



Characterizing the strongly jump-traceable sets via randomness

Noam Greenberg^{a,*}, Denis R. Hirschfeldt^b, André Nies^c

^a *School of Mathematics, Statistics and Operations Research, Victoria University of Wellington, Wellington, New Zealand*

^b *Department of Mathematics, University of Chicago, Chicago, IL, USA*

^c *Department of Computer Science, University of Auckland, Auckland, New Zealand*

Received 28 October 2010; accepted 17 June 2012

Available online 16 August 2012

Communicated by the Managing Editors of AIM

Abstract

We show that if a set A is computable from every superlow 1-random set, then A is strongly jump-traceable. Together with a result of Greenberg and Nies [Noam Greenberg, André Nies, Benign cost functions and lowness properties, *J. Symbolic Logic* 76 (1) (2011) 289–312], this theorem shows that the computably enumerable (c.e.) strongly jump-traceable sets are exactly the c.e. sets computable from every superlow 1-random set.

We also prove the analogous result for superhighness: a c.e. set is strongly jump-traceable if and only if it is computable from every superhigh 1-random set.

Finally, we show that for each cost function c with the limit condition there is a 1-random Δ_2^0 set Y such that every c.e. set $A \leq_T Y$ obeys c . To do so, we connect cost function strength and the strength of randomness notions. Together with a theorem of Greenberg and Nies (ibid.), this result gives a full correspondence between obedience of cost functions and being computable from Δ_2^0 1-random sets.

© 2012 Elsevier Inc. All rights reserved.

MSC: primary 03D32; secondary 03D25; 03D30; 03D80; 03F60; 68Q30

Keywords: Computability; Randomness; Lowness; Traceability

* Corresponding author.

E-mail addresses: greenberg@msor.vuw.ac.nz (N. Greenberg), drh@math.uchicago.edu (D.R. Hirschfeldt), andre@cs.auckland.ac.nz (A. Nies).

1. Background and motivation

There are two aspects to the information content of sets of natural numbers. In terms of computational complexity, a set of numbers is considered to code a lot of information if it is useful as an oracle for relative computation. In terms of effective randomness, difficulty to detect patterns in the set marks it as complicated, or random. The interaction between these two aspects of complexity is the focus of much current research in computability theory.

Although earlier research naturally gravitated toward the complex, recent findings have shown rich structure in the region of the simple. Properties of sets that indicate being uncomplicated are called *lowness properties*. They have proved to be essential in the understanding of random sets, and of the connections between computability and randomness along the entire spectrum of complexity.

A lowness property that is central to this study is that of K -triviality. A series of results by Downey, Hirschfeldt, Nies, and Stephan (see [18,31]) developed penetrating techniques for the study of several classes of low sets. These results have established the coincidence of several such notions, three of the important ones being: K -triviality (being far from random), lowness for randomness (not being able to detect new patterns in random sets), and being computable from a relatively random oracle. (Here, “random” means Martin-Löf random, or 1-random, as defined in Section 2.3.) This coincidence established the robustness of this class. Further results have demonstrated its usefulness and importance to the field; see, for example, [11].

The diverse characterizations of the K -trivial sets, and the techniques used to study them, have led to three paradigms for understanding lowness of a set A of natural numbers, introduced by Nies [34,35].

1. *Being weak as an oracle*. This paradigm means that A is not very useful as an oracle for Turing machines. This is the oldest way of thinking about lowness. For instance, A is of hyperimmune free degree if it does not compute fast growing functions: each function computed by A is dominated by a computable function. Some formal instances of the paradigm are expressed through A' , the halting set relative to A . For instance, the traditional notion, simply called “low”, states that A' is as simple as possible in the Turing degrees. The newer notion of superlowness states that A' is as simple as possible in the truth-table degrees.

2. *Being computed by many oracles*. Traditionally, there were no interesting answers to the question “how many sets compute A ?”; the answer is always “uncountably many”—indeed continuum many—but unless A is computable (in which case every set computes A), the collection of sets computing A has measure 0. Recently, more detailed answers have proved to be insightful, in particular in conjunction with answers to the question “what *kinds* of sets compute A ?” For example, as noted above, A is K -trivial if and only if A is computed by some set that is 1-random relative to A , in which case the class of oracles computing A is large in an effective sense relative to A .

3. *Being inert*. Shoenfield’s limit lemma states that a set A is computable from the halting set \emptyset' if and only if it has a computable approximation. (We let Δ_2^0 denote the collection of such sets.) The inertness paradigm says that a Δ_2^0 set A is close to computable if it is computably approximable with a small number of changes. For formal instances of the inertness paradigm, we use so-called *cost functions*. They measure the total number of changes of a Δ_2^0 set, and especially that of a computably enumerable set. Most examples of cost functions are based on randomness-related concepts. (Precise definitions of all of these concepts will be given below. For more background on these paradigms see [36].)

The K -trivial sets exemplify these paradigms. Every K -trivial set is superlow; as mentioned above, a set is K -trivial if and only if it is computable by a set that is 1-random relative to it; a set is K -trivial if and only if it has an approximation that obeys a canonical cost function c_K defined below.

There are two ways to give mathematical definitions of lowness properties: combinatorial and analytic. Combinatorial lowness properties, such as (traditional) lowness and superlowness, are defined by discrete tools and by traditional computability. Analytic lowness properties are defined via measure, either directly or coded by prefix-free Kolmogorov complexity, or via some type of effectively given real number. Even though the central notion of K -triviality *implies* some combinatorial lowness properties (such as superlowness), it is only known to be *equivalent* to analytic notions. In other words, currently, K -triviality has only analytic characterizations. The search for a combinatorial characterization of the K -trivial sets is considered an intriguing open problem.

Traceability is a combinatorial tool that is used to define several lowness properties. Among these notions, *strong jump-traceability*, defined by Figueira, Nies and Stephan [13], was proposed (see [27]) as a natural candidate for the desired combinatorial characterization of K -triviality. This conjecture was refuted by Cholak, Downey and Greenberg [4]; further work [1] refuted another possible characterization, in terms of the rate of growth of the traces. However, Cholak, Downey, and Greenberg did show that for computably enumerable (c.e.) sets, strong jump-traceability at least implies K -triviality, making strong jump-traceability the first known combinatorial notion to imply K -triviality. They further showed that in conjunction with computable enumerability, strong jump-traceability has some appealing structure (it induces an ideal in the Turing degrees). These results prompted interest in strong jump-traceability in its own right.

Strong jump-traceability falls under the first lowness paradigm discussed above, since it is related to weakness of the jump. In [16], Nies and Greenberg showed that strong jump-traceability can be characterized using cost-function approximations, thus giving it also a characterization within the third paradigm. They used this result to show that every strongly jump-traceable c.e. set is computed by many random oracles, a lowness property belonging to the second paradigm. Along the way, they showed that strong jump-traceability is useful in settling problems in other areas of computability, unrelated to randomness.

In the current paper, in a reverse turn of events, we show that strong jump-traceability can in fact be defined analytically, using lowness properties of the second paradigm. This result shows the robustness of strong jump-traceability. The heart of the paper, guided by the second paradigm for lowness, is the investigation of the oracular power of random sets relative to c.e. sets. In other words, the question under consideration is: which c.e. sets are computable from which random sets?

Two important early results are seminal. Chaitin [3] showed that there is a *complete* 1-random set, that is, a 1-random set that is Turing above the halting set \emptyset' . Hence, every c.e. set is computable from a 1-random set. This result was later extended by Gács [14] and Kučera [24], who showed that *every* set is computable from a 1-random set. The focus thus turned to *incomplete* 1-random sets, that is, 1-random sets that do not compute \emptyset' . Here, Kučera's basic result [25] is that every 1-random Δ_2^0 set is Turing above a noncomputable c.e. set.

The evidence that being computable from an incomplete 1-random set is a lowness property came much later. In [18], Hirschfeldt, Nies and Stephan showed that if Y is an incomplete 1-random set, and A is a c.e. set computable from Y , then in fact Y is 1-random relative to A , and A is K -trivial. It is still open whether every K -trivial set is computable from some incomplete 1-random set.

There are two ways to extend Kučera's result, both following the second paradigm for lowness. One is to investigate which c.e. sets are computable by many incomplete sets. Relevant here is the extension of Kučera's result by Hirschfeldt and Miller (see [34, Theorem 5.3.15]), who showed that if \mathcal{C} is a Σ_3^0 null class, then there is a noncomputable c.e. set computed by all 1-random elements of \mathcal{C} . In the current paper, we show, for several classes \mathcal{C} of sets, that the strongly jump-traceable c.e. sets are precisely the c.e. sets that are computable from all 1-random elements of \mathcal{C} . We do this for the classes consisting of the ω -c.e. sets, the superlow sets, and the superhigh sets. (These and other computability theoretic concepts mentioned in this introduction will be defined below.) These results give characterizations of c.e. strong jump-traceability according to the second lowness paradigm. They are simpler than the characterization according to this paradigm of the K -trivial sets as those sets A that are computable from a set that is 1-random relative to A ; for strong jump-traceability, we do not need to relativize randomness.

Another direction for extending Kučera's result is to keep our focus on a single random set and the c.e. sets that it computes. A first attempt would be to consider a Δ_2^0 1-random set Y as "strong" if all c.e. sets computable from Y share some strong lowness property. In early 2009 Greenberg proved [15] that there is a Δ_2^0 1-random set Y such that every c.e. set computable from Y is strongly jump-traceable. This result contrasts with the fact, observed in [16], that no ω -c.e. 1-random set has this property.

The next logical step is to relate the "lowness strength" of a random set with its degree of randomness. Up to now we have only mentioned the standard notion of randomness, due to Martin-Löf. We can investigate what happens if we require a higher level of randomness, that is, if the statistical tests for measuring randomness are made more stringent. The Hirschfeldt-Miller theorem serves as a limiting result, as it implies that a 1-random set is weakly 2-random (defined in Section 2.3) if and only if it is not Turing above any noncomputable c.e. set. Hence we are driven to notions of randomness that are stronger than Martin-Löf's but still compatible with being Δ_2^0 . The natural notion that arises in this context is that of Demuth randomness [6]. Demuth tests generalize Martin-Löf tests $(\mathcal{G}_m)_{m \in \omega}$ in that one can change the m -th component (a Σ_1^0 class of measure at most 2^{-m}) a computably bounded number of times. A set Z fails a Demuth test if Z is in infinitely many final versions of the \mathcal{G}_m . In this direction, in mid-2009 Kučera and Nies [23] extended Greenberg's result by showing that every c.e. set computable from a Demuth random set is strongly jump-traceable.

In the current paper, we extend the result of Kučera and Nies [23] to give a fundamental connection between the second and third lowness paradigms discussed above. We extend the notion of *benign* cost functions, used by Greenberg and Nies to characterize strong jump-traceability, and show the relationship between the strength of generalized benign cost functions and being computable from sets of corresponding degree of randomness, as measured by generalizations of Demuth randomness. Translating Kučera's result to the language of cost functions, Greenberg and Nies have shown that being computable from a 1-random Δ_2^0 set can be forced by obedience to a corresponding cost function. A variant of a classic result of Ershov implies that the strength of every reasonable cost function can be gauged by some form of generalized benignity. Putting all of these results together, we get a full correspondence between the second and third paradigms for lowness properties: obedience to cost functions is equivalent to being computable from random Δ_2^0 sets. For the first time we get an abstract equivalence between paradigms along a wide array of cases, rather than just one witnessed by particular examples.

One question that we have not completely answered is how reliant our results are on the sets investigated being computably enumerable. Several of our implications do not use this

hypothesis, but we have not eliminated it completely. There are some preliminary results in this direction. Downey and Greenberg recently managed to eliminate the assumption of computable enumerability from one of the results from [4], showing that every strongly jump-traceable set is K -trivial. They conjecture that, like K -triviality, the concept of strong jump-traceability is inherently computably enumerable, that is, that the ideal of strongly jump-traceable degrees is generated by its c.e. elements. If true, this conjecture would imply that almost all of our results carry over to the general, non-c.e. case.

2. Overview

We give more technical detail on the ideas discussed above, and survey the results of the paper. We also provide some basic definitions, and fix notation. We assume familiarity with standard computability-theoretic notions and notation.

2.1. Traceability

We begin by defining a notion that is central to this paper. An *order function* is a nondecreasing, unbounded computable function h such that $h(0) > 0$. A *trace* for a partial function $\psi: \omega \rightarrow \omega$ is a uniformly c.e. sequence $\langle T_x \rangle$ of finite sets such that $\psi(x) \in T_x$ for all $x \in \text{dom } \psi$. A trace $\langle T_x \rangle$ is *bounded* by an order function h if $|T_x| \leq h(x)$ for all x .

Let h be an order function. A set A is *h -jump-traceable* if every partial function that is partial computable in A has a trace that is bounded by h . A set A is called *jump-traceable* if it is h -jump-traceable for some order function h . A set A is called *strongly jump-traceable* if it is h -jump-traceable for every order function h .

For every set A there is a universal partial A -computable function, which we denote by J^A . (We fix $A' = \text{dom } J^A$.) As this universality is witnessed by effective coding, it follows that a set A is jump-traceable if and only if J^A has a trace that is bounded by some order function. Similarly, a set A is strongly jump-traceable if and only if for every order function h , the function J^A has a trace that is bounded by h .

The class of jump-traceable sets is much larger than the class of strongly jump-traceable sets. There is a perfect class of jump-traceable sets [32], but every strongly jump-traceable set is Δ_2^0 , and indeed K -trivial [8].

2.2. Strong reducibilities, ω -c.e. sets, and superlowiness

A *computable approximation* to a set $A \in 2^\omega$ is a uniformly computable sequence $\langle A_s \rangle_{s < \omega}$ such that for every n , we have $A_s(n) = A(n)$ for almost all s . Associated with every computable approximation $\langle A_s \rangle$ is the mind-change function $n \mapsto \#\{s: A_{s+1}(n) \neq A_s(n)\}$. A set A is *ω -c.e.* if it has some computable approximation whose associated mind-change function is bounded by a computable function.

Let A and B be sets. Recall that $A \leq_{\text{wtt}} B$ if there is a Turing reduction of A to B with a computable bound on the use of this reduction, and that $A \leq_{\text{tt}} B$ if and only if A is B 's image under a total computable map from 2^ω to itself.

The following are equivalent for a set $A \in 2^\omega$:

- (1) $A \leq_{\text{wtt}} \emptyset'$;
- (2) $A \leq_{\text{tt}} \emptyset'$;
- (3) A is ω -c.e.

A set A is *superlow* [2,28] if A' is ω -c.e., or equivalently, if every set that is c.e. relative to A is ω -c.e. This formulation points to the fact that this notion does not depend on the choice

of enumeration of partial computable functions and hence of universal machine. It also shows immediately that every superlow set is ω -c.e.

Nies [32] showed that jump-traceability and superlowness coincide on the c.e. sets, but do not imply each other on the ω -c.e. sets. (For one direction, by the superlow basis theorem, further discussed in Section 2.8, there is a superlow 1-random set. On the other hand, no jump-traceable, or even c.e. traceable, set can be diagonally non-computable, while each 1-random set is diagonally non-computable. See [9] or [34] for definitions of these concepts.)

By analogy with the traditional notions of highness and lowness, we define a set A to be *superhigh* [28] if $\emptyset'' \leq_{\text{wtt}} A'$, or equivalently, if $\emptyset'' \leq_{\text{tt}} A'$. This notion too can be characterized in terms of approximations; we discuss this fact in Section 3.

2.3. Measure and randomness

We let λ denote the usual product (“fair coin”) measure on 2^ω . A (statistical) *test* is a sequence $\langle \mathcal{G}_n \rangle_{n < \omega}$ of effective (c.e.) open subclasses of 2^ω such that $\lambda \mathcal{G}_n \leq 2^{-n}$ for all n . A set Z *passes* a test $\langle \mathcal{G}_n \rangle$ if $Z \notin \mathcal{G}_n$ for almost all n . (The idea is that $\langle \mathcal{G}_n \rangle$ determines a null class $\lim \mathcal{G}_n = \bigcap_{n < \omega} \bigcup_{m \geq n} \mathcal{G}_m$ consisting of the sets that fail the test.)

A test $\langle \mathcal{G}_n \rangle$ is a *Martin-Löf test* if the sequence $\langle \mathcal{G}_n \rangle$ is uniformly c.e., that is, if there is a computable function f such that $f(n)$ is a c.e. index for \mathcal{G}_n for all n . A set Z is called *Martin-Löf random*, or *1-random*, if it passes every Martin-Löf test. There is a *universal* Martin-Löf test; in other words, there is a Martin-Löf test $\langle \mathcal{G}_n \rangle$ such that $\lim \mathcal{G}_n$ is the collection of sets that are not 1-random.

These notions can be relativized in the usual computability-theoretic manner, to yield, for instance, the notion of 1-randomness relative to a given set.

For more on algorithmic randomness, see [9,11,34].

2.4. Characterizations of strong jump-traceability and diamond classes

The main results of this paper are the characterizations of c.e. strong jump-traceability, along the lines of the second paradigm for lowness discussed in the introduction. In Section 4, we give the main argument that establishes the following.

Theorem 2.1. *If a set A is computable from every superlow 1-random set, then A is strongly jump-traceable.*

Here we do not assume that A is c.e. We elaborate on the proof of [Theorem 2.1](#) in Section 2.8 below.

In [16], it is proved that every c.e. strongly jump-traceable set is computable from all ω -c.e. 1-random sets. Since every superlow set is ω -c.e., this result, together with [Theorem 2.1](#), gives us two characterizations of c.e. strong jump-traceability. In the following let A be a c.e. set.

Characterization Ia. *A is strongly jump-traceable $\Leftrightarrow A$ is computable from every ω -c.e. 1-random set.*

Characterization Ib. *A is strongly jump-traceable $\Leftrightarrow A$ is computable from every superlow 1-random set.*

For the next characterization of strong jump-traceability, we impose a condition of complexity on the oracle, as opposed to the previous characterizations, where we imposed conditions of simplicity.

Characterization II. *A is strongly jump-traceable \Leftrightarrow A is computable from every superhigh 1-random set.*

We remark that in [22] it was already shown that some K -trivial c.e. set is not computable from all superhigh 1-random sets. The result appeared first in the conference paper [33] in extended abstract form. **Characterization II** is proved in Sections 5 and 6.

A set is LR-hard if \emptyset' is LR-reducible to it (see [9] or [34] for a definition of LR-reducibility). The implication from left to right of **Characterization II** improves a result from [16], that every c.e., strongly jump-traceable set is computable from every LR-hard 1-random set; Simpson [39] showed that every LR-hard set is superhigh. It was already noted in [16] that the collection of c.e. sets that are computable from all LR-hard 1-random sets strictly contains the strongly jump-traceable ones; it is still open whether this collection coincides with the collection of c.e. K -trivial sets.

We note that in **Characterization Ib**, we cannot replace superlowness by lowness, because the only sets that are computable in all 1-random low sets are the computable sets. This fact can be deduced from a variant of the low basis theorem that allows for upper-cone avoidance (see [34, Theorem 1.8.39]). Likewise, in **Characterization II**, we cannot replace superhighness by highness: the 1-random, high sets Ω and $\Omega^{\emptyset'}$ form a minimal pair, so the only sets that are computable from all 1-random high sets are the computable sets. (Here Ω is Chaitin’s well-known example of a 1-random set, and $\Omega^{\emptyset'}$ is its relativization to the halting problem.)

The proofs of the implications of strong jump-traceability (**Theorem 2.1** and the right-to-left direction of **Characterization II**) are of technical interest, as they use a variant of Nies’ golden run method that is not, in advance, bounded in depth. This method was developed to show that each K -trivial set is low for K (see [31,34]; for a definition of the concept of lowness for K , see [9] or [34]). Some nonuniformity seems to be a key for such an argument (for example, one cannot effectively obtain a constant witnessing lowness for K from a K -triviality constant; see [9] or [34] for a discussion of this result). This nonuniformity is amplified in the current constructions.

The following notation will be useful as shorthand. For a class of sets \mathcal{C} , let \mathcal{C}^\diamond denote the collection of c.e. sets that are computable from all 1-random sets in \mathcal{C} . The Hirschfeldt-Miller theorem already mentioned in the introduction states that if \mathcal{C} is a null Σ_3^0 class, then \mathcal{C}^\diamond contains a noncomputable (indeed, promptly simple) set. It extends Kučera’s classic result that every Δ_2^0 1-random set computes a noncomputable c.e. set, because the singleton $\{Y\}$ is Π_2^0 for any Δ_2^0 set Y . For more background on the diamond operator, see [34, Section 8.5] or [16].

