CHARACTERIZING LOWNESS FOR DEMUTH RANDOMNESS

LAURENT BIENVENU, ROD DOWNEY, NOAM GREENBERG, ANDRÉ NIES, AND DAN TURETSKY

Abstract. We show the existence of noncomputable oracles which are low for Demuth randomness, answering a question in [16] (also Problem 5.5.19 in [36]). We fully characterize lowness for Demuth randomness using an appropriate notion of traceability. Central to this characterization is a partial relativization of Demuth randomness, which may be more natural than the fully relativized version. We also show that an oracle is low for weak Demuth randomness if and only if it is computable.

1. Introduction

Tools from computability theory are used to answer the question "when is an infinite binary string random?". By either using effective betting strategies, effectively presented null sets, or effective descriptions of initial segments, definitions such as Martin-Löf's, Schnorr's and others' give rise to a hierarchy of notions of randomness. The rich field of algorithmic randomness classifies the levels of this hierarchy and, among other pursuits, attempts to understand the behaviour of the Turing degrees of random sets. Prominent examples are the theorem, independently proved by Kučera [28] and Gács [21], that every set is Turing-reducible to a random set and, in contrast, Stephan's result [40], showing that random sets with high information content are atypical.

The interaction between computability and randomness, though, is bidirectional: it is used not only to understand randomness, but also to understand computability itself. Starting with Kučera's seminal work [28], in which he used randomness to give an injury-free solution to Post's problem, the study of randomness has been used to directly answer questions about Turing degrees and computability in general. Furthermore, it has yielded new notions, such as traceability, which turned out to be essential ingredients in our understanding of the Turing degrees. For example, the notion of strong jump traceability, which arose from algorithmic randomness, was used in [13] to answer a long-standing question regarding the inversion of pseudojump operators to c.e. degrees. Another example is Ishmukhametov's use of traceability [25] to classify the c.e. degrees which have strong minimal covers.

Central to this interaction is the notion of lowness for a randomness notion $\mathcal{C}$. An oracle A is said to be *low for* $\mathcal{C}$ if every $\mathcal{C}$-random set is also $\mathcal{C}$-random relative to A. This is a notion of computational weakness: it says that the oracle A cannot detect regularities in any $\mathcal{C}$-random set. The study of lowness for $\mathcal{C}$ gives, on the one hand, an understanding of the notion of randomness $\mathcal{C}$ and its relativization to an oracle; and on the other hand, gives us insight about what it means to have little power as an oracle. When there are non-computable oracles which are low for $\mathcal{C}$, these can be viewed as 'closed to computable' and are usually very far from the halting problem $\emptyset'$. When there are none, then the coincidence of computability and lowness for $\mathcal{C}$ captures computability itself, using analytic means.

Downey, Greenberg, Nies and Turetsky were supported by Marsden grants from the Royal Society of New Zealand, Turetsky as a postdoctoral fellow.

Given the potential benefits of the study of lowness, it is not surprising that a lot of work has gone into characterizing lowness for randomness notions. The most useful notion of randomness remains that given by Martin-Löf. Dually, lowness for Martin-Löf randomness turned out to give a fascinating class of oracles, also known as the *K-trivials*. The similarities between constructions of Kučera and Terwijn [29] and of Mučnik's (unpublished, see [15, Theorem 11.2.5]) led to Nies's characterization [35] of lowness for randomness in terms of K-triviality; the robustness of this class was further exhibited by its coincidence with lowness for weak 2-randomness (Downey, Nies, Weber and Yu [17]). In contrast, Nies [35] showed that the only oracles that are low for computable randomness are the computable sets.

An important distinction, given a notion of randomness $\mathcal{C}$, is between lowness for $\mathcal{C}$ and *lowness for $\mathcal{C}$-tests*. Most randomness notions studied in the literature are defined by specifying that a $\mathcal{C}$-random set is one which avoids a countable class of null sets, the effectively-presented (in the sense of $\mathcal{C}$) null sets. Usually, each such null set is presented as the limit of a *$\mathcal{C}$-test*, which is a sequence $\langle \mathcal{U}_n \rangle$ of open sets, whose measure quickly tends to 0 [1]. The corresponding null set is the collection of reals which belong to infinitely many sets $\mathcal{U}_n$. Such reals are said to be *captured* by the test. Often, the sets $\mathcal{U}_n$ are nested, in which case the corresponding null set is simply the intersection $\bigcap_n \mathcal{U}_n$.

We say that an oracle A is *low for $\mathcal{C}$-tests* if every $\mathcal{C}$-null set relative to A is contained (or *covered*) by a $\mathcal{C}$-null set. In other words, for every $\mathcal{C}$-test $\langle \mathcal{U}_n^A \rangle$ relative to A, there is a $\mathcal{C}$-test $\langle \mathcal{V}_n \rangle$ which covers $\langle \mathcal{U}_n^A \rangle$, in the sense that every real which is an element of infinitely many sets $\mathcal{U}_n^A$, is also an element of infinitely many sets $\mathcal{V}_n$. The point is that usually, the extra computational power of the oracles allows it to design tests which capture more reals; an oracle is low for tests if the tests relative to A do not in fact produce larger null sets than the ones which are specified without access to an oracle.

Certainly, every oracle which is low for $\mathcal{C}$-tests is also low for $\mathcal{C}$. Equivalence of these two notions is immediate if there is a universal $\mathcal{C}$-test, that is, a greatest $\mathcal{C}$-null set, which captures precisely the non-$\mathcal{C}$-random sets. Thus, for example, it is immediate that lowness for Martin-Löf randomness is the same as lowness for Martin-Löf tests. However, most notions of randomness (such as Schnorr, Kurtz or computable randomness, as well as weak 2-randomness) do not admit universal tests. Nonetheless, for every notion of randomness $\mathcal{C}$ studied so far, lowness for $\mathcal{C}$ and lowness for $\mathcal{C}$-tests coincide[2]. This intriguing phenomenon is observed empirically; we still do not have a deep unifying reason for all of these coincidences, even though Bienvenu and Miller [6] gave such a unified view for a wide class of randomness notions (including Martin-Löf randomness, computable randomness and Schnorr randomness). This equivalence is usually not proved directly, but passes through a third characterization of the two notions.

An exemplifying case is that of Schnorr randomness. First, Terwijn and Zambella [42] characterized lowness for Schnorr tests by a property called computable (or recursive) traceability. Only later, Kjos-Hanssen, Nies and Stephan [27] showed that lowness for Schnorr randomness also coincides with computable traceability, and so coincides with lowness for Schnorr tests. Unlike the case of K-triviality, this characterization is purely computability-theoretic; the definition of computable

[1] There are some exceptions to this principle, for example for the notion of weak-2-randomness.

[2] Diamondstone and Franklin [19] recently gave an example of a randomness notion – difference randomness – for which lowness for tests is strictly stronger than lowness for randomness. However, the "tests" involved in difference randomness do not belong to the usual family of tests. A difference test is indeed a sequence of sets $\langle \mathcal{U}_n \rangle$ whose measure tends to 0 quickly, but instead of being open, each $\mathcal{U}_n$ is the *set-theoretic difference* of two effectively open sets.

traceability does not involve notions from randomness. The same applies to the characterization, by Greenberg and Miller ([22], using work of Stephan and Yu [41]) of lowness for Kurtz randomness and of lowness for Kurtz tests as the intersection of the hyperimmune-free (**0**-dominated) degrees with the degrees which do not compute a function always avoiding the jump function $e \mapsto \Phi_e(e)$ (the non-DNR oracles).

Demuth randomness was first introduced by Demuth [8, 9]. It turned out to be too strong for his original purpose by work of [4, 5]. Nonetheless, it has been shown to be a very fruitful notion in algorithmic randomness by recent work of Diamondstone, Greenberg and Turetsky [11, 23], and Kučera and Nies [30].

In this paper, we study the two lowness properties (for tests and for randomness) associated to Demuth randomness. Most lowness properties for randomness can be characterized by a combinatorial notion of traceability, and indeed we shall give such a characterization for lowness for Demuth randomness (and show that it coincides with lowness for Demuth tests). The techniques we use go well beyond the state of the art in the study of lowness. For example, the forcing construction of Section 3.4 needs a much finer measure-theoretic analysis (with the use of Chernoff bounds) than the arguments of the same type that previously appeared in the literature. We hope that these techniques will be useful for eventually giving a unifying explanation for the coincidence of lowness for randomness and lowness for tests. We use the characterization to show the existence of non-computable oracles that are low for Demuth randomness, answering a question in [16] (also Problem 5.5.19 in [36]).

Demuth randomness. Demuth was primarily interested in various kinds of effective null classes because of their role in constructive mathematical analysis. For instance, he studied the differentiability of constructive functions defined on the unit interval. (The functions he considered were constructive in the Russian sense; in modern parlance they are referred to as Markov-computable). His notion of randomness was sufficiently strong to ensure that every constructive function f satisfies the Denjoy alternative at every Demuth-random point. See Kučera and Nies [31, Def. 11] for a discussion of Demuth's original definition, and [36, Section 3.6] for more background.

Compared to Martin-Löf's, Demuth's idea is to allow changing the whole n^{th} component $\mathcal{U}_n$ of a test (this is an effectively open subset of Cantor space of measure at most 2^{-n}) a computably bounded number of times. A real is then captured by the test $\langle \mathcal{U}_n \rangle$ if it lies in infinitely many of the *final versions* of the sets $\mathcal{U}_n$.

We give a formal definition. Recall that a function $f : \omega \to \omega$ is computable from the halting problem $\emptyset'$ if and only if it has a *computable approximation*, that is, a uniformly computable sequence of functions $\langle f_s \rangle$ which pointwise converges to f (i.e., for all but finitely many s, we have $f_s(x) = f(x)$). Of course this means that the *number of mind-changes* $\#\{s : f_{s+1}(x) \neq f_s(x)\}$ is finite for every x. If we further require that this number of mind-changes, as a function of x, be bounded by some computable function, then we get the notion of *ω-computably approximable* functions (or ω-c.a.); this is the class Δ^{-1}_{ω} of the Ershov hierarchy.[3]

Definition 1.1. A *Demuth test* is an effective sequence $\langle \mathcal{U}_n \rangle$ of effectively open (Σ^0_1) subsets of Cantor space such that:

(1) For all n, the measure $\lambda(\mathcal{U}_n)$ of $\mathcal{U}_n$ is bounded by 2^{-n}; and
(2) there is an ω-c.a. function mapping n to a Σ^0_1 index for $\mathcal{U}_n$.

As mentioned above, the notion of test gives notions of null sets and of randomness. A set (an element of Cantor space) X is captured by a Demuth test $\langle \mathcal{U}_n \rangle$ if $X \in \mathcal{U}_n$

[3] The popular but not quite standard term *ω-c.e.* was reserved by Ershov to denote the class Σ^{-1}_{ω}, which is the natural generalisation of the classes of n-c.e. sets.

for infinitely many n. A set is *Demuth random* if it is not captured by any Demuth test.

While Demuth randomness is now known to be strictly stronger than is necessary for characterizing the Denjoy alternative (Bienvenu, Hölzl and Nies [4, 5]), Demuth randomness turns out to be of interest on its own. Lying between weak 2-randomness and Martin-Löf randomness, it shares some pleasing properties of 1-genericity. Unlike Martin-Löf random sets, Demuth random sets cannot compute the halting problem; in fact they are all generalized low. Unlike weakly 2-random sets, Demuth random sets can be Δ^0_2. This allows Demuth random sets to interact with c.e. degrees, in the style of Kučera. For instance, the strongly jump-traceable c.e. sets were characterized (in one direction by Kučera and Nies [30], in the other by Greenberg and Turetsky [23]) as the c.e. sets computable from a Demuth random set.

Traceability. Traces for functions from ω to ω were first introduced in set theory by Bartoszyński (see [2]), where he called them slaloms. He used them for forcing results related to cardinal characteristics of the continuum.

In computability, traceability is a measure of weakness of an oracle. An oracle A will be called traceable if the values of any function ψ in some class of functions computed by A can be captured in finite sets of small size. Formally, a *trace* for a partial function ψ is a sequence $\langle T_n \rangle$ of finite sets such that for all $n \in \mathrm{dom}\psi$, $\psi(n) \in T_n$. The point is that the complexity of the trace $\langle T_n \rangle$ is smaller than the complexity of ψ; while we need A to compute ψ, the trace can be generated computably, with no access to the oracle A. Traceability, then, would say that the oracle A is so weak so that it cannot compute a function which escapes effective traces.

The different notions of traceability vary as we specify:

(1) the class of functions computed by A which are all traced;
(2) the complexity of the trace; and
(3) the rate of growth required of the size of the components of the trace.

The rate of growth is usually calibrated by Schnorr's *order functions*. Recall that an order function is a computable function from ω to $\omega \setminus \{0\}$ (so it only takes positive values) which is non-decreasing and unbounded. A trace $\langle T_n \rangle$ is *bounded* by an order function h if for all n, $\#T_n \leqslant h(n)$. We also say that $\langle T_n \rangle$ is an *h-trace*.

For example, Terwijn and Zambella [42] called an oracle A *computably traceable* if for some fixed order function h, each total function that A computes has an h-trace $\langle T_n \rangle$ such that a strong index for T_n can be effectively computed from n (the strong index not only gives us a way to compute T_n, but also gives a bound on its elements). In another example which was mentioned above, Figueira, Nies and Stephan [18] defined an oracle A to be *strongly jump traceable* if for *every* order function h, every *A-partial computable* function has a uniformly c.e. h-trace (so unlike computable traceability, here from n we only have a method for enumerating the elements of T_n, but not for computing the set T_n). For more background see [36, Sections 8.2 and 8.4].

We introduce a notion of tracing, *Demuth traceability*, which will be instrumental in characterizing lowness for Demuth randomness. In an analogue to the move from Martin-Löf randomness to Demuth randomness, Demuth traceability is a modification of computable traceability which allows finitely many changes to both the values of the functions being traced and the components of the trace, but the number of changes in both needs to be computably bounded. To formalize this notion, we recall a generalization of the notion of ω-c.a. functions due to Cole and Simpson [7].

Definition 1.2. Let A be an oracle (an element of Cantor space). A function $f\colon \omega \to \omega$ is *bounded limit-recursive* in A, in short $\mathsf{BLR}\langle A\rangle$, if there is a uniformly A-computable sequence $\langle f_s\rangle$ of functions converging to f, such that the mind-change function $n \mapsto \#\{s : f_{s+1}(n) \neq f_s(n)\}$ is bounded by a computable function.

Equivalently, f is $\mathsf{BLR}\langle A\rangle$ if it is computable from $A' \equiv_{\mathrm{T}} A' \oplus A$, where the A'-use is bounded by a computable function, but there is no such restriction on the A-use.

Note that a function is $\mathsf{BLR}\langle\emptyset\rangle$ if and only if it is ω-c.a.

Definition 1.3. An oracle A is *Demuth traceable* if there is an order function h, such that every function f which is $\mathsf{BLR}\langle A\rangle$ has an h-trace $\langle T_n\rangle$ such that there is an ω-c.a. function taking n to a c.e. index for T_n.

In passing, we remark that Cole and Simpson characterized the oracles A such that $\mathsf{BLR}\langle A\rangle = \mathsf{BLR}\langle\emptyset\rangle$ as those which are both superlow and jump traceable. The equation $\mathsf{BLR}\langle A\rangle = \mathsf{BLR}\langle\emptyset\rangle$ is the same as saying that A is Demuth traceable, but with the bound h not being an order function but the constant function 1. For more, see Section 5.

