregal and Nies [4] has revealed an interesting interaction between "predictability" in terms of traceability, and simplicity in terms of Kolmogorov complexity. Recall that a set Z is of hyperimmune-free degree, if every function computable from Z is dominated by a computable function. A is said to be computably traceable if A is "uniformly hyperimmune-free". That is, there is a computable function h such that for each $f \leq_T A$, there exists a computable sequence of canonical finite sets $D_{g(x)}$ with $|D_{g(x)}| \leq h(x)$, and such that $f(x) \in D_{g(x)}$ for all x. They showed that

Theorem 1.1. *A is low for Schnorr randomness iff A is computably traceable.*

Hence the notion of being low for Schnorr randomness coincided with a combinatorial notion, that of being computably traceable.

A very robust class exhibiting low information content is the class of *K-trivial reals*[1]. Formally a real A is K-trivial if there is some constant c such that $K(A\restriction_n) \leq K(n) + c$ for every n, where K denotes the prefix-free Kolmogorov complexity. Here $A\restriction_n$ denotes the first n bits of A. Hence an initial segment of a K-trivial real contains no more information than its own length; clearly all computable reals are K-trivial. The most well-known work on the K-trivials was the work of Nies showing the coincidence of several simple classes:

Theorem 1.2 (Nies [22,24]). *A is K-trivial iff A is low for Martin-Löf randomness iff A is low for K.*

A real A is low for K if $\exists c \forall \sigma (K(\sigma) \leq K^A(\sigma) + c)$; that is, A does not help in the compression of strings when used as an oracle. The robustness of this class was further demonstrated when various other characterizations were found; for instance the reals low for weak 2-randomness, and the bases for Martin-Löf randomness [16]. Here A is a base for Martin-Löf randomness if $A \leq_T Z$ for some Z which is random relative to A. Intuitively there cannot be many possibilities for initial segments of a base for randomness, because we can use the given Turing reduction Φ (where $A = \Phi^Z$) to lower the Kolmogorov complexity of possible initial segments of Z. This was in fact the driving force behind the "hungry sets" theorem of [16]. We refer the reader to Franklin and Stephan [14] for a Schorr random version of a base. Other notions of bases which have been studied are the LR-bases for randomness [2,3], and the JT-bases for randomness [19]. These are notions obtained from a base for randomness, by replacing Turing reducibility with different weak reducibilites.

The resemblance which the K-trivial reals bear with the computable sets makes one wonder if they are related to the low sets. Is there also a combinatorial characterization in terms of traceability like the one for Schnorr lowness? Recent developments have suggested that this was the case. In [10], Downey, Hirschfeldt, Nies and Stephan showed that the K-trivial reals were natural solutions to Post's problem in the following sense:

Theorem 1.3. *Every K-trivial real is Turing incomplete.*

They used a new method widely known as the "Decanter method". This method exploited the fact that for any given K-trivial real, we could challenge its triviality very slowly. This resembles the "drip-feeding" action of a decanter, and

[1] We identify sets of natural numbers with real numbers. It is common to use the term "sets of natural numbers" in traditional computability theory, while in algorithmic information theory it is useful to think of these as infinite binary sequences.

hence the fanciful name. Nies [23,24] then applied a non-uniform method of the Decanter method to show:

Theorem 1.4. *Every K-trivial real is superlow.*

In fact, the same proof also shows that every K-trivial real is jump traceable at an order of $n \log^2 n$. These results suggested that jump traceability was the appropriate combinatorial notion associated with K-triviality, in the same way as computable traceability was related to lowness for Schnorr randomness. Armed with this insight, Figueira, Nies and Stephan [13] defined the notion of strong jump traceability. They defined A to be *strongly jump traceable*, if A is jump traceable with respect to all order functions. Figueira, Nies and Stephan used a cost function construction to show the existence of a promptly simple strongly jump traceable c.e. set. One can view c.e. strong jump traceability as a natural strengthening of being superlow. Unlike the case of computable traceability, strong jump traceability is different from jump traceability; in fact there is an entire hierarchy of jump traceable sets ordered by the growth rates of the bounding functions on the size of the trace. This hierarchy contains infinitely many strata in either direction. That is, there is no single maximal bound for jump traceability (Figueira Nies and Stephan [13]), and neither is there a single minimal bound (Ng [21]). In fact, Greenberg and Downey [8] showed that if one got down to a level of $\log \log n$, then every jump traceable real was Δ_2^0.

