BENIGN COST FUNCTIONS AND LOWNESS PROPERTIES

NOAM GREENBERG AND ANDRÉ NIES

ABSTRACT. We show that the class of strongly jump-traceable c.e. sets can be characterised as those which have sufficiently slow enumerations so they obey a class of well-behaved cost functions, called benign. This characterisation implies the containment of the class of strongly jump-traceable c.e. Turing degrees in a number of lowness classes, in particular the classes of the degrees which lie below incomplete random degrees, indeed all LR-hard random degrees, and all ω-c.e. random degrees. The last result implies recent results of Diamondstone's and Ng's regarding cupping with superlow c.e. degrees and thus gives a use of algorithmic randomness in the study of the c.e. Turing degrees.

1. Introduction

K-triviality has become central for the investigation of algorithmic randomness. This property of a set A of natural numbers expresses that A is as far from random is possible, in that its initial segments are as compressible as possible: there is a constant b such that for all n, $K(A \upharpoonright n) \leq K(n) + b$, where K denotes prefix-free Kolmogorov complexity. The robustness of this class is expressed by its coincidence with several notions indicating that the set is computationally feeble (Nies [18]; Hirschfeldt and Nies; Hirschfeldt, Nies and Stephan [10]; for more background on these coincidences see [17, Ch. 5]).

- lowness for Martin-Löf randomness: as an oracle, the set $A \in 2^\omega$ cannot detect any patterns in a Martin-Löf random set.
- lowness for K: as an oracle, the set A cannot compress any strings beyond what can be done computably.
- being a base for Martin-Löf randomness: A is so feeble that some A-random set can compute it.

The key for this equivalence is the notion of *cost functions* and *obeying* them. Cost functions were originally introduced to expose the similarity of constructions of typical sets in the various classes. The by-now standard construction of a promptly simple K-trivial set ([6]) resembles the earlier construction of a set that is low for Martin-Löf randomness ([12]), and the construction of a set which is low for K (Mučnik 1999, see [2] or [17, 5.3.35]). The requirements, which want to enumerate numbers into the set A that is being built, are restrained from doing so not by discrete negative requirements, such as in the standard Friedberg construction of a low set, but by a cost function, which has a more continuous nature. This resemblance between the constructions of "typical" representatives in the classes mentioned above was the seed for the proofs of equivalence of these notions. Technically, this equivalence is summarised in the Main Lemma of [17, Section 5.5],

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

which indeed yields the harder implication that each K-trivial is low for K. A further application of that Main Lemma characterizes the class of K-trivial sets by the standard cost function $c_{\mathcal{K}}$: a Δ^0_2 set A is K-trivial if and only if there is a computable approximation $\langle A_s \rangle$ of A such that

$$\sum_{s<\omega} c_{\mathcal{K}}(x,s)[\![x \text{ is least such that } A_s(x) \neq A_{s+1}(x)]\!]$$

is finite, where $c_{\mathcal{K}}(x,s) = \sum_{x<y<s} 2^{-K_s(y)}$. In a sense, the canonical way to construct a K-trivial set is the only way to construct such sets.

All known characterisations of the class of K-trivial sets involve an analytic component such as Lebesgue measure or prefix-free Kolmogorov complexity. A still standing question is whether this class can be defined using purely combinatorial tools, as used in computability theory outside its interaction with algorithmic randomness. A one-time candidate was the class of *strongly jump-traceable* sets. Traceability was introduced into computability theory by Terwijn and Zambella [24] for their study of another lowness notion, that of lowness for Schnorr randomness; a variant was also used by Ishmukhametov [11] in his study of strong minimal covers in the Turing degrees. A third variant, jump-traceability, was introduced by Nies in [19]. The strong version of jump-traceability was defined by Figueira, Nies and Stephan [7]. They showed that a non-computable strongly jump-traceable c.e. set exists. For the formal definitions, recall that a *c.e. trace* for a partial function ψ is a uniformly c.e. sequence $\langle T_x \rangle$ of finite sets such that for all $x \in \operatorname{dom} \psi$ we have $\psi(x) \in T_x$. An *order function* is a computable, non-decreasing and unbounded function $h\colon \omega \to \omega \setminus \{0\}$. A c.e. trace $\langle T_x \rangle$ is *bounded* by an order function h if for all x, $|T_x| \leq h(x)$. Finally, a set A is strongly jump-traceable if for every order function h, every partial function $\psi\colon \omega \to \omega$ which is partial computable in A has a c.e. trace which is bounded by h.

In [3], Cholak, Downey and Greenberg showed that the attempt to define K-triviality using strong jump-traceability fails, but that in fact, restricted to the c.e. degrees, the strongly jump-traceable degrees form a proper sub-ideal of the ideal of K-trivial degrees. This was the first known example of such an ideal. Several questions remained open:

(1) How does the ideal of strongly jump-traceable c.e. sets relate to other ideals and classes of degrees? Most of these classes are known to be contained in the K-trivial degrees but are not yet known to be distinct from the ideal of K-trivial degrees. Here we mostly think of classes derived from algorithmic randomess, such as the collection of degrees which are bounded by an incomplete random degree. Cholak, Downey and Greenberg showed in their paper that the strongly jump-traceable degrees are all ML-non-cuppable, another example for such a class, but no further examples were known.
(2) Can the strongly jump-traceable sets be characterised by cost functions? Related to that is the question of the complexity of this ideal. Can traceability for some fixed order ensure strong jump-traceability? Is this ideal a Σ^0_3 ideal, like the K-trivial ideal, or is it more complicated?
(3) What is the status of non-c.e. strongly jump-traceable sets? The notion of K-triviality is inherently enumerable; the cost function characterisation

implies that every K-trivial set is computable from a c.e. one. Does the same hold for strong jump-traceability?

(4) Are there other characterisations of the strongly jump-traceable sets, which would indicate that this notion is robust?

(5) Are there other natural ideals between the strongly jump-traceable degrees and the K-trivial degrees? Are there natural proper sub-ideals of the strongly jump-traceable degrees?

The main results. We answer the first two questions. The solution for the second question gives a unified approach to one kind of "box-promotion" constructions. Two examples of such constructions were first given in [3].