Although there are uncountably many Demuth traceable sets (Theorem 1.5), some of our results (see Section 4) indicate that the class of Demuth traceable sets is small, especially if we intersect them with the hyperimmune-free (computably dominated) sets, those sets which only compute functions which are bounded by computable functions. This class in fact characterizes lowness for Demuth randomness:

Theorem 1.4. *The following are equivalent for an oracle A:*

(1) *A is low for Demuth tests.*
(2) *A is low for Demuth randomness.*
(3) *A is both Demuth traceable and computably dominated.*

The first step toward Theorem 1.4 was taken by Downey and Ng [16], who showed that every oracle which is low for Demuth randomness is computably dominated. As we shall shortly see, we make use of their result in the proof of Theorem 1.4. Theorem 1.4, however, also helps to settle the question of the *existence* of noncomputable oracles which are low for Demuth randomness, a fact which has eluded researchers up to now. To this end, we prove the following theorem:

Theorem 1.5. *There is a Π^0_1 class consisting of noncomputable oracles which are all Demuth traceable.*

A Π^0_1 class with no computable elements is necessarily perfect, and so we get $2^{\aleph_0}$-many Demuth traceable sets. We get even more:

Corollary 1.6. *There is a perfect set of noncomputable oracles, all of which are low for Demuth randomness.*

Proof of Corollary 1.6, assuming Theorem 1.5. Let $\mathcal{P}$ be a Π^0_1 class with no computable elements, consisting of sets which are all Demuth traceable. The hyperimmune-free basis theorem of Miller and Martin [32], performed carefully so as to preserve splittings, yields a perfect subclass $\mathcal{Q} \subset \mathcal{P}$ consisting of computably dominated oracles. Theorem 1.4 ensures that $\mathcal{Q}$ is as required. □

Partial relativization. A key concept underlying much of this work is that of *partial relativization*. As the name suggests, this is the result of relativizing to an oracle only some aspect of a computable notion, while leaving other aspects unrelativized. In effect, we study what happens under restricted access to the oracle. The idea originated implicitly in [7], was further developed in [38], and studied in [1].

An example of a partial relativization was already given in the notion of bounded limit-recursive functions, Definition 1.3. A full relativization to an oracle A of the notion of ω-c.a. functions would be similar to the definition of $\mathsf{BLR}\langle A\rangle$, except that the number of mind-changes would be required only to be bounded by an A-computable function, not by a computable function. Partial relativizations are also used to define so-called *weak reducibilities* associated with lowness notions, a prime example of which is the notion of LR-reducibility, the reducibility associated with lowness for Martin-Löf randomness.

It turns out that partial relativization of randomness notions themselves is often useful as well. While all reasonable relativizations of Martin-Löf randomness coincide, this is not the case for other notions of randomness. For example, Miyabe [33], following work of Downey, Griffiths and LaForte [14], examined a partial relativization of Schnorr randomness with only truth-table access to the oracle. Here we suggest a partial relativization of Demuth randomness.

Definition 1.7. Let A be an oracle. A $\text{Demuth}_{\text{BLR}}\langle A\rangle$*-test*, or a Demuth test *by*[4] A, is a sequence $\langle \mathcal{U}_n^A \rangle$ of $\Sigma^0_1(A)$ subsets of Cantor space such that for all n, $\lambda\left(\mathcal{U}_n^A\right) \leqslant 2^{-n}$, and there is a $\mathsf{BLR}\langle A\rangle$-function taking n to a $\Sigma^0_1(A)$-index for $\mathcal{U}_n^A$.

A set is $\text{Demuth}_{\text{BLR}}\langle A\rangle$-random if it is not captured by any $\text{Demuth}_{\text{BLR}}\langle A\rangle$-test.[5]

So the difference between $\text{Demuth}_{\text{BLR}}\langle A\rangle$ tests and (fully relativized) Demuth^A tests is that in the former, the number of changes of a component $\mathcal{U}_n^A$ of the test is bounded by a computable function, and in the latter by an A-computable function. In both, though, the function which takes a pair (n, s) to the index for the version of $\mathcal{U}_n^A$ at stage s, is A-computable. It is easy to see that every $\text{Demuth}_{\text{BLR}}\langle A\rangle$-test is also a Demuth^A-test, and so every set which is Demuth random relative to A is also $\text{Demuth}_{\text{BLR}}\langle A\rangle$-random. If A is computably dominated, then bounding by A-computable functions and bounding by computable functions are the same, and so the notions coincide. In particular, the $\text{Demuth}_{\text{BLR}}\langle\emptyset\rangle$-random sets are precisely the Demuth random sets.

There are two ways to think about the different relativizations of a notion of randomness. One is to accept that a randomness notion should specify its relativizations. That is, we extend our understanding of what a notion of randomness is, from a mere class of random reals, to a binary relation between reals and oracles, saying which reals are random relative to which oracle. Under this interpretation, Demuth randomness and $\text{Demuth}_{\text{BLR}}$ randomness are two distinct notions of randomness, which coincide on the computable oracles. It then makes sense to speak of lowness for $\text{Demuth}_{\text{BLR}}$ and for $\text{Demuth}_{\text{BLR}}$-tests. Namely, an oracle A is low for $\text{Demuth}_{\text{BLR}}$ if every Demuth random set is $\text{Demuth}_{\text{BLR}}\langle A\rangle$-random, and low for $\text{Demuth}_{\text{BLR}}$ tests if every $\text{Demuth}_{\text{BLR}}\langle A\rangle$-test can be covered by a Demuth test.

Another line of thinking still tries to choose, among the various possible relativizations of a notion of randomness, the most useful or natural one. Miyabe [33], for example, showed that the truth-table version of Schnorr randomness, which we mentioned above, satisfies van Lambalgen's theorem, while the theorem is known to fail for the full relativization of Schnorr randomness. Miyabe suggested that

[4] While full relativization of a notion to an oracle A is indicated with the phrase "relative to A" or "in A", or simply prefixed by A, partial relativization is often indicated with the phrase "by A". So for example, the functions which are $\mathsf{BLR}\langle A\rangle$ are the functions which are ω-c.a. by A.

[5] We remind the reader that for the capturing / passing criterion we take Solovay's, as we do not assume that the sequence is nested. Weak passing — avoiding the intersection $\bigcap_n \mathcal{U}_n$ — gives rise to a weaker notion of randomness, discussed later.

satisfying van Lambalgen's theorem is a criterion for identifying the "correct" relativization of a notion of randomness. In this context, recently Bienvenu, Diamondstone, Greenberg and Turetsky [3] showed that van Lambalgen's theorem holds for $\mathrm{Demuth}_{\mathrm{BLR}}$, while it fails for the full relativization of Demuth randomness. Further evidence for the usefulness of $\mathrm{Demuth}_{\mathrm{BLR}}$ is the simpler characterization of lowness:

Theorem 1.8. *The following are equivalent for an oracle A:*
(1) *A is low for* $\mathrm{Demuth}_{\mathrm{BLR}}$ *tests.*
(2) *A is low for* $\mathrm{Demuth}_{\mathrm{BLR}}$ *randomness.*
(3) *A is Demuth traceable.*

Indeed, Theorem 1.8 seems to be the fundamental one, and Theorem 1.4 is an easy corollary of Theorem 1.8, using Downey and Ng's result mentioned above:

Proof of Theorem 1.4, given Theorem 1.8. (1)→(2): As with every notion of randomness, every oracle which is low for Demuth tests is also low for Demuth randomness.

(2)→(3): Suppose that A is low for Demuth randomness. Then A is also low for $\mathrm{Demuth}_{\mathrm{BLR}}$ randomness, and so is Demuth traceable. By Downey and Ng's [16], A is also computably dominated.

(3)→(1): Suppose that A is both Demuth traceable and computably dominated. Then A is low for $\mathrm{Demuth}_{\mathrm{BLR}}$ tests. Because A is computably dominated, every Demuth test relative to A is actually a $\mathrm{Demuth}_{\mathrm{BLR}}\langle A\rangle$-test, and so is covered by a Demuth test. Hence A is low for Demuth tests. □

The following observation is due to D. Diamondstone and Nies. We say that an oracle A is Demuth cuppable if there is a Demuth random Y such that $\emptyset' \leqslant_{\mathrm{T}} A \oplus Y$.

Corollary 1.9. *Suppose A is Demuth traceable. Then A is not Demuth cuppable.*

Proof. In [36, Thm. 3.6.26] it is shown that each Demuth random set Z is generalized low, that is, $Z' \leqslant_{\mathrm{T}} Z \oplus \emptyset'$. Actually the proof yields an ω-c.a. function f dominating J^Z, the jump of Z.

If Y is Demuth random, then Y is $\mathrm{Demuth}_{\mathrm{BLR}}\langle A\rangle$-random by Theorem 1.8. The proof of [36, Thm. 3.6.26] builds a Demuth test with at most 2^m changes to the m-th version. Hence we may "partially relativize" to A this proof in order to obtain a function f in $\mathsf{BLR}\langle A\rangle$ dominating $J^{A\oplus Y}$. Since A is Demuth traceable, this function has a Δ^0_2 upper bound. So $A \oplus Y \not\geqslant_T \emptyset'$. (In fact, this argument shows that $A \oplus Y$ is generalized low.) □

The BLR transform. The main concepts in this paper are obtained by applying what we call a *BLR-transform* to a computability theoretic concept. We replace certain computable functions (but not size bounds) by ω-c.a. functions. For instance, the BLR-transform of computable traceability is BLR traceability. We replace the condition $f \leqslant_{\mathrm{T}} X$ in its definition by the condition $f \in \mathsf{BLR}\langle X\rangle$, which by the nature of BLR yields a partial relativization. With a slight adjustment of definitions, the BLR-transform of Schnorr randomness is Demuth randomness. This may explain that a lot of our results on BLR traceability and Demuth randomness are parallel to the investigations of computable traceability and Schnorr randomness [42, 27] we discussed above. The methods are also parallel but get more complex because we are dealing with more involved concepts.

1.1. **The content of the paper.** We start, in Subsection 1.2, with reviewing notation and simplifying the tests we work with. In Section 2 we prove Theorem 1.5.

In Section 3 we prove Theorem 1.8. One of the main tools we use is forcing with Demuth closed sets. This is analogous to the characterization by Kautz of weak 2-randomness in terms of generic filters for forcing with Π^0_2 closed sets of positive measure (see [15, Theorem 7.2.28]), and gives further evidence for the ease of working with Demuth randomness.

In Section 4 we give evidence for the thesis that the class of Demuth traceable oracles is small. In particular, we discuss the relationship between jump traceability and Demuth traceability. We show that the latter is strictly stronger; indeed we separate Demuth traceability from jump traceability in both the ω-c.a. degrees, and in the computably dominated degrees. This shows that in some sense, the computably dominated Demuth traceable sets are an analogue of the strongly jump traceable sets, outside the Δ^0_2 degrees. Note that in turn, the computably dominated, jump traceable sets are contained in the computably traceable sets (see [27]), and the containment is strict, because the jump traceable sets are always generalized low, while computably traceable sets need not be generalized low; see [36, after Cor. 8.4.7].

An argument of Terwijn and Zambella easily holds for Demuth traceability, showing that for the uniform bound h in the definition of Demuth traceability we may choose any bound we like. However, unlike all other traceability notions, we can strengthen Demuth traceability by requiring that the bound h be constant. In Section 5 we show that we get a strict hierarchy as the constant bound changes; as mentioned above, the first level of this hierarchy, the sets which are traceable with bound 1, are the superlow and jump traceable sets investigated by Cole and Simpson.

Finally, in Section 6 we consider a weakening of Demuth randomness, called weak Demuth randomness. Lowness in this context is starkly different from lowness for Demuth randomness: we show that the only oracles that are low for weak Demuth randomness are the computable ones.

1.2. **Clopen tests.** We fix some notation. Recall that a set of binary strings $W \subseteq 2^{<\omega}$ defines an open subset of Cantor space

$$[W]^{\prec} = \{Z \in 2^{\omega} \colon\ \exists n\ (Z\restriction_n \in W)\}.$$

Let $(W^X_e)_{e\in\omega}$ be an effective listing of sets of strings that are c.e. in an oracle X. For each index e and oracle X, we let $\mathcal{W}^X_e = [W^X_e]^{\prec}$ be the e^{th} $\Sigma^0_1(X)$ subset of Cantor space. Thus, for an oracle A, a Demuth^A test is a sequence $\langle \mathcal{U}^A_n\rangle_{n<\omega} = \langle \mathcal{W}^A_{g(n)}\rangle_{n<\omega}$, where $\lambda(\mathcal{W}^A_{g(n)}) \leqslant 2^{-n}$ and g is a function which is ω-c.a. relative to A; while such a sequence is a $\text{Demuth}_{\text{BLR}}\langle A\rangle$-test (a Demuth test *by* A) if g is a $\mathsf{BLR}\langle A\rangle$ function. If $\langle g_s\rangle$ is an A-computable approximation to the function g which witnesses that g is ω-c.a. in A, or $\mathsf{BLR}\langle A\rangle$, then for any t and n, we let $\mathcal{U}^A_n[t] = \mathcal{W}^A_{g_t(n)}$, the version of $\mathcal{U}^A_n$ at stage t.[6]

Unlike Schnorr or Martin-Löf randomness, the fact that we are allowed to change the components of a test allows us to simplify the structure of these components. Namely, we may assume that they are all clopen subsets of Cantor space, and moreover, that we have a strong index for these clopen sets. We fix an effective list $\langle C_n\rangle$ of finite subsets of $2^{<\omega}$, given by strong indices; we then let $\mathcal{C}_n = [C_n]^{\prec}$. So $\langle \mathcal{C}_n\rangle_{n<\omega}$ is an effective list of all clopen subsets of Cantor space.

Definition 1.10. A *clopen Demuth test* is a sequence $\langle \mathcal{C}_{g(n)}\rangle_{n<\omega}$, where $\lambda(\mathcal{C}_{g(n)}) \leqslant 2^{-n}$ and g is an ω-c.a. function.

Hölzl, Kräling, Stephan and Wu noticed that by passing excess measure to later test components, and (computably) increasing the number of changes allowed, clopen Demuth tests are sufficient to determine Demuth randomness.

[6]We do not require this to be clopen: we do not mean $\mathcal{W}^A_{g_t(n),t}$.

Proposition 1.11 (Thm. 11, (a)↔(b), of [24]). *Every Demuth test is covered by some clopen Demuth test.*

For a more detailed argument see [23].

In fact, the argument for Proposition 1.11 relativizes in both ways. For an oracle A, a clopen Demuth^A test is a test $\langle \mathcal{C}_{g(n)} \rangle$ as above, with g being ω-c.a. in A; and a clopen Demuth$_{\text{BLR}}\langle A\rangle$ test is one with g being $\mathsf{BLR}\langle A\rangle$. Then every Demuth^A test is covered by a clopen Demuth^A test, and every Demuth$_{\text{BLR}}\langle A\rangle$ test is covered by a clopen Demuth$_{\text{BLR}}\langle A\rangle$ test. Note that in passing from clopen Demuth tests to either form of clopen A-tests, the only ingredient which is changed is the complexity of the index function g. Contrast this with passing from Demuth tests to A-tests, where we also allow to increase the complexity of the test components, from Σ^0_1 open sets to $\Sigma^0_1(A)$ open sets; covering by clopen sets shows that this increase in complexity is not fundamental, and that the real extra power given by an oracle resides wholly in the complexity of the function giving the indices of the components of the test.

Similarly, we observe that we may use strong indices, rather than c.e. indices, in the definition of Demuth traceability. For brevity, call a trace $\langle T_n\rangle$ an *ω-c.a. trace* if $T_n = W_{g(n)}$ for some ω-c.a. function g. So an oracle A is Demuth traceable if there is an order function h such that every function which is $\mathsf{BLR}\langle A\rangle$ has an h-bounded ω-c.a. trace.

Let $\langle D_n\rangle$ be an effective sequence of all finite subsets of ω, given by strong indices. A *strong ω-c.a. trace* is a trace of the form $\langle D_{f(n)}\rangle$ for some ω-c.a. function A. For any order function h, every function which has an h-bounded ω-c.a. trace also has a strong h-bounded ω-c.a. trace; we simply allow more changes to the trace components T_n, and each time a new element is enumerated into T_n we declare a new strong index for $T_n[s]$. The number of extra changes is bounded by the product of h and the original bound on the number of changes in the indices of T_n. In short, the notion of Demuth traceability can be defined using strong ω-c.a. traces.