The characterizations above can be written as the equalities

$$(\omega\text{-c.e.})^\diamond = \text{Superlow}^\diamond = \text{Superhigh}^\diamond = \text{SJT}_{\text{c.e.}}$$

where $\text{SJT}_{\text{c.e.}}$ is the collection of c.e., strongly jump-traceable sets.

We note that every class of the form \mathcal{C}^\diamond induces an ideal in the c.e. Turing degrees. Hence any of the equalities with $\text{SJT}_{\text{c.e.}}$ above implies the result from [4] that the strongly jump-traceable sets are closed under join.

2.5. On the assumption of computable enumerability

As mentioned in the introduction, the assumption that the sets in question are computably enumerable is not used in all of our characterizations. For example, **Theorem 2.1** above does not rely on such an assumption.

- (1) We do not know yet how to make good use of strong jump-traceability of a set that is not c.e.; for example, we know only that c.e. strongly jump-traceable sets obey all benign cost functions (see below). Thus, in showing that a c.e. strongly jump-traceable set is computable from all superlow and superhigh 1-random sets we make essential use of computable enumerability.
- (2) On the other hand, showing that sets that are computable from many 1-random oracles are strongly jump-traceable, or obey certain cost functions, does not seem to make essential use of the sets being c.e. In all of these examples, the property we use is that the set is superlow and jump-traceable.

The path from computable enumerability to superlowness and jump-traceability passes through the following fundamental facts, which follow from results in [18,31].

Fact 2.2. Suppose that a set A is computable from a set that is 1-random relative to A . (We call such a set a *base for 1-randomness*.) Then A is superlow and jump-traceable.

Proof. By Hirschfeldt, Nies and Stephan [18], A is K -trivial. By Nies [31], A is superlow and jump-traceable. \square

The following appears in [18].

Fact 2.3. If A is a c.e. set, Y is an incomplete 1-random set, and $A \leq_T Y$, then Y is 1-random relative to A , so A is a base for 1-randomness.

Combining these results, we have the following.

Corollary 2.4. Every c.e. set that is computable from an incomplete 1-random set is superlow and jump-traceable.

2.6. Cost functions

The third paradigm for lowness states that a Δ_2^0 set A is close to being computable if it has a computable approximation that changes little. Cost functions are the tools that are used to measure this amount of change. For background on cost functions, see [34, Section 5.3], [16], or [38].

A *cost function* is a computable function $c(x, s)$ that takes non-negative rational values. We say that c is *monotone* if c is nonincreasing in the first variable and nondecreasing in the second variable. If $c(x, s)$ is a monotone cost function, then $x \mapsto \lim_s c(x, s) = \sup_s c(x, s)$ is nonincreasing. A cost function c satisfies the *limit condition* if $\lim_s c(x, s)$ is finite for all x and

$$\lim_{x \rightarrow \infty} \sup_s c(x, s) = 0,$$

or equivalently, for all $\varepsilon > 0$, for almost all x , we have $c(x, s) < \varepsilon$ for all s .

Given a computable approximation $\langle A_s \rangle$ of a set A and a cost function c , the *total cost* of the approximation according to c is the quantity

$$\sum_s c(x, s) \llbracket x < s \text{ is least such that } A_{s-1}(x) \neq A_s(x) \rrbracket. \tag{1}$$

We say that a computable approximation $\langle A_s \rangle$ obeys a cost function c if the total cost of $\langle A_s \rangle$ according to c is finite. The intuitive meaning is that the total amount of changes (as

measured by c) is small. We say that a Δ_2^0 set obeys a cost function c if A has some computable approximation that obeys c .

The basic result regarding cost functions is that every cost function that satisfies the limit condition is obeyed by some noncomputable (indeed, promptly simple) c.e. set. This result has its roots in constructions of Kučera and Terwijn [26], and Downey, Hirschfeldt, Nies and Stephan [10]. The standard example for a cost function is $c_{\mathcal{K}}(x, s) = \sum_{i=x+1}^s 2^{-K_s(i)}$ (where K is prefix-free Kolmogorov complexity and K_s its stage s approximation). Nies [31] characterizes the K -trivial sets along the lines of the third lowness paradigm: a set is K -trivial if and only if it obeys $c_{\mathcal{K}}$.

Greenberg and Nies [16] provided a similar result for the c.e. strongly jump-traceable sets. They introduced a special class of cost functions c with the limit condition: in an effective sense $\sup_s c(v, s)$ converges quickly to 0 as $v \rightarrow \infty$.

Definition 2.5. A monotonic cost function c is called *benign* if there is a computable function $g: \mathbb{Q}^+ \rightarrow \omega$ with the following property: if $0 = v_0 < \dots < v_n$ and $c(v_i, v_{i+1}) \geq q$ for each $i < n$, then $n \leq g(q)$.

The main result of Greenberg and Nies [16] is that a c.e. set is strongly jump-traceable if and only if it obeys every benign cost function. We will apply the harder left-to-right implication several times. We will also improve the right-to-left direction in [Corollary 2.8](#) by discarding the hypothesis that the set is c.e.

2.7. Equivalence of the second and third lowness paradigms for Δ_2^0 sets

As our last main result we show that a particular realization of the second lowness paradigm, being computable from a sufficiently random set, is in a sense equivalent to the third paradigm. One direction was already obtained by Greenberg and Nies [16]. They defined, for any computable approximation $\langle Y_s \rangle$ of a Δ_2^0 set Y , a cost function c_Y , which satisfies the limit condition, such that if Y is 1-random, then every c.e. set that obeys c_Y is computable from Y . This construction is essentially a translation of Kučera's classic argument from [25] into the language of cost functions.

Thus, an appropriate cost function forces computability from a given 1-random Δ_2^0 set; if Y is also ω -c.e., then c_Y is benign. (Greenberg and Nies then used their characterization of strong jump-traceability in terms of obedience to benign cost functions to obtain their result that $\text{SJT}_{\text{c.e.}} \subseteq (\omega\text{-c.e.})^\circ$, that is, the left-to-right part of [Characterization Ia](#) discussed in [Section 2.4](#).)

Our last result provides a converse for c.e. sets.

Theorem 2.6. *For each cost function c with the limit condition, there is a 1-random Δ_2^0 set Y such that each c.e. set $A \leq_{\text{T}} Y$ obeys c .*

Demuth randomness is a notion stronger than 1-randomness that is still compatible with being Δ_2^0 . The m th component of a test can be replaced a computably bounded number of times. In the proof of [Theorem 2.6](#), we gauge how well-behaved the cost function c is by associating with it a computable well-ordering R . The level of randomness we need to impose on Y in order to obtain the result is given by a further strengthening of Demuth randomness, where the m th component of a test can be changed finitely many times while “counting down” along the well-ordering $\omega \cdot R$. The details, along with the formal definition of Demuth randomness and its strengthenings, are deferred to [Section 7](#).

Theorem 2.6 is related to **Characterization Ia**, and to Greenberg’s result [15] that there is a 1-random Δ_2^0 set Y such that every c.e. set computable from Y is strongly jump-traceable. As mentioned in the introduction, no ω -c.e. 1-random set Y can have this property. That is, **Characterizations Ia** and **Ib** cannot be replaced by analogous ones involving a single ω -c.e., or superlow, 1-random set, or indeed finitely many such sets. This fact can be argued in two ways. In [16] it is shown that no single benign cost function can force strong jump-traceability, whereas as we already stated, if Y is ω -c.e. then c_Y is benign. Alternatively, we can cite work by Ng [29], who showed that the index set of $\text{SJT}_{\omega\text{-c.e.}}$ is Π_4^0 complete. On the other hand, for any Δ_2^0 set Y , the index set $\{e \mid W_e \leq_T Y\}$ is Σ_4^0 .

However, **Characterization Ia** does imply the following.

Proposition 2.7. *There is a monotone cost function c that satisfies the limit condition, such that every set that obeys c is strongly jump-traceable.*

Proposition 2.7 and **Theorem 2.6** together yield a new proof of Greenberg’s result [15].

Proof of Proposition 2.7. The class of ω -c.e. sets is Σ_3^0 . Thus, the proof of the result of Hirschfeldt and Miller in [34, Theorem 5.3.15] provides a cost function c with the limit condition such that every set A obeying c is Turing below each 1-random ω -c.e. set. By **Theorem 2.1**, every such set A is strongly jump-traceable. It is easily verified that c is monotone. \square

As a corollary to **Theorem 2.1**, we now improve one direction of the result of Greenberg and Nies [16] stated after **Definition 2.5**, by discarding the hypothesis that the set is c.e.

Corollary 2.8. *Suppose a Δ_2^0 set A obeys all benign cost functions. Then A is strongly jump-traceable.*

Proof. For each ω -c.e. random set Y , the set A obeys the benign cost function c_Y defined in [16]. Hence $A \leq_T Y$. By **Theorem 2.1** this fact implies that A is strongly jump-traceable. \square

2.8. Extensions to general Π_1^0 classes

Recall that a Π_1^0 class is the complement of an effectively open (that is, c.e.) subclass of 2^ω ; equivalently, it is the collection of paths through a computable subtree of $2^{<\omega}$. We make extensive use of the fact that there are Π_1^0 classes all of whose elements are 1-random. This fact follows from the existence of a universal Martin-Löf test.¹ The collection of 1-random sets is in fact a union of Π_1^0 classes. We remark that a Π_1^0 class consisting only of 1-random elements cannot be null.

The fundamental result regarding Π_1^0 classes is the Jockusch-Soare superlow basis theorem [20], which states that every nonempty Π_1^0 class contains a superlow element.

The proof of the implication from right to left in **Characterization Ib** does not make special use of randomness. We will actually prove the following in Section 4.

Theorem 2.9. *Let \mathcal{P} be a nonempty Π_1^0 class, and suppose that A is a jump-traceable set computable from every superlow member of \mathcal{P} . Then A is strongly jump-traceable.*

¹ A specific example is the class consisting of sets X such that $K(X \upharpoonright_n) \geq n - 1$ for all n ; here K denotes prefix-free Kolmogorov complexity.

Theorem 2.1, and hence Characterizations Ia and Ib in Section 2.4, follow from Theorem 2.9 with the aid of the following observation, by applying Theorem 2.9 to any Π_1^0 class that consists only of 1-random sets.

Proposition 2.10. *A set that is computable from every superlow 1-random set is jump-traceable.*

Proof. By the superlow basis theorem, and the fact that there is a Π_1^0 class containing only 1-random sets, there is a superlow 1-random set Z . Splitting Z into two halves we can write $Z = X \oplus Y$. By van Lambalgen's Theorem [42], both X and Y are 1-random; indeed, they are relatively 1-random: X is 1-random relative to Y , and Y is 1-random relative to X . Since $X, Y \leq_T Z$, both X and Y are superlow.

By the assumption on A , we have $A \leq_T X, Y$. Since Y is 1-random relative to X , and $A \leq_T X$, we get that Y is 1-random relative to A . Since also $A \leq_T Y$, Fact 2.2 implies that A is jump-traceable. \square

A further corollary of Theorem 2.9 characterizes strong jump-traceability of a c.e. set in terms of PA-completeness. (Here a set is PA-complete if it computes a completion of Peano arithmetic, or equivalently, if it computes a member of any nonempty Π_1^0 class.) Every PA-complete set computes a 1-random set, and every ω -c.e. PA-complete set computes an ω -c.e. 1-random set.²

Corollary 2.11. *A c.e. set is strongly jump-traceable if and only if it is computable from every superlow (equivalently, ω -c.e.) PA-complete set.*

Remark 2.12. For the reverse problem—characterizing the class of sets that are reducible to superlow sets that are PA complete or 1-random—there is a difference between PA completeness and 1-randomness. Indeed, every superlow set is computable from *some* superlow, PA complete set: there is a Π_1^0 class \mathcal{P} that contains only PA-complete sets (say the class of $\{0, 1\}$ -valued, diagonally noncomputable functions). By a partial relativization of the superlow basis theorem in [34, Exercise 1.8.41], if A is superlow, then there is some $Z \in \mathcal{P}$ such that $(A \oplus Z)' \leq_{tt} A'$. Now, the class of PA-complete sets is upward closed in the Turing degrees, hence $A \oplus Z$ is PA-complete, is superlow, and computes A . So, in short, the class of sets that are computable from PA-complete, superlow sets is exactly the class of superlow sets.

As mentioned earlier, this assertion is not true if we replace PA-completeness by 1-randomness; if A is a c.e. set that is computable in some superlow (indeed, incomplete) 1-random set, then A is K -trivial [18], and not every superlow set is K -trivial.

For many Π_1^0 classes \mathcal{P} , any set computable in all superlow members of \mathcal{P} must in fact be computable. For instance, it is not hard to show that there is a Π_1^0 class \mathcal{P} without computable members such that any distinct $Y, Z \in \mathcal{P}$ form a minimal pair. On the other hand, there are Π_1^0 classes \mathcal{P} that do not consist only of 1-random sets or PA-complete sets, such that the class of sets that are computable in all superlow elements of \mathcal{P} is exactly the class of strongly jump-traceable c.e. sets $\mathbf{SJT}_{c.e.}$. Consider, for example, the notion of *complex* sets from [21]. A set Z is complex if there is some order function h such that $C(Z \upharpoonright_n) \geq h(n)$ for all n (here C

² There is a Medvedev complete Π_1^0 class \mathcal{P} containing a set in every PA-complete degree. There is a Π_1^0 class \mathcal{Q} containing only random sets. Since \mathcal{Q} is Medvedev reducible to \mathcal{P} , for every $X \in \mathcal{P}$ there is some $Z \in \mathcal{Q}$ such that $Z \leq_{tt} X$. This is because a Turing functional $\Phi: \mathcal{P} \rightarrow \mathcal{Q}$ can be extended to a functional that is total on all sets, and thus a truth-table functional. If $X \in \mathcal{P}$ is ω -c.e., then $X \leq_{tt} \emptyset'$, and so for all $Z \leq_{tt} X$ we have $Z \leq_{tt} \emptyset'$, i.e. Z is ω -c.e. For background on Medvedev reducibility, see [40].

denotes plain Kolmogorov complexity). It was shown in [21] that a set is complex if and only if there is some diagonally noncomputable function f that is weak-truth-table reducible to A . In [38], techniques from [16] are elaborated to show that every c.e., strongly jump-traceable set is computable in any ω -c.e. complex set. Hence if h is a sufficiently slow-growing order function, then the class \mathcal{P}_h of sets Z such that $C(Z \upharpoonright_n) \geq h(n)$ for all n is a nonempty Π_1^0 class with the desired property.

Replacing superlowiness by superhighness does not yield a theorem analogous to [Theorem 2.9](#). The reason is that in proving the left-to-right direction of [Characterization II](#), we use the fact that Π_1^0 classes of 1-random sets are not null, which allows for Kučera coding into these classes. Not all Π_1^0 classes admit such coding. However, Medvedev complete classes, such as the class of complete extensions of Peano arithmetic, or of $\{0, 1\}$ -valued diagonally noncomputable functions, do admit such coding; indeed coding into these classes is easier than into classes of 1-random sets, because the coding locations can be obtained effectively, essentially by Gödel's incompleteness theorem. Hence, a simpler form of the argument in [Section 6](#) would yield the following.

Theorem 2.13. *Every c.e. set that is computable from every superhigh PA-complete set is strongly jump-traceable.*

Note that this theorem, and the right-to-left direction of [Characterization II](#), can also be viewed as characterizations of the limits of upper-cone avoidance in “codable” Π_1^0 classes in the context of coding. We discuss this idea in [Section 6](#).

3. Restrained approximations

Recall that [Corollary 2.4](#) allows us, in the proof of the right-to-left directions of our characterizations, to replace the assumption that the set A is c.e. by the assumption that it is superlow and jump-traceable. In this section, on the way to proving these directions in [Sections 4](#) and [6](#), we comment on this property of the set A . We show how this property is exploited to obtain useful approximations for functions that are partial computable in A .

Note that if we assume that A is c.e., then the condition of jump-traceability of [Theorem 2.9](#) is guaranteed by the coincidence result in [32], that a c.e. set is superlow if and only if it is jump-traceable.

We remark that this coincidence extends to the n -c.e. sets by a result of Ng [30]. It is not hard to build a 2-c.e. jump-traceable (and hence superlow) set that is not Turing below a c.e. jump-traceable set. This fact shows that the class of superlow and jump-traceable degrees is in an essential sense larger than the class of superlow c.e. degrees, and thereby motivates the extension of some of our results to this case.

3.1. Bounded limit-recursive functions

Before we begin, we need to partially relativize the characterization of ω -c.e. functions mentioned in [Section 2.2](#). This relativization will be of use in one direction of [Theorem 3.5](#) below, and later when we discuss superhighness in [Sections 5](#) and [6](#).

The following definition, due to Cole and Simpson [5], is only a partial relativization of the notion of ω -c.e.-ness, because the bound on the number of mind changes remains computable.

Definition 3.1. Let X be a set. A function $f: \omega \rightarrow \omega$ is *bounded limit-recursive* in X (we write $f \in \text{BLR}(X)$) if there is an X -computable approximation $\langle f_s \rangle$ to f such that the associated mind-change function $n \mapsto \#\{s: f_{s+1}(n) \neq f_s(n)\}$ is bounded by a computable function.

Thus a function is ω -c.e. if and only if it is in $\text{BLR}(\emptyset)$. The following result, [5, Theorem 6.4], generalizes the characterization of ω -c.e. sets. Note that it holds only for sets, not for functions in general.

Fact 3.2. Let $X \subseteq \omega$. The following are equivalent for a set $A \subseteq \omega$:

- (1) $A \leq_{\text{tt}} X'$;
- (2) $A \leq_{\text{wtt}} X'$;
- (3) $A \in \text{BLR}(X)$.

Hence a set X is superhigh if and only if $\emptyset'' \in \text{BLR}(X)$.

The next fact, [5, Corollary 6.15], characterizes the conjunction of superlowiness and jump-traceability.

Fact 3.3. A set X is superlow and jump-traceable if and only if $\text{BLR}(X) = \text{BLR}(\emptyset)$, that is, if and only if every function that is in $\text{BLR}(X)$ is ω -c.e.

3.2. Functionals

We define a *partial computable functional* to be a partial computable function $\Gamma: 2^{<\omega} \times \omega \rightarrow \omega$ such that for all $x < \omega$, the domain of $\Gamma(-, x)$ is an antichain of $2^{<\omega}$ (in other words, this domain is prefix-free). The idea is that the functional is the collection of minimal oracle computations of an oracle Turing machine. For any $A \in 2^{\leq\omega}$ and $x < \omega$, we let $\Gamma^A(x) = y$ if there is some initial segment τ of A such that $\Gamma(\tau, x) = y$. Then Γ^A is an A -partial computable function, and every A -partial computable function is of the form Γ^A for some partial computable functional Γ . We write $\Gamma^A(x) \downarrow$ if $x \in \text{dom } \Gamma^A$; otherwise we write $\Gamma^A(x) \uparrow$. The use of a computation $\Gamma^A(x) = y$ is the length of the unique initial segment τ of A such that $\Gamma(\tau, x) = y$.

If $\langle A_s \rangle$ is a computable approximation to a Δ_2^0 set A , and $\langle \Gamma_s \rangle$ is an effective enumeration of (the graph of) a partial computable functional, then we let $\Gamma^A[s] = \Gamma_s^{A_s}$. Note that Γ_s is a finite set, and so $\text{dom } \Gamma^A[s]$ is computable, rather than just c.e. By convention, if $\Gamma_s(\tau, x) = y$ then $|\tau|, x, y < s$.

3.3. Existence of restrained approximations

Let $\langle A_s \rangle$ be a computable approximation to a Δ_2^0 set A , and let $\langle \Gamma_s \rangle$ be an enumeration of a partial computable functional. We say that $\langle A_s, \Gamma_s \rangle$ is an approximation to the A -partial computable function Γ^A .

Suppose that $\Gamma^A(x) \downarrow [s]$; let u be the use of that computation. We say that this computation is *destroyed* at stage $s + 1$ if $A_{s+1} \upharpoonright_u \neq A_s \upharpoonright_u$.

Definition 3.4. An approximation $\langle A_s, \Gamma_s \rangle$ to an A -partial computable function is a *restrained A -approximation* if there is some computable function g such that for all x , the number $g(x)$ bounds the number of stages s such that a computation $\Gamma^A(x) \downarrow [s]$ is destroyed at stage $s + 1$.

Theorem 3.5. The following are equivalent for a set $A \in 2^\omega$:

- (1) A is both superlow and jump-traceable.
- (2) Every A -partial computable function has a restrained A -approximation.