We begin with a few definitions, following [17]. A *monotone cost function* is a computable function c which associates with every number $x < \omega$ and stage $s < \omega$ a "cost" $c(x, s)$, a non-negative rational number, of changing the approximation $A_s(x)$ for membership of x in A at stage s. We require that for each x, the sequence $\langle c(x, s)\rangle_{s<\omega}$ is non-decreasing, and converges to a limit $c(x)$; we require that the limit cost function $c(x)$ is non-increasing with x, and indeed, that for any fixed stage s, the cost $\langle c(x, s)\rangle_{x<\omega}$ at stage s is non-increasing.

We say that a computable approximation $\langle A_s\rangle$ of a Δ^0_2 set A *obeys* a cost function c if the sum

$$\sum_{s<\omega} c(x, s)[\![x \text{ least such that } A_s(x) \neq A_{s+1}(x)]\!]$$

is finite. We say that a Δ^0_2 set A *obeys* a cost function c if there is some computable approximation $\langle A_s\rangle$ for A which obeys c. In this terminology, Nies's result above is that a Δ^0_2 set is K-trivial if and only if it obeys the standard cost function $c_{\mathcal{K}}$. We note, by the way, that if A is a c.e. set which obeys a cost function c, then A has a computable *enumeration* which obeys c [20].

We usually require our cost functions to satisfy the limit condition $\lim_{x\to\infty} c(x) = 0$, where again $c(x) = \lim_s c(x, s)$. In this paper we introduce a class of cost function that satisfy the limit condition in a restrained and predictable manner.

Definition 1.1. A (monotone) cost function c is *benign* if there is a computable function g such that for every positive rational ε, $g(\varepsilon)$ bounds the size of any collection $\mathcal{I}$ of pairwise disjoint intervals of natural numbers such that for all $[x, s) \in \mathcal{I}$ we have $c(x, s) \geq \varepsilon$.

For example, the standard cost function $c_{\mathcal{K}}$ is benign: for any $\varepsilon > 0$, any $\mathcal{I}$ as in the definition cannot have size greater than $1/\varepsilon$, because the witnesses, in the universal machine, for $c_{\mathcal{K}}(x, s) \geq \varepsilon$ for $[x, s) \in \mathcal{I}$ must all be distinct; this is because the intervals in $\mathcal{I}$ are disjoint.

Here is another way to understand the definition of benignity. Let $\varepsilon > 0$. Set $y^\varepsilon_0 = 0$, and if y^ε_k is defined, and there is some s such that $c(y^\varepsilon_k, s) \geq \varepsilon$, then set y^ε_{k+1} to be the least such s. If c satisfies the limit condition $\lim_x c(x) = 0$, then this process has to halt after finitely many iterations. Then c is benign iff there is a computable (in ε) bound on the number of iterations of this process.

Our main theorem settles the first part of Question (2) above.

Theorem 1.2. *A c.e. set A is strongly jump-traceable if and only if it obeys all benign cost functions.*

Since $c_{\mathcal{K}}$ is benign, this implies the main result of [3], that every strongly jump-traceable c.e. set A is K-trivial.

As for the second part of Question (2), we show the following.

Theorem 1.3. *For any benign cost function c, there is a c.e. set A which obeys c and is not strongly jump-traceable.*

In light of Theorem 1.2, this says that for every benign cost function c there is another, more stringent, and yet still benign, cost function d, such that there is a c.e. set which obeys c but not d. Theorem 1.3 implies Cholak, Downey and Greenberg's result from [3] that the ideal of strongly jump-traceable degrees is strictly contained in the ideal of K-trivial degrees. The techniques we use elaborate on their techniques in the language of procedures and cost functions.

In light of the connection between strong jump-traceability and benign cost functions, exhibited by Theorem 1.2 and Proposition 2.2, Theorem 1.3 is related to Ng's result from [14], that no single order function h can ensure strong jump-traceability. In Section 3 we show that Ng's result follows from Theorem 1.3. Indeed, the results are close: alternatively, we could obtain Theorem 1.3 from Ng's theorem; we include a proof of Theorem 1.3 for completeness of presentation. We remark that utilising techniques similar to the ones of Section 3, Ng went on to show that the index-set for the collection of strongly jump-traceable c.e. sets is Π^0_4-complete, and so certainly is not Σ^0_3.

Applying Theorem 1.2. The main theorem allows us to unify constructions which show that the strongly jump-traceable degrees are a subclass of most of the subclasses of the K-trivial degrees which are considered in the study of algorithmic randomness.

Recall that $Y \leq_{LR} X$ if every X-random set is Y-random. A set X is called *LR-hard* if $\emptyset' \leq_{LR} X$. The class of LR-hard sets is denoted LRH. Kjos-Hanssen, Miller, Solomon (see [22] or [17, 5.6.30]) showed that a set is LR-hard if and only if it is (uniformly) almost everywhere dominating. Nies [17, 6.3.14] showed that an LR-hard random set can be Turing incomplete.

Theorem 1.4. *Every strongly jump-traceable c.e. set is computable from any LR-hard random set.*

As an immediate corollary we see that in the c.e. degrees, the collection of strongly jump-traceable degrees is contained in the collection of degrees which are bounded by incomplete random degrees. The motivation for Theorem 1.4 is the following key open problem in algorithmic randomness [13, Question 4.6]: does the collection of c.e. sets which are bounded by some incomplete random set coincide with the collection of c.e., K-trivial sets? The former collection is known to be contained in the collection of K-trivial c.e. sets, but the reverse implication remains open, and is considered one of the major open problems in the field. It is our hope that Theorem 1.4 will contribute to our understanding of bounding by incomplete random sets.

Hirschfeldt showed that if A is an incomplete c.e. set, X is an incomplete random set, and $\emptyset' \leq_T A \oplus X$, then X is LR-hard (see [17, Theorem 8.5.15]). Hence Theorem 1.4 implies the result from [3], that no strongly jump-traceable c.e. set is ML-cuppable. Again, it is an important open problem whether the class of ML-non-cuppable sets coincides with the K-trivials [13, Question 4.8].

The authors were surprised to discover the following result.