We will make use of a fact, mentioned above, which is proved by the same argument given by Terwijn and Zambella's [42] – that in the definition of Demuth traceability, the choice of order function does not matter. That is, if A is Demuth traceable, then for every order function h, every $\mathsf{BLR}\langle A\rangle$ function has an h-bounded ω-c.a. trace.

2. A perfect class of Demuth traceable sets

In this section we prove Theorem 1.5: we show that the class of noncomputable Demuth traceable sets contains a nonempty Π^0_1 class. As we shall see later (Section 4), this strengthens a theorem of Nies (see [36, Thm. 8.4.4 and Exercise 8.4.6]) stating that the jump traceable sets contain a perfect Π^0_1 class. As noted above, any Π^0_1 class with no computable elements (also called a *special* Π^0_1 class) is perfect.

Proof of Theorem 1.5. We will build a class $\mathcal{P}$. To ensure that every $X \in \mathcal{P}$ is Demuth traceable, we will build a trace for every $f \in \mathsf{BLR}\langle X\rangle$. To do this, we will need an enumeration of all such functions f, which we obtain by enumerating the functionals which generate them from the oracles X.

Specifically, we fix a computable enumeration of pairs $\{(\Gamma_e, g_e)\}_{e\in\omega}$ such that for each e,

- g_e is a partial computable function;
- Γ_e is a functional and Γ^X_e is total for every oracle $X \in 2^\omega$;
- $\#\{t \mid \Gamma^X_e(n,t) \neq \Gamma^X_e(n,t+1)\} < g_e(n)$ for every n such that $g_e(n)\downarrow$ and every oracle X.

We let $f_e^X(n) = \lim_t \Gamma_e^X(n,t)$. We arrange this enumeration such that if $f \in \mathsf{BLR}\langle X\rangle$ for some X, then $f = f_e^X$ for some e.

We will build ω-c.a. traces $\{\langle T_n^e\rangle_{n\in\omega}\}_{e\in\omega}$. For all i, we need to meet the requirement:

R_i**:** ϕ_i is not a computable description of a set in $\mathcal{P}$.

For every pair (e, n) with $e < n$, we need to meet the requirement:

$Q_{e,n}$**:** If $g_e(n)\downarrow$, then $f_e^X(n) \in T_n^e$ for all sets $X \in \mathcal{P}$.

These requirements will suffice to prove the theorem.

The basic idea: To trace f_e^X with a sequence $\langle T_n^e\rangle_{n\in\omega}$ we use restraint. When we see a $\sigma_0 \in 2^{<\omega}$ such that $\Gamma_e^{\sigma_0}(n, t_0) = c_0$ converges for some t_0 and c_0, we require that all elements of $\mathcal{P}$ extend σ_0 by removing all elements which do not, and we put $T_n^e = \{c_0\}$. When we later see a σ_1 extending σ_0 which makes $\Gamma_e^{\sigma_1}(n, t_1) = c_1$ converge for some $t_1 > t_0$ and some $c_1 \neq c_0$, we then require that all elements of $\mathcal{P}$ extend σ_1. We make $T_n^e = \{c_1\}$, changing its index. In this fashion, we will not change T_n^e more than $g_e(n)$ times, so it will be an ω-c.a. trace, as required.

Of course, following this strategy for every e and n will make $\mathcal{P}$ contain only a single element, which would then be computable. So at some point we must relax the construction a little to allow multiple elements. Note also that the basic strategy above would ensure that T_n is a singleton, which is far stronger than we require. So instead of having only a single string σ, which all elements of $\mathcal{P}$ must extend, our strategy will keep some finite number of strings $\sigma^0, \dots, \sigma^{m-1}$, and all elements of $\mathcal{P}$ must extend one of them. Whenever we see an extension γ of one of these σ^i that causes a new value of $\Gamma_e^\gamma(n,t)$, we restrict to extensions of γ, just as in the basic strategy, but we only do so above σ^i.

In this way, the set T_n^e will have size at most m, and will change at most $m \cdot g_e(n)$ times (in the full construction the number of changes will be higher, because of the interaction of strategies, but it will still be computable). We will arrange to keep $m \leqslant 2^n$, and so this will be an ω-c.a. 2^n-trace. The strategies which contribute to the growth of m are the non-computability strategies; each will potentially double the value of m. So we will need to arrange the priorities of the strategies such that there are at most n non-computability strategies with higher priority than the $Q_{e,n}$-strategy.

However, as mentioned above, the actual number of changes to T_n^e will depend on the interaction of strategies. Specifically, it will depend not just on $g_e(n)$, but also on the $g_{e'}(n')$ of higher priority strategies. To ensure there is a computable bound on the number of changes, it is essential that these $g_{e'}(n')$ all converge. So we assign priorities to these strategies as the construction runs; initially, the $Q_{e,n}$-strategy will not have a priority and will not be attended to by the construction. When $g_e(n)$ converges, the $Q_{e,n}$-strategy is assigned a priority lower than every previously assigned priority. In this way, we can calculate the bound on the number of changes to T_n^e as soon as the $Q_{e,n}$-strategy is assigned a priority.

Now, however, we must revisit our earlier commitment to have at most n non-computability strategies with higher priority than the $Q_{e,n}$-strategy. Since the $Q_{e,n}$-strategy could be assigned an arbitrarily late priority, to meet this commitment we must be prepared to drop the priority of non-computability strategies when the $Q_{e,n}$-strategy is assigned a priority. It will be the case, however, that every non-computability strategy eventually stops dropping in priority.

Formalizing the above:

Each strategy will receive from the previous strategy some finite collection of strings $\{\alpha_j\}$, all of the same length, and it will create some finite extensions $\{\beta_k\}$

(all of the same length) such that for every j there is at least one k with $\alpha_j \subseteq \beta_k$, and all sets in $\bigcup_k [\beta_k]$ meet the strategy's requirement.

R_i-strategies will initially define exactly two β_k for every α_j, but may later remove one.

$Q_{e,n}$-strategies will define exactly one β_k for every α_j. They may need to redefine β_k some finite number of times, but each new definition will be an extension of the previous.

At the end of every stage s, we let $\{\hat{\beta}_k\}$ be the outputs of the last strategy to act at stage s, and define the tree P_s to be all strings comparable with one of the $\hat{\beta}_k$. $\mathcal{P}$ will be $\bigcap_s [P_s]$.

Description of R_i-strategy:

This is the standard non-computability requirement on a tree.

(1) Let $\{\alpha_j\}_{j<m}$ be the output of the previous strategy. For every $j < m$, let $\beta_{j,0} = \alpha_j \hat{\ } 0$ and $\beta_{j,1} = \alpha_j \hat{\ } 1$.
(2) Wait for $\phi_i(|\alpha_0|)$ to converge; while waiting, let the outputs be $\{\beta_{j,0}, \beta_{j,1}\}_{j<m}$.
(3) When $\phi_i(|\alpha_0|)$ converges...
   - ...if $\phi_i(|\alpha_0|) = 0$, let the outputs be $\{\beta_{j,1}\}_{j<m}$.
   - ...if $\phi_i(|\alpha_0|) \neq 0$, let the outputs be $\{\beta_{j,0}\}_{j<m}$.

Description of $Q_{e,n}$-strategy:

Until $g_e(n)$ converges, this strategy takes no action. We ignore for the moment the computable bound on the number of times the index of T_e^n changes.

Let $\{\alpha_j\}_{j<m}$ be the output of the previous strategy. We will keep several values to assist the strategy: ℓ_s will be the number of times the output has been redefined by stage s; $c_s(j)$ will be the current guess for $f_e(n)$ on an extension of α_j. We initially have $\ell_s = 0$ and $c_s(j) = -1$ for all j. Unless otherwise defined, $\ell_{s+1} = \ell_s$ and $c_{s+1}(j) = c_s(j)$.

For every j, let $\beta_j^0 = \alpha_j$. We initially let the outputs be $\{\beta_j^0\}_{j\in\omega}$ and define $T_n^e = \emptyset$. We run the following strategy, where s is the current stage:

(1) Wait for a string $\gamma \in P_s$ with γ extending one of the $\beta_j^{\ell_s}$ and $\Gamma_e^\gamma(n,s) \neq c_s(j)$.
(2) When such a string is found for $\beta_j^{\ell_s}$:
   (a) Define $\beta_j^{\ell_s+1} = \gamma$ and $c_{s+1}(j) = \Gamma_e^\gamma(n,s)$.
   (b) For every $k < m$ with $k \neq j$, choose $\beta_k^{\ell_s+1} \in P_s$ extending $\beta_k^{\ell_s}$ of the same length as $\beta_j^{\ell_s+1}$.
   (c) Redefine $T_n^e = \{c_{s+1}(k) \mid k < m\}$.
   (d) Define $\ell_{s+1} = \ell_s + 1$.
(3) Return to Step 1.

Full construction:

We make the assumption that for every s, there is precisely one pair (e,n) with $e < n$ and $g_{e,s+1}(n)\downarrow$ but $g_{e,s}(n)\uparrow$. We give the $Q_{e,n}$-strategies priority based on the order in which the $g_e(n)$ converge: if $g_e(n)$ converges before $g_{e'}(n')$, then the $Q_{e,n}$-strategy has stronger priority than the $Q_{e',n'}$-strategy. If $g_e(n)$ never converges, then $Q_{e,n}$ never has a priority, but this is fine because it never acts.

We prioritize the R_i-strategies based on the priorities of the $Q_{e,n}$-strategies: R_i has weaker priority than $R_{i'}$ for any $i' < i$, and also weaker priority than any $Q_{e,n}$-strategy with $n \leqslant i$. It is given the strongest priority consistent with these restrictions.

Since we only consider $n > e \geqslant 0$, the R_0-strategy will always have strongest priority. It receives $\alpha_0 = \langle\rangle$ as the "output of the previous strategy".

At stage s, let (e, n) be the pair such that $g_e(n)$ has newly converged. We initialize R_n and all strategies which had weaker priority than R_n. The priorities of the various R_i are then redetermined. We then let all Q-strategies with priorities and all R_i-strategies with $i < s$ act, in order of priority.

Whenever a strategy redefines its output, all weaker priority strategies are initialized.

Verification:

We proceed as a sequence of claims.

Claim 2.1. For each i, the priority of the R_i-strategy changes at most $i(i+1)/2$ many times.

Proof. The priority of the R_i-strategy is only changed when some $g_e(n)$ converges with $e < n \leqslant i$. There are at most $i(i+1)/2$ many such pairs (e, n). □

Claim 2.2. Let $\{\alpha_j\}_{j<m}$ be the strings which the R_i-strategy receives from the previous strategy. Then m is at most 2^i.

Proof. Induction on i. For $i = 0$, the only received string is the empty string.

For $i+1$, we observe that the Q-strategies output the same number of strings as they receive, and so the number of strings received by the R_{i+1}-strategy is the same as the number of strings in the output of the R_i-strategy. But the R_i-strategy either outputs the same number of strings as it receives or twice as many, depending on whether it reached Step (3) or not. □

Claim 2.3. Suppose $g_e(n)\downarrow$, and let $\{\alpha_j\}_{j<m}$ be the strings which the $Q_{e,n}$-strategy receives from the previous strategy. Then m is at most 2^n.

Proof. The $Q_{e,n}$-strategy has stronger priority than the R_n-strategy, and by construction the number of strings received can only increase for weaker priority strategies. □

Claim 2.4. Fix e, n, s_0 and s_1 such that $g_{e,s_0}(n)\downarrow$, $s_0 < s_1$, and the $Q_{e,n}$-strategy was never initialized at a stage between s_0 and s_1. Then the strategy redefines its output at most $(g_e(n))^{2^n}$ many times between stages s_0 and s_1.

Proof. Suppose not. Since one of the $c(j)$ changes each time the output is redefined, by the pigeon-hole principle, there must be stages $s_0 < t_0 < \cdots < t_{g_e(n)} < s_1$ and a $j < m$ with $c_{t_k}(j) \neq c_{t_k+1}(j)$ for every $k \leqslant g_e(n)$. Then for any set $X \in [\beta_j^{\ell_{t_{g_e(n)}}}]$, $|\{t \mid \Gamma_e^X(n,t) \neq \Gamma_e^X(n,t+1)\}| \geqslant g_e(n)$, contrary to assumption. □

Claim 2.5. Suppose $g_e(n)$ converges at stage s_0. Let $\{(e_k, n_k)\}_{k<s_0}$ be those pairs such that the Q_{e_k,n_k}-strategy has stronger priority than the $Q_{e,n}$-strategy. Then the $Q_{e,n}$-strategy can be initialized at most $3^n \prod_k (1 + g_{e_k}(n_k))^{2^{n_k}}$ many times.

Proof. There are at most n many R_i-strategies of stronger priority. Each R_i-strategy can cause initialization twice without being initialized itself: once by changing its output, and once when its priority weakens. Note that if the priority of an R_i-strategy weakens after stage s_0, the new priority is necessarily weaker than that of the $Q_{e,n}$-strategy.

Each Q_{e_k,n_k}-strategy can cause initialization $(g_{e_k}(n_k))^{2^{n_k}}$ many times without being initialized itself. The result follows. □

Claim 2.6. At every stage, $\#T_n^e \leqslant 2^n$.

Proof. By construction, T_n^e contains at most m many elements. By a previous claim, m is at most 2^n. □

Claim 2.7. If g_e is total, $\{T^e_n\}_{n\in\omega}$ is an ω-c.a. trace.

Proof. Let s_0 be the stage at which $g_e(n)$ converges, and let $\{(e_k, m_k)\}_{k<s_0}$ be those pairs such that the Q_{e_k,m_k}-strategy has stronger priority than the $Q_{e,n}$-strategy. By construction, T^e_n only changes when the $Q_{e,n}$-strategy redefines its output. By previous claims, this happens at most

$$(g_e(n))^{2^n} \cdot 3^n \prod_{k<s_0} (1 + g_{e_k}(m_k))^{2^{m_k}}$$

many times. Note that this value is uniformly computable in n. □

Claim 2.8. The R_i-strategy is only initialized finitely many times.

Proof. By induction. Wait for a stage such that the priority of the R_i-strategy has finished changing, and all stronger priority strategies have will never again be initialized or change their outcomes. Then the R_i-strategy will never again be initialized. □

It is immediate from the construction that all strategies ensure their requirements. This completes the proof of the theorem. □

3. Lowness for Demuth$_{\mathrm{BLR}}$ randomness

In this section we prove Theorem 1.8, the equivalence of:

(1) Lowness for Demuth$_{\mathrm{BLR}}$ tests;
(2) Lowness for Demuth$_{\mathrm{BLR}}$ randomness; and
(3) Demuth traceability.

Now two of the implications are easy, and we dispose of them swiftly. The implication (1)→(2) holds for any notion of randomness.

We prove that (3)→(1): let A be a Demuth traceable set; we show that every Demuth test by A (a Demuth$_{\mathrm{BLR}}\langle A\rangle$ test) is covered by a Demuth test. By the discussion following Proposition 1.11, it suffices to show that every clopen Demuth$_{\mathrm{BLR}}\langle A\rangle$ test is covered by a Demuth test. Let $\langle \mathcal{C}_{f(n)}\rangle_{n\in\omega}$ be a clopen Demuth$_{\mathrm{BLR}}\langle A\rangle$ test, so f is $\mathsf{BLR}\langle A\rangle$. By replacing $\mathcal{C}_{f(n)}$ with $\mathcal{C}_{f(2n+1)} \cup \mathcal{C}_{f(2n+2)}$, we may assume that $\lambda(\mathcal{C}_{f(n)}) \leqslant 4^{-n}$ for each n.

Now, let $\langle T_n\rangle_{n\in\omega}$ be an ω-c.a. trace for f, bounded by $h(n) = 2^n$. There is an ω-c.a. function q such that $\mathcal{C}_{q(n)} = \bigcup_{i\in T_n} \mathcal{C}_i$. Then $\langle \mathcal{C}_{q(n)}\rangle_{n\in\omega}$ is a Demuth test covering the given test $\langle \mathcal{C}_{f(n)}\rangle_{n\in\omega}$.