Proof. The easier implication is (2) \Rightarrow (1). Let C be a set that is c.e. in A . There is an A -partial computable function θ such that $C = \text{dom } \theta$. Let $\langle A_s, \Gamma_s \rangle$ be a restrained A -approximation to θ , witnessed by a computable function g . Let $C_s(x) = 1$ if $\Gamma^A(x) \downarrow [s]$; otherwise let $C_s(x) = 0$. Then $\langle C_s \rangle$ and g show that C is ω -c.e. Hence A is superlow.

Let θ be A -partial computable. Let $\langle A_s, \Gamma_s \rangle$ be a restrained A -approximation to θ , witnessed by a computable function g . Let

$$T_x = \left\{ \Gamma^A(x)[s] : s < \omega \ \& \ \Gamma^A(x) \downarrow [s] \right\}.$$

Then $|T_x| \leq g(x) + 1$ for every x , and $\theta(x) = \Gamma^A(x) \in T_x$ for all $x \in \text{dom } \theta$.

Hence every A -partial computable function has a trace bounded by some computable function. As discussed in Section 2.1, A is jump-traceable.

We now turn to the proof of the converse implication (1) \Rightarrow (2). Let A be a superlow, jump-traceable set, and let θ be an A -partial computable function. Let $\tilde{\Gamma}$ be a Turing functional such that $\tilde{\Gamma}^A = \theta$.

For $x \in \text{dom } \theta$, let $f(x) = A \upharpoonright_u$, where $u > 0$ is the use of the computation $\tilde{\Gamma}^A(x)$; for all $x \notin \text{dom } \theta$, let $f(x)$ be the empty string. Then $f \in \text{BLR}(A)$: indeed, $\text{dom } \theta$ is A -c.e., and so we can approximate f computably in A , changing our mind only once for $x \in \text{dom } \theta$, and not at all for $x \notin \text{dom } \theta$.

By Fact 3.3, f is ω -c.e. Let $\langle f_s \rangle$ be a computable approximation to f , with a mind-change function that is bounded by some computable function g . Let $x_s \leq s$ be the largest x such that

$$\forall y, z \leq x [f_s(y) \subseteq f_s(z) \text{ or } f_s(z) \subseteq f_s(y)],$$

and let $\sigma_s = \bigcup_{y \leq x_s} f_s(y)$. Then $\lim_s x_s = \infty$ and for each n , for almost all s we have $A \upharpoonright_n \subseteq \sigma_s$. Let $A_s(y) = \sigma_s(y)$ for $y < |\sigma_s|$ and $A_s(y) = 0$ otherwise. Then $\langle A_s \rangle$ is a computable approximation to A .

Let $\langle \tilde{\Gamma}_s \rangle$ be some computable enumeration of the Turing functional $\tilde{\Gamma}$. Now define an enumeration $\langle \Gamma_s \rangle$ of a partial computable functional $\Gamma \subseteq \tilde{\Gamma}$ as follows: at stage s , if the axiom $(\sigma, x) \mapsto y$ is already in $\tilde{\Gamma}_s$, enumerate that axiom into Γ_s if $x \leq x_s$ and $\sigma \subseteq f_s(x)$. Then $\Gamma^A = \theta$. To show that $\langle A_s, \Gamma_s \rangle$ is a restrained approximation, note that if $\Gamma^A(x) \downarrow [s]$ with use u , and that computation is destroyed at stage $s + 1$, then $u \leq |f_s(x)|$, and $f_s(y) \neq f_{s+1}(y)$ for some $y \leq x$. So the number of times this event can happen is bounded by $\sum_{y \leq x} g(y)$. \square

4. SJT_{c.e.} coincides with Superlow $^\circ$

In this section we prove Theorem 2.9. As explained above, together with the results in [16], this theorem provides Characterizations Ia and Ib (discussed in Section 2.4) of the strongly jump-traceable c.e. sets. We first fix some notation.

4.1. Notation for classes of sets

For a finite binary string $\sigma \in 2^{<\omega}$, we let $[\sigma]$ denote the sub-basic clopen subclass of 2^ω consisting of all infinite binary strings that extend σ . If W is a c.e. subset of $2^{<\omega}$, then we let $[W]^\prec = \bigcup_{\sigma \in W} [\sigma]$ be the effectively open subset of 2^ω determined by W . (A clopen class $[C]^\prec$ is determined by a finite set of strings C .)

A Π_1^0 class is the complement of some effectively open subclass of 2^ω . A Π_1^0 index for a Π_1^0 class \mathcal{P} is a c.e. index for a c.e. set $W \subseteq 2^{<\omega}$ such that $\mathcal{P} = 2^\omega \setminus [W]^\prec$.

A Π_1^0 class \mathcal{P} admits an approximation $\mathcal{P} = \bigcap_t \mathcal{P}_t$, where $\langle \mathcal{P}_t \rangle$ is a computable sequence of clopen subsets of 2^ω . Namely, we let $\mathcal{P}_t = 2^\omega \setminus [W_s]^\prec$, where $\langle W_s \rangle$ is an effective enumeration of the c.e. set W such that $\mathcal{P} = 2^\omega \setminus [W]^\prec$. (Alternatively, we can fix a computable tree $T \subseteq 2^{<\omega}$ such that \mathcal{P} is the collection of paths through T , and let \mathcal{P}_t be the union of $[\sigma]$ where $\sigma \in T$ has length t .) Given a Π_1^0 index for \mathcal{P} , we can effectively obtain the approximation $\langle \mathcal{P}_t \rangle$.

We use the compactness of 2^ω in an essential way: if \mathcal{P} is an empty Π_1^0 class, then there is some t such that \mathcal{P}_t is empty.

4.2. Discussion of the proof of Theorem 2.9

Let A be a jump-traceable set that is computable from every superlow member of a nonempty Π_1^0 class \mathcal{P} . As mentioned earlier, by the superlow basis theorem, A is superlow. By Theorem 3.5, every A -partial computable function has a restrained A -approximation.

We will show that for every order function h , every A -partial computable function has a trace bounded by $x \mapsto 2^{h(x)}$. This fact suffices for the strong jump-traceability of A since h can be an arbitrary order function. For the rest of this section, fix an order function h , and fix an A -partial computable function θ . Let $\langle A_s, \Gamma_s \rangle$ be a restrained A -approximation for θ , witnessed by a computable function g (as in Definition 3.4).

The strategy for obtaining a trace for θ is to try, and fail, to construct a superlow set $Z \in \mathcal{P}$ such that $A \not\leq_T Z$. Let $\langle \Phi_e \rangle$ be an effective enumeration of all Turing functionals. For each e , we attempt to meet the requirement $A \neq \Phi_e(Z)$. Overall, the construction consists of a recursive calling of strategies (or procedures); the strategy R^e which attempts to meet the e th requirement $A \neq \Phi_e(Z)$ is located at the e th level of the structure of all called strategies.

We recall the proof of the Jockusch-Soare superlow basis theorem [20]. A superlow element of a given nonempty Π_1^0 class \mathcal{Q} is obtained by recursively defining a sequence of decreasing subclasses of \mathcal{Q} , each deciding the next element of the jump. Given \mathcal{Q} , we let $\mathcal{Q}(0) = \mathcal{Q}$, and

$$\mathcal{Q}(n+1) = \begin{cases} \mathcal{Q}(n), & \text{if } n \in X' \text{ for all } X \in \mathcal{Q}(n) \\ \{X \in \mathcal{Q}(n) : n \notin X'\} & \text{otherwise.} \end{cases}$$

Then $\bigcap_n \mathcal{Q}(n)$ is a singleton $\{Z\}$ where Z is superlow. To see this, we approximate the sequence $\langle \mathcal{Q}(n) \rangle$. For a finite binary string $\alpha \in 2^{<\omega}$, recursively define a subclass $\mathcal{Q}(\alpha)$ of \mathcal{Q} as follows: let $\mathcal{Q}(\langle \rangle) = \mathcal{Q}$; given $\mathcal{Q}(\alpha)$, let $\mathcal{Q}(\alpha 1) = \mathcal{Q}(\alpha)$, and let $\mathcal{Q}(\alpha 0) = \{X \in \mathcal{Q}(\alpha) : |\alpha| \notin X'\}$. At stage s of an effective construction, we define $\alpha_s \in 2^{<\omega}$ to be the leftmost binary string α of length s such that $(\mathcal{Q}(\alpha))_s$ is nonempty. If $s < t$, then $(\mathcal{Q}(\alpha))_s \supseteq (\mathcal{Q}(\alpha))_t$ for all α , so α_t does not lie to the (lexicographic) left of α_s . Hence the total number of stages s such that $\alpha_s \upharpoonright_n \neq \alpha_{s+1} \upharpoonright_n$ is at most 2^n .

One avoidance, that is, meeting the requirements $A \neq \Phi_e(Z)$, can also be obtained in a similar fashion (“forcing with Π_1^0 classes”): we intersect the given class with one of the classes $\mathcal{R}_{e,\tau} = \{X : \Phi_e(X) \not\supseteq \tau\}$ for some finite initial segment τ of A . Thus we attempt to intersperse these classes with classes as above for the superlowness of Z . The assumption on A implies that this attempt will fail. The failure is due to the fact that at some level e , all attempts to diagonalize $\Phi_e(A)$ away from an initial segment of A yield empty Π_1^0 classes. This fact gives us a method for confirming “believable” computations $\Gamma^A(x) \downarrow [s]$, and hence building a trace for θ .

The combinatorial content of the construction is showing how to effectively approximate this final outcome, as in the computable approximation of the forcing proof of the superlow basis

theorem. We need to show that if the attempts to build a trace fail, that is, if all the requirements are met, then the set Z constructed is indeed superlow.

Fix a level e . For each x , a strategy S_x^e is responsible for confirming computations $\Gamma^A(x)\downarrow[s]$. Say such a computation appears, with some use u . The strategy S_x^e tests whether $A_s \upharpoonright_u$ is really an initial segment of A by attempting to meet the e -th requirement by intersecting the current class with the class $\mathcal{R}_{e, A_s \upharpoonright_u}$. The strategy then waits for the resulting intersection to become empty; if A moves in the meantime, the computation $\Gamma^A(x)[s]$ is destroyed and no harm is done. As long as the class is not empty, it seems like the e -th requirement is met, and so a new strategy for meeting the $(e + 1)$ -st requirement is called in the meantime, starting a new superlow basis construction within that Π_1^0 class. If the resulting class turns out to be empty, $A_s \upharpoonright_u$ is confirmed and the computation $\Gamma^A(x)[s]$ traced.

To show that the construction succeeds, we then argue for a contradiction and assume that at all levels e , some strategy S_x^e succeeds in meeting the e -th requirement. The key, as mentioned, is to ensure that the resulting set Z is superlow, even though the superlowness strategies are distributed over all the levels of the construction. Premature changes in A may cause difficulties here. Say a strategy S_x^e calls a procedure R^{e+1} while trying to certify a computation $\Gamma^A(x)[s]$; this run of R^{e+1} may then be cancelled due to an A change that destroys that computation. This cancellation may in turn change our approximation to Z' . To put a computable bound on the number of times such an event can occur, we use the fact that $\langle A_s, \Gamma_s \rangle$ is restrained.

4.3. Golden pairs

Say that the construction above succeeds at a level e . The following definition captures the relevant properties of the final Π_1^0 class \mathcal{Q} that is passed to the successful run of R^e , and of the associated Turing functional Φ_e . We again use the notation $\langle \mathcal{Q}(n) \rangle$ to denote the sequence of Π_1^0 classes obtained in the proof of the superlow basis theorem.

Definition 4.1. A pair \mathcal{Q}, Φ , consisting of a nonempty Π_1^0 class and a Turing functional, is a *golden pair* for Γ and h if for almost all x such that $\Gamma^A(x)\downarrow$, with some use u , for all $X \in \mathcal{Q}(h(x))$ we have $\Phi(X) \supseteq A \upharpoonright_u$.

The proof that θ has a trace bounded by 2^h is split into two separate propositions. The first verifies that golden pairs indeed yield traces.

Proposition 4.2. *If there is a golden pair for Γ and h , then $\theta = \Gamma^A$ has a trace $\langle V_x \rangle$ such that $|V_x| \leq 2^{h(x)}$ for each x .*

The second proposition asserts the existence of a golden pair.

Proposition 4.3. *If A is computable from every superlow member of \mathcal{P} , then there are a Π_1^0 class $\mathcal{Q} \subseteq \mathcal{P}$ and a functional Φ such that \mathcal{Q}, Φ is a golden pair for Γ and h .*

Proof of Proposition 4.2. Let \mathcal{Q}, Φ be a golden pair for Γ and h . We let the Π_1^0 class $\mathcal{Q}(n)[s]$ be the stage s approximation to $\mathcal{Q}(n)$. It is defined inductively like $\mathcal{Q}(n)$, but assessed with the information present at stage s . That is, if $\Phi_e^\sigma(e)\downarrow$ for each σ such that $[\sigma] \subseteq (\mathcal{Q}(n)[s])_s$ (that is, $n \in X'$ for all $X \supset \sigma$), then we let $\mathcal{Q}(n+1)[s] = \mathcal{Q}(n)[s]$; otherwise, we let

$\mathcal{Q}(n + 1)[s] = \{X \in \mathcal{Q}(n)[s] : n \notin X'\}$. As mentioned above, for every n , there are at most 2^n many Π_1^0 classes that are ever chosen to be $\mathcal{Q}(n)[s]$.³

We enumerate a number y into a set V_x at stage s if at that stage we discover that there is a binary sequence τ such that $\Gamma_s(\tau, x) = y$ and such that $\Phi(X)$ extends τ for every $X \in (\mathcal{Q}(h(x))[s])_s$ (which means that for all strings σ of length s such that $[\sigma] \subseteq (\mathcal{Q}(h(x))[s])_s$, we have $\Phi(\sigma) \supseteq \tau$).

It suffices to show that some finite variant of $\langle V_x \rangle$ is a trace for θ that is bounded by 2^h . The sequence $\langle V_x \rangle$ is uniformly c.e. For any version of $\mathcal{Q}(h(x))[s]$, at most one number y gets enumerated into V_x , so $|V_x|$ is bounded by the number $2^{h(x)}$ of possible choices for $\mathcal{Q}(h(x))[s]$. Finally, for almost all $x \in \text{dom } \theta$, for large enough s , for every $X \in \mathcal{Q}(h(x))[s] = \mathcal{Q}(h(x))$ we have $\Phi(X) \supseteq \tau = A \upharpoonright_u$, where u is the use of the computation $\Gamma^A(x)$. Then $\theta(x) \in V_x$ for almost all $x \in \text{dom } \theta$. \square

4.4. A golden pair exists

The heart of the proof of [Theorem 2.9](#) is the proof of [Proposition 4.3](#): that under the assumptions on A and \mathcal{P} , a golden pair exists for A and h . As already mentioned in the introduction, the mechanism is a nonuniform argument in the spirit of the golden run method from [\[31\]](#), except that the procedure-calling structure now has unbounded depth.

The argument was sketched already in our discussion leading to the [Definition 4.1](#) of golden pairs. For every e , a procedure R^e , provided with some Π_1^0 subclass \mathcal{P}^e of \mathcal{P} as input, attempts to show that \mathcal{P}^e, Φ_e is a golden pair for A and h . For each $x < \omega$, if $\Gamma^A(x) \downarrow$ with use u , then a subprocedure S_x^e wants to either give permanent control to the next level $e + 1$, or show that the golden pair condition holds at x for $\mathcal{Q} = \mathcal{P}^e$: for all $X \in \mathcal{P}^e(h(x))$ we have $\Phi_e(X) \supseteq A \upharpoonright_u$.

The procedures and the construction

A typical procedure calling structure at any stage of the construction is

$$R^0 \rightarrow S_y^0 \rightarrow \dots \rightarrow R^e \rightarrow S_x^e \rightarrow R^{e+1} \rightarrow \dots$$

The instructions for our procedures are simple.

Procedure R^e . This procedure runs with input \mathcal{P}^e (a Π_1^0 class) and a parameter $n < \omega$. While R^e is running, every number x such that $h(x) > n$ is marked as either *fresh* or *confirmed*. At the inception of R^e , all numbers x such that $h(x) > n$ are marked as fresh.

If R^e has control at some stage s , and there is some x that is fresh at stage s and such that $\Gamma^A(x) \downarrow [s]$ with use $u < s$, then for the least such x , we call a subprocedure S_x^e with input $\tau = A_s \upharpoonright_u$.

Procedure S_x^e . A run of this procedure is provided with a string τ —an initial segment of the current state of A —that witnesses that $\Gamma^A(x) \downarrow [s]$. It acts as follows.

(a) Start a run of R^{e+1} , with the input

$$\mathcal{P}^{e+1} = \{X \in \mathcal{P}^e(h(x)) : \Phi_e(X) \not\supseteq \tau\},$$

and parameter $h(x)$.

³ Again, recall that for a Π_1^0 class \mathcal{Q} , we let \mathcal{Q}_s be the clopen set which consists of the elements which have not been removed from \mathcal{Q} by stage s , according to a canonical enumeration of Π_1^0 classes. On the other hand, by $\mathcal{Q}(n)[s]$ we mean the Π_1^0 class which is chosen to be $\mathcal{Q}(n)$ at stage s ; at a later stage t , $\mathcal{Q}(n)[t]$ may be an entirely different Π_1^0 class.

As long as we do not see that for every $X \in \mathcal{P}^e \langle h(x) \rangle$ we have $\Phi_e(X) \supseteq \tau$, that is, as long as \mathcal{P}^{e+1} appears to be nonempty (and so S_x^e has not yet succeeded), we halt all activity for R^e and let the run of R^{e+1} take its course.

- (b) If we see that \mathcal{P}^{e+1} is empty, we cancel the run of R^{e+1} (and any of its subprocedures), and return control to R^e , marking x as confirmed.

A run of S_x^e started at a stage s with input $\tau = A_s \upharpoonright_u$ believes that $\mathcal{P}^e \langle h(x) \rangle [s] = \mathcal{P}^e \langle h(x) \rangle$ (and indeed that $\mathcal{P}^e \langle h(x) \rangle [s] = \mathcal{P}^e \langle h(x) \rangle [t]$ for all $t > s$), and that $\tau \subset A$. If either of these beliefs is incorrect, then we let t be the least stage at which we discover this incorrectness: either $\tau \not\subset A_t$, or $\mathcal{P}^e \langle h(x) \rangle [t] \neq \mathcal{P}^e \langle h(x) \rangle [t - 1]$. If S_x^e is still running at stage t , then we immediately cancel it (along with the run of R^{e+1} it called and all of its subprocedures), and return control to R^e . If S_x^e has already returned control to R^e , then we re-mark x as fresh at stage t .

The entire construction is started by calling R^0 with input $\mathcal{P}^0 = \mathcal{P}$ and parameter 0.

Verification

We show that there is some e such that \mathcal{P}^e, Φ_e is a golden pair for Γ and h (for some stable version of \mathcal{P}^e). A *golden run* is a run of a procedure R^e that is never cancelled, such that every subprocedure S_x^e that is called by that run eventually returns or is cancelled.

Claim 4.4. *If there is a golden run of R^e with input \mathcal{Q} , then \mathcal{Q}, Φ_e is a golden pair for Γ and h .*

Proof. Suppose the golden run of R^e is called with parameter n . Note that its input \mathcal{Q} is the final version of \mathcal{P}^e . Since h is an order function, for almost all x we have $h(x) > n$.

We show that for every x , only finitely many runs of S_x^e are ever called. Let $x < \omega$ be such that $h(x) > n$. If $x \notin \text{dom } \theta$, then since $\langle A_s, \Gamma_s \rangle$ is a restrained approximation, we have $\Gamma^A(x) \downarrow [s]$ for only finitely many stages s . Thus, in this case, there is a stage after which no run of S_x^e is called.

Suppose that $x \in \text{dom } \theta$. Let u be the use of the computation $\Gamma^A(x)$. For sufficiently late s we have $A_s \upharpoonright_u \subset A$ and $\mathcal{Q} \langle h(x) \rangle [t] = \mathcal{Q} \langle h(x) \rangle [s]$ for all $t > s$. If a run of S_x^e is called at such a late stage s , then it will never be cancelled. When it returns, x will be marked confirmed, and never re-marked fresh; hence no later run of S_x^e will ever be called.

A similar argument shows that if $x \in \text{dom } \theta$ and $h(x) > n$, then a run of S_x^e will indeed be called and never cancelled: We can wait for a stage s that is late enough so that the conditions above hold and, in addition, $\Gamma_s(A_s \upharpoonright_u, x) = \theta(x)$ and no run S_y^e for any $y < x$ is ever called after stage s . If x is marked fresh at such a stage s , then a run of S_x^e will be called and never cancelled. Since the run of R^e is golden, such a run will return, and x will be marked confirmed and never re-marked fresh.