Figueira, Nies and Stephan asked if strong jump traceability was the coveted combinatorial characterization of the K-trivials. Cholak, Downey and Greenberg [6] answered this for the c.e. case by showing that the c.e. strongly jump traceable sets form a proper sub-ideal of the c.e. K-trivials. In fact, they showed that if A was c.e. and jump traceable at order $\sim \sqrt{\log n}$, then A was also K-trivial. This gave the first example of a combinatorial property which implies K-triviality. Even though neither notion of jump traceability gives us an exact characterization of the K-trivials, the associated results provide a good idea of the upper and lower bounds on the order of jump traceability which would capture K-triviality. By analyzing the proofs which give the lowerbound $\sim \sqrt{\log n}$ and upperbound $\sim n \log^2 n$, two possible characterizations had been suggested. Greenberg suggested that A is K-trivial iff A is jump traceable for all orders h with $\sum_{n \in \omega} \frac{1}{h(n)} < \infty$. This was refuted by Barmpalias, Downey and Greenberg [1], and independently by Ng [20]. The second conjecture is that the collection of orders should be the class of all orders h satisfying $\sum_{n \in \omega} 2^{-h(n)} < \infty$, and is still open.

Several other lowness notions have been studied with respect to other concepts of randomness. We list a few notable examples. Downey, Greenberg, Mihailović and Nies [9] showed that the computably traceable sets were exactly those which were low for computable measure machines. Here, a computable measure machine is a prefix-free machine with a computable halting probability, and A is low for computable measure machines (c.m.m.) if for each c.m.m. M relative to A, there is a c.m.m. N and a constant c such that $K_M^A(\sigma) \geq K_N(\sigma) - c$ for every σ.

Nies [24] showed that the only sets which were low for computable randomness[2], were the computable sets. The combined work of Greenberg, Miller, Stephan and Yu [15,27] revealed that the sets which were low for Kurtz randomness, were exactly the hyperimmune-free and non-DNR degrees. These were also the sets which were low for weak 1-genericity, which showed yet another interaction between lowness notions in classical computability, and randomness. For more examples we refer the reader to Chapter 8 of Nies' book [25].

In the next section, we contribute with another result in this direction. We consider lowness with respect to a less well-known notion of randomness, known as Demuth random. This was introduced by Demuth [7] and was originally motivated by topics in constructive analysis. This appears to be a very natural (strong) randomness notion to study, and not much work has yet been done on this class.

Definition 1.5. *A Demuth test is a sequence of c.e. open sets* $\{W_{g(x)}\}_{x\in\mathbb{N}}$ *such that* $\mu W_{g(x)} < 2^{-x}$ *for every* x*, and* g *is* ω*-c.e. We say that* Z *passes the test if* $\forall^\infty x Z \notin W_{g(x)}$*. A real is Demuth random if it passes every Demuth test.*

A Demuth test is a sequence of c.e. open sets, but the function giving the weak indices is an ω-c.e. function. Informally if we were building such a test (to try and catch some real number) we have additional power over building a ML-test because we can change the name of $W_{g(x)}$ a bounded number of times. That is, we can remove a certain part (or even all) of what we have enumerated into $W_{g(x)}$ so far, as long as we only do it a computably bounded number of times. The definition of passing a Demuth test is as in the Solovay sense, and we cannot always require that $W_{g(x)} \supseteq W_{g(x+1)}$. There is no universal Demuth test, although there is a single special test $\{W_{\hat{g}(x)}\}$ which is universal in the sense that every real passing the special test is Demuth random. However the function $\hat{g}(x)$ has to emulate every ω-c.e. function, and so $\hat{g}(x)$ is Δ^0_2. Hence the special test is not a Demuth test.

Clearly the Demuth randoms lie between 2-randomness and ML-randomness. It is not hard to construct a Δ^0_2 Demuth random using the special test, and obviously no ω-c.e. set can be Demuth random. Hence the containments are proper. Demuth randoms exhibit properties which can be found in both 1- and 2-randomn reals. For instance every Demuth random (like the 2-randoms) are GL_1 and hence of hyperimmune degree by a result of Miller and Nies (Theorem 8.1.19 of [25]). Here a real is of hyperimmune degree if it is not of hyperimmune-free degree. Since there are hyperimmune-free weakly 2-randoms, this implies that Demuth randomness and weak 2-randomness are incomparable notions. However unlike the 2-randoms, the Demuth test notion is essentially computably enumerable.

We contribute two theorems to the understanding of this notion of randomness. First, we prove that every Demuth random is array computable. This notion was introduced by Downey, Jockusch and Stob [11] to describe the class of reals below which certain multiple permitting arguments could not be carried out.