Theorem 1.5. *If a c.e. set is strongly jump-traceable, then it is computable from any ω-c.e. random set.*

In contrast with the situation regarding the LR-hard random sets, we can show (Theorem 5.3) that there is a K-trivial set which is not computable in some ω-c.e. random set. Indeed, recent research by the authors together with Hirschfeldt shows the converse of the foregoing theorem, and hence the coincidence of the c.e. strongly jump-traceable sets and the sets which are computable from every ω-c.e. random set [9].

Theorem 1.5 can be improved as follows. In [8], the authors define a binary relation which is a very strong version of weak truth-table reducibility: $Y \leq_{T(tu)} X$ (Y is reducible to X with tiny use) if for every order function h, there is a reduction of X to Y whose use function is bounded by h. By making our cost functions stringent, we can in fact show that if A is c.e. and strongly jump-traceable, and X is an ω-c.e. random set, then A is reducible to X with tiny use (Proposition 5.2).

The analogy between Theorem 1.4 and Theorem 1.5 leads to the following definition. For a class $\mathcal{C}$ of sets, we let $\mathcal{C}^\Diamond$ be the class of all c.e. sets A which are computable in every random set in $\mathcal{C}$. Thus, these two theorems can be stated as the inclusions

$$\mathsf{SJT}_{c.e.} \subseteq \mathsf{LRH}^\Diamond$$

and

$$\mathsf{SJT}_{c.e.} \subseteq (\omega\text{-c.e.})^\Diamond,$$

where $\mathsf{SJT}_{c.e.}$ denotes the collection of strongly jump-traceable c.e. sets.

We recall that Hirschfeldt and Miller (see [23] or [17, 5.3.15]) used a cost function construction to show that $\mathcal{C}^\Diamond$ contains a promptly simple c.e. set for each null Σ^0_3 class $\mathcal{C}$. Thus the content of Theorems 1.4 and 1.5 is that if for a given $\mathcal{C}$, Hirschfeldt and Miller's construction happens to use a benign cost function, then every strongly jump-traceable c.e. set can be viewed as being produced by their construction.

Applying randomness in degree theory. The class $(\omega\text{-c.e.})^\Diamond$ has an unexpected application. Recall that a set B is *superlow* if $B' \leq_{\mathrm{wtt}} \emptyset'$. A problem in c.e. degree theory, which turned out to be quite difficult to solve, was whether superlow cuppability coincided with low cuppability (which in turn was shown to be equivalent to having a promptly simple degree in the classic [1]). This question was recently settled in the negative by Diamondstone [5]. In parallel, Ng [15] showed that in analogy with the almost deep degree of [4], there is a non-computable c.e. degree which joins every superlow c.e. degree to a superlow degree; he called such degrees *almost superdeep*. We show in Section 5, using the class $(\omega\text{-c.e.})^\Diamond$, that the degree of every strongly jump-traceable c.e. set A is almost superdeep, thus extending the results of both Diamondstone and Ng. In fact we show a much stronger property: $A \oplus B$ is superlow for every superlow set B, without the restriction that B be c.e. It is reasonable to conjecture that this property characterizes $\mathsf{SJT}_{c.e.}$. In contrast, Ng has built an almost superdeep degree that is not strongly jump-traceable.

The remaining questions. To end this introduction, we discuss the status of Questions (3)-(5) from Page 2. In yet unpublished work, Downey and Greenberg showed that in contrast with the jump-traceable sets, every strongly jump-traceable set is Δ^0_2, indeed is K-trivial. Theorem 1.2 seems to indicate that like K-triviality, strong jump-traceability is a notion which is inherently enumerable. Downey and Greenberg currently conjecture that every strongly jump-tracable set is computable in a c.e. one; this would imply that all the results in this paper extend to all strongly jump-traceable sets. It seems likely that the characterisation of strong jump-traceability in terms of cost functions would play an important role in a possible verification of this conjecture. In contrast, Nies thinks that the conjecture will be answered in the negative.

For the fourth question, recent results of the authors together with Hirschfeldt [9] show that the class of strongly jump-traceable degrees is indeed robust, as it coincides with some of the "diamond classes" defined above, such as $(\omega\text{-c.e.})^\Diamond$, $\mathsf{Superlow}^\Diamond$ and $\mathsf{Superhigh}^\Diamond$. Again these results use benign cost functions in a fundamental way to show that each strongly jump-traceable c.e. set is in the diamond class. However, no natural classes which lie strictly between the strongly jump-traceable and the K-trivial degrees have yet been found. Ng [16] has defined and investigated a class which is strictly contained in the strongly jump-traceable degrees. This class is obtained by partially relativising strong jump-traceability to all c.e. sets, and seems to be Π^0_5-complete. In contrast with the strongly jump-traceable degrees, the degrees in this small class cannot be promptly simple. Another line of inquiry examines what happens to the classes $\mathcal{C}^\Diamond$ when $\mathcal{C}$ is increased beyond the class of ω-c.e. sets. We say that a Δ^0_2 set Y is ω^2-c.e. if it can be computably approximated while counting down the natural well-order of $\omega \times \omega$ at each change. Nies has recently shown that $(\omega^2\text{-c.e.})^\Diamond$ is a proper subideal of $(\omega\text{-c.e.})^\Diamond = \mathsf{SJT}_{c.e.}$ [21]. Not much is known otherwise.

2. Proof of the main theorem

In this section we prove Theorem 1.2. We first prove the easy direction: if A is a c.e. set which obeys every benign cost function, then A is strongly jump-traceable. This is implied by the following Proposition 2.1. Recall that for any set A, J^A denotes a universal A-partial computable function; to show that a set is strongly jump-traceable, it is sufficient to show that for every order function h, J^A has a c.e. trace which is bounded by h. Note that if A obeys every benign cost function, then it obeys $c_{\mathcal{K}}$, and so is K-trivial. It follows ([19],[18]) that A is *jump-traceable*: every A-partial computable function has a c.e. trace which is bounded by some order function. Hence it is sufficient to prove the following.

Proposition 2.1. *Let A be a c.e., jump-traceable set, and let h be an order function. Then there is a benign cost function c such that if A obeys c, then J^A has a c.e. trace which is bounded by h.*

Proof. Since A is c.e., tracing $J^A(n)$ is equivalent to tracing the correct $J^A(n)$ computation. In other words, let $\psi^A(n)$ be the stage at which $J^A(n)$ converges with an A-correct computation. Since A is jump-traceable, there is a c.e. trace $\langle S_n \rangle$ for ψ^A which is bounded by some order function g.