Let $x \in \text{dom } \theta$ be such that $h(x) > n$. Let s be the stage at which the last run of S_x^e is called. As we just argued, this run is not cancelled; it returns at some stage $t > s$, and x is confirmed at all stages after t . We thus have $\mathcal{Q} \langle h(x) \rangle = \mathcal{Q} \langle h(x) \rangle [s]$, and $A_s \upharpoonright_u \subset A$, where u is the use of the computation $\Gamma^A(x)$. At stage t we witness the fact that $\Phi_e(X) \supseteq A_s \upharpoonright_u$ for all $X \in \mathcal{Q} \langle h(x) \rangle$.

Thus \mathcal{Q}, Φ_e is a golden pair as required. \square

It remains to show that there is a golden run of some R^e . We first need to do some counting, to establish a computable bound $N(x)$ on the number of times a procedure S_x^e (for any e) is called. We then argue as follows. Suppose there is no golden run, so every run of every R^e is either eventually cancelled, or calls some run of S_x^e that is never cancelled but never returns. By induction on e we can see that for every e , there is a run of R^e that is never cancelled, with a final version of \mathcal{P}^e . The sequence of Π_1^0 classes $\mathcal{P}^0, \mathcal{P}^1, \dots$ is nested, and so its intersection $\bigcap_e \mathcal{P}^e$

is nonempty. Let $Z \in \bigcap_e \mathcal{P}^e$. We will show that we can use approximations to the trees \mathcal{P}^e to computably approximate Z' , and that we can use our computable bounds on the number of times procedures can be called to ensure a computable bound on the number of changes. Hence Z is superlow. By our hypothesis on A , there will be some e such that $\Phi_e(Z) = A$. Consider the run of S_x^e that is never cancelled nor returns, which defines the last version of \mathcal{P}^{e+1} . It defines

$$\mathcal{P}^{e+1} = \{X \in \mathcal{P}^e \langle h(x) \rangle : \Phi_e(X) \not\geq \tau\},$$

where $\tau \subset A$ (since S_x^e is never cancelled). But this definition contradicts the fact that $Z \in \mathcal{P}^{e+1}$.

We now give the details of this argument. Recall that $g(x)$ is the computable function from Definition 3.4 bounding how often a computation $\Gamma^A(x)$ can be destroyed.

Claim 4.5. *For each e and x , every run of R^e calls at most $g(x) + 2^{h(x)}$ many runs of S_x^e .*

Proof. Suppose that at stage s , a run of S_x^e is cancelled while the run of R^e that called it is not cancelled. Let \mathcal{P}^e be the input of this run of R^e , and let τ be the input of S_x^e .

One of the following possibilities holds:

- (a) $\mathcal{P}^e \langle h(x) \rangle [s] \neq \mathcal{P}^e \langle h(x) \rangle [s - 1]$; or
- (b) $\tau \subset A_{s-1}$ but $\tau \not\subset A_s$.

The first possibility occurs fewer than $2^{h(x)}$ many times. The second, by the fact that $\langle A_s, \Gamma_s \rangle$ is a restrained approximation for θ , occurs at most $g(x)$ many times. \square

Claim 4.6. *There is a computable bound $N(x)$ on the number of times a procedure S_x^e is called for any e .*

Proof. We calculate, by recursion on e and x , a bound $M(e, x)$ on the number of times any run of R^e calls a run of S_x^e . We use Claim 4.5. Since there is only one run of R^0 , we can let $M(0, x) = g(x) + 2^{h(x)}$. For $e > 0$ we let $M(e, x)$ be the product of $g(x) + 2^{h(x)}$ with a bound on the number of runs of R^e that are called by some S_y^{e-1} with parameter $h(y) < h(x)$.

Since $h(y) < h(x)$ implies $y < x$, the number of runs of R^e with a parameter less than $h(x)$ is bounded by

$$\sum_{y < x} M(e - 1, y).$$

This completes the recursive definition of M . Now, by induction on e , the parameter of any run of R^e is at least e . So we can let $N(x) = \sum_{e < h(x)} M(e, x)$. \square

Now suppose for a contradiction that there is no golden run. So every run of every R^e is either eventually cancelled, or calls some run of S_x^e that is never cancelled but never returns. As mentioned above, by induction on e we can see that for every e , there is a run of R^e that is never cancelled, with a final version of \mathcal{P}^e .

The sequence of Π_1^0 classes $\mathcal{P}^0, \mathcal{P}^1, \dots$ is nested, and so its intersection $\bigcap_e \mathcal{P}^e$ is nonempty. Let $Z \in \bigcap_e \mathcal{P}^e$.

Claim 4.7. *Z is superlow.*

Proof. Let $n > 0$, and let e be the least number such that the permanent run of R^e is started with a parameter greater than n . As mentioned during the proof of Claim 4.6, the parameter of any run of R^e is at least e , so such an e exists.

Whether $n \in Z'$ depends only on $\mathcal{P}^{e-1}\langle n + 1 \rangle$. So we can approximate an answer to the question of whether $n \in Z'$ by tracking, at a stage s , the definition of $\mathcal{P}^d\langle n + 1 \rangle$ at that stage, where d is the greatest number such that the current (at stage s) run of R^d was started with a parameter $h(x) \leq n$.

The current version of $\mathcal{P}^d\langle n + 1 \rangle$ can change because we call the procedure S_x^e for some $h(x) \leq n$. Otherwise, it can change due to the approximation feature of the proof of the superlow basis theorem (see the proof of Proposition 4.2). Thus the number of changes is bounded by

$$2^{n+1} \sum_{h(x) \leq n} N(x),$$

which is a computable bound. Thus the above procedure gives an ω -c.e. approximation to Z' . \square

By the assumption on A , we have $A \leq_T Z$. Hence there is some e such that $\Phi_e(Z) = A$. Consider the run of S_x^e that is never cancelled nor returns, which defines the last version of \mathcal{P}^{e+1} . It defines

$$\mathcal{P}^{e+1} = \{X \in \mathcal{P}^e\langle h(x) \rangle : \Phi_e(X) \not\subseteq \tau\},$$

where $\tau \subset A$. As already explained above, this definition contradicts the fact that $Z \in \mathcal{P}^{e+1}$. This completes the proof of Proposition 4.3 and so of Theorem 2.9.

Remark 4.8. To show that A is strongly jump-traceable, it is sufficient to show that for every order function h , a universal A -partial computable function θ has a trace bounded by h . The reader may wonder why we bother with every A -partial computable function, rather than just a universal one. Let J be a partial computable functional such that for all sets X , the function J^X is a universal X -partial computable function. Even though $\theta = J^A$ is universal, the restrained A -approximation for θ gives a partial computable functional Γ such that $\Gamma^A = \theta$, but for other sets X it will not be the case that Γ^X is universal for X -partial computable functions. In the proof, it is the approximation $\Gamma^A[s]$ that we use, not $J^A[s]$, so we might as well work with a general function, rather than just a universal one.

5. $\text{SJT}_{\text{c.e.}}$ is contained in Superhigh $^\diamond$

In this section, we provide the left-to-right direction of Characterization II (see Section 2.4) of the strongly jump-traceable c.e. sets: every c.e., strongly jump-traceable set is computable from every superhigh 1-random set.

In fact, we prove a slightly stronger result, Theorem 5.1, by replacing the class of superhigh sets by a larger null Σ_3^0 class \mathcal{H} , introduced by Simpson [39], which is related to PA-completeness. We actually show that every strongly jump-traceable c.e. set is in \mathcal{H}^\diamond .

To define \mathcal{H} , recall that a function f is diagonally non-computable (d.n.c.) relative to Y if for all $x \in \text{dom } J^Y$, we have $f(x) \neq J^Y(x)$. (Recall also that J denotes a partial computable functional such that for every set Y , the function J^Y is a universal Y -partial computable function.)

Let \mathcal{P} be the $\Pi_1^0(\emptyset')$ class of $\{0, 1\}$ -valued functions that are d.n.c. relative to \emptyset' . By a result of Jockusch [19] relativized to \emptyset' , the class

$$\{Z : \exists f \leq_T Z \oplus \emptyset' [f \in \mathcal{P}]\}$$

is null. The class $\text{GL}_1 = \{Z: Z' \equiv_{\text{T}} Z \oplus \emptyset'\}$ contains every 2-random and hence is conull (see, for instance, [9]). Thus, the following class is also null:

$$\mathcal{H} = \{Z: \exists f \leq_{\text{tt}} Z' [f \in \mathcal{P}]\}.$$

This class contains **Superhigh** because \emptyset'' truth-table computes a function that is d.n.c. relative to \emptyset' .

Since \mathcal{H} is Σ_3^0 and null, we already know, by the result of Hirschfeldt and Miller mentioned in the introduction, that the class \mathcal{H}° contains a noncomputable set. We now strengthen this fact.

Theorem 5.1. *Every c.e., strongly jump-traceable set is in \mathcal{H}° , that is, is computable from every 1-random set in \mathcal{H} .*

Fix a truth-table reduction Δ . We will define a benign cost function c such that for each set A , and each 1-random set Z ,

$$\Delta(Z') \text{ is } \{0, 1\}\text{-valued d.n.c. relative to } \emptyset' \text{ and } A \text{ obeys } c \quad \Rightarrow \quad A \leq_{\text{T}} Z.$$

Theorem 5.1 then follows from the result from [16], that every c.e., strongly jump-traceable set obeys every benign cost function.

5.1. Discussion

We first explain in intuitive terms how to obtain the cost function c . The overall strategy has roots in the proof in [4] that every c.e., strongly jump-traceable set is not ML-cupable (see [9] or [34] for a definition of this concept), and in the proof in [16] that every c.e., strongly jump-traceable set is computable from every LR-hard 1-random set.

Suppose that we are given a c.e. set A , and we wish to show that $A \leq_{\text{T}} Z$ for all 1-random sets Z such that $\Delta(Z') \in \mathcal{P}$. We implicitly devise a Turing functional that reduces A to such sets Z . Since there are uncountably many such sets Z , and they are not all definable in any way, we have to work with finite initial segment of such Z —equivalently, with clopen classes of such Z . We can describe our strategy as a two-pronged attack. First, we require evidence that some clopen class \mathcal{C} consists of sets Z such that $\Delta(Z') \in \mathcal{P}$. If we find such evidence, at some stage s , then we decide that the sets in \mathcal{C} compute some initial segment of A_s . Second, if we later discover that this computation is incorrect because A has changed, *and* if it still seems like the sets in \mathcal{C} satisfy $\Delta(Z') \in \mathcal{P}$, then we try to make these sets non-1-random. In terms of the Kolmogorov complexity definition of 1-randomness, essentially what we do is give initial segments of sets in \mathcal{C} short descriptions; the technical device we actually use is a Solovay test \mathcal{G} , which we describe below. (A *Solovay test* is a c.e. collection of clopen sets C_0, C_1, \dots such that the total weight $\sum_i \lambda C_i$ is finite. It is easy to check that if a set X is 1-random and \mathcal{S} is a Solovay test, then X can be in only finitely many elements of \mathcal{S} ; see [9] or [34] for a proof.) Viewed backwards, this derandomization allows us to correct the functional. The cost function c is defined by tracking our beliefs and thus “pricing” the changes in the set A according to the amount of correction that would be required, were A to change.

The combinatorial heart of the argument is the exact designation of when we believe that a clopen class \mathcal{C} consists of sets Z such that $\Delta(Z') \in \mathcal{P}$. This is the basic tension: on the one hand, if indeed $\Delta(Z') \in \mathcal{P}$, then we need to ensure that we believe this fact for infinitely many initial segments of Z . On the other hand, we cannot run wild and issue too many short descriptions: the total weight of those descriptions has to be finite. In other words, by derandomizing strings, we

may ask for corrections in the functional, but this right is limited—we cannot ask for too much. If we believe too many strings, the total weight of the Solovay test will not be finite.

To decide whether to believe a clopen class \mathcal{C} , we define a function $\alpha \leq_T \mathcal{O}'$, by giving it a computable approximation $\langle \alpha_s \rangle$. We believe \mathcal{C} at stage s if the stage s approximation to $\Delta(Z')$ for sets $Z \in \mathcal{C}$ differs from α_s on designated locations (or really, from the coding of α in $J^{\mathcal{O}'}$ at that stage). By designating a large number of such locations, we can “keep ahead of the game” by changing α_s if it appears that we believe clopen classes that are too large (in the sense of measure). This prophylactic approach is really the main point of the argument.

5.2. The proof of Theorem 5.1

We now give the details. Let (I_e) be the sequence of consecutive intervals of ω of length $e + 1$. Thus $\min I_e = e(e + 1)/2$. As mentioned, we define a function α , partial computable in \mathcal{O}' (which will actually be total). By universality of $J^{\mathcal{O}'}$, and by the recursion theorem, we are given a computable function p that reduces α to $J^{\mathcal{O}'}$: for all x , $\alpha(x) \simeq J^{\mathcal{O}'}(p(x))$.

Let $s < \omega$. To define α_s , we first let $\mathcal{C}_{e,s}$ be the clopen set of oracles Z such that $\Delta(Z')$ agreed with $1 - \alpha$ on I_e at some stage t after the last change of $\alpha \upharpoonright_{I_e}$. That is, let

$$\mathcal{C}_{e,s} = \{Z: \exists t \ v \leq t \leq s \ \forall x \in I_e [1 - \alpha_t(x) = \Delta(Z'_t, p(x))]\}, \tag{2}$$

where $v \leq s$ is greatest such that $v = 0$ or $\alpha_v \upharpoonright_{I_e} \neq \alpha_{v-1} \upharpoonright_{I_e}$. Here Z'_t is the (finite) set of elements which have been enumerated into Z' by stage t of an enumeration of Z' with oracle Z . For each $e < s$, if $\lambda \mathcal{C}_{e,s-1} \leq 2^{-e+1}$ let $\alpha_s \upharpoonright_{I_e} = \alpha_{s-1} \upharpoonright_{I_e}$. Otherwise, change $\alpha \upharpoonright_{I_e}$: define $\alpha_s \upharpoonright_{I_e}$ in such a way that $\lambda \mathcal{C}_{e,s} \leq 2^{-e}$.

Claim 5.2. $\alpha(x) = \lim_s \alpha_s(x)$ exists for each x .

Proof of the Claim. We rely on a measure theoretic fact first used in a related context (see [34, Exercise 1.9.15]). Suppose $n \in \omega$ and we are given measurable classes \mathcal{B}_i for $1 \leq i \leq N$, and $\lambda \mathcal{B}_i \geq 2^{-e}$ where $e \in \omega$. If $k \in \omega$ is such that $N > 2^e k$, then there is a set $F \subseteq \{1, \dots, N\}$ such that $|F| = k + 1$ and $\bigcap_{i \in F} \mathcal{B}_i \neq \emptyset$. Beyond proving the claim, this fact will later yield a computable bound in x on the number of changes of $\alpha_s(x)$.

Suppose that $v_1 < \dots < v_N$ are consecutive stages at which $\alpha \upharpoonright_{I_e}$ changes. Note that for each $i < n$, the measure $\lambda \mathcal{C}_e$ increases by at least 2^{-e} from stage v_i to v_{i+1} . Therefore $\lambda \mathcal{B}_i \geq 2^{-e}$ for each $i \leq N$, where

$$\mathcal{B}_i = \{Z: Z'_{v_{i+1}} \upharpoonright_k \neq Z'_{v_i} \upharpoonright_k\},$$

and $k = \text{use } \Delta(\max p(I_e))$. Note that the intersection of any $k + 1$ many of the \mathcal{B}_i is empty. Thus $N \leq 2^e k$ by the measure theoretic fact mentioned above. \square

In fact, we have a computable bound g on the number of changes of $\alpha \upharpoonright_{I_e}$, given by $g(e) = 2^e \text{use } \Delta(\max p(I_e))$.

We define a cost function c by $c(x, s) = 2^{-x}$ for each $x \geq s$; if $x < s$, and $e \leq x$ is least such that $e = x$ or $\alpha_s \upharpoonright_{I_e} \neq \alpha_{s-1} \upharpoonright_{I_e}$, let

$$c(x, s) = \max(c(x, s - 1), 2^{-e}).$$

To show that c is benign, suppose that $0 = v_0 < v_1 < \dots < v_n$ and $c(v_i, v_{i+1}) \geq 2^{-e}$ for each $i < n$. Then $\alpha_s \upharpoonright_{I_e} \neq \alpha_{s-1} \upharpoonright_{I_e}$ for some s such that $v_i < s \leq v_{i+1}$. Hence $n \leq g(e)$.

To complete the proof of **Theorem 5.1**, let A be a c.e. set that is strongly jump-traceable. By Greenberg and Nies [16], there is a computable enumeration $\langle A_s \rangle_{s \in \omega}$ of A that obeys c .

The rest of the argument actually works for a computable approximation $\langle A_s \rangle_{s \in \omega}$ to a Δ_2^0 set A . We build a Solovay test \mathcal{G} as follows: when $A_{t-1}(x) \neq A_t(x)$, we put $\mathcal{C}_{e,t}$ defined in (2) into \mathcal{G} where e is largest such that $\alpha \upharpoonright I_e$ has been stable from x to t . Then $2^{-e} \leq c(x, t)$. Since $\lambda \mathcal{C}_{e,t} \leq 2^{-e+1} \leq 2c(x, t)$ and the computable approximation of A obeys c , the set \mathcal{G} is indeed a Solovay test.

Choose s_0 such that $\sigma \not\subseteq Z$ for each $[\sigma]$ enumerated into \mathcal{G} after stage s_0 . To show $A \leq_T Z$, given an input $y \geq s_0$, using Z as an oracle, compute $s > y$ such that $1 - \alpha_s(x) = \Delta(Z'_s; x)$ for each $x < y$. Then we claim that $A_s(y) = A(y)$. Assume not, so that $A_t(y) \neq A_{t-1}(y)$ for some $t > s$, and let $e \leq y$ be largest such that $\alpha \upharpoonright I_e$ has been stable from y to t . Then by stage $s > y$ the set Z is in $\mathcal{C}_{e,s} \subseteq \mathcal{C}_{e,t}$, so we put Z into \mathcal{G} at stage t , which is a contradiction.

6. Superhigh $^\circ$ is contained in SJT $_{c.e.}$

In this section, we prove the right-to-left direction of **Characterization II** of the strongly jump-traceable c.e. sets: every c.e. set that is computable from every superhigh 1-random set is strongly jump-traceable.

As in Section 5, we prove a somewhat stronger result. For any set $G \subseteq \omega$, we replace the class of superhigh sets by the class

$$\mathcal{C}_G = \{Y : G \leq_{tt} Y'\}.$$

This class is a subclass of the superhigh sets if $\emptyset'' \leq_T G$. No matter what G is, we show that every set in $(\mathcal{C}_G)^\circ$ is strongly jump-traceable.

6.1. The path from computable enumerability to superlowiness and jump-traceability

Fix $G \subseteq \omega$. We want to prove that every c.e. set A that is computable from every 1-random set in \mathcal{C}_G is strongly jump-traceable. As mentioned in Section 2.5, the assumption on A that we actually use, rather than A being c.e., is that A is superlow and jump-traceable.

Theorem 6.1. *Let A be a superlow, jump-traceable set, let $G \subseteq \omega$, and suppose that for any 1-random set Z such that $G \leq_{tt} Z'$ we have $A \leq_T Z$. Then A is strongly jump-traceable.*

In order to replace c.e. by superlow and jump-traceable, we need the following lemma.

Lemma 6.2. *Let $G \subseteq \omega$. If A is a c.e. set that is computable from every 1-random set Z such that $G \leq_{tt} Z'$, then A is superlow and jump-traceable.*

Lemma 6.2 follows from Corollary 2.4 and the following consequence of Kjos-Hanssen and Nies [22, Theorem 3.5].