[2] A real is computably random if it succeeds on every computable martingale.

This again suggests that Demuth randoms are like the 2-randoms, having low computational strength. In particular, the Demuth randoms below $\emptyset'$ form an interesting class, being both low *and* array computable but not superlow.

Theorem 1.6. *Each Demuth random Z is array computable.*

Proof. We observe the proof that every Demuth random is GL_1 already does it; this can be found in Chapter 3 of Nies [25]. The proof actually produces an ω-c.e. function g which dominates the function $\Theta^Z(x) :=$ least s such that $\Phi_x^Z(x)[s]\downarrow$ (which of course implies that $Z' \leq_T Z \oplus \emptyset'$). By usual convention the output value of $\Phi_x^Z(x)$ is $< \Theta^Z(x)$. Since every function computable in Z can be coded into the diagonal, we have a computable function p such that for every e, and almost every $y > e$, we have $\Phi_e^Z(y) = \Phi_{p(e,y)}^Z(p(e,y)) < \Theta^Z(p(e,y)) < g(p(e,y)) < \tilde{g}(y)$, where $\tilde{g}(y) := \max\{g(p(0,y)), g(p(1,y)), \cdots, g(p(y,y))\}$ is ω-c.e. as well. □

Next, we study the notion of lowness with respect to Demuth randomness. A relativized Demuth test involves full relativization. That is, a Demuth test relative to A is a sequence $\{W_{g(x)}^A\}$ where $\mu W_{g(x)}^A < 2^{-x}$ for every x. Here, $g(x) = \lim \tilde{g}(x,s)$ for some A-computable function $\tilde{g}$, and the number of $\tilde{g}$-mind changes is bounded by an A-computable function. A real Z is Demuth random relative to A if it passes every Demuth test relative to A. We say that A is *low for Demuth randomness* if every Demuth random is Demuth random relative to A. In the next section we prove that every real low for Demuth randomness is of hyperimmune-free degree.

However it is still unknown if there is any set which is non-computable and low for Demuth randomness. A construction of such a real will have to build a hyperimmune-free degree, and if one uses the standard forcing method then one has to address the issue of constructing the effective objects required in the proof. We conjecture that every real which is low for Demuth randomness is computable.

2 No Set of Hyperimmune Degree Can be Low for Demuth Randomness

We work in the Cantor space 2^ω with the usual clopen topology. The basic open sets are of the form $[\sigma]$ where σ is a finite string, and $[\sigma] = \{X \in 2^\omega \mid X \supset \sigma\}$. We fix some effective coding of the set of finite strings, and identify finite strings with their code numbers. We treat W_x as a c.e. open set, consisting of basic clopen sets. We say that $[\sigma] \in W_x$ to mean that the code number of σ is in W_x, and we say that a string $\tau \in W_x$ if $\tau \supseteq \sigma$ for some $[\sigma] \in W_x$. Equivalently we say that τ is *captured by* W_x. The same definition holds if we replace τ by an infinite binary string. We prove:

Theorem 2.1. *No set of hyperimmune degree can be low for Demuth randomness.*

Proof. Suppose A is of hyperimmune degree. Let h^A be a function total computable in A and non-decreasing, which escapes domination by all total computable functions. That is, for all total computable g, $\exists^\infty x(g(x) < h^A(x))$. We build a $Z \leq_T A'$ which is Demuth random, but not Demuth random relative to A. To do this, we give an A-computable approximation $\{Z_s\}$ to Z. The construction will try to achieve two goals. The first is to make Z Demuth random by making Z avoid all Demuth tests. The second goal is to ensure that for infinitely many x, there are at most $h^A(x)$ many mind changes of $Z_s \mid_x$. Hence we can easily use the approximation Z_s to build a Demuth test relative to A capturing Z infinitely often. Hence Z cannot be Demuth random relative to A.

2.1 The Motivation

Before we describe the strategy used to prove Theorem 2.1, let us see why an attempted construction of a c.e. set A which is low for Demuth randomness fails. Let us consider a single (relativized) Demuth test $\{V_x^A\}$, played by the opponent, where the index for V_x^A can change $h^A(x)$ times. Now we have to cover $\{V_x^A\}$ with a plain Demuth test $\{U_x\}$, by making sure that $V_x^A \subseteq U_x$ for every x. If $h^A(x) = 0$ for all x, then we could just follow the construction of a c.e. set which is low for random. We would enumerate y into A (to make A non-computable), if the penalty we have to pay for making the enumeration of y is small. Even when h^A is computable, we can always arrange the enumerations so that $V_x^A \subseteq U_x$ eventually, because we could use $h^A(x)$ as the bound for the index change of U_x.