Suppose that $J^{A_r}(n)\downarrow$ with use u. We say that this computation is *certified* if there is some t, $u < t < r$, such that $A_r \upharpoonright u = A_t \upharpoonright u$ such that $t \in S_n$ at stage r. We want to make sure that the cost of all $x < u$ at stage r is at least $1/h(n)$.

Hence we let

$$c(x,s) = \max\left\{\frac{1}{h(n)} : \exists\, r \le s \ \left(J^{A_r}(n) \text{ is certified, with use } u > x\right)\right\}.$$

Note that this definition indeed makes c monotone.

We first argue that c is benign. Let $\varepsilon > 0$ and suppose that $\mathcal{I}$ is a set of pairwise disjoint intervals of natural numbers such that for all $[x,s) \in \mathcal{I}$, $c(x,s) \ge \varepsilon$. Find some n^* such that $h(n^*) > 1/\varepsilon$. Let $[x,s) \in \mathcal{I}$. Then there is some $n < n^*$ and $r \le s$ such that $J^{A_r}(n)$, with use $u > x$, is certified. Say that $t < r$ witnesses that $J^{A_r}(n)$ is certified. The key is that $t \in (x,s)$ (as $x < u < t < r \le s$), and so, since the intervals in $\mathcal{I}$ are pairwise disjoint, and $t \in S_n$, we have

$$|\mathcal{I}| \le \sum_{n<n^*} |S_n| \le \sum_{n<n^*} g(n).$$

Since n^* is obtained effectively from ε and g is computable, this bound on $|\mathcal{I}|$ is also effective.

Now suppose that A obeys c. Let $\left\langle \widehat{A}_s \right\rangle$ be a computable enumeration of A such that

$$\sum_{s<\omega} c(x,s)[\![x \text{ least such that } \widehat{A}_s(x) \ne \widehat{A}_{s+1}(x)]\!] < 1.$$

Enumerate a trace $\langle T_n\rangle$ for J^A as follows: enumerate $J^{\widehat{A}_s}(n)$ into T_n at stage s if there is some $r < s$ such that $A_r \upharpoonright u = \widehat{A}_s \upharpoonright u$, where u is the use of the computation $J^{\widehat{A}_s}(n)$, and this computation gets certified at stage r.

Let $n < \omega$ and let $s_0 < s_1 < \cdots < s_{|T_n|-1}$ be the stages at which we enumerate numbers into T_n; say that the computation $J^{\widehat{A}_{s_k}}(n)$ gets certified at stage $r_k < s_k$. Let u_k be the use of that computation. For each $k < |T_n| - 1$ there is some stage $w_k \in [s_k, r_{k+1})$ such that $\widehat{A}_{w_k} \upharpoonright u_k \ne \widehat{A}_{w_k+1} \upharpoonright u_k$, as by stage s_{k+1}, the computation $J^{\widehat{A}_{s_k}}(n)$ is injured by some number below u_k entering A. By design and since $w_k \ge s_k$,

$$c(u_k - 1, w_k) \ge c(u_k - 1, s_k) \ge 1/h(n).$$

Hence $|T_n| - 1 \le h(n)$. Now replacing h by $h+1$ completes the proof. $\square$

In fact, the proposition can be strengthened: the cost function c can be chosen independently of A.

Proposition 2.2. *For every order function h there is a benign cost function c such that for any c.e. set A which obeys c, J^A has a c.e. trace which is bounded by h.*

The proof of Proposition 2.2 uses the notion of a *universal* trace $\langle S_n\rangle$ for an order function p. Let $\widetilde{p} = \left\lfloor \sqrt{p} \right\rfloor$. There is an effective listing $\left\langle S^1, S^2, S^3, \dots \right\rangle$ of all c.e. traces which are bounded by $\widetilde{p}$. Let $S_n = \bigcup_{e<\widetilde{h}(n)} S^e_n$. Then $\langle S_n\rangle$ is a c.e. trace which is bounded by p, and for every partial function ψ, if ψ has a c.e. trace which is bounded by $\widetilde{p}$, then for almost all $n \in \operatorname{dom}\psi$, $\psi(n) \in S_n$, so $\langle S_n\rangle$ *almost traces* ψ.

Proof of Proposition 2.2. By [19, Prop 5.9] (also see [17, Thm. 8.4.15]), there is an order function $\widetilde{g}$ such that for every K-trivial set A, every A-partial computable function has a c.e. trace which is bounded by $\widetilde{g}$. Let $g = \widetilde{g}^2$, and let $\langle S_n\rangle$ be a universal trace for g.

Now let c be the cost function which is obtained by running the proof of Proposition 2.1 using the c.e. trace $\langle S_n \rangle$ and all possible c.e. oracles A. Namely, we say that a computation $J^{W_{e,r}}(n)$ is certified if $W_{e,r} \upharpoonright u = W_{e,t} \upharpoonright u$ where u is the use of the computation, and $t \in S_n$ at stage r. We let

$$c(x,s) = \max \left\{ \frac{1}{h(n)} \,:\, \exists\, e, r \leq s \; \left(J^{W_{e,r}}(n) \text{ is certified, with use } u > x \right) \right\}.$$

Again this definition makes c monotone, and the argument in the proof of Proposition 2.1 shows that c is benign.