For the rest of this section, we prove (2)→(3): that lowness for Demuth$_{\mathrm{BLR}}$ randomness implies Demuth traceability. Given a set A which is not Demuth traceable, we need to construct a set Z which is Demuth random but not Demuth random by A (not Demuth$_{\mathrm{BLR}}\langle A\rangle$-random).

We will define a particular type of open sets, namely *Demuth open sets*, and their complements, *Demuth closed sets*, that reflect the behavior of Demuth tests. The forcing argument has two parts:

(a) Firstly, we show that passing a Demuth test can be interpreted as being in some appropriate Demuth closed set of positive measure. We will therefore use the family of Demuth closed sets of positive measure (ordered by inclusion) as our notion of forcing $\mathbb{P}_{\mathrm{Dem}}$. Once we have proved that a finite intersection of Demuth closed sets is again Demuth closed, we will be able to argue that any sufficiently generic filter G in $\mathbb{P}_{\mathrm{Dem}}$ determines a set $Z_G \subseteq \omega$ that passes all Demuth tests, i.e., a Demuth random set.

(b) Thereafter, we will use the hypothesis that A is not Demuth traceable in order to show that any sufficiently generic filter G of $\mathbb{P}_{\mathrm{Dem}}$ determines a set $Z \subseteq \omega$ that is not $\mathrm{Demuth}_{\mathrm{BLR}}\langle A\rangle$-random. To do so, for every function f which is $\mathsf{BLR}\langle A\rangle$ we devise a clopen $\mathrm{Demuth}_{\mathrm{BLR}}\langle A\rangle$ test $\langle \mathcal{U}_n\rangle = \left\langle \mathcal{B}_{n,f(n)}\right\rangle$, with the property that if up to null sets, almost all the components $\mathcal{U}_n$ are contained in some Demuth open set with measure smaller than 1, then f has an ω-c.a. trace with some fixed bound. This construction will require a probability-theoretic argument involving Chernoff's upper tail bound. Once this is established, we see that if f witnesses that A is not Demuth traceable, then we can generically meet infinitely many components $\mathcal{U}_n$, and so the Demuth random set we construct will not be $\mathrm{Demuth}_{\mathrm{BLR}}\langle A\rangle$ random, witnessing that A is not low for Demuth randomness.

3.1. **Demuth open sets and their basic properties.**

Definition 3.1. An open set $\mathcal{U} \subseteq 2^\omega$ is *Demuth open* if there is an ω-c.a. function $\epsilon \mapsto \mathcal{D}_\epsilon$ such that for all rational $\epsilon > 0$, $\mathcal{D}_\epsilon$ is a clopen subset of $\mathcal{U}$ such that

$$\lambda(\mathcal{U} \setminus \mathcal{D}_\epsilon) \leqslant \epsilon.$$

Lemma 3.2. *Let $\langle \mathcal{C}_n\rangle_{n<\omega}$ be a clopen Demuth test. Then for all $m < \omega$,*

$$\bigcup_{n>m} \mathcal{C}_n$$

is Demuth open.

As an immediate corollary we see that any clopen set is Demuth open.

Proof. Let $\mathcal{U} = \bigcup_{n>m} \mathcal{C}_n$.

Let $\epsilon > 0$ be rational. We can compute the least $k < \omega$ such that $2^{-k} \leqslant \epsilon$. We then let

$$\mathcal{D}_\epsilon = \bigcup_{n\in(m,k]} \mathcal{C}_n.$$

Certainly $\mathcal{D}_\epsilon$ is clopen, the map $\epsilon \mapsto \mathcal{D}_\epsilon$ is ω-c.a., and since

$$\mathcal{U} \setminus \mathcal{D}_\epsilon \subseteq \bigcup_{n>k} \mathcal{C}_n,$$

we have

$$\lambda(\mathcal{U} \setminus \mathcal{D}_\epsilon) \leqslant \sum_{n>k} \lambda(\mathcal{C}_n) \leqslant \sum_{n>k} 2^{-n} = 2^{-k} \leqslant \epsilon. \qquad \square$$

Lemma 3.3. *The union of finitely many Demuth open sets is Demuth open.*

Proof. By induction, it suffices to verify that if $\mathcal{U}$ and $\mathcal{V}$ are both Demuth open, then so is $\mathcal{U} \cup \mathcal{V}$. Let $\epsilon \mapsto \mathcal{D}_\epsilon$ and $\epsilon \mapsto \mathcal{C}_\epsilon$ witness, respectively, that $\mathcal{U}$ and $\mathcal{V}$ are Demuth open. Then the map $\epsilon \mapsto \mathcal{D}_{\epsilon/2} \cup \mathcal{C}_{\epsilon/2}$ is ω-c.a., and for any rational $\epsilon > 0$, we have

$$(\mathcal{U} \cup \mathcal{V}) \setminus (\mathcal{D}_{\epsilon/2} \cup \mathcal{C}_{\epsilon/2}) \subseteq (\mathcal{U} \setminus \mathcal{D}_{\epsilon/2}) \cup (\mathcal{V} \setminus \mathcal{C}_{\epsilon/2}),$$

so

$$\lambda\big((\mathcal{U} \cup \mathcal{V}) \setminus (\mathcal{D}_{\epsilon/2} \cup \mathcal{C}_{\epsilon/2})\big) \leqslant \epsilon. \qquad \square$$

3.2. **Obtaining a trace from a Demuth open cover.** We now show how Demuth open sets relate to Demuth traceability. We code a given function f into a sequence of sets such that any Demuth open set of measure < 1 covering almost all of these sets yields an ω-c.a. trace for f.

We fix an array of independent clopen sets which will be used to code functions. For $n < \omega$ and $k < \omega$, let

$$\mathcal{B}_{n,k} = \left\{X \in 2^\omega \,:\, \forall x < n \; [X(n,k,x) = 0]\right\}.$$

Since for distinct pairs (n, k) and (n', k'), the sets $\mathcal{B}_{n,k}$ and $\mathcal{B}_{n',k'}$ mention distinct locations, the collection of all sets $\mathcal{B}_{n,k}$ is independent. Of course it is important that for all k, $\lambda(\mathcal{B}_{n,k}) = 2^{-n}$.

For subsets $\mathcal{A}$ and $\mathcal{B}$ of Cantor space, we write $\mathcal{A} \subseteq^* \mathcal{B}$ to denote that $\lambda(\mathcal{A}\setminus\mathcal{B}) = 0$.

Lemma 3.4. *Let $\mathcal{U}$ be a Demuth open set such that $\lambda(\mathcal{U}) < 1$. Let $f\colon \omega \to \omega$, and suppose that $\mathcal{B}_{n,f(n)} \subseteq^* \mathcal{U}$ for almost all n. Then f has an ω-c.a. trace, bounded by $h(n) = 2^{4n+5}$.*

Proof. Fix an ω-c.a. function $\epsilon \mapsto \mathcal{D}_\epsilon$ witnessing that $\mathcal{U}$ is Demuth open.

For $n < \omega$, let

$$T_n = \left\{k < \omega \,:\, \lambda\left(\mathcal{B}_{n,k} \setminus \mathcal{D}_{2^{-3n}}\right) \leqslant 2^{-3n}\right\}.$$

Since $\epsilon \to \mathcal{D}_\epsilon$ is ω-c.a., there is an ω-c.a. function g such that $T_n = W_{g(n)}$. We will show that $f(n) \in T_n$ and $\#T_n \leqslant 2^{4n+5}$ for almost all n.

Let $n < \omega$ such that $\mathcal{B}_{n,f(n)} \subseteq^* \mathcal{U}$. Let $\epsilon = 2^{-3n}$. Since

$$\lambda\left(\mathcal{U} \setminus \mathcal{D}_\epsilon\right) \leqslant \epsilon,$$

and since

$$\mathcal{B}_{n,f(n)} \setminus \mathcal{D}_\epsilon \subseteq \left(\mathcal{B}_{n,f(n)} \setminus \mathcal{U}\right) \cup \left(\mathcal{U} \setminus \mathcal{D}_\epsilon\right),$$

we must have

$$\lambda\left(\mathcal{B}_{n,f(n)} \setminus \mathcal{D}_\epsilon\right) \leqslant \epsilon,$$

so $f(n) \in T_n$.

To finish the proof of Lemma 3.4, it remains to show $\#T_n \leqslant 2^{4n+5}$ for almost all n. This follows from the next proposition which is true for any probability measure μ on a space $\mathcal{X}$. In our application the space will be Cantor space with the usual product measure λ.

For any measurable non-null set $\mathcal{R} \subseteq \mathcal{X}$, we let $\mu_\mathcal{R}$ be the conditional probability μ given $\mathcal{R}$ holds: for all measurable $\mathcal{E}$,

$$\mu_\mathcal{R}(\mathcal{E}) = \frac{\mu(\mathcal{R} \cap \mathcal{E})}{\mu(\mathcal{R})}.$$

Informally, the proposition says that if $\mu(\mathcal{R}) \geqslant 1/2$ and localizing to $\mathcal{R}$ increases the measure of every member of an independent collection of sets that have measure 2^{-n} even slightly, then this collection is small.

Proposition 3.5. *Let $n > 0$. Let $\mathfrak{B}$ be a μ-independent collection of subsets of $\mathcal{X}$, each of which has μ-measure 2^{-n}. Let $\mathcal{R} \subseteq \mathcal{X}$ such that $\mu(\mathcal{R}) \geqslant 1/2$. Suppose that for all $\mathcal{B} \in \mathfrak{B}$, $\mu_\mathcal{R}(\mathcal{B}) \geqslant 2^{-n} + 2^{-2n}$. Then $\#\mathfrak{B} \leqslant 2^{4n+5}$.*

The proof of Proposition 3.5 is somewhat technical and appeals to probability-theoretic arguments. We postpone it until Subsection 3.5. Here we show how to complete the proof of Lemma 3.4 assuming this proposition.

Suppose that $n < \omega$, and as before let $\epsilon = 2^{-3n}$. Let $\mathfrak{B}$ be the collection of sets $\mathcal{B}_{n,k}$ where $k \in T_n$. For $k \in T_n$, because $\mathcal{D}_\epsilon \subseteq \mathcal{U}$, we have

$$\mathcal{B}_{n,k} \setminus \mathcal{U} \subseteq \mathcal{B}_{n,k} \setminus \mathcal{D}_\epsilon,$$

and hence

$$\lambda\left(\mathcal{B}_{n,k} \setminus \mathcal{U}\right) \leqslant 2^{-3n}.$$

Since $\lambda(\mathcal{U}) < 1$, if n is large enough we have $\lambda(\mathcal{U}) < 1 - 2^{-n}$. Since $\mathcal{U}$ is open, there is a clopen set $\mathcal{C}$ disjoint from $\mathcal{U}$ such that $\lambda\mathcal{C} \geqslant 2^{-n}$. Hence we can choose a clopen set $\mathcal{G} \subseteq 2^\omega \setminus \mathcal{C}$ such that, where $\mathcal{R} = \mathcal{U} \cup \mathcal{G}$, we have

$$1/2 \leqslant \mu(\mathcal{R}) \leqslant 1 - 2^{-n}.$$

Now let $\mathcal{B} \in \mathfrak{B}$, and let $\delta = 2^{-n}$. By assumption, we have

$$\mu(\mathcal{R} \cap \mathcal{B}) = \mu(\mathcal{B}) - \mu(\mathcal{B} \setminus \mathcal{R}) \geqslant \delta - \delta^3.$$

Since $\mu(\mathcal{R}) \leqslant 1 - \delta$, we have

$$\mu_{\mathcal{R}}(\mathcal{B}) = \frac{\mu(\mathcal{R} \cap \mathcal{B})}{\mu(\mathcal{R})} \geqslant \frac{\delta - \delta^3}{1 - \delta} = \delta(1 + \delta)$$

Thus for all $\mathcal{B} \in \mathfrak{B}$, $\mu_{\mathcal{R}}(\mathcal{B}) \geqslant 2^{-n} + 2^{-2n}$. Since all the hypotheses of Proposition 3.5 are met, we may conclude that $\#T_n \leqslant 2^{4n+5}$ for almost all n. □

3.3. **Forcing with Demuth closed sets.** We now introduce the notion of forcing which naturally generates Demuth random sets. As described above, we will later show that the Demuth random sets generated by the notion of forcing are not $\text{Demuth}_{\text{BLR}}\langle A \rangle$-random for any oracle A which is not Demuth traceable.

For background on forcing with closed sets of positive measure, recall that a set is weakly 2-random if and only if it is 2-generic for forcing with Π^0_2 classes of positive measure, which may be taken to be closed (Kautz [26], see [15, Theorem 7.2.28]). This is an effective version of Solovay's random real forcing, much like the notion of n-genericity is the effective version of Cohen forcing. The following section shows that Demuth randomness resembles weak 2-randomness in this aspect as it has a similar characterization.

Definition 3.6. A *Demuth closed* set is a complement of a Demuth open set. *Demuth forcing* $\mathbb{P}_{\text{Dem}}$ is the notion of forcing consisting of Demuth closed sets of positive measure, ordered by inclusion.

Lemma 3.7. *If $G \subset \mathbb{P}_{\text{Dem}}$ is a sufficiently generic filter, then $\bigcap G$ is a singleton.*

Proof. Let $G \subset \mathbb{P}_{\text{Dem}}$ be a filter. Since G consists of compact subsets of 2^ω and has the finite intersection property, $\bigcap G$ is nonempty.

For $n < \omega$, consider the set

$$\left\{\mathcal{F} \in \mathbb{P}_{\text{Dem}} \,:\, \exists \tau \in 2^{<\omega} \;\; [|\tau| = n \;\&\; \mathcal{F} \subseteq [\tau]]\right\}. \tag{1}$$

It is easy to see that for all n, the set (1) is dense in $\mathbb{P}_{\text{Dem}}$, and so if G is sufficiently generic, for all n, there is some binary string τ of length n such that for all $X \in \bigcap G$, $\tau \subset X$. □

For a sufficiently generic filter G of $\mathbb{P}_{\text{Dem}}$, let Z_G be the unique element of $\bigcap G$. The next lemma is key to our construction.

Lemma 3.8. *If G is a sufficiently generic filter of $\mathbb{P}_{\text{Dem}}$, then Z_G is Demuth random.*

Proof. By Proposition 1.11, it suffices to show that if G is sufficiently generic, then Z_G passes every clopen Demuth test.

Let $\langle \mathcal{C}_n \rangle_{n<\omega}$ be a clopen Demuth test. We show that the set

$$\left\{ \mathcal{F} \in \mathbb{P}_{\mathrm{Dem}} \,:\, \exists m \left[\mathcal{F} \cap \left(\bigcup_{n>m} \mathcal{C}_n \right) = \emptyset \right] \right\} \tag{2}$$

is dense in $\mathbb{P}_{\mathrm{Dem}}$; the lemma then follows.

Let $\mathcal{F} \in \mathbb{P}_{\mathrm{Dem}}$. Since $\lambda(\mathcal{F}) > 0$, there is some $m < \omega$ such that

$$\lambda \left(\bigcup_{n>m} \mathcal{C}_n \right) < \lambda(\mathcal{F}),$$

so

$$\lambda \left(\mathcal{F} \setminus \bigcup_{n>m} \mathcal{C}_n \right) > 0.$$

By Lemmas 3.2 and 3.3, $\mathcal{F} \setminus \bigcup_{n>m} \mathcal{C}_n$ is Demuth closed. And so $\mathcal{F} \setminus \bigcup_{n>m} \mathcal{C}_n$ is an extension of $\mathcal{F}$ in the set (2). $\square$

This completes the first part of the argument: if G is sufficiently generic, then Z_G is Demuth random. It remains to show that if A is not Demuth traceable, then for a sufficiently generic G, Z_G is not Demuth random by A ($\mathrm{Demuth}_{\mathrm{BLR}}\langle A \rangle$ random).

3.4. Forcing failure of lowness.

Suppose that A is not Demuth traceable. As mentioned in Subsection 1.2, for any order function h there is some function f which is $\mathsf{BLR}\langle A \rangle$ but has no h-bounded ω-c.a. trace. Obtain such a function f for the order function $h(n) = 2^{4n+5}$.