Theorem 6.3. *For any $G \subseteq \omega$ there is an incomplete 1-random set Z such that $G \leq_{tt} Z'$.*

As a gentle introduction to **Theorem 6.1**, we include a proof of **Theorem 6.3**. It turns out that the proof of **Theorem 6.1** is closely related to our proof of **Theorem 6.3**, in a fashion even stronger than the way the proof of **Theorem 2.9** relates to the proof of the superlow basis theorem. For the proof of **Theorem 6.3**, we start with a Π_1^0 class \mathcal{S} consisting of 1-random sets, enumerate a set A , and use a generalized version of Kučera coding to build some $Z \in \mathcal{S}$ that codes G , in the

sense that $G \leq_{tt} Z'$, but that avoids A , in the sense that $A \not\leq_T Z$. The proof of [Theorem 6.1](#) is a reversal, of sorts, of the same situation, in which A is given, but we try to construct such a set $Z \in \mathcal{S}$ nonetheless. Our failure to avoid A is then translated, as was done in Section 4, into a golden pair, and so into an enumeration of a trace for the given A -partial computable function. In this way, the proof of [Theorem 6.3](#) serves as a blueprint for the proof of [Theorem 6.1](#). Also, in some sense, this argument shows that strong jump-traceability is exactly the level at which the power of upper-cone avoidance in conjunction with coding fails in “codable” Π_1^0 classes such as classes of 1-random sets or Medvedev complete classes (i.e., Π_1^0 classes that have the highest possible degree in the Medvedev lattice of mass problems).

We remark that for $G = \emptyset''$, which is the case we are interested in to prove the right-to-left direction of [Characterization II, Theorem 6.3](#) can be proved by a 1-random pseudo-jump inversion technique, to obtain a Δ_2^0 1-random set Z . See [[34, Theorem 6.3.14](#)].

6.2. Kučera coding

We start with a review of Kučera coding into Π_1^0 classes of 1-random sets. For a string τ and a class $\mathcal{B} \subseteq 2^\omega$, let $\mathcal{B} \upharpoonright \tau = \{X \in 2^\omega : \tau X \in \mathcal{B}\}$. If \mathcal{B} is a measurable class, then $\lambda(\mathcal{B} \upharpoonright \tau) = 2^{|\tau|} \lambda(\mathcal{B} \cap [\tau])$.

Recall that a string $\tau \in 2^{<\omega}$ is called *extendible* in a Π_1^0 class \mathcal{P} if $\mathcal{P} \cap [\tau]$ is nonempty, or equivalently, if $\mathcal{P} \upharpoonright \tau$ is nonempty. If \mathcal{P} has positive measure, this notion can be strengthened: for any $r < \omega$, we say that $\tau \in 2^{<\omega}$ is *r-extendible* in \mathcal{P} if $\lambda(\mathcal{P} \upharpoonright \tau) \geq 2^{-r}$.

Let \mathcal{P} be a Π_1^0 class of positive measure, and let $r < \omega$ be sufficiently large so that $\lambda\mathcal{P} \geq 2^{-r}$. We define an embedding of the full binary tree into subclasses of \mathcal{P} of positive measure defined as the intersections of \mathcal{P} with basic clopen classes. That is, for every finite binary string α we define a string $\text{kuc}_r(\mathcal{P}, \alpha)$ such that:

- if $\alpha \subset \beta$, then $\text{kuc}_r(\mathcal{P}, \alpha) \subset \text{kuc}_r(\mathcal{P}, \beta)$; if $\alpha \perp \beta$, then $\text{kuc}_r(\mathcal{P}, \alpha) \perp \text{kuc}_r(\mathcal{P}, \beta)$;
- for all $\alpha \in 2^{<\omega}$, the string $\text{kuc}_r(\mathcal{P}, \alpha)$ is $r + |\alpha|$ -extendible in \mathcal{P} .

We define $\text{kuc}_r(\mathcal{P}, \alpha)$ by recursion on α , based on the following lemma.

Lemma 6.4 (*Kučera; see also [[34, Lemma 3.3.1](#)]*). *Suppose that \mathcal{P} is a Π_1^0 class, $l < \omega$, and $\tau \in 2^{<\omega}$ is l -extendible in \mathcal{P} . Then there are at least two strings $\sigma \supset \tau$ of length $|\tau| + l + 1$ that are $l + 1$ -extendible in \mathcal{P} .*

We let $\text{kuc}_r(\mathcal{P}, \emptyset)$ be the leftmost string τ of length r that is r -extendible in \mathcal{P} . If $\text{kuc}_r(\mathcal{P}, \alpha)$ has been defined, then we let $\text{kuc}_r(\mathcal{P}, \alpha 0)$ be the leftmost extension of $\text{kuc}_r(\mathcal{P}, \alpha)$ of length $|\text{kuc}_r(\mathcal{P}, \alpha)| + r + |\alpha| + 1$ that is $r + |\alpha| + 1$ -extendible in \mathcal{P} , and let $\text{kuc}_r(\mathcal{P}, \alpha 1)$ be the rightmost such extension.

For all α , the length of $\text{kuc}_r(\mathcal{P}, \alpha)$ is

$$r + |\alpha|r + \binom{|\alpha|}{2} - 1.$$

We define

$$\ell(n, r) = r(n + 1) + \binom{n}{2} - 1;$$

so $|\text{kuc}_r(\mathcal{P}, \alpha)| = \ell(|\alpha|, r)$ for all α . The point is that the map $\alpha \mapsto \text{kuc}_r(\mathcal{P}, \alpha)$ is not computable, but ℓ is.

This simple version of Kučera coding is sufficient to prove the Kučera-Gács Theorem, that every set is computable from a 1-random set. For, let \mathcal{P} be a Π_1^0 class consisting of 1-random sets. We know that $\lambda\mathcal{P} > 0$, so fix some r such that $\lambda\mathcal{P} \geq 2^{-r}$. Let $G \in 2^\omega$ and let

$$Z = \bigcup_n \text{kuc}_r(\mathcal{P}, G \upharpoonright_n).$$

The reason for $G \leq_T Z$ is that we can effectively determine, given $Z \upharpoonright_{\ell(n,r)} = \text{kuc}_r(\mathcal{P}, G \upharpoonright_n)$, whether $Z \upharpoonright_{\ell(n+1,r)}$ is the leftmost or the rightmost extension of $Z \upharpoonright_{\ell(n,r)}$ of its length that is $r + n + 1$ -extendible in \mathcal{P} , because the set of l -extendible strings is co-c.e., uniformly in l .

This last argument points to an effective approximation to the coding strings. Recall the descending approximation $\langle \mathcal{P}_t \rangle$ to \mathcal{P} by clopen sets (from Section 4.1). Of course, $\lambda\mathcal{P}_t \geq \lambda\mathcal{P}$ for all t , so if $\lambda\mathcal{P} \geq 2^{-r}$, then for all t and all α , the string $\text{kuc}_r(\mathcal{P}_t, \alpha)$ is defined, and indeed effectively obtained, uniformly in α and t (and in r). In fact, $\langle \text{kuc}_r(\mathcal{P}_t, \alpha) \rangle_{t < \omega}$ is a computable approximation, with a computably bounded number of changes, to the function $\alpha \mapsto \text{kuc}_r(\mathcal{P}, \alpha)$. For, if $\text{kuc}_r(\mathcal{P}_t, \alpha)$ is stable along an interval of stages, then in this interval we see at most $2^{\ell(|\alpha|+1,r) - \ell(|\alpha|,r)}$ many changes in $\text{kuc}_r(\mathcal{P}_t, \alpha 0)$ (and the same holds for $\text{kuc}_r(\mathcal{P}, \alpha 1)$), again because the set of $r + |\alpha| + 1$ -extendible extensions of $\text{kuc}_r(\mathcal{P}_t, \alpha)$ of length $\ell(|\alpha| + 1, r)$ is (uniformly) co-c.e. Inductively, we obtain the following.

Lemma 6.5. *For any α , the number of stages t such that*

$$\text{kuc}_r(\mathcal{P}_{t+1}, \alpha) \neq \text{kuc}_r(\mathcal{P}_t, \alpha)$$

is bounded by $2^{\ell(|\alpha|,r)}$.

6.3. Lower bound functions

If we want to combine coding with Friedberg-Muchnik style diagonalization, one Π_1^0 class is not sufficient: we need to pass to Π_1^0 subclasses that avoid computations that currently look correct. To use Kučera coding on each of these classes, we need, effectively in the index of a class, a positive lower bound on its measure. The lower bound function is the map giving this lower bound.

In the proofs of both Theorems 6.3 and 6.1, we begin with a Π_1^0 class \mathcal{S} of 1-random sets and enumerate a c.e. set V ; from each $v \in V$ we effectively compute an index for a Π_1^0 subclass $\mathcal{P}^{(v)}$ of \mathcal{S} . A lower bound function for $\langle \mathcal{P}^{(v)} \rangle_{v \in V}$ is a (total) computable function q such that for all $v \in V$, if $\mathcal{P}^{(v)}$ is nonempty, then $\lambda\mathcal{P}^{(v)} \geq 2^{-q(v)}$.

Lemma 6.6. *Let \mathcal{S} be a Π_1^0 class of 1-random sets. Any effective list $\langle \mathcal{P}^{(v)} \rangle_{v \in V}$ of Π_1^0 subclasses of \mathcal{S} has a lower bound function. Moreover, a computable index for the function can be computed effectively from an index for the enumeration $\langle \mathcal{P}^{(v)} \rangle$.*

Proof. This result is best proved using a basic result on prefix-free Kolmogorov complexity: By the Kraft-Chaitin Theorem (see for instance [34, Theorem 2.2.17]), there is a coding constant c_0 such that $\lambda\mathcal{P}^{(v)} \leq 2^{-K(v)-c_0} \rightarrow \mathcal{P}^{(v)} = \emptyset$ (see [34, Exercise 3.3.3 and its solution]). Fix $d \in \omega$ such that $K(v) \leq 2 \log v + d$. Let $q(v) = 2 \log v + c_0 + d$. The constant c_0 can be obtained effectively from the enumeration $\langle \mathcal{P}^{(v)} \rangle$ because the Kraft-Chaitin Theorem is uniform. \square

Hence, by the recursion theorem, we may assume that a lower bound function for the classes enumerated during the construction is known to us during the construction; we fix such a function q .

Since the descending, clopen, effective approximation $\langle \mathcal{P}_t \rangle$ to a Π_1^0 class \mathcal{P} is obtained effectively from a canonical index for \mathcal{P} , we get such an approximation $\langle \mathcal{P}_t^{(v)} \rangle$ uniformly for all $v \in V$. We may assume that for all $v \in V$ and $t < \omega$, if $\lambda \mathcal{P}_t^{(v)} < 2^{-q(v)}$, then $\mathcal{P}_t^{(v)}$ is empty. To omit an index, for all $v \in V$ such that $\mathcal{P}^{(v)}$ is nonempty, we let $\text{kuc}(\mathcal{P}^{(v)}, \alpha) = \text{kuc}_{q(v)}(\mathcal{P}^{(v)}, \alpha)$, and similarly, $\text{kuc}(\mathcal{P}_t^{(v)}, \alpha) = \text{kuc}_{q(v)}(\mathcal{P}_t^{(v)}, \alpha)$, which is defined if and only if $\mathcal{P}_t^{(v)}$ is nonempty, a condition that is effectively detectable. The map $(v, t, \alpha) \mapsto \text{kuc}(\mathcal{P}_t^{(v)}, \alpha)$ is computable on its domain, which is itself computable.

6.4. Proof of Theorem 6.3

We start with a Π_1^0 class of 1-random sets \mathcal{S} ; this class has positive measure. We enumerate a c.e. set A , against which we try to diagonalize. For coding, we approximate coding strings σ_γ by giving their stage s versions $\sigma_{\gamma,s}$. For diagonalization, we approximate Π_1^0 classes \mathcal{S}_γ , subclasses of \mathcal{S} , such that for all γ and all $Z \in \mathcal{S}_\gamma$, we have $\Phi_{|\gamma|}(Z) \neq A$. For compatibility of coding and diagonalization, we ensure that for all γ we have $\mathcal{S}_\gamma \subset [\sigma_\gamma]$, and that for both $j < 2$, the string σ_{γ^j} is extendible in \mathcal{S}_γ .

At stage s we define $\mathcal{S}_{\gamma,s}$, an approximation to \mathcal{S}_γ . As mentioned above, we will enumerate a c.e. set V . Each class $\mathcal{S}_{\gamma,s}$ will be of the form $\mathcal{P}^{(v)}$ for some $v \in V$. If $\mathcal{S}_{\gamma,s-1} = \mathcal{P}^{(v)}$, and at stage s we decide not to change this class, that is, we decide that $\mathcal{S}_{\gamma,s} = \mathcal{S}_{\gamma,s-1}$, then of course we have $\mathcal{S}_{\gamma,s} = \mathcal{P}^{(v)}$ for the same v . If we decide to pick a new class, so that $\mathcal{S}_{\gamma,s} \neq \mathcal{S}_{\gamma,s-1}$, then we enumerate a new element u into V at stage s , and define $\mathcal{S}_{\gamma,s} = \mathcal{P}^{(u)}$. The number u will equal $\langle \gamma, k \rangle$, where k is the number of previous versions of $\mathcal{S}_{\gamma,t}$, and we identify $2^{<\omega}$ with ω in a natural way.

Note the multiplicity of the subscript s : $\mathcal{S}_{\gamma,s}$ is a Π_1^0 class; the clopen class that is its stage s approximation will be denoted by $(\mathcal{S}_{\gamma,s})_s$.

Construction

At stage s , we define $\mathcal{S}_{\gamma,s}$ and $\sigma_{\gamma,s}$ by recursion on γ . Starting with $\gamma = \emptyset$, we let $\sigma_{\emptyset,s} = \emptyset$.

Now suppose that $\sigma_{\gamma,s}$ is defined. If $\gamma \neq \emptyset$, let $\gamma^- = \gamma \upharpoonright_{|\gamma|-1}$ be γ with the last bit chopped off. By induction, $\mathcal{S}_{\gamma^-,s}$ is already defined. If $\gamma = \emptyset$, let $\mathcal{S}_{\gamma^-,s} = \mathcal{S}$.

There are three possibilities.

- (1) If $s = 0$, or if $s > 0$ but $\sigma_{\gamma,s} \neq \sigma_{\gamma,s-1}$, then we pick a new value for $\mathcal{S}_{\gamma,s}$. Let v be the new index for $\mathcal{S}_{\gamma,s}$, which we enumerate into V .

We let

$$\mathcal{S}_{\gamma,s} = \{Z \in \mathcal{S}_{\gamma^-,s} \cap [\sigma_{\gamma,s}] : \neg (\Phi_{|\gamma|}(Z, v) \downarrow = 0)\}.$$

- (2) If $s > 0$ and $\sigma_{\gamma,s} = \sigma_{\gamma,s-1}$, but $(\mathcal{S}_{\gamma,s-1})_s$ is empty, then let $\mathcal{S}_{\gamma,s} = \mathcal{S}_{\gamma^-,s} \cap [\sigma_{\gamma,s}]$.
- (3) Otherwise, let $\mathcal{S}_{\gamma,s} = \mathcal{S}_{\gamma,s-1}$.

After $\mathcal{S}_{\gamma,s}$ is defined, for both $j < 2$, we let $\sigma_{\gamma^j,s} = \text{kuc}(\mathcal{S}_{\gamma,s}, \langle j \rangle)$.

Verification

Claim 6.7. For all γ , both $\sigma_{\gamma,s}$ and $\mathcal{S}_{\gamma,s}$ stabilize to final values σ_γ and \mathcal{S}_γ . The approximations are both ω -c.e.

Proof. By induction on γ . We always have $\sigma_{\emptyset,0} = \emptyset$.

Let γ be any string, and suppose that in an interval $[t_0, t_1]$ of stages, the value of $\sigma_{\gamma,s}$ is constant. Then $\mathcal{S}_{\gamma,s}$ is changed at most once between stages t_0 and t_1 .

Suppose now that in an interval $[t_0, t_1]$ of stages, the class $\mathcal{S}_{\gamma,s}$ is not redefined. Let $v \in V$ be the number such that $\mathcal{S}_{\gamma,s} = \mathcal{P}^{(v)}$ for all $s \in [t_0, t_1]$. Let $j < 2$. By Lemma 6.5, in stages between t_0 and t_1 , the value of $\sigma_{\gamma j,s}$ changes at most $2^{\ell(1,q(v))}$ many times.

Hence both $\sigma_{\gamma,s}$ and $\mathcal{S}_{\gamma,s}$ reach a limit. To see that the number of changes is bounded computably in γ , we again argue by recursion on γ . If $\mathcal{S}_{\gamma,s}$ changes at most m many times, then we recall that the V -indices for \mathcal{S}_γ are $(\gamma, 0), (\gamma, 1), \dots$, and so for both $j < 2$, the number of times $\sigma_{\gamma j,s}$ changes is bounded by

$$\sum_{k \leq m} 2^{\ell(1,q((\gamma,k)))}$$

which is computable. \square

It is clear from the instructions that for all γ , we have $\mathcal{S}_\gamma \subset \mathcal{S}_{\gamma^-}$ and $\mathcal{S}_\gamma \subset [\sigma_\gamma]$.

Claim 6.8. *For all γ , we have $\mathcal{S}_{\gamma^-} \cap [\sigma_\gamma] \neq \emptyset$.*

Proof. By induction on γ . For $\gamma = \emptyset$ the claim follows from $\sigma_\emptyset = \emptyset$ and $\mathcal{S} \neq \emptyset$.

Suppose that $\mathcal{S}_{\gamma^-} \cap [\sigma_\gamma] \neq \emptyset$. Let $j < 2$. We show that $\mathcal{S}_\gamma \cap [\sigma_{\gamma j}] \neq \emptyset$.

First, we see that $\mathcal{S}_\gamma \neq \emptyset$, which follows from the instructions. Let s_0 be the stage at which $\sigma_{\gamma,s}$ stabilizes. At stage s_0 , we pick a new value for \mathcal{S}_γ . If at a later stage s_1 we discover that \mathcal{S}_{γ,s_0} is empty, then we switch to the final value $\mathcal{S}_\gamma = \mathcal{S}_{\gamma^-} \cap [\sigma_\gamma]$, which by induction is nonempty. Otherwise, $\mathcal{S}_\gamma = \mathcal{S}_{\gamma,s_0}$ is nonempty.

Now that we know that \mathcal{S}_γ is nonempty, we know that each Kučera string $\text{kuc}(\mathcal{S}_\gamma, \alpha)$ is extendible in \mathcal{S}_γ , in particular $\sigma_{\gamma j} = \text{kuc}(\mathcal{S}_\gamma, \langle j \rangle)$. \square

Now define a c.e. set A as follows: at a stage $s > 0$, if $\sigma_{\gamma,s} = \sigma_{\gamma,s-1}$ but $(\mathcal{S}_{\gamma,s-1})_s = \emptyset$ (that is, case (2) of the construction holds for γ at stage s), then enumerate the V -index of $\mathcal{S}_{\gamma,s-1}$ (that is, the number $v \in V$ such that $\mathcal{S}_{\gamma,s-1} = \mathcal{P}^{(v)}$) into A .

Claim 6.9. *For all γ and all $Z \in \mathcal{S}_\gamma$, we have $\Phi_{|\gamma|}(Z) \neq A$.*

Proof. Fix γ , and let s_0 be the stage at which σ_γ stabilizes. Let v be the V -index of \mathcal{S}_{γ,s_0} .

If there are no changes in $\mathcal{S}_{\gamma,s}$ after stage s_0 , that is, if $\mathcal{S}_\gamma = \mathcal{S}_{\gamma,s_0}$, then $v \notin A$, and by the definition of \mathcal{S}_{γ,s_0} , for no $Z \in \mathcal{S}_\gamma$ do we have $\Phi_{|\gamma|}(Z, v) = 0$.

Otherwise, at some stage $s_1 > s_0$ we redefine $\mathcal{S}_{\gamma,s_1} = \mathcal{S}_{\gamma^-} \cap [\sigma_\gamma]$, and there are no further changes in \mathcal{S}_γ . But this redefinition is done only because at stage s_1 we discover that $\mathcal{S}_{\gamma,s_0} = \emptyset$. By the definition of \mathcal{S}_{γ,s_0} , we thus have $\Phi_{|\gamma|}(Z, v) = 0$ for all $Z \in \mathcal{S}_\gamma$. But in this case, $v \in A$. \square

Let $G \in 2^\omega$. Define $Z = \bigcup_n \sigma_G \upharpoonright_n$. If $n \leq m < \omega$ then by Claim 6.8,

$$\emptyset \neq \mathcal{S}_G \upharpoonright_m \cap [\sigma_G \upharpoonright_{m+1}] \subseteq \mathcal{S}_G \upharpoonright_n \cap [\sigma_G \upharpoonright_{m+1}],$$

so by compactness, $Z \in \mathcal{S}_G \upharpoonright_n$ for all n . By Claim 6.9, $A \not\leq_T Z$, so Z is incomplete.

Claim 6.10. $G \leq_{tt} Z'$.

Proof. By Fact 3.2 (due to Cole and Simpson), it is equivalent to show that $G \in \text{BLR}(Z)$.