Since both c and $c_{\mathcal{K}}$ are benign, so is the cost function $c + c_{\mathcal{K}}$. If A is a c.e. set which obeys $c + c_{\mathcal{K}}$, then it obeys both c and $c_{\mathcal{K}}$. It follows that A is K-trivial, so the converging time function ψ^A for J^A has a c.e. trace bounded by $\widetilde{g}$; so ψ^A is almost traced by $\langle S_n \rangle$. The last paragraph of the proof of Proposition 2.1 now shows that J^A is almost traced by a c.e. trace $\langle T_n \rangle$ which is bounded by h. Of course a finite modification gives a full trace. □

We now turn to the proof of the harder direction of Theorem 1.2: we show that if A is c.e. and strongly jump-traceable, then A obeys every benign cost function. The argument is a generalisation of the box-promotion method proof from [3] which shows that every strongly jump-traceable c.e. set A is K-trivial. Indeed, we prove a converse of Proposition 2.2:

Proposition 2.3. *For any benign cost function c, there is an order function h with the following property: if A is c.e. set such that J^A has a c.e. trace bounded by h, then A obeys c.*

Instead of using the recursion theorem as in [3, 17], we rely on universal traces. We first note that for every order function h' there is an order function h such that for any set X, if J^X has a c.e. trace bounded by h, then every X-partial computable function has a c.e. trace which is bounded by h' (we say that X is *h'-jump-traceable*). This is because every X-partial computable function is coded in the jump function, and we can uniformly limit the rate of growth of the functions which give the coding locations. So if $\langle T_n \rangle$ is a universal trace for $\widetilde{h} = (h')^2$, and J^X has a c.e. trace bounded by h, then every X-partial computable function is almost traced by $\langle T_n \rangle$. In the following, it suffices to define the computable function $\widetilde{h}$; the function h in the theorem is then given by the remark above.

The general idea of any box-promotion construction with c.e. oracle A is to certify some appropriate A-configurations up to varying degrees of certainty. To this end, we define an A-partial computable function Φ^A; to certify $A_s \upharpoonright u$ we define, at stage s, $\Phi^{A_s}(z)$ for various z with use u and output s; the configuration is then certified at a later stage t if $A_t \upharpoonright u = A_s \upharpoonright u$ and $s \in T_z$ at stage t. The degree of certainty this certification gives us depends on the bound $\widetilde{h}(z)$ we have for the size of T_z; we know that we cannot make more than $\widetilde{h}(z) - 1$ many mistakes. So if, for example, $\widetilde{h}(z) = 1$, and $A_s \upharpoonright u$ is certified at stage t, then we know that $A \upharpoonright u = A_s \upharpoonright u$. Unfortunately, though, for almost all z we have $\widetilde{h}(z) > 1$.

Specifically, to find an enumeration $\left\langle \widehat{A}_s \right\rangle$ of A which obeys c, we want to speed up a given enumeration $\langle A_s \rangle$ of A and only accept sufficiently certified configurations of A. To ensure obedience to c, if $\widehat{A}_s(x)$ changes on some x such that

$c(x, s) \geq 2^{-n}$, say, then we want to make progress, in the sense that the previous version of $A \upharpoonright x+1$ was certified by some T_z such that $\widetilde{h}(z) \leq n$. The idea is to ensure that because of this limit on $|T_x|$, this won't happen more than n times. Hence the sum

$$\sum_{s<\omega} c(x,s)[\![x \text{ is least such that } \widehat{A}_s(x) \neq \widehat{A}_{s+1}(x)]\!]$$

will be bounded by $\sum_{n<\omega} n2^{-n}$, which is finite.

The part of the construction which deals with those x's for which $c(x, s) \geq 2^{-n}$, call it requirement R_n, may ignore those x's for which $c(x, s) \geq 2^{-(n-1)}$, as these need to be certified in even stronger "boxes" T_z. All of these certification processes need to work in concert; in general, at a given stage s, we will have $u_1 < u_2 < u_3 < \dots$ such that $A_s \upharpoonright u_1$ has to be certified with strength 2^{-1} (by R_1), $A_s \upharpoonright u_2$ has to be certified with strength 2^{-2} (by R_2), etc. The problem is that not every $\Phi^A(z)$ is traced by T_z; there are finitely many exceptions. Hence for every $d < \omega$, a version of the construction indexed by d will guess that $\Phi^A(z)$ is traced by T_z for each z such that $\widetilde{h}(z) \geq d$. Almost all versions will be successful. To keep the various versions from interacting, each version will control its own (infinite) collection of inputs z. That is, for every z, only one version of the construction will attempt to make definitions of $\Phi^A(z)$.

A common feature of all box-promotion constructions is that certification takes place along a whole block of boxes which together form a "meta-box". The point is that to ensure that R_n only certifies $n-1$ many wrong initial segments of A, we need each failure to correspond to an enumeration into the *same* T_z. On the other hand, if a correct initial segment is tested on some T_z, then this z is never again available for testing other, longer initial segments of A. The idea is that if one meta-box B used by R_n is *promoted* (by some $s \in T_z$ for all $z \in I$ discovered to be wrong), then we break B up into many sub-boxes, and so on. The fact that c is benign, witnessed by a computable bound function g, allows us to set in advance the size of the necessary meta-boxes, thus making $\widetilde{h}$ computable. A meta-box for R_n can be broken up at most n times, so the necessary size for an original R_n meta-box is $(g(2^{-n}))^{n+1}$.

Definition of $\widetilde{h}$ and the initial meta-boxes. Let $\langle I_n \rangle_{n\geq 1}$ be consecutive, pairwise disjoint intervals of ω such that $|I_n| = n(g(2^{-n}))^{n+1}$. For all $z \in I_n$, let $\widetilde{h}(z) = n$.

Next, we split each I_n into intervals $I_n^1, I_n^2, \dots, I_n^n$, each of size $(g(2^{-n}))^{n+1}$. The interval I_n^d will be used by the d-version of R_n, which we denote by R_n^d. So we set $B_{n,0}^d$, the initial meta-box for R_n^d, to be I_n^d.

At any stage s, $B_{n,s}^d$ will be an interval of ω whose size is a power of $g(2^{-n})$. For $k \in \{1, 2, \dots, g(2^{-n})\}$, we let $B_{n,s}^d(k)$ be the k^{th} subinterval of $B_{n,s}^d$ of $B_{n,s}^d$ of size $|B_{n,s}^d|/g(2^{-n})$.

d-stages and certification. As mentioned above, the d-version of the construction guesses that for all $n \geq d$, for all $z \in I_n^d$, if $\Phi^A(z)\downarrow$ then $\Phi^A(z) \in T_z$. The *d-stages* s_i^d are defined by recursion; these are the stages at which this guess looks correct.