Recall the sets $\mathcal{B}_{n,k}$ from above. The fact that f is $\mathsf{BLR}\langle A \rangle$, and that $\lambda(\mathcal{B}_{n,f(n)}) = 2^{-n}$, means that $\langle \mathcal{B}_{n,f(n)} \rangle_{n\in\omega}$ is a $\mathrm{Demuth}_{\mathrm{BLR}}\langle A \rangle$ test.

Lemma 3.9. *If G is sufficiently A-generic, then there are infinitely many n such that $Z_G \in \mathcal{B}_{n,f(n)}$.*

And so Z_G fails the test $\langle \mathcal{B}_{n,f(n)} \rangle$, and so is not $\mathrm{Demuth}_{\mathrm{BLR}}\langle A \rangle$ random; this completes the proof of Theorem 1.8.

Proof. For $m < \omega$, we show that the set

$$\left\{ \mathcal{F} \in \mathbb{P}_{\mathrm{Dem}} \,:\, \exists n > m \,\left[\mathcal{F} \subseteq \mathcal{B}_{n,f(n)} \right] \right\} \tag{3}$$

is dense in $\mathbb{P}_{\mathrm{Dem}}$; the lemma would follow.

Fix $m < \omega$, and let $\mathcal{F} \in \mathbb{P}_{\mathrm{Dem}}$. By the assumption that f does not have an h-bounded ω-c.a. trace, Proposition 3.4, applied to the Demuth open set $\mathcal{U} = 2^\omega \setminus \mathcal{F}$ tells us that there is some $n > m$ such that

$$\lambda \left(\mathcal{B}_{n,f(n)} \cap \mathcal{F} \right) > 0.$$

As was noted above, every clopen set is Demuth closed, and so by Lemma 3.3, $\mathcal{B}_{n,f(n)} \cap \mathcal{F}$ is Demuth closed. It follows that $\mathcal{B}_{n,f(n)} \cap \mathcal{F}$ is an extension of $\mathcal{F}$ in the set (3). $\square$

3.5. **Independent sets and Chernoff bounds.** We complete the proof of Theorem 1.8 by establishing Proposition 3.5 which we had postponed.

Proof of Proposition 3.5. Let $\delta = 2^{-n}$ and $N = \#\mathfrak{B}$ which we may assume to be finite. For a set $\mathcal{E} \subseteq \mathcal{X}$, we let $1_{\mathcal{E}}$ be the characteristic function of $\mathcal{E}$. Define a function $h\colon \mathcal{X} \to \mathbb{R}$ by

$$h = \sum_{\mathcal{B}\in\mathfrak{B}} 1_{\mathcal{B}}.$$

We let

$$\mathcal{K} = \{x \in \mathcal{X} \,:\, h(x) > \delta(1+\delta/2)N\}\,.$$

By the assumption on the measure of the elements of $\mathfrak{B}$ in $\mathcal{R}$, we have

$$\int h \,\mathrm{d}\mu_{\mathcal{R}} = \sum_{\mathcal{B}\in\mathfrak{B}} \int 1_{\mathcal{B}} \,\mathrm{d}\mu_{\mathcal{R}} = \sum_{\mathcal{B}\in\mathfrak{B}} \mu_{\mathcal{R}}(\mathcal{B}) \geqslant \delta(1+\delta)N.$$

On the other hand, of course,

$$\int h \,\mathrm{d}\mu_{\mathcal{R}} = \int_{\mathcal{K}} h \,\mathrm{d}\mu_{\mathcal{R}} + \int_{\mathcal{X}\setminus\mathcal{K}} h \,\mathrm{d}\mu_{\mathcal{R}}.$$

For all $x \in \mathcal{X}\setminus\mathcal{K}$, we have $h(x) \leqslant \delta(1+\delta/2)N$, so

$$\int_{\mathcal{X}\setminus\mathcal{K}} h \,\mathrm{d}\mu_{\mathcal{R}} \leqslant \mu_{\mathcal{R}}(\mathcal{X}\setminus\mathcal{K})\delta(1+\delta/2)N.$$

For all $x \in \mathcal{X}$, we have $h(x) \leqslant N$, so

$$\int_{\mathcal{K}} h \,\mathrm{d}\mu_{\mathcal{R}} \leqslant \mu_{\mathcal{R}}(\mathcal{K})N.$$

Let $p = \mu_{\mathcal{R}}(\mathcal{K})$, so $\mu_{\mathcal{R}}(\mathcal{X}\setminus\mathcal{K}) = 1-p$. The inequalities established so far yield

$$\delta(1+\delta)N \leqslant pN + (1-p)\delta(1+\delta/2)N,$$

whence we obtain

$$p \geqslant \frac{\delta^2}{2(1-\delta-\delta^2/2)};$$

as $\delta \leqslant 1/2$ we have $1-\delta-\delta^2/2 \in (0,1)$, so

$$p \geqslant \delta^2/2 = 2^{-(2n+1)}.$$

Chernoff's upper tail bound [39] states that for any $\epsilon \in (0,1)$, letting $a = \int h \,\mathrm{d}\mu$, we have

$$\mu\left(\{x \in \mathcal{X} \,:\, h(x) \geqslant (1+\epsilon)a\}\right) < e^{-\epsilon^2 a/4}.$$

Applying the bound to $\epsilon = \delta/2$, since

$$\int h \,\mathrm{d}\mu = \sum_{\mathcal{B}\in\mathfrak{B}} \mu(\mathcal{B}) = \delta N,$$

we obtain

$$\mu(\mathcal{K}) < e^{-\delta^3 N/16}.$$

Of course,

$$\mu(\mathcal{K}) \geqslant \mu(\mathcal{K}\cap\mathcal{R}) = \mu_{\mathcal{R}}(\mathcal{K})\cdot\mu(\mathcal{R}) = p\cdot\mu(\mathcal{R}),$$

so overall we obtain

$$p\cdot\mu(\mathcal{R}) < e^{-\delta^3 N/16}.$$

Since $\mu(\mathcal{R}) \geqslant 1/2$, we have $p\cdot\mu(\mathcal{R}) \geqslant 2^{-(2n+2)}$. Taking the natural logarithm and then the negative, we obtain

$$\delta^3 N < -16\ln 2^{-(2n+2)} = -16\log_2 2^{-(2n+2)}\ln 2 < 16(2n+2) \leqslant 2^{n+5},$$

the last step recalling that $n \geqslant 1$. Hence

$$N < \frac{2^{n+5}}{\delta^3} = 2^{n+5}2^{3n} = 2^{4n+5}$$

as required. □

4. Demuth tracebility and jump traceability

We have shown that the class of computably dominated Demuth traceable sets coincides with lowness for Demuth randomness. We now give a computability-theoretic analysis of Demuth traceability by relating it to the more familiar notion of jump traceability. We observe that Demuth traceability implies jump traceability, but while the notions coincide on the c.e. degrees, they differ on both the ω-c.a. degrees and on the computably dominated degrees. Thus, lowness for Demuth randomness properly implies being computably dominated and jump traceable.

4.1. **Jump traceability.** An oracle A is *jump traceable* [34] if every A-partial computable function ψ has a uniformly c.e. trace bounded by some order function. The reason that we do not require a uniform bound h on the traces for all functions which are A-partial computable is that there is a universal A-partial computable function. Letting J^A be such a function (for example, $J^A(e) = \varphi_e^A(e)$), we see that an oracle A is jump traceable if and only if J^A has a uniformly c.e. trace bounded by an order function. However, unlike c.e. traceability, computable traceability and Demuth traceability, the fact that we need to trace partial functions means that the standard Terwijn-Zambella for the irrelevance of the choice of the order function fails for jump traceability. Indeed, the classes of h-jump-traceable sets vary significantly with the growth-rate of the order function h; for sufficiently fast-growing functions h, there is a perfect set of h-jump-traceable sets, while for sufficiently slow-growing h, all h-jump-traceable sets are Δ^0_2. In particular, jump traceability and strong jump traceability, defined in the introduction, differ substantially.

Proposition 4.1. *Every Demuth traceable set is jump traceable.*

Proof. Suppose that A is Demuth traceable. Let $f(n) = J^A(n)$ if $n \in \mathrm{dom} J^A$, and otherwise let $f(n) = 0$. Then f is $\mathsf{BLR}\langle A\rangle$. Let $(T_n)_{n\in\omega}$ be an ω-c.a. trace for f, bounded by an order function h, and suppose that g is an order function which witnesses that the index function for $\langle T_n\rangle$ is ω-c.a. Let T_n^* be the union of all versions of T_n over stages. Then $(T_n^*)_{n\in\omega}$ is a uniformly c.e. trace for f, and so for J^A, bounded by $h \cdot g$. □

In the rest of this section we will show that Demuth traceability strictly implies jump traceability. However, they agree on the c.e. degrees. Indeed, on the c.e. degrees, Demuth traceability is equivalent to a strong form of Demuth traceability, in which the bound h can be taken to be the constant function 1.

Recall that a set A is *superlow* if $A' \leqslant_{\mathrm{tt}} \emptyset'$. Nies [34] showed that jump traceability and superlowness coincide on the c.e. sets (see also [36, 8.4.23]), while neither class includes the other within the ω-c.a. sets.

Note that a set A is Demuth traceable with bound 1 if and only if every $\mathsf{BLR}\langle A\rangle$ function is ω-c.a., in other words if $\mathsf{BLR}\langle A\rangle = \mathsf{BLR}\langle\emptyset\rangle$. Cole and Simpson [7, Cor. 6.15] studied the class of oracles with the latter property, which they dubbed the oracles *low for BLR*. They showed [7, Cor. 6.15] that this class coincides with the intersection of jump traceability and superlowness. Thus:

Fact 4.2. *A set is both jump traceable and superlow if and only if it is Demuth traceable with bound 1.*

Since each jump traceable c.e. set is superlow, we obtain:

Proposition 4.3. *The following are equivalent for a c.e. set A:*

(1) *A is Demuth traceable.*
(2) *A is Demuth traceable with bound 1.*
(3) *A is jump traceable.*

4.2. **Separating jump traceability from Demuth traceability in the ω-c.a. degrees.** We show that the class of Demuth traceable sets is strictly smaller than the class of jump traceable sets. We first separate these classes within the ω-c.a. sets.

Theorem 4.4. *There is an ω-c.a. set which is jump traceable but not Demuth traceable.*

Proof. We must build an ω-c.a. set A and a $\mathsf{BLR}\langle A\rangle$ function which escapes all ω-c.a. traces. To diagonalize against all such traces, we will need an enumeration of them, so let $\langle (T^e_n)_{n\in\omega}, g^e\rangle_{e\in\omega}$ be an enumeration of all partial ω-c.a. traces with bound $h(n) = n$, where $g^e(n)$ is the bound on the number of times the index for T^e_n can change; the functions g^e are partial computable. We construct an ω-c.a. set A, a Turing functional Γ and a c.e. trace $\langle V_n\rangle_{n\in\omega}$ meeting the following requirements:

G**:** Γ^A is an approximation of a total $\mathsf{BLR}\langle A\rangle$ function.
R_e**:** $\lim_s \Gamma^A(e, s) \notin T^e_e$.
N_i**:** $\#V_i \leqslant 2^{i^4}$; if $J^A(i)\downarrow$, then $J^A(i) \in V_i$.

Our basic strategy for meeting requirement R_e is to choose many distinct strings as possible initial segments of A, and define $\Gamma(e)$ differently along each string. Since $|T^e_e| \leqslant e$, by counting there will always be at least one of these strings σ with $\Gamma^\sigma(e) \notin T^e_e$. We make that string our current initial segment of A. If at some later stage $\Gamma^\sigma(e)$ enters T^e_e, we change to a different initial segment.

Our basic strategy for meeting requirement N_i is restraint, similar to the proof of Theorem 1.5. Whenever we see a σ which causes $J^\sigma(i)$ to converge, we restrain A to be an extension of that σ and enumerate $J^\sigma(i)$ into V_i. Of course, this restraint will cause injury to later R_e-strategies, and here is where we use the fact that Γ^A is merely an approximation to a $\mathsf{BLR}\langle A\rangle$ function; whenever a higher priority N_i-strategy acts, we can restart the R_e-strategy by simply redefining $\Gamma(e)$, so long as we have a computable bound on the number of times we do so.

Similarly, an R_e-strategy changing between possible initial segments of A will interfere with a later N_i-strategy's attempts to restrain A, but here we use the fact that V_i need not be a singleton. As long as higher priority strategies only change between at most 2^{i^4} possible initial segments of A, the N_i-strategy can restrain to a single σ above each of these initial segments and enumerate all of these possible $J^\sigma(i)$ into V_i. Note that the number of strings the R_e-strategy will change between is dependent on $|T^e_e|$; for our calculations, it is important that the R_e-strategy diagonalizes at $\Gamma^A(e)$ instead of at some arbitrary $\Gamma^A(e')$ with $e' > e$. It is for this reason that the R_e-strategy redefines $\Gamma(e)$ when injured instead of simply choosing a larger e' to work with.

The R_e-strategy may need to change initial segments many times; potentially as many as $g^e(e) \cdot e$ times. To ensure that A is ω-c.a., we cause all these initial segments to agree for the first $g^e(e)$ bits, and only differ after that point. So the R_e-strategy is only causing changes to late bits, and so the number of changes to a bit can remain computably bounded.

We order the requirements as $R_0 < N_0 < R_1 < N_1 < \dots$, and at every stage s we run strategies for every requirement R_e and N_i with $e, i < s$ in increasing order. Every strategy receives from the previous strategy a finite collection of incomparable strings X_s which are possible initial segments of A and a specified string $\sigma_s \in X_s$

which is the current initial segment of A_s. The strategy is responsible for creating its own set X'_s of incomparable strings and its own $\sigma'_s \in X'_s$ satisfying:

- For every $\tau \in X_s$, there is a $\tau' \in X'_s$ with $\tau' \succ \tau$; and
- $\sigma'_s \succ \sigma_s$.

The strategy for R_0 always receives $X_s = \{\langle\rangle\}$ and $\sigma_s = \langle\rangle$.

Strategy for R_e:

First we wait for $g^e(e)$ to converge. While we wait, we let $X'_s = \{\tau\widehat{\ }0 \mid \tau \in X_s\}$, $\sigma'_s = \sigma_s\widehat{\ }0$, and define $\Gamma(e, s) = 0$ with use 0.

Once $g^e(e)$ has converged, we let $\rho_0, \dots, \rho_{2^e-1}$ be the strings of length e, ordered lexicographically. We define $X'_s = \{\tau\widehat{\ }\left(0^{g^e(e)}\right)\widehat{\ }\rho_i \mid \tau \in X_s, i < 2^e\}$. We define $\Gamma^{\tau\widehat{\ }\left(0^{g^e(e)}\right)\widehat{\ }\rho_i}(e, s) = i$ for all $\tau \in X_s$ and $i < 2^e$. For every string τ which is incomparable with every string in X'_s, we define $\Gamma^\tau(e, s) = 0$.

We let j be least such that $j \notin T^e_{e,s}$ (since $\#T^e_e \leqslant e$, we know $j < 2^e$) and let $\sigma'_s = \sigma_s\widehat{\ }\left(0^{g^e(e)}\right)\widehat{\ }\rho_j$.

Strategy for N_i:

Given X_s, we shall construct X'_s to contain precisely one $\tau' \succ \tau$ for every $\tau \in X_s$.

Given $\tau \in X_s$, we search for a $\rho \succ \tau$ with $|\rho| < s$ such that $J^\rho_s(i)\downarrow$. If there is no such ρ, we define $\tau' = \tau\widehat{\ }0$. Otherwise, we let t be least such that there is a $\rho \succ \tau$ with $|\rho| < t$ and $J^\rho_t(i)\downarrow$, let τ' be the least such ρ under some ordering, and enumerate $J^{\tau'}_t(i)$ into V_i.

We define $X'_s = \{\tau' \mid \tau \in X_s\}$ and let σ' be the unique element of X'_s extending σ.