The problem is that an enumeration into A not only increases the amount we have to put into U_x, but also gives the opponent a chance to redefine $h^A(x)$. Suppose he has defined $h^A(x)$ with use b_x. At some stage we will have to commit ourselves to a number $g(x)$, and promise never to change the index for U_x more than $g(x)$ times. We would of course declare that $g(x) > h^A(x)$, but once we do that, the opponent could challenge us to change $A \mid_{b_x}$ to ensure the non-computability of A. We have to eventually change $A \mid_{b_x}$ at some x, and allow the opponent to make $h^A(x) > g(x)$, and then we are stuck.

Note that the opponent will be likely to have a winning strategy, if h^A escapes domination by all computable functions. He could then carry out the above for each e, patiently waiting for an x such that $h^A(x) > \varphi_e(x)$, and then defeat the e^{th} Demuth test. This is the basic idea used in the following proof, where we will play the opponent's winning strategy.

2.2 Listing All Demuth Tests

In order to achieve the first goal, we need to specify an effective listing of all Demuth tests. It is enough to consider all Demuth tests $\{U_x\}$ where $\mu(U_x) < 2^{-3(x+1)}$. Let $\{g_e\}_{e\in\mathbb{N}}$ be an effective listing of all partial computable functions of a single variable. For every g in the list, we will assume that in order to output $g(x)$, we will have to first run the procedures to compute $g(0), \cdots, g(x-1)$, and wait for all of them to return, before attempting to compute $g(x)$. We may also assume that g is non-decreasing. This minor but important restriction on g ensures that:

(i) $dom(g)$ is either $\mathbb{N}$, or an initial segment of $\mathbb{N}$,
(ii) for every x, $g(x+1)$ converges strictly after $g(x)$, if ever.

By doing this, we will not miss any total non-decreasing computable function. It is easy to see that there is a total function $k \leq_T \emptyset'$ that is universal in the following sense:

1. if $f(x)$ is ω-c.e. then for some e, $f(x) = k(e, x)$ for all x,
2. for all e, the function $\lambda x k(e, x)$ is ω-c.e.,
3. there is a uniform approximation for k such that for all e and x, the number of mind changes for $k(e, x)$ is bounded by

$$\begin{cases} g_e(x) \text{ if } g_e(x)\downarrow, \\ \quad 0 \quad \text{otherwise.} \end{cases}$$

Let $k(e, x)[s]$ denote the approximation for $k(e, x)$ at stage s. Denote $U^e_x = W_{k(e,x)}$, where we stop enumeration if $\mu(W_{k(e,x)}[s])$ threatens to exceed $2^{-3(x+1)}$. Then for each e, $\{U^e_x\}$ is a Demuth test, and every Demuth test is one of these. To make things clear, we remark that there are two possible ways in which $U^e_x[s] \neq U^e_x[s+1]$. The first is when $k(e, x)[s] = k(e, x)[s+1]$ but a new element is enumerated into $W_{k(e,x)}$. The second is when $k(e, x)[s] \neq k(e, x)[s+1]$ altogether; if this case applies we say that U^e_x has a *change of index at stage* $s + 1$.

2.3 The Strategy

Now that we have listed all Demuth tests, how are we going to make use of the function h^A? Note that there is no single universal Demuth test; this complicates matters slightly. The e^{th} requirement will ensure that Z passes the first e many (plain) Demuth tests. That is,

$$\mathcal{R}_e : \text{ for each } k \leq e, Z \text{ is captured by } U^k_x \text{ for only finitely many } x.$$

$\mathcal{R}_e$ will do the following. It starts by picking a number r_e, and decides on $Z\restriction_{r_e}$. This string can only be captured by U^k_x for $x \leq r_e$, so there are only finitely many pairs $\langle k, x\rangle$ to be considered since we only care about $k \leq e$. Let S_e denote the collection of these open sets. If any $U^k_x \in S_e$ captures $Z\restriction_{r_e}$, we would change our mind on $Z\restriction_{r_e}$. If at any point in time, $Z\restriction_{r_e}$ has to change more than $h^A(0)$ times, we would pick a new follower for r_e, and repeat, comparing with $h^A(1), h^A(2), \cdots$ each time. The fact that we will eventually settle on a final follower for r_e, will follow from the hyperimmunity of A; all that remains is to argue that we can define an appropriate computable function *at each* $\mathcal{R}_e$, in order to challenge the hyperimmunity of A.