We let $s_0^d = 0$. Given s_i^d, let s_{i+1}^d be the least stage $s > s_i^d$ such that for every $n \in [d, i]$, for all $z \in I_n^d$, either $\Phi^{A_s}(z)\uparrow$ or $\Phi^{A_s}(z) \in T_{z,s}$. We ensure that the d-version of the construction only makes definitions of Φ^A at *d-stages*. Hence, if the d-version of the construction guesses correctly, there will be infinitely many *d-stages*.

For a *d-stage* $s = s^d_{i+1}$, let $\bar{s} = s^d_i$ be the previous *d-stage*. We say that $A_s \upharpoonright u$ is *certified* if

$$A_s \upharpoonright u = A_{\bar{s}} \upharpoonright u.$$

Definition of Φ. Fix $d < \omega$ and $n \geq d$. Let $i > n$ and $s = s^d_i$. We describe the action of R^d_n at stage s. Recall the sequence $\left\langle y^{2^{-n}}_k \right\rangle$ from the introduction, which we rename $\langle y^n_k \rangle$: $y^n_0 = 0$, and if y^n_k is defined, then y^n_{k+1} is the least s such that $c(y^n_k, s) \geq 2^{-n}$, if such a stage s exists; otherwise, y^n_{k+1} is not defined. At a stage s we can compute all y^n_k for which $y^n_k \leq s$. We know that $y^n_{g(2^{-n})}$ is not defined.

The aim is that by the end of stage s, if $k \geq 1$, $y^n_k \leq \bar{s}$ is defined and $A_s \upharpoonright y^n_k$ is certified, then we will have $\Phi^{A_s}(z)\downarrow$ with use y^n_k for all $z \in B^d_{n,s}(k)$: we say that $A_s \upharpoonright y^n_k$ is *tested* in $B^d_{n,s}(k)$. The inductive hypothesis on the construction is that this indeed holds for all such k at the end of stage $\bar{s}$, whereas if y^n_k is not defined at stage $\bar{s}$, or it is defined but $A_{\bar{s}} \upharpoonright y^n_k$ is not certified, then for all $z \in B^d_{n,\bar{s}}(k)$ we have $\Phi^{A_{\bar{s}}}(z)\uparrow$.

First, to see if R^d_n can make progress, we check if there is some witness $k \geq 1$ such that $A_{\bar{s}} \upharpoonright y^n_k$ was tested in $B^d_{n,\bar{s}}(k)$, and such that

$$A_s \upharpoonright y^n_k \neq A_{\bar{s}} \upharpoonright y^n_k.$$

If so, then R^d_n can *promote* its meta-box B^d_n: We reset

$$B^d_{n,s} = B^d_{n,\bar{s}}(k),$$

where k is the least such witness. We note that in this case, for every $z \in B^d_{n,s}$ we have, before we make any new definitions, $\Phi^{A_s}(z)\uparrow$, because at stage $\bar{s}$ we have $\Phi^{A_{\bar{s}}}(z)\downarrow$ with use y^n_k. Hence we can define $\Phi^{A_s}(z)$ for such z as we like: for all $l \in [1, k)$, $A_s \upharpoonright y^n_l$ is certified, and so for all $z \in B^d_{n,s}(l)$ we define $\Phi^{A_s}(z) = s$ with use y^n_l.

Now if R^d_n does not promote its meta-box at stage s, then $B^d_{n,s} = B^d_{n,\bar{s}}$; for all $k \geq 1$ such that $A_{\bar{s}} \upharpoonright y^n_k$ was tested at stage $\bar{s}$, $A_s \upharpoonright y^n_k$ is still certified, and is still tested in $B^d_{n,s}(k) = B^d_{n,\bar{s}}(k)$. If there are k such that $y^n_k \leq s$ and $A_s \upharpoonright y^n_k$ is certified, but $A_{\bar{s}} \upharpoonright y^n_k$ was not tested at stage $\bar{s}$, then for all $z \in B^d_{n,s}(k) = B^d_{n,\bar{s}}(k)$ we have $\Phi^{A_{\bar{s}}}(z)\uparrow$, so we can define $\Phi^{A_s}(z) = s$ with use y^n_k for all such z.

This ends the construction. Before we define the enumeration $\left\langle \widehat{A}_s \right\rangle$ of s and show that the enumeration obeys c, we need to make sure that the construction is consistent, in that the instructions can always be carried out. We can easily verify that the inductive hypothesis holds at the end of stage s: if $y^n_k < s$ and $A_s \upharpoonright y^n_k$ is certified, then it is tested in $B^d_{n,s}(k)$; otherwise, for all $z \in B^d_{n,s}(k)$ we have $\Phi^{A_s}(z)\uparrow$. Another issue is to verify that each requirement R^d_n can always promote its meta-box B^d_n when that is required, that is, it can divide $B^d_{n,s}(k)$ into at least $g(2^{-n})$ many sub-intervals. This follows from the size of the original meta-box $B^d_{n,0} = I^d_n$ and the following lemma:

Lemma 2.4. *The procedure R^d_n does not promote its meta-box more than n times.*

Proof. Let $r < t$ be two stages at which R^d_n promotes its meta-box. Note that for all $s \geq r$, for all $z \in B^d_{n,s}(z)$, if $\Phi^{A_s}(z)\downarrow$ then $\Phi^{A_s}(z) \geq r$: when $B^d_{n,r}$ is redefined at stage r, we have $\Phi^{A_r}(z)$ undefined, and all new definitions, at stage r or afterwards, are made with the value being the stage number. Let k be such that $B^d_{n,t} = B^d_{n,\bar{t}}(k)$. By the conditions for promotion, we have $A_{\bar{t}} \upharpoonright y^n_k$ certified and tested, so $\Phi^{A_{\bar{t}}}(z) \in T_{z,t}$ (by the definition of a *d-stage*, which t is). Hence there is a number $s \in [r, t)$ in T_z for all such z.

Also, if t is the first stage at which B^d_n is promoted, then the same argument shows that there is a number smaller than t in T_z for all $z \in B^d_{n,t}$.