Verification:

First observe that for a fixed R_e-strategy, if $s_0 < s_1$ and $X_s = X_{s_0}$ for every $s_0 < s \leqslant s_1$, then there can be at most one stage t with $s_0 < t < s_1$ and $X'_t \neq X'_{t+1}$ — the stage at which $g^e(e)$ converges.

Similarly, for a fixed N_i-strategy, if $s_0 < s_1$ and $X_s = X_{s_0}$ for every $s_0 < s \leqslant s_1$, there can be at most $\#X_s$ many stages t at which $X'_t \neq X'_{t+1}$. Further, $\#X_s$ is bounded by 2^{i^2}.

Thus by induction, for any R_e- or N_i-strategy, X_s is eventually fixed.

We perform a similar analysis for σ_s: for a fixed R_e-strategy, if $s_0 < s_1$ and $\sigma_s = \sigma_{s_0}$ for every $s_0 < s \leqslant s_1$, then there can be at most $1 + g^e(e) \cdot e$ many stages t with $s_0 < t < s_1$ and $\sigma'_t \neq \sigma'_{t+1}$ — first when $g^e(e)$ converges, and then every such stage after that indicates that a new element was enumerated into the current version of T^e_e. If $g^e(e)$ does not converge, then there can be no such stages.

For a fixed N_i-strategy, if $s_0 < s_1$ and $\sigma_s = \sigma_{s_0}$ for every $s_0 < s \leqslant s_1$, there can be at most one stage t with $s_0 < t < s_1$ and $\sigma'_t \neq \sigma'_{t+1}$ — the stage at which a convergent jump computation is found on an extension of σ_{s_0}.

Thus, by induction, for any R_e- or N_i-strategy, σ_s is eventually fixed. Further, by construction, for any fixed R_e-strategy, $|\sigma_s| \geqslant e$. Thus if we let $\sigma^e = \lim_s \sigma_s$ for the R_e-strategy, $A = \bigcup_e \sigma^e$ is a Δ^0_2 set.

Claim 4.5. A is an ω-c.a. set.

Proof. For any value $n \in \omega$, we approximate $A(n)$ by considering the R_{n+1}-strategy and the values of $\sigma_s(n)$. As reasoned earlier, the number of times σ_s can change is bounded by

$$2^n \cdot \prod_{\substack{e<n+1 \\ g^e(e)\downarrow}} 2 + g^e(e) \cdot e.$$

But if $e < n+1$ and $g^e(e){\downarrow}\geqslant n$, then we can omit it from the above product when considering $\sigma_s(n)$: all the extensions that the R_e-strategy is changing between begin with $g^e(e)$ many 0s, and thus agree on n. So the number of times $\sigma_s(n)$ can change is bounded by

$$2^n \cdot \prod_{\substack{e<n+1 \\ g^e(e)\downarrow<n}} 2 + g^e(e)\cdot e \leqslant 2^n(2+n^2)^{n+1}. \qquad \square$$

Clearly $\Gamma^Y(e,s)$ is defined for all $e < s$ and oracles Y, and so in particular for the oracle A. Also, if $\Gamma^\tau(e,s) \neq \Gamma^\tau(e,s+1)$ for some string τ, it indicates that $X'_s \neq X'_{s+1}$ for the R_e-strategy. By the earlier reasoning, this can happen at most $2^{e+1}\cdot\prod_{i<e} 2^{i^2}$ many times, which is a bound uniformly computable from e. Thus Γ^A is an approximation to a total $\mathsf{BLR}\langle A\rangle$ function.

By construction, $\lim_s \Gamma^A(e,s) \notin T^e_e$. As observed in Subsection 1.2, this means that A is not Demuth traceable.

By construction, if $J^A(i){\downarrow}$, then the N_i-strategy will have acted to enumerate $J^A(i)$ into V_i. Whenever an element is enumerated into V_i, it indicates that $X'_s \neq X'_{s+1}$ for the N_i-strategy. By the earlier reasoning, this can happen at most $2^{i+1}\cdot\prod_{j\leqslant i} 2^{j^2} \leqslant 2^{i^4}$ many times. So A is jump traceable. $\square$

4.3. **Separating jump traceability from Demuth traceability in the computably dominated degrees.** We now separate Demuth traceability from jump traceability within the computably dominated sets. In particular, this means that there is a set which is low for Schnorr randomness but not for Demuth randomness.

Theorem 4.6. *There is a set which is jump traceable and computably dominated, but not Demuth traceable.*

Proof. The proof is in some sense an elaboration on the proof of Theorem 4.4; rather than a Δ^0_2 set, we build a Δ^0_3 set using an approximation argument inside a Π^0_1 class. We build a Π^0_1-class $\mathcal{P}$, a functional Γ and a computably dominated set $X \in \mathcal{P}$, with Γ^X demonstrating that X is not Demuth traceable by being an approximation for a $\mathsf{BLR}\langle X\rangle$ function which has no ω-c.a. trace.

Let T be the computable tree $T := \{\sigma\in\omega^{<\omega} \mid (\forall n < |\sigma|)[\sigma(n)\leqslant n]\}$. We will define a limit-computable embedding $g: T\to 2^{<\omega}$ with $\mathcal{P}$ the image of $[T]$ under g. For every s, we will define $g(\sigma,s)$ for all $\sigma\in T$ with $|\sigma|\leqslant s$. Then $g(\sigma) = \lim_s g(\sigma,s)$. An important property of g is the following: if $|\sigma| < s$ and $g(\sigma,s) = g(\sigma,s+1)$, then $g(\sigma\widehat{\ }j,s)\subseteq g(\sigma\widehat{\ }j,s+1)$.

For every $n\leqslant s$, and every $\sigma\in T$ with $|\sigma| = n$, we define $\Gamma^{g(\sigma,s)}(n,s) = \sigma$. To show that Γ^X is $\mathsf{BLR}\langle X\rangle$ (that is, ω-c.a. by X), we will later demonstrate a computable bound on the number of times $g(\sigma,s)$ changes.

Jump traceability strategy:

We ensure jump traceability as follows: for every $\sigma\in T$ with $|\sigma| = n$, we wait until a stage $s+1$ when we see a $\tau\in T$ extending σ with $|\tau|\leqslant s$ and $J^{g(\tau,s)}_s(n){\downarrow}$. Then we define $g(\sigma,s+1) = g(\tau,s)$, enumerate $J^{g(\tau,s)}_s(n)$ into V_n and cease our action on behalf of σ unless $g(\sigma^-)$ changes.

The sequence $(V_n)_{n\in\omega}$ will trace the jump of every $X\in\mathcal{P}$, and if $g(\sigma^-)$ never changed, then we would have $|V_n|\leqslant(n+1)!$. Instead we will have $|V_n|\leqslant(n+1)!\cdot h(n-1)$, where h is a computable bound on the changes of g which we establish later.

Basic non-Demuth-traceable strategy:

Given an ω-c.a. trace $(T^k_n)_{n\in\omega}$ with $|T^k_n| \leqslant n$, the idea for defeating this trace is the following: suppose $\sigma \in T$ with $|\sigma| \geqslant n-1$. Then by counting, $\sigma\frown j \notin T^k_n$ for some $j \leqslant n$, and so $g(\sigma\frown j)$ forces that $\lim_s \Gamma^X(e,s)$ is not traced by $(T^k_n)_{n\in\omega}$.

Basic strategy for being computably dominated:
This strategy is the most complex; we use a modification of the full approximation argument found in [12]. Our version is considerably simplified, however, since we are not attempting to make X have rank 1 in $\mathcal{P}$. For every functional Φ_i, if Φ^X_i is total for our set X, we must construct a computable function f_i which dominates Φ^X_i.

Suppose $\sigma_0, \ldots, \sigma_k$ are the elements of T of length n (for some n). We consider the σ_j in order. As long as there is no τ_j extending σ_j with $|\tau_j| \leqslant s$ and $\Phi^{g(\tau_j,s)}_{i,s}(0)\downarrow$, then $g(\sigma_j, s)$ forces the satisfaction of the requirement for Φ_i. When we see such a τ_j, we set $g(\sigma_j, s+1) = g(\tau_j, s)$ and move on to σ_{j+1}.

If we have found a τ_j for every σ_j, we define $f_i(0) = \max_{j\leqslant k}\{\Phi^{g(\tau_j,s)}_{i,s}(0)\}$. In this case, $\langle\rangle$ forces that $f_i(0) \geqslant \Phi^X_i(0)$.

Of course, once we have forced that $\Phi^X_i(0)\downarrow\leqslant f_i(0)$, we must move on to considering $\Phi^X_i(1)$. We repeat the same strategy as above, but we make an important observation: the σ_j used for $\Phi^X_i(1)$ need not be the same as those used for $\Phi^X_i(0)$ (that is, we need not use the same length n). It is important that we constantly increase n, otherwise $g(\sigma)$ would not converge for $|\sigma| > n$ and $\mathcal{P}$ would consist of only $k+1$ elements.

Priority tree:
Suppose that the strategy for Φ^X_0 is considering σ_j, searching for a $\tau_j \supseteq \sigma_j$ with $\Phi^{g(\sigma_j,s)}_{0,s}(0)\downarrow$. If it never finds such a τ_j, then X must go through $g(\sigma_j)$. The strategy for avoiding the ω-c.a. trace $(T^0_n)_{n\in\omega}$ must act above σ_j. On the other hand, if the strategy for Φ^X_0 always finds a τ_j for every σ_j considered, then the requirement for Φ^X_0 is satisfied for any $X \in \mathcal{P}$, and so the strategy for $(T^0_n)_{n\in\omega}$ is free to act at $\langle\rangle$.

Similarly, if $g(\sigma\frown j)$ forces that $(T^k_n)_{n\in\omega}$ does not trace Γ^X, then the strategy for Φ^X_1 cannot act on all of T, but must restrict itself to considering T above $\sigma\frown j$.

For this reason, we will make a priority tree of these strategies, with the strategies at level $2i$ devoted to Φ^X_i and the strategies at level $2i+1$ devoted to $(T^i_n)_{n\in\omega}$ (the jump-traceability strategies will not appear on the priority tree). Each strategy of the first type will have two outcomes: `inf` and `fin`. Each strategy of the second type will have only a single outcome: `outcome`. Strategy α will inherit from its predecessor a string $\sigma_\alpha \in T$ to work above; the root strategy will use $\sigma = \langle\rangle$.

Full non-BLR-traceable strategy:
For α a $(T^k_n)_{n\in\omega}$-strategy, let $t_k(n)$ be the (partial) computable bound on the number of times T^k_n can change. The only action α takes is to define σ_α (and initialise strategies extending $\alpha\frown$`outcome`). While we wait for $t_k(n)$ to converge, we let $\sigma_{\alpha\frown\texttt{outcome}} = \sigma_\alpha$. Once $t_k(n)$ has converged, at every stage s, let j be least such that $g(\sigma_\alpha\frown j, s) \notin T^k_{n,s}$. We let $\sigma_{\alpha\texttt{outcome}} = \sigma\frown j$. If this is different from the last time α was accessible, we initialise all strategies extending $\alpha\frown$`outcome`.

Full strategy for being computably dominated:
For α a Φ^X_0-strategy, let s_α be the stage at which α was first visited after most recently being initialised. Its behavior is as follows:

(1) Set $m = 0$.

(2) Let $\sigma_0, \dots, \sigma_k$ be the elements of T above σ_α and of length $s_\alpha + m$.
(3) Let $j \leqslant k$ be least with $\Phi_{i,s}^{g(\sigma_j,s)}(m)\uparrow$. If there is no such j, define $f_\alpha(m) = \max_{j \leqslant k}\{\Phi_{i,s}^{g(\tau_j,s)}(m)\}$, increment m and return to Step 2.
(4) Wait for a stage s when there is a $\tau_j \supseteq \sigma_j$ with $|\tau_j| \leqslant s$ and $\Phi_{i,s}^{g(\tau_j,s)}(m)\downarrow$.
(5) Define $g(\sigma_j, s+1) = g(\tau_j, s)$. and return to Step 3.

When α is initialised, $\sigma_{\alpha\frown\texttt{inf}}$ is set to σ_α. While waiting at Step 4, α has outcome $\texttt{fin}$ and $\sigma_{\alpha\frown\texttt{fin}} = \sigma_j$. When α leaves Step 4, all strategies beneath $\alpha\frown\texttt{fin}$ are initialised. Whenever α returns to Step 2, it has outcome $\texttt{inf}$ for one stage.

Construction: At stage $s = 2t$, we run the jump traceability strategy for $J^X(n)$ for all $n \leqslant s$ in increasing order, stopping if any of these strategies act.

At stage $s = 2t+1$, we run all accessible non-Demuth-traceable and computable domination strategies up to level s, in order of priority, stopping if any of these strategies act.

After running the appropriate strategies, if some strategy defined $g(\sigma, s+1) = g(\tau, s)$ for some σ and τ, we choose appropriate values for $g(\pi, s+1)$ for all $\pi \supset \sigma$ with $|\pi| < s+1$, and we define $g(\rho, s+1) = g(\rho, s)$ for all $\rho \not\supseteq \sigma$. If no strategy acted, we define $g(\rho, s+1) = g(\rho, s)$ for all ρ. We also choose appropriate values for $g(\pi, s+1)$ for all $|\pi| = s+1$.

For every strategy α on the priority tree, if $g(\sigma_\alpha, s+1) \neq g(\sigma_\alpha, s)$, we initialise α.

Verification: We define the true path as usual.

Claim 4.7. There is a computable function h such that for each σ, we have $h(|\sigma|) \geqslant \#\{s \mid g(\sigma, s) \neq g(\sigma, s+1)\}$.

Proof. We construct h recursively.

The only strategies which can change $g(\sigma)$ without changing $g(\sigma')$ for any $|\sigma'| < |\sigma|$ are the jump traceability strategy for $J^X(|\sigma|)$ and computable domination strategies α with $s_\alpha + m = |\sigma|$. The first can act at most $(|\sigma|+1)!$ times without $g(\sigma')$ changing for some $|\sigma'| < |\sigma|$. There are at most $|\sigma|^2$ of the latter (because at most s^2 strategies have been visited by stage s), and without $g(\sigma')$ changing for some $|\sigma'| < |\sigma|$, each can act at most $(|\sigma|+1)!$ times before $s_\alpha + m$ is larger than $|\sigma|$.

So

$$h(|\sigma|) = h(|\sigma|-1) + (h(|\sigma|-1)+1) \cdot ((|\sigma|+1)! + |\sigma|^2 \cdot (|\sigma|+1)!)$$

suffices. □

Note that this is a bound on the number of times $\Gamma^X(|\sigma|, s)$ can change for any $X \in P$. It follows by induction that every strategy along the true path is initialised only finitely many times. Let X be the limit of σ_α for α along the true path. It is now immediate from the construction that every strategy along the true path and every jump-traceability strategy ensures its requirement is met. □

5. Constant bounds on the traces

We observed above that unlike other traceability notions, considering constant bounds in the definition of Demuth traceability does not force the oracle to be computable. Indeed, we gave a characterization of those oracles which are Demuth traceable with bound 1. When considering constant bounds, we show that every increase of a constant bound also enlarges the class of Demuth traceable oracles with that bound.

Theorem 5.1. *For every n, there is a Δ^0_2-set A which is Demuth traceable with bound $n+1$ but not bound n.*

Proof. Fix n. Let $(T^i, f_i)_{i\in\omega}$ be an enumeration of all partial ω-c.a. traces with bound n; here $f_i(m)$ is the bound on the number of times the index for T^i_m can change. Let $(\Gamma_e, g_e)_{e\in\omega}$ be an enumeration of all BLR functionals; here $g_e(m)$ is the bound on $\#\{s \mid \Gamma^X_e(m,s) \neq \Gamma^X_e(m,s+1)\}$ for any oracle X. We construct the desired A, a $\mathsf{BLR}\langle A\rangle$ function Δ^A, and ω-c.a. traces $(V^e)_{e\in\omega}$ with bound $n+1$ using strategies to meet the following requirements:

$R_{e,m}$**:** If $g_e(m)\downarrow$, then $\lim_s \Gamma^A_e(m,s) \in V^e_m$.
P_i**:** There is a number j such that either $f_i(j)\uparrow$ or $\lim_s \Delta^A(j,s) \notin T^i_j$.