To construct a stage s -approximation to G , note that if γ and δ are incomparable strings, then $\sigma_{\gamma,s}$ and $\sigma_{\delta,s}$ are also incomparable. Hence

$$G_s = \bigcup \{ \gamma : \sigma_{\gamma,s} \subset Z \}$$

is well-defined (it may be finite). It is (uniformly) Z -computable, because for all γ and s , we have $|\sigma_{\gamma,s}| \geq |\gamma|$.

Let $x < \omega$. By Claim 6.7, there is a computable bound on the number of stages at which any of the strings $\sigma_{\gamma,s}$ for any string γ of length $x + 1$ may change. The approximation $G_s(x)$ may change only at such stages. Hence $\langle G_s \rangle$ witnesses that $G \in \text{BLR}(Z)$. \square

6.5. Discussion of the proof of Theorem 6.1

The rest of this section is dedicated to the proof of Theorem 6.1. Fix $G \in 2^\omega$. We assume that A is a superlow and jump-traceable set, computable from every 1-random set Z such that $G \leq_{tt} Z'$. As in Section 4, fix an order function h , and an A -partial computable function θ . By Theorem 3.5, we can fix a restrained A -approximation $\langle A_s, \Gamma_s \rangle$ to θ , witnessed by a computable function g .

As discussed above, this proof will follow the idea of the proof of Theorem 2.9, but mirroring the proof of Theorem 6.3. Thus we again begin with a nonempty Π_1^0 class \mathcal{S} that contains only 1-random sets. We will adapt the definition of a golden pair to the current setting; such a pair will arise from a failure to construct some $Z \in \mathcal{S}$ that both codes G and does not compute A . Again we will approximate strings σ_γ that are extendible in \mathcal{S} and serve as coding strings, and again we will make the approximation $\langle \sigma_{\gamma,s} \rangle_{s \in \omega}$ an ω -c.e. one. As in the proof of Theorem 2.9, subclasses in which we attempt to diagonalize against A will be tied to computations $\Gamma^A(x)[s]$ that are under a process of verification.

The fact that the proof of the superlow basis theorem is linear in nature, but the proof of the coding Theorem 6.3 is not, makes the structure of runs of procedures more complicated in the current proof. Rather than having a linear structure, we now have an (infinitely branching) tree of runs of procedures at any given stage. Thus, while before we had at most one procedure of each type per level e , now many of them run in parallel. When a procedure R^e is called, an initial segment $\eta \subset G$ will have been coded into Z' already. So we now have versions $R^{e,\eta}$ for various strings η . A subprocedure S_x^e has to live with the coding into Z' of a further string of length $h(x)$. Thus, we have versions $S_x^{e,\eta\alpha}$ for each α of length $h(x)$. During the construction, this feature leads to some extra cancellations, as we need to be able to replace a run $S_y^{e,\eta\beta}$ by $S_x^{e,\eta\alpha}$ for $\alpha \subset \beta$. In the definition of golden pairs we will fix an η and refer only to runs $S_x^{e,\eta\alpha}$ where $\eta\alpha \subseteq G$.

For coding, we again work with an effective list $\mathcal{P}^{(v)}$ of subclasses of \mathcal{S} that we enumerate. Every class we define will be on this list. We mostly leave this as an implicit part of the construction.

6.6. Golden pairs

Definition 6.11. A pair \mathcal{Q}, Φ , consisting of a nonempty Π_1^0 class $\mathcal{Q} = \mathcal{P}^{(v)}$ and a Turing functional Φ , is a *golden pair* for Γ, h , and G , with parameter $\eta \subset G$, if for almost all x such that $\Gamma^A(x) \downarrow$, with use u , for the α of length $h(x)$ such that $\eta\alpha \subset G$, if $X \in \mathcal{Q} \cap [\text{kuc}(\mathcal{Q}, \alpha)]$, then $\Phi(X) \supseteq A \upharpoonright_u$.

Proposition 6.12. If there is a golden pair for Γ, h , and G , with parameter $\eta \subset G$, then $\theta = \Gamma^A$ has a c.e. trace $\langle V_x \rangle_{x \in \omega}$ such that $|V_x| \leq 2^{h^2(x)}$ for all x .

Proof. Let $\mathcal{Q} = \mathcal{P}^{(v)}$ and Φ be such a golden pair. At stage s we enumerate a number y into V_x if there is a string α of length $h(x)$ such that at that stage we discover that there is a binary sequence τ for which $\Gamma_s(\tau, x) = y$ and $\Phi(X)$ extends τ for every $X \in \mathcal{Q}_s \cap [\text{kuc}(\mathcal{Q}_s, \alpha)]$.

To establish the bound on $|V_x|$, let $n = h(x)$. By Lemma 6.5, for all strings α of length n , the string $\text{kuc}(\mathcal{Q}, \alpha)$ changes at most $2^{\ell(n, q(v))}$ many times. We have $\ell(n, q(v)) \leq (3/4)n^2$ for almost all n . Taking the union over all strings α of length n , we obtain $|V_x| \leq 2^n 2^{\ell(n, q(v))}$, which is bounded by 2^{n^2} for almost all n .

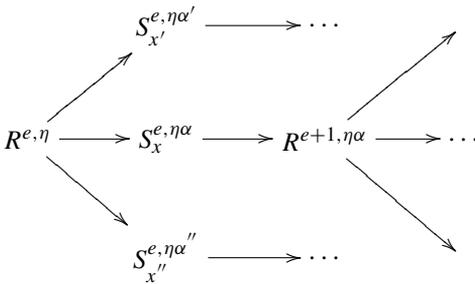
To establish tracing, if $x \in \text{dom } \theta$, let α be of length $h(x)$ such that $\eta\alpha \subseteq G$. Let t_x be so large that $\text{kuc}(\mathcal{Q}, \alpha) = \text{kuc}(\mathcal{Q}_s, \alpha)$ for all $s \geq t_x$. Since \mathcal{Q}, Φ is golden, for almost all x and for large enough $s \geq t_x$, we can see that for every $X \in \mathcal{Q} \cap \text{kuc}(\mathcal{Q}, \alpha)$ we have $\Phi(X) \supseteq \tau = A \upharpoonright_u$, where u is the use of $\Gamma^A(x)$. Thus $\theta(x) \in V_x$ for almost all x in the domain of θ . Mending the sequence on finitely many inputs yields a trace as required. \square

Thus, the rest of the proof is devoted to showing that a golden pair exists.

Proposition 6.13. *There is a Π_1^0 class $\mathcal{Q} \subseteq \mathcal{S}$ and a Turing functional Φ such that \mathcal{Q}, Φ is a golden pair for Γ, h , and G .*

6.7. The procedures and the construction

The procedure calling structure is now



At each stage, for each level e and each string η such that $R^{e,\eta}$ is running, the strings α such that some $S_x^{\eta\alpha}$ is running and has not returned form a prefix-free set.

Procedure $R^{e,\eta}$. This procedure runs with input $\mathcal{P}^{e,\eta}$ (again, a Π_1^0 class of the form $\mathcal{P}^{(v)}$ for some $v \in V$), and a parameter n . During its run, for every string α of length greater than n and every x such that $|\alpha| = h(x)$, the pair (α, x) is marked either *fresh* or *confirmed*. Initially, all such pairs are fresh. A string α such that $|\alpha| > n$ requires attention at stage s if there is some x such that $|\alpha| = h(x)$ and such that:

- no procedure $S_y^{e,\eta\beta}$ is currently running for any $y < x$ and $\beta \subseteq \alpha$;
- $\Gamma^A(x) \downarrow [s]$; and
- (α, x) is currently fresh.

For any string α that requires attention at stage s and is minimal among such strings under the prefix relation (that is, no proper initial segment of α also requires attention), we choose x to be the least number that witnesses that α requires attention, and call a run of the procedure $S_x^{e,\eta\alpha}$ with input $A_s \upharpoonright_u$, where u is the use of the computation $\Gamma^A(x)[s]$. We also cancel any run of any procedure $S_y^{e,\eta\beta}$ where $y > x$ and $\alpha \subseteq \beta$. This concludes the instructions for $R^{e,\eta}$.

For $\mathcal{P} = \mathcal{P}^{(v)}$ and a string α let

$$\mathcal{P}(\alpha) = \mathcal{P} \cap [\text{kuc}(\mathcal{P}, \alpha)].$$

We also let $\mathcal{P}_s(\alpha) = \mathcal{P}_s \cap [\text{kuc}(\mathcal{P}_s, \alpha)]$ at stage s .

Procedure $S_x^{e,\eta\alpha}$. This procedure is provided with a string $\tau \subset A_s$ such that $\Gamma_s(\tau, x) \downarrow$, and acts as follows.

(a) Start a run of $R^{e+1,\gamma}$, where $\gamma = \eta\alpha$, with input

$$\mathcal{P}^{e+1,\gamma} = \{X \in \mathcal{P}^{e,\eta}(\alpha) : \Phi_e(X) \not\subseteq \tau\} \tag{3}$$

and parameter $h(x)$. If this is the i th run of a procedure of this type $S_x^{e,\eta\alpha}$ ($i \geq 1$), enumerate $v = \langle e, x, i \rangle$ into V and let $\mathcal{P}^{(v)} = \mathcal{P}^{e+1,\gamma}$.

(b) If at some stage $\mathcal{P}^{e+1,\gamma}$ becomes empty, cancel the run of $R^{e+1,\gamma}$ and return, marking (α, x) as confirmed.

As before, a run of $S_x^{e,\eta\alpha}$ with input τ believes that $\tau \subset A$ and that the current version of $\mathcal{P}^{e,\eta}(\alpha)$ is correct. Thus, suppose that $S_x^{e,\eta\alpha}$ is called (with input τ) at stage s , and $t > s$ is the least stage at which either $\tau \not\subset A_t$ or $\text{kuc}(\mathcal{P}_t^{e,\eta}, \alpha) \neq \text{kuc}(\mathcal{P}_s^{e,\eta}, \alpha)$. If the same run of $S_x^{e,\eta\alpha}$ is still running at stage t , then it is immediately cancelled (together with all of its subprocedures); otherwise, (α, x) is re-marked as fresh.

This concludes the instructions for $S_x^{e,\eta\alpha}$.

The construction is started by calling $R^{0,\emptyset}$ with input $\mathcal{P}^{(0)} = \mathcal{S}$. (Recall that \mathcal{S} is a nonempty Π_1^0 class containing only 1-random sets, defined at the beginning of this proof.)

6.8. Verification

We show that there is an e and $\eta \subset G$ such that $\mathcal{P}^{e,\eta}, \Phi_e$ is a golden pair for Γ, h , and G for the final version of $\mathcal{P}^{e,\eta}$. First we do the necessary counting of how often procedures can be called. We begin with the analog of Claim 4.5. The situation is more complicated here because the number of cancellations of a run $S_x^{\eta\alpha}$ depends on the length of the coding string $\text{kuc}(\mathcal{P}^{e,\eta}, \alpha)$, and hence on the lower bound on the measure of the Π_1^0 class this runs works in. Recall the computable function q from Lemma 6.6.

Claim 6.14. *There is a computable function $B(x, r)$ such that a run $R^{e,\eta}$, with input $\mathcal{P}^{e,\eta} = \mathcal{P}^{(v)}$, calls at most $B(x, q(v))$ many runs of any $S_x^{e,\eta\alpha}$.*

Proof. Fix $x < \omega$ and a string α of length $h(x)$. Any call of $S_x^{e,\eta\alpha}$ by the run of $R^{e,\eta}$ beyond the first one is done because a previous run was cancelled, or because a later change caused (α, x) to be re-marked as fresh. This situation can have one of three causes:

- (i) the previous run of $S_x^{e,\eta\alpha}$ had input τ , and later we saw that $\tau \not\subset A_t$;
- (ii) $\text{kuc}(\mathcal{P}_s^{e,\eta}, \alpha)$ has changed;
- (iii) some run $S_y^{e,\eta\beta}$ was started, where $y < x$ and $\beta \subseteq \alpha$.

The number of times (i) occurs is bounded by $g(x)$; recall that g witnesses that $\langle A_s, \Gamma_s \rangle$ is a restrained approximation. The number of times (ii) can occur is bounded by $2^{\ell(|\alpha|, q(v))}$, by Lemma 6.5. Let $\tilde{B}(0, r) = 1 + g(0) + 2^{\ell(h(0), r)}$, and for $x > 0$, let $\tilde{B}(x, r) = 1 + g(x) + 2^{\ell(h(x), r)} + \sum_{y < x} \tilde{B}(y, r)$. Then $\tilde{B}(x, q(v))$ bounds the number of calls of $S_x^{e,\eta\alpha}$ by the single run of $R^{e,\eta}$. Summing over all strings α of length $h(x)$, we see that $B(x) = 2^{h(x)} \tilde{B}(x, r)$ is a bound as required. \square

We proceed to a fact similar to Claim 4.6.

Claim 6.15. (1) *There is a computable bound $M(e, x)$ on the number of calls of any procedure of the form $S_x^{e,\gamma}$.*

(2) *There is a computable bound $r(e, x)$ on the V -index of any class of the form $\mathcal{P}^{e+1,\gamma}$ that is called by a procedure $S_x^{e,\gamma}$. Hence, with the aid of the function q , we get a computable lower bound on the measure of all such classes.*

Proof. Both parts are computed simultaneously by recursion on e .

(2) for e, x follows from (1) for the same pair e, x . The i th call of any $S_x^{e,\gamma}$ provides its run $R^{e+1,\gamma}$ with input $\mathcal{P}^{e+1,\gamma} = \mathcal{P}^{(v)}$, where $v = \langle e, x, i \rangle$. Thus we can let $r(e, x) = \max_{i \leq M(e,x)} \langle e, x, i \rangle$.

For $e = 0$, there is a single run of $R^{0,\emptyset}$ that is never cancelled, with input $\mathcal{P}^{0,\emptyset} = \mathcal{P}^{(0)}$. So $M(0, x) = B(x, q(0))$ is an upper bound as desired.

Now assume that $e > 0$ and that (1) and (2) have been computed for all pairs (e', x') that lexicographically precede the pair (e, x) . By Claim 6.14, we may let $M(e, x)$ be the product of

- (i) a bound on the number of runs of $R^{e,\eta}$ that are called by some $S_y^{e-1,\eta}$ with parameter $h(y) < h(x)$, and
- (ii) a bound on the number of times a single run $R^{e,\eta}$ can call $S_x^{e,\gamma}$.

Both bounds are obtained by the fact that every run of some $R^{e,\eta}$ that calls some $S_x^{e,\gamma}$ is in turn called by a run of $S_y^{e,\eta}$ for some y such that $h(y) < h(x)$, as $R^{e,\eta}$'s parameter is $h(y)$. Since h is monotone, $y < x$.

Hence a bound (i) for the number of such runs $R^{e,\eta}$ is given by $\sum_{y < x} M(e - 1, y)$, and by Claim 6.14, a bound for (ii) is $\max_{y < x} \max_{v < r(e-1,y)} B(x, q(v))$. \square

We say that a run of $R^{e,\eta}$ is a *golden run* if $\eta \subset G$, the run is never cancelled, and every subprocedure $S_x^{e,\eta\alpha}$ with $\eta\alpha \subset G$ that is called by that run eventually returns or is cancelled.

Claim 6.16. *If there is a golden run of $R^{e,\eta}$ with input (the final version of) $\mathcal{P}^{e,\eta}$, then $\mathcal{P}^{e,\eta}, \Phi_e$ is a golden pair for Γ, h , and G , in the sense of Definition 6.11, with parameter η .*

Proof. Let n be the parameter of this golden run. For almost all x we have $h(x) > n$. To show the golden pair condition for such an x , suppose that $\Gamma^A(x)$ converges. Suppose $\eta\alpha \subseteq G$ where $|\alpha| = h(x)$. Choose s_0 so large that $\Gamma^A(x)$ and $\text{kuc}(\mathcal{P}^{e,\eta}, \alpha)$ are stable from stage s_0 on, and (by Claim 6.15 and the hypothesis that the run of $R^{e,\eta}$ is golden) all runs $S_y^{e,\eta\beta}$ for $y < x$ have returned or are cancelled. Following the instructions for $R^{e,\eta}$, we may now start a run $S_x^{e,\eta\alpha}$ (even if some $S_y^{e,\eta\gamma}$ where $y > x$ and $\alpha \subseteq \gamma$ is running and must be cancelled), and this run is not cancelled. Since this final run of $S_x^{e,\eta\alpha}$ returns, the Π_1^0 class in (3) becomes empty. Hence the golden pair condition for x holds of $\mathcal{P}^{e,\eta}, \Phi_e$. \square

Claim 6.17. *Suppose that there is no golden run. Then for every e , there is some $\gamma \subset G$ such that there is a run of $R^{e,\gamma}$ that is never cancelled.*

Proof. By induction on e . For $e = 0$ we have $\gamma = \emptyset$. Assume the claim holds for e , witnessed by some $\gamma \subset G$. Since the run of $R^{e,\gamma}$ that is never cancelled is not golden, there is some α such that $\eta\alpha \subset G$ and such that there is a call of $S_x^{e,\eta\alpha}$ for some x that is never cancelled but never returns. Then $S_x^{e,\eta\alpha}$ calls a run of $R^{e+1,\eta\alpha}$ that is never cancelled. \square

It remains to show that there is a golden run. For this we use the hypothesis that A is computable from every 1-random set Z such that $G \leq_{tt} Z'$. We define the coding strings σ_γ for $\gamma \in 2^{<\omega}$. Let $\sigma_{\emptyset,s} = \emptyset$.

- (1) If $\sigma_{\eta,s}$ has been defined and procedure $R^{e,\eta}$ is running at stage s with input \mathcal{P} , then for all $\alpha \neq \emptyset$ such that no procedure $S^{e,\eta\beta}$ is running for any $\beta \subsetneq \alpha$, let $\sigma_{\eta\alpha,s} = \text{kuc}(\mathcal{P}, \alpha)$.
- (2) If α is maximal under the prefix relation so that $\sigma_{\eta\alpha,s}$ is now defined, it must be the case that $S_x^{e,\eta\alpha}$ is currently running (for some x) and has called a run of $R^{e+1,\eta\alpha}$. This situation puts us back in case (1) with η replaced by $\eta\alpha$, and the recursive definition can continue.

We verify that $\gamma \subset \delta$ implies $\sigma_{\gamma,s} \subset \sigma_{\delta,s}$ for each s and $|\delta| \leq s$. The fact to verify is that if some $R^{e,\eta}$ is running at stage s with input $\mathcal{P}^{e,\eta}$, and $S_x^{e,\eta\alpha}$ is also running at this stage and provides to $R^{e+1,\eta\alpha}$ the input $\mathcal{P}^{e+1,\eta\alpha}$, then $\text{kuc}(\mathcal{P}_s^{e,\eta}, \alpha) \subseteq \text{kuc}(\mathcal{P}_s^{e+1,\eta\alpha}, \emptyset)$. The reason this fact holds is that we define $\mathcal{P}^{e+1,\eta\alpha}$ as a subclass of $\mathcal{P}^{e,\eta} \cap [\rho]$, where $\rho = \text{kuc}(\mathcal{P}_s^{e,\eta}, \alpha)$. We may of course assume that this containment holds also for the clopen approximations to these classes at stage s , that is, that $\mathcal{P}_s^{e+1,\eta\alpha} \subseteq \mathcal{P}_s^{e,\eta} \cap [\rho]$. Now the desired extension $\rho \subseteq \text{kuc}(\mathcal{P}_s^{e+1,\eta\alpha}, \emptyset)$ holds because of measure considerations. Let v be the V -index of $\mathcal{P}^{e+1,\eta\alpha}$. Then by definition, $\text{kuc}(\mathcal{P}^{e+1,\eta\alpha}, \emptyset)$ has length $q(v)$. From $\mathcal{P}^{e+1,\eta\alpha} \subset [\rho]$ we conclude that $\lambda \mathcal{P}^{e+1,\eta\alpha} \leq 2^{-|\sigma|}$. But we also know that $\lambda \mathcal{P}^{e+1,\eta\alpha} \geq 2^{-q(v)}$. Hence $q(v) \geq |\sigma|$. Since $\mathcal{P}_s^{e+1,\eta\alpha} \subset [\rho]$, we conclude that indeed $\rho \subseteq \text{kuc}(\mathcal{P}_s^{e+1,\eta\alpha}, \emptyset)$.

Claim 6.18. *For every γ , the number of stages s such that $\sigma_{\gamma,s+1} \neq \sigma_{\gamma,s}$ is finite, and in fact is computably bounded in γ (and hence in $|\gamma|$).*

Proof. We can have $\sigma_{\gamma,s+1} \neq \sigma_{\gamma,s}$ for two reasons.