The meta-boxes are nested, so if B^d_n were promoted $n+1$ times, say for the $n+1$st time at stage s, we'd have $n+1$ distinct numbers in T_z for all $z \in B^d_{n,s}$. This contradicts the fact that $B^d_{n,s} \subset I^d_n$ and for all $z \in I^d_n$ we have $n = \widetilde{h}(z) \geq |T_z|$. □

We turn to define $\left\langle \widehat{A}_s \right\rangle$ and show that this enumeration of A obeys c.

Fix some d such that for all $n \geq d$, for all $z \in I^d_n$, if $\Phi^A(z)\downarrow$ then $\Phi^A(z) \in T_z$. So there are infinitely many *d-stages*. From now, we drop the superscript d from R^d_n, *d-stage*, s^d_i, $B^d_{n,s}$, etc.

By recursion we define a sub-sequence of *stages*. Let $q(0) = 0$, and given $q(r)$, let $q(r+1)$ be the least *stage* s greater than $q(r)$ at which $A_s \upharpoonright q(r)$ is certified. For all $r < \omega$, let $\widehat{A}_r = A_{q(r+2)} \upharpoonright r$. For all r, let x_r be the least x such that $\widehat{A}_{r-1}(x) \neq \widehat{A}_r(x)$ (so $x_r < r$). Let n_r be the unique n such that

$$2^{-n} \leq c(x_r, r) < 2^{-(n-1)}.$$

Hence, showing that $\left\langle \widehat{A}_r \right\rangle$ obeys c is equivalent to showing that

$$\sum_r 2^{-n_r}$$

is finite.

Lemma 2.5. *For any r, there is some* stage $s \in (q(r+1), q(r+2)]$ *at which R_{n_r} promotes its meta-box.*

Proof. Let $n = n_r$ and $x = x_r$. Let k be the greatest such that y^n_k is defined and $y^n_k \leq r$. We have $x < y^n_k$, for otherwise, by monotonicity of c, we'd have

$$c(y^n_k, r) \geq c(x, r) \geq 2^{-n}$$

which would imply that y^n_{k+1} is defined and is not greater than r.

Hence $y^n_k \leq r \leq q(r)$. The choice of x and the fact that $x < y^n_k$ shows that

$$A_{q(r+2)} \upharpoonright y^n_k \neq A_{q(r+1)} \upharpoonright y^n_k.$$

However, by the definition of $q(r+1)$ and the fact that $y^n_k \leq q(r)$, $A_{q(r+1)} \upharpoonright y^n_k$ is certified, so it is tested on $B_{n,q(r+1)}(k)$ at stage $q(r+1)$. We get a change on $A_s \upharpoonright y^n_k$ by stage $q(r+2)$, so if s is the least *stage* beyond $q(r+1)$ at which $A_s \upharpoonright y^n_k$ is not certified, then $s \leq q(r+2)$ and R_n promotes its meta-box at stage s. □

It follows that for all n,

$$\{r \,:\, n_r = n\}$$

has size at most n, and so

$$\sum_r 2^{-n_r} \leq \sum_n n2^{-n}$$

which is finite. This completes the proof of Proposition 2.3.

3. No single benign cost function suffices for strong jump-traceability

In this section we prove Theorem 1.3: if c is a benign cost function, then there is some c.e. set A which obeys c but is not strongly jump-traceable.

Let g be a computable bound function which witnesses that c is benign. In this construction, for notational convenience, we replace g by $g+1$, so $g(\varepsilon)$ is strictly greater than the number of pairwise disjoint intervals $[x,s)$ such that $c(x,s) \geq \varepsilon$.

To prove the theorem, we enumerate a c.e. set A; the enumeration $\langle A_s \rangle$ which we define will obey c. To ensure that A is not strongly jump-traceable, we design an order function h and build a functional Ψ; we meet the requirements R_e, which say that the e^{th} c.e. trace $\langle S^e_x \rangle$ with bound h does not trace Ψ^A. The idea is that R_e will work with potential witnesses in some interval I_e of natural numbers; we will define $h(x) = e$ for all $x \in I_e$, and so R_e will want to change the value of $\Psi^A(x)$ for some $x \in I_e$ at least e times. To ensure that $\langle A_s \rangle$ obeys c, we sometimes need to abandon a witness, because the cost of redefining $\Psi^A(x)$, by enumerating the use of the computation into A, becomes too large. Since c is benign we can calculate in advance the total number of possible such abandonments R_e may need to concede. This yields a bound on the size of I_e. Hence h is computable. We defer the precise definition of the I_e and the associated order function h until later.

To make the situation clear, we consider the first few requirements. An attempt to meet R_1 would have a witness $x \in I_1$ for which we first define, at some stage s_0, $\Psi^A(x) = s_0$ with use $s_0 + 1$. At a later stage s_1, s_0 appears in S^1_x, and we want to enumerate s_0 into A and redefine $\Psi^A(x) = s_1$, meeting the requirements since $|S^1_x| \leq 1$. If the cost $c(s_0, s_1)$ is greater than the quota, say $1/2$, allocated to R_1, then we need to abandon x and start afresh with a new witness. This can happen only fewer than $g(1/2)$ many times, so we need $|I_1| \geq g(1/2)$.

Now consider R_2. The process is similar, except that if x is not abandoned at stage s_1, then s_1 may still appear in S^2_x at a yet later stage s_2, at which point we want to enumerate s_1 into A. We are now in a double bind, because enumerating s_0 into A at stage s_1 has already cost R_2 the amount of $c(s_0, s_1)$, which was smaller than R_2's quota (say another $1/2$), but yet positive. If $c(s_1, s_2)$ is greater than what's left to spend ($1/2 - c(s_0, s_1)$), then we need to abandon x and start with a fresh witness, with a net loss of $c(s_0, s_1)$ for R_2 and no gain whatsoever. The strategy is to take into consideration all possible such failures and "spread out the investment". Instead of being willing to spend it all each time, R_2 declares a quantity of $1/4$ which is reserved to spending at a stage like s_2, i.e., when it is ready to meet the requirement. This means that it may abandon the witness at the stage s_2 fewer than $g(1/4)$ many times. Between such abandonments, it may spend one unreturned cost at a stage s_1; so the amount it is willing to spend at a stage s_1 should be no more than $1/4g(1/4)$. So between abandoning witnesses at an s_2 stage, we may abandon fewer than $g(\frac{1}{4g(1/4)})$ witnesses. It follows that we

need
$$|I_2| \geq g(1/4)\, g(\frac{1}{4g(1/4)}).$$

In general, R_e's total capital allotment is $e2^{-e}$, and it is willing to overall spend 2^{-e} at each level.