Meeting these requirements for every e, m and i will suffice to prove the theorem.

Basic idea:

The basic strategy for $R_{e,m}$ is restraint: when we see a string $\sigma \prec A_s$ with $\Gamma^\sigma_e(m,s) \neq \Gamma^\sigma_e(m,s+1)$, we restrain $A_s\restriction_{|\sigma|}$ and set $V^e_m = \{\Gamma^\sigma_e(m,s+1)\}$. This will happen at most $g_e(m)$ many times.

The basic strategy for P_i is to choose a j and $n+1$ incomparable strings $\sigma_0, \ldots, \sigma_n$, and define $\Delta^{\sigma_k}(j,s) = k$. Then by counting there is always a $k \leqslant n$ with $k \notin T^i_{j,s}$, so at stage s we choose the least such k and define $\sigma_k \prec A_s$. This will cause a change in A at most $f_i(j)\cdot n$ many times.

Of course, there is conflict between these two basic strategies, as one wishes to restrain A and the other wishes to change A, and so we must resolve this. If $v_e(m)$ (the bound on the number of times the index of V^e_m can change) is defined after $f_i(j)$ converges, then it can be defined large enough for $R_{e,m}$ to handle the finitely many changes caused by P_i. If $d(j)$ (the bound on $\#\{s \mid \Delta^A(j,s) \neq \Delta^A(j,s+1)\}$) is defined after $g_e(m)$ converges, then it can be defined large enough to allow P_i to switch to larger strings whenever the old ones are restrained by $R_{e,m}$.

Since the trace V^e need only exist if g_e is total, we can assume that $v_e(m)$ is defined at the same stage $g_e(m)$ converges. However, d must be total no matter what, so if $g_e(m)$ converges after $d(j)$ is defined but before $f_i(j)$ converges, neither of the previous two cases hold. By appropriate managing of priority, we can ensure that for each (e,m), there is at most one (i,j) for which this holds. This P_i will be changing between $n+1$ different versions of A, while V^e_m can have size $n+1$, so it can contain one element for each of the versions. Of course, V^e_m will need to change versions whenever $\Gamma_e(m,s)$ changes along one of these $(n+1)$ different versions of A, but that is at most $(n+1)\cdot g_e(m)$ different versions of V^e_m, and recall that n is fixed in the construction. Thus $v_e(m)$ can be made large enough to account for this.

Organizing the construction:

We must assign priorities dynamically. We prioritize the $R_{e,m}$-strategies based on the order the $g_e(m)$ converge — if $g_e(m)$ converges before $g_{e'}(m')$, then the $R_{e,m}$-strategy has higher priority than the $R_{e',m'}$-strategy. We assume that exactly one of these converges at every stage. Strategies for which $g_e(m)\uparrow$ are never assigned a priority.

At stage s, let i_0 be least such there is a j with $i_0 \leqslant j < s$ and $f_{i_0,s}(j)\downarrow$, and let j_0 be the least such j for this i_0. The P_{i_0}-strategy will work with j_0 and has highest priority amongst the P_i-strategies at stage s. Every strategy $R_{e,m}$ such that $g_e(m)$ converged by stage j_0 has higher priority than P_{i_0}, while the rest have lower.

Let s_0 be the stage at which $f_{i_0}(j_0)$ converged. Let $i_1 > i_0$ be least such that there is a j with $\max(i_1, s_0, j_0) \leqslant j < s$ and $f_{i_1,s}(j)\downarrow$, and let j_1 be least for this i_1. Then i_1 will work with j_1 and has next highest priority amongst the P_i-strategies at stage s. Every $R_{e,m}$ such that $g_e(m)$ converged by stage j_1 has higher priority than P_{i_1}, while the rest have lower.

We continue in this fashion, assigning priorities at stage s to as many P_i as possible; those which remain will not have a priority at stage s. Note that by construction, for every $R_e(m)$ there is at most one P_{i_k}, such that $g_e(m)$ converged at stage s', $f_{i_k}(j_k)$ converged at stage s_k, and $j_k < s' < s_k$.

We then let every strategy which has a priority act in order of priority. Every strategy will receive from the previous strategy a set of incomparable strings B which are potential initial segments of A, with one of those strings distinguished as the current initial segment. The highest priority strategy receives $\{\langle\rangle\}$ with $\langle\rangle$ distinguished. Each strategy is then responsible for constructing a set of incomparable strings B' with every string in B' extending a string in B. One of the strings in B' must be distinguished, and it must extend the distinguished string in B.

If P_i is working with j at stage s, it will define $\Delta^X(j,s)$ for every oracle X. For every remaining j, we define $\Delta^X(j,s) = 0$ for all oracles X.

Strategy P_i:

We fix some collection $\sigma_0, \ldots, \sigma_n$ of incomparable strings.

Let τ be the distinguished string in the received set B (all other strings in B will be ignored). We define $\Delta^{\tau\frown\sigma_k}(j,s) = k$ for every $k \leqslant n$, define $\Delta^\rho(j,s) = 0$ for every ρ incomparable with all the $\tau\frown\sigma_k$, and let

$$B' = \{\tau\frown\sigma_k \mid k \leqslant n\}.$$

Let k be least such that $k \notin T^i_{j,s}$. We distinguish $\tau\frown\sigma_k$ as the current initial segment of A.

Strategy for $R_{e,m}$:

Let B be the set received from the previous strategy. For every $\tau \in B$, we shall search for a $\rho \succ \tau$ that maximizes

$$\#\{s' \mid \Gamma^\rho_e(m,s') \neq \Gamma^\rho_e(m,s'+1)\}.$$

Let ρ_τ be least such under some ordering. Then we define

$$B' = \{\rho_\tau \mid \tau \in B\}$$

and

$$V^e_{m,s} = \{\Gamma^{\rho_\tau}_e(m,s) \mid \tau \in B\}.$$

We distinguish in B' whichever ρ_τ extends the distinguished element of B.

Verification:

By induction, every P_i-strategy for which f_i is total will eventually settle on a j and cease changing priority. By induction again, every strategy only redefines its B' or distinguished element finitely many times. Thus we can define $\sigma(\ell,s)$ to be the distinguished element of the strategy with priority ℓ at stage s, and we know that $\sigma(\ell) = \lim_s \sigma(\ell,s)$ exists. Further, $\sigma(\ell,s) \subset \sigma(\ell+1,s)$ by construction, and so $A = \bigcup_\ell \sigma(\ell)$ exists and is Δ^0_2.

Claim 5.2. The $R_{e,m}$-strategy ensures its requirement.

Proof. Let ℓ be the eventual priority of the $R_{e,m}$-strategy, and let s be the last stage at which this strategy acts. Then $\Gamma^{\sigma(\ell)}_e(m,s) \in V^e_m$ by construction, and $\sigma(\ell) \prec A$. If there is some $\rho \succeq \sigma(\ell)$ and some stage $t > s$ with $\Gamma^{\sigma(\ell)}_e(m,s) \neq \Gamma^\rho_e(m,t)$, this would contradict the action of $R_{e,m}$. Thus $\lim_s \Gamma^A_e(m,s) \in V^e_m$.

It remains only to show that there is a bound on the number of changes to V^e_m which is uniformly computable in m. Let s_0 be the stage at which $g_e(m)$ converged.

There are only three ways in which the $R_{e,m}$-strategy's set B can be changed after stage s_0: a higher priority $R_{e',m'}$ can act; a P_i-strategy working with a $j < s_0$ such that $f_i(j)$ converged before stage s_0 can act; and a P_i-strategy can begin working with a $j < s_0$. Note that by construction, at any given stage there can be at most one P_i-strategy working with a $j < s_0$ such that $f_i(j)$ converged after stage s_0, and the action of this strategy does not affect $R_{e,m}$'s set B.

If $R_{e',m'}$ is of higher priority than $R_{e,m}$, $g_{e'}(m')$ converged before stage s_0. $R_{e',m'}$ will act at most $(n+1)\cdot g_{e'}(m')$ many times between stages when some higher priority strategy acts.

If P_i is working with a $j < s_0$, and $f_i(j)$ converged before s_0, P_i will act at most $(n+1)\cdot f_i(j)$ many times between stages when some higher priority strategy acts.

If P_i begins working with a $j < s_0$ at some stage after s_0, then necessarily $i < s_0$. Further, if later a different strategy $P_{i'}$ begins working with a $j' < s_0$, then necessarily $i' < i$ or $i' = i$ and $j' < j$. Thus this can occur at most s_0^2 many times after stage s_0.

Between stages when the $R_{e,m}$-strategy's B changes, the strategy will act at most $(n+1)\cdot g_e(m)$ many times. Thus an upper bound for the number of times V_m^e changes is

$$\left[\prod (n+1)\cdot g_{e'}(m')\right]\cdot\left[\prod (n+1)\cdot f_i(j)\right]\cdot s_0^2\cdot(n+1)\cdot g_e(m),$$

where the first product ranges over those e', m' with $g_{e'}(m')$ converged before stage s_0, and the second product ranges over those i, j with $f_i(j)$ converged before stage s_0. Thus the above expression can be computed at stage s_0. □

Claim 5.3. Δ^A is a $\mathsf{BLR}\langle A\rangle$ function.

Proof. Fix j. By construction, there are only three situations in which $\Delta^A(j,s) \neq \Delta^A(j,s+1)$: j is claimed by some P_i at stage s but not at stage s_1; j is claimed by some P_i at stage $s+1$ but not at stage s; and j is claimed by some P_i at stages s and $s+1$, and some higher priority strategy acts at stage $s+1$. Notably, $\Delta^A(j,s)$ is not affected by $T^i_{j,s}$.

By the manner in which we assign priorities, if P_i claims j, then $i \leqslant j$. If later P_i stops claiming j, then P_i will never again claim j. If later some P'_i claims j, then $i' < i$. So the first and second cases can only occur $(j+1)$ many times each.

If some P_i has claimed j at stage $s+1$ and $R_{e,m}$ is higher priority than P_i, then $g_e(m)$ converged by stage j. If some $P_{i'}$ is higher priority and is working with some j', then $j' < j$ and $f_{i'}(j')$ converged by stage j. By the same argument as in the previous claim, we can bound the number of times a higher priority strategy acts by

$$\left[\prod (n+1)\cdot g_e(m)\right]\cdot\left[\prod (n+1)\cdot f_{i'}(j')\right],$$

where the first product ranges over those e, m such that $g_e(m)$ converged by stage j, and the second product ranges over those i', j' such that $f_{i'}(j')$ converged by stage j. So

$$2\cdot(j+1)+\left[\prod (n+1)\cdot g_e(m)\right]\cdot\left[\prod (n+1)\cdot f_{i'}(j')\right]$$

serves as a bound for $\#\{s \mid \Delta^A(j,s) \neq \Delta^A(j,s+1)\}$. □

Claim 5.4. The P_i-strategy ensures its requirement.

Proof. If f_i is not total, this is trivial. Failing that, let j be the value that P_i eventually works with. Then by counting, there is a $k \leqslant n$ such that $k \notin T^i_j$. By construction, $\lim_s \Delta^A(j,s)$ is the least such k. □

This completes the proof. □

6. Lowness for weak Demuth randomness

The Solovay condition for being captured by a test $\langle \mathcal{U}_n \rangle$, namely being in infinitely many components $\mathcal{U}_n$, is the natural one to use when the tests are not nested. If the test is nested, then the capturing condition is equivalent to being in all components. Nevertheless, strengthening the notion of capturing even when the tests are not nested gives rise to a weaker notion of randomness which turns out to be useful. This definition also goes back to Demuth; see [31] for more background.

Definition 6.1. A set Z *weakly passes* a test $\langle \mathcal{U}_n \rangle$ if $Z \notin \bigcap_n \mathcal{U}_n$ (equivalently, the test *strongly captures* the set Z if $Z \in \mathcal{U}_n$ for all n). A set Z is *weakly Demuth random* if it weakly passes every Demuth test.

Because the universal Martin-Löf test is nested, and is a Demuth test, we see that weak Demuth randomness is an intermediate notion between Demuth randomness and Martin-Löf randomness:

$$\text{Demuth random} \to \text{weak Demuth random} \to \text{ML-random}.$$

In [30] it is shown that a weakly Demuth random set is never superhigh. On the other hand, they build a high Δ^0_2 set that is weakly Demuth random (while a Demuth random is always generalized low_1). Here, we show that similar to the situation vis-a-vis computable randomness, computability itself can be characterized as lowness for weak Demuth randomess.

Theorem 6.2. *A set is low for weak Demuth randomness if and only if it is computable.*

Proof. One direction is immediate. The interesting direction is showing that lowness for weak Demuth randomness implies computability. The way we do this is by showing that lowness for weak Demuth randomness implies both K-triviality and being computably dominated. We then obtain the desired result by using the facts that every K-trivial set is Δ^0_2, and that the only computably dominated Δ^0_2 sets are the computable sets. Thus, the theorem is proved once we obtain Propositions 6.3 and 6.4 below.

Proposition 6.3. *Every set which is low for weak Demuth randomness is K-trivial.*

This follows immediately from a result of Downey et al. [17, Thm. 4.2]: if every weakly 2-random is ML-random in an oracle A, then A is K-trivial. Nonetheless, we give a direct proof based on a different literature result [6], in order to adapt the proposition to the case of partial relativization.

Proof. Bienvenu and Miller [6] introduced a partial relativization of Martin-Löf randomness. For an oracle V, a $\mathrm{ML}\langle V \rangle$ *test* is a sequence $[W_{g(n)}]^{\prec}_{n\in\omega}$, where $g \leqslant_{\mathrm{T}} V$, such that $\lambda[W_{g(n)}]^{\prec} \leqslant 2^{-n}$. The difference from the full relativization of Martin-Löf randomness is that the test components must be Σ^0_1, not merely $\Sigma^0_1(V)$. As the index function for this test is not necessarily computable, the union of a tail of such a test may not be Σ^0_1, and so weakly passing such tests is not equivalent to Solovay passing them. Nonetheless, a set Z is *$\mathrm{ML}\langle V \rangle$-random* if it weakly passes every $\mathrm{ML}\langle V \rangle$ test $\langle \mathcal{U}_n \rangle$, that is, if $Z \notin \bigcap_n [W_{g(n)}]^{\prec}$ for each such test.

Bienvenu and Miller showed the following for any pair of oracles V and A: if each $\mathrm{ML}\langle V \rangle$-random set is ML-random relative to A, then A is K-trivial. Now letting A be a set which is low for weak Demuth randomness, the proposition follows from their result using $V = \emptyset'$, once we notice that every set which is $\mathrm{ML}\langle \emptyset' \rangle$ is weakly Demuth random (in fact a set is $\mathrm{ML}\langle \emptyset' \rangle$ if and only it is weakly-2-random), and that in turn every set which is weakly Demuth random relative to A is also Martin-Löf random relative to A. The former follows from the fact that every ω-c.a. function is computable in $\emptyset'$, and the latter is a relativization to A of the fact, noticed above, that every weakly Demuth random set is Martin-Löf random. □

We remark that the relativization of the implication from weak Demuth randomness to Martin-Löf randomness also holds if we only partially relativize weak Demuth randomness. In other words, the proof of Proposition 6.3 shows that if A is low for weak $\text{Demuth}_{\text{BLR}}$ randomness, then A is K-trivial. Although we do not include a proof, this implication reverses; if A is K-trivial, then A is low for weak $\text{Demuth}_{\text{BLR}}$ randomness.

The next proposition, which completes the proof of the theorem, is an analog of a result of Downey's and Ng's [16], which we mentioned and used above, that lowness for Demuth randomness implies being computably dominated. This proposition uses the power of the full relativization of weak Demuth randomness.

Proposition 6.4. *Every set which is low for weak Demuth randomness is computably dominated.*

Proof. Let A be a set computing a function f which is not dominated by any computable function. We construct a weakly Demuth random set Z which is not weakly Demuth random relative to A.