- (i) A run of $S_x^{e,\eta}$ is called for some $\eta \subseteq \gamma$.
- (ii) A string $\text{kuc}(\mathcal{P}, \alpha)$ involved in the definition of $\sigma_{\gamma,s}$ changes from stage s to stage $s + 1$.

For any run of $S_x^{e,\eta}$ for $\eta \subseteq \gamma$ we must have $e, h(x) \leq |\gamma|$, so a computable bound on the number of changes of type (i) is given by Claim 6.15(1).

If a string $\text{kuc}(\mathcal{P}, \alpha)$ is involved in the definition of $\sigma_{\gamma,s}$, then we must have $\mathcal{P} = \mathcal{P}^{e,\eta}$ for some $e \leq |\gamma|$ and $\eta\alpha \subseteq \gamma$. Assuming that $e > 0$, this run of $R^{e,\eta}$ was called by some $S_x^{e,\eta}$ where again $h(x) \leq |\gamma|$. By Claim 6.15(2), $\mathcal{P} = \mathcal{P}^{(v)}$, where $v \leq r(e - 1, y)$ for some $y < x$. Thus, effectively in γ , we get a bound on $q(v)$ for the V -index v of \mathcal{P} , and so with the aid of Lemma 6.5 (and again Claim 6.15(1)), a bound on the number of times a change as in (ii) may happen. \square

For all γ , let $\sigma_\gamma = \lim_s \sigma_{\gamma,s}$. Let $Z = \bigcup_{\gamma < G} \sigma_\gamma$. By a proof identical to the proof of Claim 6.10, $G \leq_{tt} Z'$. By the assumption on A , we have $A \leq_T Z$. Hence $\Phi_e(Z) = A$ for some e .

Assume for a contradiction that there is no golden run. By Claim 6.17, there is a run of $R^{e,\eta}$, for some $\eta \subset G$, that is never cancelled. Since this run is not golden, there is some α such that $\eta\alpha \subset G$ and such that there is a run of $S_x^{e,\eta\alpha}$ for some x , called by $R^{e,\eta}$, that is never cancelled and never returns. This run defines

$$\mathcal{P}^{e+1,\eta\alpha} = \{X \in \mathcal{P}^{e,\eta} \langle \alpha \rangle : \Phi_e(X) \not\geq \tau\},$$

where $\mathcal{P}^{e,\eta}$ is the input of $R^{e,\eta}$. These classes are never altered, as $S_x^{e,\eta\alpha}$ is never cancelled, so we have $\tau \subset A$. Since the run of $S_x^{e,\eta\alpha}$ never return, the class $\mathcal{P}^{e+1,\eta\alpha}$ is nonempty.

Claim 6.19. $Z \in \mathcal{P}^{e+1,\eta\alpha}$.

Proof. Since $S_x^{e,\eta\alpha}$ is never cancelled, after its inception, no run $S_y^{e,\eta\beta}$ for any $\beta \subsetneq \alpha$ is ever called by $R^{e,\eta}$. Hence $\sigma_{\eta\alpha} = \text{kuc}(\mathcal{P}^{e,\eta}, \alpha)$.

Let $e' > e + 1$. By Claim 6.17, there is some string $\gamma \subset G$ for which there is a run of $R^{e',\gamma}$ that is never cancelled. We have $|\gamma| \geq e'$. Then σ_γ is extendible in $\mathcal{P}^{e',\gamma}$ and $\mathcal{P}^{e',\gamma} \subseteq \mathcal{P}^{e+1,\eta\alpha}$. The result follows by compactness. \square

We now get the desired contradiction, since $\Phi_e(X) \neq A$ for all $X \in \mathcal{P}^{e+1,\eta\alpha}$. This completes the proof of Proposition 6.13 and hence of Theorem 6.1.

7. Demuth random sets and cost functions

Consider the situation that $A \leq_T Y$ where A is c.e. and Y is a 1-random Δ_2^0 set. In this section, we develop the connection between the strength of cost functions A can obey and the degree of randomness of Y . This analysis will yield a proof of Theorem 2.6.

We gauge the degree of randomness of Y via the notion of Demuth randomness and its variants. Recall from the first section the idea behind Demuth randomness. Tests are generalized in that one can change the m -th component for a computably bounded number of times. We will introduce stronger randomness notions, but still compatible with being Δ_2^0 , by relaxing the condition that the number of changes be computably bounded. Instead, each time there is a change to the current version of the m -th component, we count down along a computable well-ordering R .

7.1. R -approximations

We begin with some intuitive background. Ershov [12] introduced a theory of a -c.e. functions for an ordinal notation a (see [41] for a recent survey). The following simpler variant suffices for our purposes. We replace the ordinal notations by arbitrary computable well-orders. Let $g(n, s)$ be a computable approximation to a function f . Suppose that the number of mind changes at n is bounded by $h(n)$ where h is a computable function. We can think of the situation as follows: an “approximator” promises to give us a computable approximation to f . The approximator also has to give us evidence that the approximation will indeed stabilize on every input. Thus, for every n , at stage 0 the approximator puts a marker labelled n on the number $h(n)$ in the standard ordering $<$ of the natural numbers. Each time the approximator wants to change its approximation to $f(n)$, that is, at a stage s such that $g(n, s) \neq g(n, s - 1)$, the approximator needs to move the n th marker at least one number to the left, that is, decrease its value in the ordering $<$. Since this ordering is a well-ordering of the natural numbers, this process ensures the stabilization of the approximation $g(n, s)$. The effectiveness of the entire setup is also due to the fact that $<$ is a computable well-ordering, and that the moves of the markers are given effectively. The notion of an R -approximation is identical, except that we replace $<$ by some other computable well-ordering of the natural numbers.

For the rest of the section, we assume that every computable well-ordering R we mention is infinite; indeed we assume that its field is ω .

Definition 7.1. Let $R = (\omega, <_R)$ be a computable well-ordering. An R -approximation is a computable function

$$g = \langle g_0, g_1 \rangle: \omega \times \omega \rightarrow \omega \times \omega$$

such that for each x and each $s > 0$,

$$g(x, s) \neq g(x, s - 1) \rightarrow g_1(x, s) <_R g_1(x, s - 1).$$

In this case, g_0 is a computable approximation to a total Δ_2^0 function f . We say that g is an R -approximation to f . A Δ_2^0 function f is called R -c.e. if it has an R -approximation.

Lemma 7.2. *Let R be a computable well-ordering. Every ω -c.e. function is R -c.e.*

Sketch of proof. If S and R are computable well-orderings, and S is computably embeddable into R , then every S -c.e. set is R -c.e., because the effective embedding of S into R can be used to translate any S -approximation to an R -approximation.

Now, since we assume that R is infinite, we can effectively embed $(\omega, <)$ into R by recursively choosing bigger and bigger elements in the sense of $<_R$. If the order-type of R is not a limit ordinal, then we first fix a limit point of R , and then build our embedding entirely below that limit point. \square

The following lemma is related to Ershov’s result that each Δ_2^0 function is a -c.e. for some notation a of ω^2 (see [41, Theorem 4.3]). In our simpler setting, a well-ordering of type ω suffices.

Lemma 7.3. *For each computable approximation $g_0: \omega \times \omega \rightarrow \omega$ to a Δ_2^0 function f , there is a computable well-ordering R of order type ω and a computable function $g_1: \omega \times \omega \rightarrow \omega$ such that $\langle g_0, g_1 \rangle$ is an R -approximation to f .*

Proof. It suffices to define a computable well-ordering R with an infinite computable field $V \subseteq \omega \times \omega$. Let

$$V = \omega \times \{0\} \cup \{\langle x, s \rangle : s > 0 \ \& \ g_0(x, s) \neq g_0(x, s - 1)\}.$$

For $\langle x, s \rangle, \langle y, t \rangle \in V$, we declare that $\langle x, s \rangle <_R \langle y, t \rangle$ if $x < y$, or $x = y$ and $s > t$. Then R is of order type ω because g_0 is a computable approximation. Let $g_1(x, u) = \langle x, s \rangle$ where $s \leq u$ is largest such that $\langle x, s \rangle \in V$. Then $\langle g_0, g_1 \rangle$ is an R -approximation to f . \square

Later on we will need the following fact.

Lemma 7.4. *For every computable well-ordering R , there is a uniformly \emptyset' -computable listing $\langle f^e \rangle$ of all R -c.e. functions.*

Proof. Define a *partial R -approximation* to be a partial computable function $\psi = \langle \psi_0, \psi_1 \rangle : \omega^2 \rightarrow \omega^2$ such that $\text{dom } \psi$ is closed downward in both variables, and such that for all n and $s > 0$, if $(n, s) \in \text{dom } \psi$ and $\psi(n, s) \neq \psi(n, s - 1)$ then $\psi_1(n, s) <_R \psi_1(n, s - 1)$. There is an effective listing $\langle \psi^e \rangle$ of all partial R -approximations.

Write $\psi^e(n, t) \downarrow [s]$ to denote that $(n, t) \in \text{dom } \psi^e$ and that this fact is discovered after s steps of computation of some universal machine. We may assume that $\text{dom } \psi^e [s]$ is closed downward in both variables. Given e, n , and s , let t be greatest such that $\psi^e(n, t) \downarrow [s]$, and let $g^e(n, s) = \psi_0(n, t)$. If there is no such t , then let $g^e(n, s) = 0$.

Now $\langle g^e \rangle$ is a uniformly computable sequence of functions, and the function f^e defined by letting $f^e(n) = \lim_s g^e(n, s)$ is total for all e , so $\langle f^e \rangle$ is uniformly \emptyset' -computable. If ψ^e is not total, then $\text{dom } \psi^e$ is finite, whence $f^e(n) = 0$ for almost all n . If ψ^e is total then $f^e(n) = \lim_s \psi_0(n, s)$ for all n . \square

7.2. *R*-Demuth random sets

Definition 7.5. Let $R = (\omega, <_R)$ be a computable well-ordering. An *R*-Demuth test is a sequence of c.e. open sets $(\mathcal{L}_m)_{m \in \omega}$ such that $\forall m \lambda \mathcal{L}_m \leq 2^{-m}$, and there is a function f with an *R*-approximation such that $\mathcal{L}_m = [W_{f(m)}]^\prec$. A set Z passes the test if $Z \notin \mathcal{L}_m$ for almost all m . We say that Z is *R*-Demuth random if Z passes each *R*-Demuth test.

Thus, a set is Demuth random if and only if it is $(\omega, <)$ -Demuth random. By Lemma 7.2, for every computable well-ordering R , every *R*-Demuth random set is Demuth random.

Proposition 7.6. For every computable well-ordering R , there is a Δ_2^0 set Y that is *R*-Demuth random.

The proof of Proposition 7.6 is a variant of the construction of a Δ_2^0 Demuth random set; see [34, Theorem 3.6.25]. The proof is divided into two parts. There is no universal *R*-Demuth test, but nevertheless, we first show that there is a “special” test (\mathcal{G}_n) such that every set passing this test is *R*-Demuth random. Then we show that there is a Δ_2^0 set that passes this special test. In the following we write \mathcal{H}_e for $[W_e]^\prec$.

Definition 7.7. A special test is a sequence of c.e. open sets $(\mathcal{G}_m)_{m \in \omega}$ such that $\lambda \mathcal{G}_m \leq 2^{-2m-1}$ for each m , and there is a function $g \leq_T \emptyset'$ such that $\mathcal{G}_m = \mathcal{H}_{g(m)}$. Z passes the test if $Z \notin \mathcal{G}_m$ for almost all m .

Lemma 7.8. Let $R = (\omega, <_R)$ be a computable well-ordering. There is a special test (\mathcal{G}_n) such that every set that passes (\mathcal{G}_n) is *R*-Demuth random.

Proof. A set Z is *R*-Demuth random iff for each *R*-Demuth test $(\mathcal{U}_m)_{m \in \omega}$, Z passes the *R*-Demuth tests $(\mathcal{U}_{2m})_{m \in \omega}$ and $(\mathcal{U}_{2m+1})_{m \in \omega}$. Thus it suffices to build a special test $(\mathcal{G}_m)_{m \in \omega}$ that emulates all *R*-Demuth tests $(\mathcal{S}_n)_{n \in \omega}$ such that $\lambda \mathcal{S}_n \leq 2^{-2n}$ for each n . The idea is now to put together all *R*-Demuth tests of this kind. This construction does not result in a universal *R*-Demuth test because the enumeration of all *R*-Demuth tests cannot be done effectively. However, it can be done effectively relative to a \emptyset' oracle, thus yielding a special test.

The following definition will also be useful later. Given an effectively open class $\mathcal{W} = [W]^\prec$ (for some c.e. set W) and a positive rational number ε , we can (uniformly in a c.e. index for W and ε) obtain a c.e. index for an effectively open class, which we denote by $\mathcal{W}^{(\leq \varepsilon)}$, such that

- (a) $\mathcal{W}^{(\leq \varepsilon)} \subseteq \mathcal{W}$;
- (b) $\lambda \mathcal{W}^{(\leq \varepsilon)} \leq \varepsilon$;
- (c) if $\lambda \mathcal{W} \leq \varepsilon$, then $\mathcal{W}^{(\leq \varepsilon)} = \mathcal{W}$.

The idea is simply to copy W , but prevent the enumeration of any string into $\mathcal{W}^{(\leq \varepsilon)}$ that would make the measure go beyond ε .

By Lemma 7.4, there is a uniformly \emptyset' -computable listing (f^e) of the *R*-c.e. functions. Thus, there is a function $\tilde{q} \leq_T \emptyset'$ such that $\tilde{q}(e, n) = f^e(n)$ for each e, n . Now let $q \leq_T \emptyset'$ be a function such that

$$\mathcal{H}_{q(e,n)} = \mathcal{H}_{\tilde{q}(e,n)}^{(\leq 2^{-2n})}$$

and let

$$\mathcal{G}_m = \bigcup_{e < m} \mathcal{H}_{q(e,e+m+1)}.$$

Then $\lambda \mathcal{G}_m \leq \sum_{e < m} 2^{-2(e+m+1)} \leq 2^{-2m-1}$. If Z passes the special test $\langle \mathcal{G}_n \rangle$ then it passes each R -Demuth test. \square

The proof of Proposition 7.6 is completed with the following lemma.

Lemma 7.9. *If $\langle \mathcal{G}_n \rangle$ is a special test, then some Δ_2^0 set Z passes $\langle \mathcal{G}_n \rangle$.*

Proof. This lemma is Claim 2 of the proof of [34, Theorem 3.6.25]. We give a sketch for completeness. For $n < \omega$, let $\mathcal{L}_n = \bigcup_{m \leq n} \mathcal{G}_m$.

Recall (from Section 6) that for a class \mathcal{W} and a string τ , we let $\mathcal{W} \upharpoonright \tau = \{X : \tau X \in \mathcal{W}\}$. If \mathcal{W} is measurable then $\lambda(\mathcal{W} \upharpoonright \tau) = 2^{|\tau|} \lambda(\mathcal{W} \cap [\tau])$. Note that $\lambda(\mathcal{W} \upharpoonright \tau)$ is the average of $\lambda(\mathcal{W} \upharpoonright \tau 0)$ and $\lambda(\mathcal{W} \upharpoonright \tau 1)$.

With oracle \emptyset' , we recursively build a set Z such that

$$\lambda(\mathcal{L}_n \upharpoonright Z \upharpoonright_n) \leq 1 - 2^{-n-1}$$

for all n . This inequality holds for $n = 0$ since $\mathcal{L}_0 = \mathcal{G}_0$ and $\lambda \mathcal{G}_0 \leq 1/2$. If $Z \upharpoonright_n$ has been defined and the inequality holds, then \emptyset' can determine a 1-bit extension $Z \upharpoonright_{n+1}$ of $Z \upharpoonright_n$ such that $\lambda(\mathcal{L}_n \upharpoonright Z \upharpoonright_{n+1}) \leq 1 - 2^{-n-1}$.

Since $\mathcal{L}_{n+1} = \mathcal{L}_n \cup \mathcal{G}_{n+1}$ and $\lambda(\mathcal{G}_{n+1}) \leq 2^{-2n-3}$, we have $\lambda(\mathcal{G}_{n+1} \upharpoonright Z \upharpoonright_{n+1}) \leq 2^{n+1} 2^{-2n-3} = 2^{-n-2}$. Thus $\lambda(\mathcal{L}_{n+1} \upharpoonright Z \upharpoonright_{n+1}) \leq 1 - 2^{-n-2}$.

To finish we show that $Z \notin \mathcal{G}_n$ for all n . If $Z \in \mathcal{G}_n$ then $Z \in \mathcal{L}_n$, so there is some $m \geq n$ such that $[Z \upharpoonright_m] \subseteq \mathcal{L}_n$ since \mathcal{L}_n is open; but $\mathcal{L}_n \subseteq \mathcal{L}_m$, and $[Z \upharpoonright_m] \not\subseteq \mathcal{L}_m$. \square

Let R^ω be the computable well-ordering of type $|R|^\omega$ obtained from R in the canonical way. Suppose $\langle \mathcal{G}_n \rangle$ is the special test obtained in Lemma 7.8. Analyzing the proofs of the foregoing lemmas shows that the R -Demuth random set Z we build is R^ω -c.e.

7.3. R -benignity

Given a monotone cost function c , we define a computable function $q, s \mapsto w_c(q, s)$ where q is a non-negative rational and $s \in \omega$. Let $w_c(q, 0) = 0$. For $s > 0$ let

$$w_c(q, s) = \begin{cases} s & \text{if } c(w_c(q, s-1), s) \geq q \\ w_c(q, s-1) & \text{otherwise.} \end{cases} \tag{4}$$

We now count the number of times $w_c(q, s)$ changes. Clearly, c satisfies the limit condition $\lim_x \sup_s c(x, s) = 0$ if and only if $w_c(q, s)$ changes only finitely often for each q , and c is benign in the sense of Definition 2.5 if and only if this number of changes is in fact computably bounded in q . We use R -approximations to get a hold on cost functions that satisfy the limit condition but are not necessarily benign.

Definition 7.10. Let R be a computable well-ordering. A monotone cost function c is R -benign if there is a computable function $f: \omega^2 \rightarrow \omega$ such that $\langle w_c, f \rangle$ is an R -approximation.

That is, we not only require that the function $q \mapsto \lim_s w_c(q, s)$ be R -c.e., we actually require that the canonical approximation $w_c(q, s)$ be the first component of an R -approximation. By the monotonicity of c , it is sufficient to replace all rational numbers q by a computable sequence of rational numbers $\langle q_n \rangle$ descending to 0. Clearly, a cost function is benign if and only if it is $(\omega, <)$ -benign. From Lemma 7.3 we have that a monotone cost function c satisfies the limit condition if and only if there is some computable well-ordering R such that c is R -benign.

7.4. *Main result of this section*

Recall that the product $A \cdot B$ of two linear orderings A and B is the linear ordering obtained by replacing every point in B by a copy of A . In other words, it is the right-lexicographic ordering on $A \times B$.

Theorem 7.11. *Let R be a computable well-ordering, and let c be an R -benign cost function. Let Y be any $\omega \cdot R$ -Demuth random set. Then every c.e. set A that is computable from Y obeys c .*

Before proving [Theorem 7.11](#), we show how it implies [Theorem 2.6](#).

Proof of Theorem 2.6. Let c be a monotone cost function that satisfies the limit condition. By [Lemma 7.3](#), there is some computable well-ordering R such that c is R -benign. By [Proposition 7.6](#), there is some Δ_2^0 set Y that is $\omega \cdot R$ -Demuth random. By [Theorem 7.11](#), every c.e. set computable from Y obeys c . \square

Proof of Theorem 7.11. We define the numbers $w_c(q, s)$ by [\(4\)](#) and use the shorthand

$$v(m, s) = w_c(2^{-2m}, s).$$

Note that $v(m) = \lim_s v(m, s)$ exists for each m by the hypothesis on c . Furthermore, we may assume that the function $\lambda m \cdot v(m)$ is unbounded; otherwise, every c.e. set obeys c .