Suppose that $r^0_e, r^1_e, \cdots$ are the followers picked by $\mathcal{R}_e$. The required computable function P would be something like $P(n) = \sum_{k\leq e}\sum_{x\leq r^n_e} g_k(x)$, for if $P(N) < h^A(N)$ for some N, then we would be able to change $Z\restriction_{r^N_e}$ enough times on the N^{th} attempt. There are two considerations. Firstly, we do not know which of $g_0, \cdots, g_e$ are total, so we cannot afford to wait on non converging computations when computing P. However, as we have said before, we can have a different P at each requirement, and the choice of P can be non-uniform. Thus, P could just sum over all the total functions amongst $g_0, \cdots, g_e$.

The second consideration is that we might not be able to compute $r_e^0, r_e^1, \cdots$, if we have to recover r_e^n from the construction (which is performed with oracle A). We have to somehow figure out what r_e^n is, externally to the construction. Observe that however, if we restrict ourselves to non-decreasing $g_0, g_1, \cdots$, it would be sufficient to compute an upperbound for r_e^n. We have to synchronize this with the construction: instead of picking r_e^n when we run out of room to change $Z \upharpoonright_{r_e^{n-1}}$, we could instead pick r_e^n the moment enough of the $g_k(x)$ converge and demonstrate that their sum exceeds $h^A(r_e^{n-1})$. To recover a bound for say, r_e^1 externally, we compute the first stage t such that *all of the* $g_k(x)[t]$ have converged for $x \leq r_e^0$ and g_k total.

2.4 Notations Used for the Formal Construction

The construction uses oracle A. At stage s we give an approximation $\{Z_s\}$ of Z, and at the end we argue that $Z \leq_T A'$. The construction involves finite injury of the requirements. $\mathcal{R}_1$ for instance, would be injured by $\mathcal{R}_0$ finitely often while $\mathcal{R}_0$ is waiting for hyperimmune permission from h^A. We intend to satisfy $\mathcal{R}_e$, by making $\mu(U_x^e \cap [Z \upharpoonright_r])$ small for appropriate x, r. At stage s, we let $r_e[s]$ denote the follower used by $\mathcal{R}_e$. At stage s of the construction we define Z_s up till length s. We do this by specifying the strings $Z_s \upharpoonright_{r_0[s]}, \cdots, Z_s \upharpoonright_{r_k[s]}$ for an appropriate number k (such that $r_k[s] = s - 1$). We adopt the convention of $r_{-1} = -1$ and $\alpha \upharpoonright_{-1} = \alpha \upharpoonright_0 = \langle\rangle$ for any string α. We let $S_e[s]$ denote all the pairs $\langle k, x\rangle$ for which $\mathcal{R}_e$ wants to make Z avoid U_x^k at stage s. The set $S_e[s]$ is specified by

$$S_e[s] = \{\langle k, x\rangle \mid k \leq e \ \wedge \ r_{k-1}[s] + 1 \leq x \leq r_e[s]\}.$$

Define the sequence of numbers

$$M_n = \sum_{j=n}^{2n} 2^{-(1+j)};$$

these will be used to approximate Z_s. Roughly speaking, the intuition is that $Z_s(n)$ will be chosen to be either 0 or 1 depending on which of $(Z_s \upharpoonright_n)^\frown 0$ or $(Z_s \upharpoonright_n)^\frown 1$ has a measure of $\leq M_n$ when restricted to a certain collection of U_x^e.

If P is an expression we append $[s]$ to P, to refer to the value of the expression as evaluated at stage s. When the context is clear we drop the stage number from the notation.

2.5 Formal Construction of *Z*

At stage $s = 0$, we set $r_0 = 0$ and $r_e \uparrow$ for all $e > 0$, and do nothing else. Suppose $s > 0$. We define $Z_s \upharpoonright_{r_k[s]}$ inductively; assume that has been defined for some k. There are two cases to consider for $\mathcal{R}_{k+1}$:

1. $r_{k+1}[s] \uparrow$: set $r_{k+1} = r_k[s] + 1$, end the definition of Z_s and go to the next stage.

2. $r_{k+1}[s]\downarrow$: check if $\sum_{\langle e,x\rangle\in S_{k+1}[s]} 2^{r_{k+1}} g_e(x)[s] \leq h^A(r_{k+1}[s])$. The sum is computed using converged values, and if $g_e(x)[s]\uparrow$ for any e, x we count it as 0. There are two possibilities:
   (a) *sum* $> h^A(r_{k+1})$: set $r_{k+1} = s$, and set $r_{k'}\uparrow$ for all $k' > k+1$. End the definition of Z_s and go to the next stage.
   (b) *sum* $\leq h^A(r_{k+1})$: pick the leftmost node $\sigma \supseteq Z_s\restriction_{r_k[s]}$ of length $|\sigma| = r_{k+1}[s]$, such that $\sum_{\langle e,x\rangle\in S_{k+1}[s]} \mu(U^e_x[s]\cap[\sigma]) \leq M_{r_{k+1}[s]}$. We will later verify that σ exists by a counting of measure. Let $Z_s\restriction_{r_{k+1}[s]}= \sigma$.