We shall construct a Demuth test $\langle \mathcal{U}_n \rangle_{n\in\omega}$ relative to A and a weakly Demuth random set Z such that $Z \in \bigcap_n \mathcal{U}_n$ (so Z is not weakly Demuth random relative to A). Let $(\langle \mathcal{V}_n^k \rangle_{n\in\omega}, g_k)_{k\in\omega}$ be an enumeration of all partial computable Demuth tests, where $g_k(n)$ is the bound on the number of times the version of $\mathcal{V}_n^k$ can change. We need to ensure the following requirements hold:

N_k**:** If g_k is total, there is some n with $Z \notin \mathcal{V}_n^k$.

We do not directly create a strategy for every N_k requirement. Instead, we associate a strategy with every $\mathcal{U}_n$. If this strategy sees that $g_k(n+2)\downarrow < f(n)$, it will work to meet N_k. It does this by instructing all strategies associated with $\mathcal{U}_\ell$ for $\ell > n$ to construct their $\mathcal{U}_\ell$ avoiding $\mathcal{V}_{n+2}^k$.

To implement this, the strategy associated with $\mathcal{U}_n$ defines a set $\mathcal{B}_{n+1}$ which the strategy associated with $\mathcal{U}_{n+1}$ must work to avoid. We cannot hope to build $\mathcal{U}_{n+1}$ so that it avoids $\mathcal{B}_{n+1}$ in its entirety, so instead we settle for only having small overlap with it. The strategy then passes the overlap to the next strategy as part of $\mathcal{B}_{n+2}$. In order to keep small overlap with $\mathcal{B}_{n+1}$, $\mathcal{U}_{n+1}$ may need to change as many as $4g_k(n+2)$ times. Thus we can use $f(n)$ to compute an upper bound on the number of changes.

If g_k is total, we argue that Z weakly passes the Demuth test $\left\langle \langle V_n^k \rangle_{n\in\omega}, g_k \right\rangle$ as follows. By assumption there is some n satisfying $g_k(n+2) < f(n)$, and thus the strategy associated with $\mathcal{U}_n$ will work to meet N_k. Our set Z will be a set in $\bigcap_n \mathcal{U}_n \setminus \mathcal{B}_n$. Thus since $Z \notin \mathcal{B}_{n+1}$, Z weakly passes the Demuth test.

Strategy for $\mathcal{U}_n$:

The strategy for $\mathcal{U}_n$ only acts at stages $s \geqslant n$. It will keep a value $k_s(n)$, which may be undefined.

At stage s, if $s = n$ or $\mathcal{U}_{n-1,s} \neq \mathcal{U}_{n-1,s-1}$, we define $\mathcal{U}_{n,s}$ to be some clopen set in $\mathcal{U}_{n-1,s}$ disjoint from $\mathcal{B}_{n,s}$ and of size 2^{-n}.

Otherwise, if $\lambda(\mathcal{U}_{n,s} \cap \mathcal{B}_{n,s}) \leqslant 2^{-(n+2)}$, we define $\mathcal{U}_{n,s} = \mathcal{U}_{n,s-1}$. If $\lambda(\mathcal{U}_{n,s} \cap \mathcal{B}_{n,s}) > 2^{-(n+2)}$, we define $\mathcal{U}_{n,s}$ to be some clopen set in $\mathcal{U}_{n-1,s}$ disjoint from $\mathcal{B}_{n,s}$ and of size 2^{-n}.

Note that since $\lambda(\mathcal{U}_{n-1,s}) = 2^{-(n-1)}$ and $\lambda(\mathcal{B}_{n,s}) \leqslant 2^{-n}$, there is always sufficient measure to choose $\mathcal{U}_{n,s}$ as described.

Finally, if there is some $k < n$ such that $g_k(n+2)\downarrow < f(n)$, and for every $m < n$, $k \neq k_s(m)$, we choose the least such k and set $k_s(n) = k$ and $\mathcal{B}_{n+1,s} = \mathcal{U}_{n,s} \cap (\mathcal{B}_{n,s} \cup \mathcal{V}^k_{n+2})$. Otherwise, we leave $k_s(n)$ undefined and set $\mathcal{B}_{n+1,s} = \mathcal{U}_{n,s} \cap \mathcal{B}_{n,s}$.

If $k_s(n)$ is undefined, clearly $\lambda(\mathcal{B}_{n+1,s}) \leqslant 2^{-(n+1)}$ by construction. Otherwise, $\mathcal{B}_{n+1,s} \subseteq (\mathcal{U}_{n-1,s} \cap \mathcal{B}_{n-1,s}) \cup \mathcal{V}^k_{n+2}$, and thus $\lambda(\mathcal{B}_{n+1,s}) \leqslant 2^{-(n+2)} + 2^{-(n+2)} = 2^{-(n+1)}$.

Construction:

At every stage s, we begin by setting $\mathcal{B}_{0,s} = \emptyset$. Then, for $n \leqslant s$ in increasing order, we run the strategy for $\mathcal{U}_n$.

Verification:

We proceed with a sequence of claims.

Claim 6.5. The set $\{s \mid k_s(n) \neq k_{s+1}(n)\}$ has size at most 4^n.

Proof. By construction, $k_s(n) \neq k_{s+1}(n)$ implies that either some $g_k(n+2)\downarrow$ at stage $s+1$ with $k < n$ or some $k_s(m) \neq k_{s+1}(m)$ with $m < n$. The former can clearly happen at most n times, while by induction the latter can happen at most 4^m for each $m < n$. Thus we have the bound

$$n + \sum_{m<n} 4^m \leqslant 4^n. \qquad \square$$

Claim 6.6. If g_k is total, then there is some n such that $\lim_s k_s(n) = k$. Hence the requirement N_k is met.

Proof. Since the function $n \mapsto g_k(n+2)$ is total computable, by assumption there are infinitely many n such that $g_k(n+2) < f(n)$. Let $n_0, \dots, n_k$ be the first $k+1$ such n. Let s_0 be a stage such that $g_k(n_i+2)$ has converged for every $i \leqslant k$. By construction, for every $s \geqslant s_0$, $k_s(n_i) = k$ for some $i \leqslant k$. By pigeon hole, there is some i such that $k_s(n_i) = k$ for infinitely many s. But by the previous claim, $k_s(n_i)$ can only change finitely many times. Thus $n = n_i$ is as desired. $\square$

Claim 6.7. There is a total A-computable function h such that $\#\{s \mid \mathcal{U}_{n,s} \neq \mathcal{U}_{n,s+1}\} \leqslant h(n)$.

Proof. We build this bound by recursion. Since $\mathcal{B}_{0,s} = \emptyset$ for all s, we can take $h(0) = 0$.

For $n > 0$, $\mathcal{B}_{n,s}$ is a union of finitely many test elements $\mathcal{V}^{k_i}_{n_i,s}$, with $k_i = k_s(n_i-2)$ and $n_i - 2 < n$. Each $\mathcal{V}^{k_i}_{n_i}$ can change versions at most $g_{k_i}(n_i)$ many times, and by construction $g_{k_i}(n_i) < f(n_i-2)$. Further, each $k_s(n_i-2)$ can change at most 4^{n_i-2} many times.

If none of the $k_s(m)$ with $m < n$ have changed between stages s_0 and s_1, and none of the $\mathcal{V}^{k_i}_{n_i}$ have changed their versions between stages s_0 and s_1, and $\mathcal{U}_{n-1,s} = \mathcal{U}_{n-1,s+1}$ for all $s_0 \leqslant s < s_1$, then since $\lambda(\mathcal{B}_{n,s}) \leqslant 2^{-n}$ and $\mathcal{U}_n$ is changed whenever $\lambda(\mathcal{U}_{n,s} \cap \mathcal{B}_{n,s}) > 2^{-(n+2)}$, $\mathcal{U}_n$ will have to be changed at most 3 times between stages s_0 and s_1. Thus we can bound the number of changes by

$$4 \cdot (h(n-1)+1) \cdot \sum_{m<n} 4^m f(m).$$

This is clearly A-computable. $\square$

Claim 6.8. The class $\mathcal{B}_n = \limsup_s \mathcal{B}_{n,s}$ is a Σ^0_1-class.

Proof. Let s_0 be a stage such that all $k_s(m)$ have converged for $m < n$ by stage s_0, and for every $m < n$ with $k = k_{s_0}(m)$ defined, $\mathcal{V}^k_{m+2}$ will never again change versions. Then $\limsup_s \mathcal{B}_{n,s}$ is simply the finite union of the Σ^0_1-classes $\mathcal{V}^k_{m+2}$. $\square$

Claim 6.9. Let $\mathcal{U}_n = \lim_s \mathcal{U}_{n,s}$. For $m < n$, $(\mathcal{U}_m - \mathcal{B}_m) \supset (\mathcal{U}_n - \mathcal{B}_n)$.

Proof. By construction, $\mathcal{U}_m \supset \mathcal{U}_n$ and $\mathcal{B}_m \cap \mathcal{U}_n \subseteq \mathcal{B}_n$. □

By compactness, $\bigcap_n (\mathcal{U}_n - \mathcal{B}_n)$ is nonempty. Let Z be a point in the intersection. Clearly Z is not A-weakly Demuth random, since it fails the test $\langle \mathcal{U}_n \rangle_{n\in\omega}$.

Claim 6.10. *Z is weakly Demuth random.*

Proof. For any weak Demuth test $\langle \mathcal{V}_n^k \rangle_{n\in\omega}$, let n be such that $k = \lim_s k_s(n)$. Then $\mathcal{V}_{n+2}^k \subseteq \mathcal{B}_{n+1}$ by construction, and so $Z \notin \mathcal{V}_{n+2}^k$. Thus Z weakly passes this Demuth test. □

Thus Z is the set we desire, completing the proof of the proposition and of the theorem. □

□

References

[1] G. Barmpalias, J. S. Miller, and A. Nies. Randomness notions and partial relativization. *Israel J. Math.*, 191(2):791–816, 2012.

[2] T. Bartoszyński. Combinatorial aspects of measure and category. *Fund. Math.*, 127(3):225–239, 1987.

[3] L. Bienvenu, D. Diamondstone, N. Greenberg, and D. Turetsky. Van Lambalgen's theorem for Demuth randomness. Proceedings of the 12th Asian Logic Colloquium (2013), 251-270.

[4] L. Bienvenu, R. Hoelzl, J. Miller, and A. Nies. The Denjoy alternative for computable functions. Symposium on theoretical aspects of computer science (STACS) 2012, 543-554.

[5] L. Bienvenu, R. Hoelzl, J. Miller, and A. Nies. Demuth, Denjoy, and Density. Submitted, 2013.

[6] L. Bienvenu and J. Miller. Randomness and lowness notions via open covers. *Ann. Pure Appl. Logic*, 163(5), 506–518, 2012.

[7] J. Cole and S. Simpson. Mass problems and hyperarithmeticity. *J. Math. Log.*, 7(2):125–143, 2007.

[8] O. Demuth. Some classes of arithmetical real numbers. *Comment. Math. Univ. Carolin.*, 23(3):453–465, 1982.

[9] O. Demuth. Remarks on the structure of tt-degrees based on constructive measure theory. *Comment. Math. Univ. Carolin.*, 29(2):233–247, 1988.

[10] R. Downey and N. Greenberg. Strong jump traceability II: the general case. *Israel J. Math.*, 2011. In press.

[11] D. Diamondstone, N. Greenberg and D. Turetsky. Inherent enumerability of strong jump-traceability. Submitted.

[12] Rod Downey. On Π^0_1 classes and their ranked points. *Notre Dame J. Formal Logic*, 32(4):499–512 (1992), 1991.

[13] R. Downey and N. Greenberg. Pseudo-jump inversion and SJT-hard sets. Submitted.

[14] R. Downey, E. Griffiths, and G. Laforte. On Schnorr and computable randomness, martingales, and machines. *Math. Log. Q.*, 50(6):613–627, 2004.

[15] R. Downey and D. Hirschfeldt. *Algorithmic randomness and complexity.* Theory and Applications of Computability. Springer, New York, 2010.

[16] R. Downey and K.M. Ng. Lowness for Demuth randomness. In *Mathematical Theory and Computational Practice*, volume 5635 of *Lecture Notes in Computer Science*, pages 154–166. Springer Berlin / Heidelberg, 2009. Fifth Conference on Computability in Europe, CiE 2009, Heidelberg, Germany, July 19-July 24.

[17] R. Downey, A. Nies, R. Weber, and L. Yu. Lowness and Π^0_2 nullsets. *J. Symbolic Logic*, 71(3):1044–1052, 2006.

[18] S. Figueira, A. Nies, and F. Stephan. Lowness properties and approximations of the jump. *Ann. Pure Appl. Logic*, 152:51–66, 2008.

[19] J. Franklin and D. Diamondstone. Lowness for difference tests. To appear in the Notre Dame J. Formal Logic.

[20] J. Franklin and K. M. Ng. Difference randomness. *Proceedings of the American Mathematical Society. Proc. Amer. Math. Soc.*, 139(1):345–360, 2011.

[21] P. Gács. Every sequence is reducible to a random one. *Inform. and Control*, 70:186–192, 1986.

[22] N. Greenberg and J. Miller. Lowness for Kurtz randomness. *J. Symbolic Logic*, 74(2):665–678, 2009.

[23] N. Greenberg and D. Turetsky. Strong jump-traceability and Demuth randomness. Submitted.
[24] R. Hölzl, T. Kräling, S. Stephan, and G. Wu. Initial segment complexities of randomness notions. To appear.
[25] S. Ishmukhametov. Weak recursive degrees and a problem of Spector. In *Recursion theory and complexity (Kazan, 1997)*, volume 2 of *de Gruyter Ser. Log. Appl.*, pages 81–87. de Gruyter, Berlin, 1999.
[26] S. Kautz. *Degrees of random sets.* Ph.D. Dissertation, Cornell University, 1991.
[27] B. Kjos-Hanssen, A. Nies, and F. Stephan. Lowness for the class of Schnorr random sets. *SIAM J. Computing*, 35(3):647–657, 2005.
[28] A. Kučera. An alternative, priority-free, solution to Post's problem. In *Mathematical foundations of computer science, 1986 (Bratislava, 1986)*, volume 233 of *Lecture Notes in Comput. Sci.*, pages 493–500. Springer, Berlin, 1986.
[29] A. Kučera and S. Terwijn. Lowness for the class of random sets. *J. Symbolic Logic*, 64:1396–1402, 1999.
[30] A. Kučera and A. Nies. Demuth randomness and computational complexity. *Ann. Pure Appl. Logic*, 162:504–513, 2011.
[31] A. Kučera and A. Nies. Demuth's path to randomness (extended abstract). Computation, Physics and Beyond. Lecture Notes in Computer Science, 2012, Volume 7160, 159-173.
[32] W. Miller and D. A. Martin. The degree of hyperimmune sets. *Z. Math. Logik Grundlag. Math.*, 14:159–166, 1968.
[33] K. Miyabe. Truth-table Schnorr randomness and truth-table reducible randomness. *MLQ Math. Log. Q.*, 57(3):323–338, 2011.
[34] A. Nies. Reals which compute little. In *Logic Colloquium '02*, Lecture Notes in Logic, pages 260–274. Springer–Verlag, 2002.
[35] A. Nies. Lowness properties and randomness. *Adv. in Math.*, 197:274–305, 2005.
[36] A. Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.
[37] A. Nies. Computably enumerable sets below random sets. *Ann. Pure Appl. Logic*, 2011. To appear.
[38] A. Nies. New directions in computability and randomness. Talk at the CCR 2009 in Luminy, available at this link, 2009.
[39] A. N. Shiryayev. *Probability*, volume 95 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1984. Translated from the Russian by R. P. Boas.
[40] F. Stephan. Marin-Löf random and PA-complete sets. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 342–348. Assoc. Symbol. Logic, La Jolla, CA, 2006.
[41] F. Stephan and L. Yu. Lowness for weakly 1-generic and Kurtz-random. In *Theory and applications of models of computation*, volume 3959 of *Lecture Notes in Comput. Sci.*, pages 756–764. Springer, Berlin, 2006.
[42] S. Terwijn and D. Zambella. Algorithmic randomness and lowness. *J. Symbolic Logic*, 66:1199–1205, 2001.