Let Φ be a Turing functional such that $A = \Phi^Y$ for some $\omega \cdot R$ -Demuth random set Y . To ensure that A obeys c we want to restrict the changes of $A \upharpoonright_{v(m,s)}$. To do so we define an $\omega \cdot R$ -Demuth test based on the following: let $\mathcal{L}_m[s]$ go through all the c.e. open sets

$$\{Z: A_s \upharpoonright_{v(m,s)} \subseteq \Phi^Z\}.$$

That is, as long as both $v(m, s)$ and $A_s \upharpoonright_{v(m,s)}$ remain unchanged we have the same version of \mathcal{L}_m , but when one of them changes, so does \mathcal{L}_m . The idea is that the measure of the final version of \mathcal{L}_m has to exceed 2^{-m} for almost all m , otherwise Y would fail the $\omega \cdot R$ -Demuth test obtained by stopping the enumeration of \mathcal{L}_m when its measure attempts to exceed that bound. To see that $\langle \mathcal{L}_m \rangle$ indeed has an $\omega \cdot R$ -computable approximation, we observe that every $v(m, s)$ -change is accompanied by a downward step in R , and that between such changes, i.e. while $v(m, s)$ is constant, there can be only $v(m, s)$ many changes to $A_s \upharpoonright_{v(m,s)}$, as A is c.e.

For the formal details, fix a computable function h_0 such that $[W_{h_0(m,s)}]^\prec = \mathcal{L}_m[s]$ for each s .

Claim 1. *One can extend h_0 to an $\omega \cdot R$ -approximation $\langle h_0, h_1 \rangle$.*

By the hypothesis that c is R -benign, there is a computable function $f: \omega \times \omega \rightarrow \omega$ such that $\langle v, f \rangle$ is an R -approximation. The idea is now to follow this R -approximation if $v(m, s)$ changes, and use the first components of the pairs in $\omega \cdot R$ to count the changes of $A \upharpoonright_{v(m,s)}$ while $v(m, s)$ is stable. For the formal details, we define the computable function h_1 by

$$h_1(m, s) = \langle l, f(m, s) \rangle,$$

where the counter $l \in \omega$ is initialized at $v(m, s)$ when $v(m, s) \neq v(m, s - 1)$. Subsequently, each time $A \upharpoonright_{v(m,s)}$ changes while $v(m, s)$ remains at this value, we decrease this counter. This completes the proof of [Claim 1](#).

Recall from the proof of [Proposition 7.6](#) that for a c.e. open set \mathcal{W} and a positive rational ε , we let $\mathcal{W}^{(\leq \varepsilon)}$ denote a uniformly obtained c.e. open set contained in \mathcal{W} that equals \mathcal{W} if the measure of \mathcal{W} does not exceed ε .

Let $\mathcal{H}_m[s] = (\mathcal{L}_m[s])^{(\leq 2^{-m})}$. Let \mathcal{H}_m denote the final version of $\mathcal{H}_m[s]$. Then, by Claim 1, $(\mathcal{H}_m)_{m \in \omega}$ is an $\omega \cdot R$ -Demuth test.

By the hypothesis on Y , there is m^* such that $Y \notin \mathcal{H}_m$ for each $m \geq m^*$. Then, since $A \upharpoonright_{v(m)} \subseteq \Phi^Y$, we have $\lambda \mathcal{L}_m > 2^{-m}$ for each $m \geq m^*$; otherwise Y would enter \mathcal{H}_m .

It remains to obtain a computable enumeration $(\widehat{A}_s)_{s \in \omega}$ of A obeying c . We first define an infinite computable sequence of stages: let $s_0 = m^*$, and

$$s_{i+1} = \mu s > s_i \forall m [m^* \leq m \leq s_i \rightarrow \lambda \mathcal{L}_{m,s}[s] > 2^{-m}].$$

Let

$$g(i) = \max\{v(m, s_i) : m^* \leq m \leq s_i\}.$$

Now, consider a stage s such that $s_i \leq s < s_{i+1}$. Given x , since the function $\lambda m \cdot v(m)$ is unbounded, there is a least positive $j \geq i$ such that $g(j - 1) > x$. Let

$$\widehat{A}_s(x) = A_{s_{j+2}}(x).$$

Claim 2. *The computable enumeration $(\widehat{A}_s)_{s \in \omega}$ obeys c .*

We have to show that the total cost of changes for this enumeration, as defined in (1) in Section 2.6, is finite. Suppose that at a stage s , the number x is least such that $\widehat{A}_{s-1}(x) \neq \widehat{A}_s(x)$. Then $s = s_j$ for some j such that $g(j - 1) > x$. So, we can choose a least $m_s = m < s_j$ such that $x < v(m, s)$. We may assume that $x > v(m^* + 1)$. Then $m > m^*$ and $v(m - 1, s) \leq x$. Recall that $v(m - 1, s) = w_c(2^{-2m+2})$. Then, by definition and the monotonicity of c we have $c(x, s) < 2^{-2m+2}$.

Since $g(j - 1) > x$, we have $A_{s_{j+1}}(x) \neq A_{s_{j+2}}(x)$, so all the versions $\mathcal{L}_m[s]$ for $s > s_{j+1}$ are disjoint from $\mathcal{L}_m[s_{j+1}]$. Then, since $\lambda \mathcal{L}_m[s_{j+1}] > 2^{-m}$, a situation as above for a particular value m can occur at most 2^m many times. (That is, there are at most 2^m stages s such that $m_s = m$.) Thus, the total cost of changes at numbers $x > v(m^* + 1)$ for this computable enumeration of A is bounded by $\sum_m 2^m 2^{-2m+2} = 8$. \square

When R is reasonably nice, we can replace $\omega \cdot R$ by R itself in Theorem 7.11, so getting a more pleasing correspondence between benignity of the cost function c and the Demuthness of the random set Y . Suppose that R is a computable well-ordering obtained from a notation for a limit ordinal. In fact, what we use is that: (1) the order-type of R is a limit ordinal; (2) the set of limit points of R is computable; (3) the successor function on R is computable. We explain how to modify the proof of Theorem 7.11 to see that if c is R -benign, Y is R -Demuth random and $A \leq_T Y$ is c.e., then A obeys c .

The sequence (\mathcal{L}_m) is defined precisely as in the proof of Theorem 7.11, and so the only change is to Claim 1; we need to show that $(\mathcal{L}_m[s])$ can be extended to an R -computable approximation, rather than merely an $\omega \cdot R$ -one. Recall that before, we only used the fact that while $v(m, s)$ is constant, $A_s \upharpoonright_{v(m,s)}$ can only change $v(m, s)$ many times. But the function taking i to the i th version of $v(m, s)$ is not computable. Nonetheless, we can bound the number of changes computably. That is, letting $u_i(m)$ be the i th version of $v(m, s)$, we construct a total computable function $p(m, i)$ such that for all m and i such that $u_i(m)$ is defined, $p(m, i)$ bounds the number of stages s at which $v(m, s) = u_i(m)$ and $A_{s+1} \upharpoonright_{u_i(m)} \neq A_s \upharpoonright_{u_i(m)}$.

To get the function p , define an A -partial computable function ψ by letting $\psi^A(m, i) \downarrow = A \upharpoonright_v$ if $v = u_i(m)$. So $\psi(m, i) \downarrow$ only if $u_i(m)$ is defined, that is, if there are at least i many versions of $u_i(m)$. Now A is computable from the Demuth random set Y , which is Turing incomplete

(see [34, Theorem 3.6.26]), and as mentioned above often, this means that A is K -trivial and so both superlow and jump-traceable. **Theorem 3.5** states that ψ has a restrained A -approximation $\langle \Gamma_s, A_s \rangle$. We can take the bound for this approximation to be the function p , as long as we require that if $\Gamma^A(m, i) \downarrow = \sigma[s]$ then $v(m, s) \geq u_i(m)$ and that $\sigma = A_s \upharpoonright_{u_i(m)}$. We use the enumeration $\langle A_s \rangle$ in the construction.

Now given p , we give an R -indexing h_1 for the approximation $\langle \mathcal{L}_m[s] \rangle$ as follows. Suppose that at a stage s we get a new value for $v(m, s)$, which is $u_i(m)$ (actually $s = v(m, s)$). Using the function f so that $\langle v, f \rangle$ is an R -approximation, we find a limit point γ in R and $n < \omega$ so that $f(m, s) = \gamma + n$ (addition in R). We then let

$$h_1(m, s) = \gamma + n + p(i, m) + p(i + 1, m) + \cdots + p(i + n, m),$$

again with addition in R . Then, as long as $v(m, s)$ does not change, we move down one step at a time whenever $A \upharpoonright_{u_i(m)}$ changes. The function p gives us enough room so that h_1 is as required.

Remark 7.12. Suppose that the computable well-ordering R is not of limit order-type, say of order type $\alpha + n$, where n is finite and α is limit, and let S be a computable well-ordering of type α obtained from R in a natural way. Let c be an R -benign monotone cost function, as witnessed by the R -approximation $\langle w_c, f \rangle$. Suppose that for each $\varepsilon > 0$, there are a non-negative rational $q < \varepsilon$ and an s such that $f(q, s)$ is in the “ n part” of R . Then it is not hard to check that c is in fact benign. Otherwise, we can adjust f to obtain an S -approximation. Thus, every R -benign monotone cost function is in fact S -benign.

So, if R is a computable well-ordering such that there are R -benign cost functions that are not S -benign for any computable proper initial segment S of R , then R has limit order-type.

In particular, the analysis above applies to $R = (\omega, <)$. It follows that each c.e. set A Turing below a Demuth random set obeys every benign cost function. As mentioned in the first section of the paper, Kučera and Nies [23] had previously obtained the equivalent result that such a set A is strongly jump-traceable.

Remark 7.13. By further adapting the techniques above to plain Demuth tests, we may obtain a new proof of the harder right-to-left direction of **Characterization 1a** for the case of a c.e. set A : if A is below each ω -c.e. 1-random set, then A obeys each benign cost function c (and hence A is strongly jump-traceable).

Firstly, given A and c , define the computable functions p and f as above. Let $r(m) = p(m, f(m, 0)) \cdot f(m, 0)$. The proof of **Lemma 7.8** shows that there is a Demuth test $\langle \mathcal{G}_n \rangle$ (taking the role of the special test there) such that each set passing $\langle \mathcal{G}_n \rangle$ passes each Demuth test $\langle \mathcal{H}_m \rangle$ with at most $r(m)$ many changes to versions of the m -th component. Secondly, since $\langle \mathcal{G}_n \rangle$ is now a Demuth test, by the proof of **Lemma 7.9** there is an ω -c.e. set Y passing it. Finally, by hypothesis, $A = \Phi(Y)$ for some Turing functional Φ . Define the Demuth test $\langle \mathcal{H}_m \rangle$ as in the proof of **Theorem 7.11**. Then the number of times a version $\mathcal{H}_m[s]$ changes is bounded by $r(m)$, so Y passes this Demuth test. Now the proof of **Theorem 7.11** shows that A obeys c .

7.5. Recent developments

Some developments relevant to the paper have occurred since its original submission in October 2010.

Superlowness and superhighness are Demuth test compatible. Nies [37] defined a class $\mathcal{C} \subseteq 2^\omega$ to be *Demuth test-compatible* if each Demuth test is passed by a member of \mathcal{C} . If \mathcal{C} is Demuth

test compatible, then so is $\mathcal{C} \cap \text{MLR}$. In [37, Theorem 4.4] he shows that superlow and superhighness are Demuth test compatible, using some of the methods from our proofs of Theorems 2.9 and 6.1, respectively. By the same argument as in Remark 7.13 above, this yields new proofs that a c.e. set that is Turing below each superlow [superhigh] ML-random set is strongly jump traceable.

Weak Demuth randomness. A set Z is called *weakly Demuth random* if it weakly passes each Demuth test (\mathcal{G}_n) , in the sense that $Z \not\subseteq \mathcal{G}_n$ for some n . Clearly, no ω -c.e. (and hence no superlow) set is weakly Demuth random. Kučera and Nies [23, Corollary 3.6] have shown that no superhigh set is weakly Demuth random by adapting the method in our proof of Theorem 5.1. Thus, the following simultaneously generalizes the two facts that a c.e. strongly jump traceable set is below all superlow, and all superhigh ML-random sets.

Proposition 7.14. *Every c.e. strongly jump traceable set A is below each ML-random set Y that is not weakly Demuth random.*

Sketch of proof. It is not hard to see that weak Demuth tests are equivalent in strength to Π_2^0 null classes of the form $\bigcap_x V_x$, where V_x is effectively open uniformly in x , $V_x \supseteq V_{x+1}$, and the cost function $c(x, s) = \lambda V_{x,s}$ is benign. Thus $Y \in \bigcap V_x$ for such a sequence $\langle V_x \rangle$. Since A obeys the corresponding benign cost function c by Greenberg and Nies [16], we have $A \leq_T Y$ by the proof of the Hirschfeldt-Miller theorem in [34]. \square

The restriction to c.e. sets. Diamondstone, Greenberg and Turetsky [7] have recently shown that every strongly jump-traceable set is computable from a c.e. one. This means that several results concerning c.e. strongly jump-traceable sets can be extended to all strongly jump-traceable sets. In particular, Characterizations Ia and Ib, and Proposition 7.14 now hold for all sets: a set A is strongly jump-traceable if and only if it is computable from all ω -c.e. random sets if and only if it is computable from all superlow random sets if and only if it is computable from all random sets which are not weakly Demuth random. As far as Characterization II is concerned, we now know that every strongly jump-traceable set is computable from every superhigh random set, but for the converse, we need to know that every set which is computable from every superhigh random set is superlow and jump-traceable; this is still open.

More on Demuth randomness. Finally, Greenberg and Turetsky [17] have shown the converse to the Kučera-Nies result from [23], namely that every strongly jump-traceable set is computable from a Demuth random set. Putting these results together, we obtain yet another characterization of strong jump-traceability inside the c.e. sets using randomness and Turing reducibility: a c.e. set is strongly jump-traceable if and only if it is computable from some Demuth random set.

Acknowledgments

The first and third authors were partially supported by the Marsden Fund of New Zealand. The second author was partially supported by the National Science Foundation of the United States.

References

- [1] George Barmpalias, Rodney Downey, Noam Greenberg, K -trivial degrees and the jump-traceability hierarchy, Proc. Amer. Math. Soc. 137 (6) (2009) 2099–2109.
- [2] Mark Bickford, Chris F. Mills, Lowness properties of r.e. sets, Manuscript, UW Madison, 1982.
- [3] Gregory Chaitin, A theory of program size formally identical to information theory, J. ACM 22 (1975) 329–340.

- [4] Peter Cholak, Rod Downey, Noam Greenberg, Strong jump-traceability I: the computably enumerable case, *Adv. Math.* 217 (5) (2008) 2045–2074.
- [5] Joshua A. Cole, Stephen G. Simpson, Mass problems and hyperarithmeticality, *J. Math. Log.* 7 (2) (2007) 125–143.
- [6] Osvald Demuth, Some classes of arithmetical real numbers, *Comment. Math. Univ. Carolin.* 23 (3) (1982) 453–465.
- [7] David Diamondstone, Noam Greenberg, Daniel D. Turetsky, Inherent enumerability of strong jump-traceability. (submitted for publication).
- [8] Rod Downey, Noam Greenberg, Strong jump-traceability II: K -triviality, in: *Israel J. Mathematics* (in press).
- [9] Rod Downey, Denis Hirschfeldt, *Algorithmic Randomness and Complexity*, Springer-Verlag, Berlin, 2010, 855 pages.
- [10] Rod Downey, Denis Hirschfeldt, André Nies, Frank Stephan, Trivial reals, in: *Proceedings of the 7th and 8th Asian Logic Conferences*, Singapore University Press, Singapore, 2003, pp. 103–131.
- [11] Rod Downey, Denis R. Hirschfeldt, André Nies, Sebastiaan A. Terwijn, Calibrating randomness, *Bull. Symbolic Logic* 12 (3) (2006) 411–491.
- [12] Yuri L. Ershov, A hierarchy of sets, I, *Algebra Logika* 7 (1) (1968) 47–74. English Translation, Consultants Bureau, NY, pp. 25–43.
- [13] Santiago Figueira, André Nies, Frank Stephan, Lowness properties and approximations of the jump, *Ann. Pure Appl. Logic* 152 (1–3) (2008) 51–66.
- [14] Peter Gács, Every set is reducible to a random one, *Inf. Control* 70 (1986) 186–192.
- [15] Noam Greenberg, A random set which only computes strongly jump-traceable c.e. sets, *J. Symbolic Logic* 76 (2) (2011) 700–718.
- [16] Noam Greenberg, André Nies, Benign cost functions and lowness properties, *J. Symbolic Logic* 76 (1) (2011) 289–312.
- [17] Noam Greenberg, Daniel D. Turetsky, Strong jump-traceability and Demuth randomness (in press).
- [18] Denis R. Hirschfeldt, André Nies, Frank Stephan, Using random sets as oracles, *J. Lond. Math. Soc.* (2) 75 (3) (2007) 610–622.
- [19] Carl G. Jockusch Jr., Degrees of functions with no fixed points, in: *Logic, Methodology and Philosophy of Science, VIII (Moscow, 1987)*, in: *Stud. Logic Found. Math.*, vol. 126, North-Holland, Amsterdam, 1989, pp. 191–201.
- [20] Carl G. Jockusch Jr., Robert I. Soare, Π_1^0 classes and degrees of theories, *Trans. Amer. Math. Soc.* 173 (1972) 33–56.
- [21] Bjørn Kjos-Hanssen, Wolfgang Merkle, Frank Stephan, Kolmogorov complexity and the recursion theorem, in: *Symposium on Theoretical Aspects of Computer Science 2006*, in: *LNCS*, vol. 3884, Springer, 2006, pp. 149–161.
- [22] Bjørn Kjos-Hanssen, André Nies, Superhighness, *Notre Dame J. Form. Log.* 50 (4) (2010) 445–452. 2009.
- [23] Antonin Kučera, André Nies, Demuth randomness and computational complexity, *Ann. Pure Appl. Logic* 162 (2011) 504–513.
- [24] Antonin Kučera, Measure, Π_1^0 classes, and complete extensions of PA, in: *Recursion Theory Week*, in: *Lecture Notes in Mathematics*, vol. 114, Springer, Oberwolfach, Berlin, 1984–1985, pp. 245–259.
- [25] Antonin Kučera, An alternative, priority-free, solution to Post’s problem, in: *Mathematical Foundations of Computer Science, 1986 (Bratislava, 1986)*, in: *Lecture Notes in Comput. Sci.*, vol. 233, Springer, Berlin, 1986, pp. 493–500.
- [26] Antonin Kučera, Sebastiaan A. Terwijn, Lowness for the class of random sets, *J. Symbolic Logic* 64 (1999) 1396–1402.
- [27] Joseph S. Miller, André Nies, Randomness and computability: open questions, *Bull. Symbolic Logic* 12 (3) (2006) 390–410.
- [28] Jeanleah Mohrher, A refinement of low n and high n for the r.e. degrees, *Z. Math. Logik Grundlag. Math.* 32 (1) (1986) 5–12.
- [29] Keng Meng Ng, On strongly jump-traceable reals, *Ann. Pure Appl. Logic* 154 (2008) 51–69.
- [30] Keng Meng Ng, *Computability, Traceability and Beyond*. Ph.D. Dissertation, Victoria Univ. of Wellington, 2009.
- [31] André Nies, Lowness properties and randomness, *Adv. Math.* 197 (1) (2005) 274–305.
- [32] André Nies, Reals which compute little, in: *Logic Colloquium ’02*, in: *Lect. Notes Logic*, vol. 27, Assoc. Symbol. Logic, La Jolla, CA, 2006, pp. 261–275.
- [33] André Nies, Superhighness and strong jump traceability, in: *ICALP 2009*, in: *Lecture Notes in Computer Science*, vol. 5555, Springer-Verlag, 2009, pp. 726–739.
- [34] André Nies, *Computability and Randomness*, Oxford University Press, 2009, 444+xv pages.
- [35] André Nies, Applying randomness to computability., in: *Tutorial at ASL Summer Meeting*, Sofia, 2009.

- [36] André Nies, Interactions of computability and randomness, in: Proceedings of the International Congress of Mathematicians, Hyderabad, 2010, pp. 30–57.
- [37] André Nies, Computationally enumerable sets below random sets, *Ann. Pure Appl. Logic* (2011) Available online 28 December 2011.
- [38] André Nies, Calculus of cost functions (in preparation).
- [39] Stephen G. Simpson, Almost everywhere domination and superhighness, *Math. Logic Quart.* 53 (4–5) (2007) 462–482.
- [40] Andrea Sorbi, The Medvedev lattice of degrees of difficulty, in: *Computability, Enumerability, Unsolvability*, in: LMS Lecture Notes, vol. 224, Cambridge University Press, 1996, pp. 289–312.
- [41] Frank Stephan, Yue Yang, Liang Yu, Turing degrees and the Ershov hierarchy, in: Proceedings of the 10th Asian Logic Conference, World Scientific, Singapore, 2009, pp. 300–321.
- [42] Michiel van Lambalgen, The axiomatization of randomness, *J. Symbolic Logic* 55 (1990) 1143–1167.