We say that $\mathcal{R}_{k+1}$ has *acted.* If 2(a) is taken, then we say that $\mathcal{R}_{k+1}$ has *failed the sum check.* This completes the description of Z_s.

2.6 Verification

Clearly, the values of the markers $r_0, r_1, \cdots$ are kept in increasing order. That is, at all stages s, if $r_k[s]\downarrow$, then $r_0[s] < r_1[s] < \cdots < r_k[s]$ are all defined. From now on when we talk about Z_s, we are referring to the fully constructed string at the end of stage s. It is also clear that the construction keeps $|Z_s| < s$ at each stage s.

Lemma 2.2. *Whenever step 2(b) is taken, we can always define $Z_s\restriction_{r_{k+1}[s]}$ for the relevant k and s.*

Proof. We drop s from notations, and proceed by induction on k. Let Υ be the collection of all possible candidates for $Z_s\restriction_{r_{k+1}}$, that is, $\Upsilon = \{\sigma : \sigma \supseteq Z\restriction_{r_k} \wedge |\sigma| = r_{k+1}\}$. Suppose that $k \geq 0$:

$$\sum_{\sigma\in\Upsilon}\sum_{\langle e,x\rangle\in S_{k+1}} \mu(U^e_x\cap[\sigma]) = \sum_{\langle e,x\rangle\in S_{k+1}}\sum_{\sigma\in\Upsilon}\mu(U^e_x\cap[\sigma])$$
$$\leq \sum_{\langle e,x\rangle\in S_{k+1}} \mu(U^e_x\cap[Z\restriction_{r_k}]) \leq \sum_{\langle e,x\rangle\in S_k} \mu(U^e_x\cap[Z\restriction_{r_k}]) + \sum_{x=r_k+1}^{r_{k+1}}\sum_{e\leq k+1}\mu(U^e_x)$$
$$\leq M_{r_k} + \sum_{x=r_k+1}^{r_{k+1}} 2^{-2x} \text{ (since } k\leq r_k) \leq M_{r_k} + \sum_{x=2r_k+1}^{r_k+r_{k+1}} 2^{-(1+x)}$$
$$= \sum_{x=r_{k+1}}^{2r_{k+1}} 2^{-(1+x)}2^{r_{k+1}-r_k} \text{ (adjusting the index } x) = M_{r_{k+1}}|\Upsilon|.$$

Hence, there must be some σ in Υ which passes the measure check in 2(b) for $Z\restriction_{r_{k+1}}$. A similar, but simpler counting argument follows for the base case $k = -1$, using the fact that the search now takes place above $Z\restriction_{r_k}= \langle\rangle$. □

Lemma 2.3. *For each e, the follower $r_e[s]$ eventually settles.*

Proof. We proceed by induction on e. Note that once $r_{e'}$ has settled for every $e' < e$, then $\mathcal{R}_e$ will get to act at every stage after that. Hence there is a stage s_0 such that

(i) $r_{e'}$ has settled for all $e' < e$, and
(ii) r_e receives a new value at stage s_0.

Note also that $\mathcal{R}_e$ will get a chance to act at every stage $t > s_0$, and the only reason why r_e receives a new value after stage s_0, is that $\mathcal{R}_e$ fails the sum check. Suppose for a contradiction, that $\mathcal{R}_e$ fails the sum check infinitely often after s_0.

Let $q(n-1)$ be the stage where $\mathcal{R}_e$ fails the sum check for the n^{th} time after s_0. In other words, $q(0), q(1), \cdots$ are precisely the different values assigned to r_e after s_0. Let $\mathcal{C}$ be the collection of all $k \leq e$ such that g_k is total, and d be a stage where $g_k(x)[d]$ has converged for all $k \leq e$, $k \notin \mathcal{C}$ and $x \in dom(g_k)$. We now define an appropriate computable function to contradict the hyperimmunity of A. Define the total computable function p by: $p(0) = 1 + \max\{s_0, d,$ the least stage t where $g_k(r_e[s_0])[t] \downarrow$ for all $k \in \mathcal{C}\}$. Inductively define $p(n+1) = 1+$ the least t where $g_k(p(n))[t] \downarrow$ for all $k \in \mathcal{C}$. Let $P(n) = \sum_{k \leq e} \sum_{x \leq p(n)} 2^{p(n)} g_k(x)[p(n+1)]$, which is the required computable function.

One can show by a simple induction, that $p(n) \geq q(n)$ for every n, using the fact that $\mathcal{R}_e$ is given a chance to act at every stage after s_0, as well as the restrictions we had placed on the functions $\{g_k\}$. Let N be such that $P(N) \leq h^A(N)$. At stage $q(N+1)$ we have $\mathcal{R}_e$ failing the sum check, so that $h^A(N) < h^A(q(N)) < \sum_{\langle k,x\rangle \in S_e} 2^{q(N)} g_k(x)$, where everything in the last sum is evaluated at stage $q(N+1)$. That last sum is clearly $< P(N) \leq h^A(N)$, giving a contradiction. □

Let $\hat{r}_e$ denote the final value of the follower r_e. Let $Z = \lim_s Z_s$. We now show that Z is not Demuth random relative to A. For each e and s, $Z_{s+1+\hat{r}_e} \upharpoonright_{\hat{r}_e}$ is defined, by Lemma 2.2.

Lemma 2.4. *For each e, $\#\{t \geq 1 + \hat{r}_e : Z_t \upharpoonright_{\hat{r}_e} \neq Z_{t+1} \upharpoonright_{\hat{r}_e}\} \leq h^A(\hat{r}_e)$.*

Proof. Suppose that $Z_{t_1} \upharpoonright_{\hat{r}_e} \neq Z_{t_2} \upharpoonright_{\hat{r}_e}$ for some $1 + \hat{r}_e \leq t_1 < t_2$. We must have $r_{e'}$ already settled at stage t_1, for all $e' \leq e$. Suppose that $Z_{t_2} \upharpoonright_{\hat{r}_e}$ is to the left of $Z_{t_1} \upharpoonright_{\hat{r}_e}$, then let e' be the least such that $Z_{t_2} \upharpoonright_{\hat{r}_{e'}}$ is to the left of $Z_{t_1} \upharpoonright_{\hat{r}_{e'}}$. The fact that $\mathcal{R}_{e'}$ didn't pick $Z_{t_2} \upharpoonright_{\hat{r}_{e'}}$ at stage t_1, shows that we must have a change of index for U^a_b between t_1 and t_2, for some $\langle a,b\rangle \in S_{e'} \subseteq S_e$. Hence, the total number of mind changes is at most $2^{\hat{r}_e} \sum_{\langle a,b\rangle \in S_e} g_a(b)$, where divergent values count as 0. $2^{\hat{r}_e}$ represents the number of times we can change our mind from left to right consecutively without moving back to the left, while $\sum_{\langle a,b\rangle \in S_e} g_a(b)$ represents the number of times we can move from right to left. Since $\mathcal{R}_e$ never fails a sum check after $\hat{r}_e$ is picked, it follows that the number of mind changes has to be bounded by $h^A(\hat{r}_e)$. □

By asking A' appropriate 1-quantifier questions, we can recover $Z = \lim_s Z_s$. Hence Z is well-defined and computable from A'. To see that Z is not Demuth random in A, define the A-Demuth test $\{V^A_x\}$ by the following: run the construction and enumerate $[Z_s \upharpoonright_x]$ into V^A_x when it is first defined. Subsequently each time we get a new $Z_t \upharpoonright_x$, we change the index for V^A_x, and enumerate the new $[Z_t \upharpoonright_x]$ in. If we ever need to change the index $> h^A(x)$ times, we stop and do nothing. By Lemma 2.4, Z will be captured by $V^A_{\hat{r}_e}$ for every e.

Lastly, we need to see that Z passes all $\{U_x^e\}$. Suppose for a contradiction, that $Z \in U_x^e$ for some e and $x > \hat{r}_e$. Let δ be such that $Z \in [\delta] \in U_x^e$, and let $e' \geq e$ such that $\hat{r}_{e'} > |\delta|$. Go to a stage in the construction where δ appears in U_x^e and never leaves, and $r_{e'} = \hat{r}_{e'}$ has settled. At every stage t after that, observe that $\langle e, x\rangle \in S_{e'}$, and that $\mathcal{R}_{e'}$ will get to act, at which point it will discover that $\mu(U_x^e \cap [Z\restriction_{\hat{r}_{e'}}]) = 2^{-\hat{r}_{e'}} > M_{\hat{r}_{e'}}$. Thus, $\mathcal{R}_{e'}$ never picks $Z\restriction_{\hat{r}_{e'}}$ as an initial segment for Z_t, giving us a contradiction. $\square$

References

1. Barmpalias, G., Downey, R., Greenberg, N.: K-trivial degrees and the jump-traceability hierarchy. Proceedings of the American Mathematical Society (to appear)
2. Barmpalias, G., Lewis, A., Ng, K.M.: The importance of Π_1^0-classes in effective randomness (submitted)
3. Barmpalias, G., Lewis, A., Stephan, F.: Π_1^0 classes, LR degrees and Turing degrees. Annals of Pure and Applied Logic
4. Bedregal, B., Nies, A.: Lowness properties of reals and hyper-immunity. In: WoLLIC 2003. Electronic Lecture Notes in Theoretical Computer Science, vol. 84 (2003)
5. Bickford, M., Mills, C.: Lowness properties of r.e. sets. Theoretical Computer Science (typewritten unpublished manuscript)
6. Cholak, P., Downey, R., Greenberg, N.: Strong jump-traceability I: The computably enumerable case. Advances in Mathematics 217, 2045–2074 (2008)
7. Demuth, O.: Remarks on the structure of tt-degrees based on constructive measure theory. Commentationes Mathematicae Universitatis Carolinae 29(2), 233–247 (1988)
8. Downey, R., Greenberg, N.: Strong jump-traceability II: The general case (in preparation)
9. Downey, R., Greenberg, N., Mihailović, N., Nies, A.: Lowness for computable machines. In: Computational Prospects of Infinity. Lecture Notes Series of the Institute for Mathematical Sciences, NUS, vol. 15, pp. 79–86 (2008)
10. Downey, R., Hirschfeldt, D., Nies, A., Stephan, F.: Trivial reals. In: Proceedings of the 7th and 8th Asian Logic Conferences, pp. 103–131. World Scientific, Singapore (2003)
11. Downey, R., Jockusch, C., Stob, M.: Array nonrecursive sets and multiple permitting arguments. Recursion Theory Week 1432, 141–174 (1990)
12. Downey, R., Ng, K.M.: Splitting into degrees with low computational strength (in preparation)
13. Figueira, S., Nies, A., Stephan, F.: Lowness properties and approximations of the jump. In: Proceedings of the Twelfth Workshop of Logic, Language, Information and Computation (WoLLIC 2005). Electronic Lecture Notes in Theoretical Computer Science, vol. 143, pp. 45–57 (2006)
14. Franklin, J., Stephan, F.: Schnorr trivial sets and truth-table reducibility. Technical Report TRA3/08, School of Computing, National University of Singapore (2008)
15. Greenberg, N., Miller, J.: Lowness for Kurtz randomness. Journal of Symbolic Logic (to appear)
16. Hirschfeldt, D., Nies, A., Stephan, F.: Using random sets as oracles (to appear)
17. Kjos-Hanssen, B., Nies, A.: Superhighness (to appear)

18. Kjos-Hanssen, B., Nies, A., Stephan, F.: Lowness for the class of Schnorr random sets. Notre Dame Journal of Formal Logic 35(3), 647–657 (2005)
19. Ng, K.M.: Ph.D Thesis (in preparation)
20. Ng, K.M.: Strong jump traceability and beyond (submitted)
21. Ng, K.M.: On strongly jump traceable reals. Annals of Pure and Applied Logic 154, 51–69 (2008)
22. Nies, A.: On a uniformity in degree structures. In: Complexity, Logic and Recursion Theory. Lecture Notes in Pure and Applied Mathematics, February 1997, pp. 261–276 (1997)
23. Nies, A.: Reals which compute little. CDMTCS Research Report 202, The University of Auckland (2002)
24. Nies, A.: Lowness properties and randomness. Advances in Mathematics 197, 274–305 (2005)
25. Nies, A.: Computability And Randomness. Oxford University Press, Oxford (2006) (to appear)
26. Slaman, T., Solovay, R.: When oracles do not help. In: Fourth Annual Conference on Computational Learning Theory, pp. 379–383 (1971)
27. Stephan, F., Liang, Y.: Lowness for weakly 1-generic and Kurtz-random. In: Cai, J.-Y., Cooper, S.B., Li, A. (eds.) TAMC 2006. LNCS, vol. 3959, pp. 756–764. Springer, Heidelberg (2006)
28. Terwijn, S., Zambella, D.: Algorithmic randomness and lowness. Journal of Symbolic Logic 66, 1199–1205 (2001)