The actions of R_e at level k, $1 \leq k \leq e$, will be carried out through a procedure P^e_k. R_e also uses procedures P^e_0 which don't incur any cost. Each procedure P^e_k for $k > 0$ calls a procedure P^e_{k-1} and expects it to return, at some stage s_k, with a witness x and some s_{k-1} such that $\Psi^A(x) = s_{k-1}$ with use s_{k-1}, and such that $|S^e_x| \geq k$ at stage s_k. There are two possibilities:

(a) It violates that computation by enumerating s_{k-1} into A, redefining $\Psi^A(x) = s_k$ with use $s_k + 1$. Then it waits for s_k to appear in S^e_x, so that $|S^e_x| \geq k+1$ and P^e_k can return to the procedure P^e_{k+1} which called it. If $k = e$ then the requirement is met.

(b) The cost $c_{s_k}(s_{k-1})$ of enumerating s_{k-1} at stage s_k is too big, bigger than a threshold δ^e_k. In this case x gets cancelled, and a new run of P^e_{k-1} is called.

We calculate the necessary thresholds δ^e_k and a bound n^e_k on the total number of times a procedure P^e_k can be called. We call P^e_e once, so let $n^e_e = 1$. Hence $\delta^e_e = 2^{-e}$, and $n^e_{e-1} = g(\delta^e_e)$. Inductively, given n^e_k for $k > 0$, we set
$$\delta^e_k = \frac{2^{-e}}{n^e_k}$$
and
$$n^e_{k-1} = n^e_k\, g\,(\delta^e_k)\,.$$
Recall that we are splitting ω into consecutive intervals $\langle I_e \rangle$, where I_e is the reservoir of inputs to Ψ^A that R_e has access to. If we now require that $|I_e| = n^e_0$, there are sufficiently many inputs for R_e. Recall also that we define the order function h via $h(x) = e$ for all $x \in I_e$.

We now describe the action of each procedure.

A procedure P^e_0, called at some stage s_0, chooses a fresh $x \in I_e$, defines $\Psi^A(x) = s_0$ with use $s_0 + 1$, and waits for s_0 to appear in S^e_x. When this happens, the procedure returns, with output x and s_0.

A procedure P^e_k, for $1 \leq k \leq e$, calls a procedure P^e_{k-1}. When that procedure returns at stage s_k, with a witness x such that $\Psi^A(x) = s_{k-1}$ with use $s_{k-1}+1$, we compare $c(s_{k-1}, s_k)$ and δ^e_k:

- If $c(s_{k-1}, s_k) > \delta^e_k$, then we cancel x, and call a new run of P^e_{k-1}.
- Otherwise, we enumerate s_{k-1} into A, and redefine $\Psi^A(x) = s_k$ with use s_k+1. We wait for s_k to show up in S^e_x. When this happens, the procedure returns, with the witness x.

When a procedure P^e_k defines $\Psi^A(s_k)$ with use $s_k + 1$, while waiting for s_k to appear in S^e_x, if $A \restriction s_k + 1$ changes due to the action of procedures working for other requirements, then P^e_k redefines $\Psi^A(x)$ with the same value and use. In this way the different requirements R_e essentially don't interfere with each other.

The construction starts procedure P^e_e at stage e.

The verification follows the basic plan.

Lemma 3.1. *If $k > 0$, then a single run of a procedure P^e_k calls at most $g(\delta^e_k)$ many procedures P^e_{k-1}.*

Proof. Let $t_1, t_2, \dots, t_m$ be the stages at which a P^e_{k-1} procedure returns, with witnesses $x_1, x_2, \dots x_m$, but the run of P^e_k does not return, necessarily because $c(r_l, t_l) \geq \delta^e_k$, where $r_l = \Psi^{A_{t_l}}(x_l)$ is the stage at which the run of P^e_{k-1} which returns at t_l has defined $\Psi^A(x_l)$. Since $r_l > t_{l-1}$, the intervals in

$$\{[r_l, t_l) \,:\, l = 1, 2, \dots, m\}$$

are pairwise disjoint, so $m < g(\delta^e_k)$ by the hypothesis that c is benign via $g - 1$. $\square$

Lemma 3.2. *For each $k \leq e$, at most n^e_k runs of P^e_k are ever called.*

Proof. This is proved by reverse induction on k.

For $k = e$: we only call P^e_e once, and we defined $n^e_e = 1$.

Let $k < e$, and assume the lemma holds for $k + 1$: no more that n^e_{k+1} calls of procedure P^e_{k+1} are made. By Lemma 3.1, every run of P^e_{k+1} calls at most $g(\delta^e_{k+1})$ many runs of procedure P^e_k, so the total number of runs of P^e_k which are called is at most

$$n^e_{k+1} g(\delta^e_{k+1}) = n^e_k. \qquad \square$$

Corollary 3.3. *A run of P^e_0 can always choose a fresh $x \in I_e$ as a witness.*

Proof. By Lemma 3.2, at most n^e_0 runs of P^e_0 are ever called. Each such run requires one witness $x \in I_e$. We defined $|I_e| = n^e_0$ so we never run out of witnesses. $\square$

Lemma 3.4. *If a run of P^e_k returns with a witness x at some stage, then $|S^e_x| \geq k$ at that stage.*

Proof. This is proved by (forward) induction on k. It is clear for $k = 0$. Suppose that this holds for $k-1$. Suppose that a run of P^e_k returns at some stage s_{k+1}. Then there was a stage s_k at which a procedure P^e_{k-1}, called by this run of P^e_k, returned with the same witness x, and by induction, at that stage s_k, we had $|S^e_x| \geq k - 1$. The run of P^e_k then defined $\Psi^A(x) = s_k$. Note that s_k is not in S^e_x at stage s_k. On the other hand, s_k appears in S^e_x at stage s_{k+1}, so at that later stage we must have $|S^e_x| \geq k$. $\square$

Lemma 3.5. *Each requirement R_e is met.*

Proof. Lemma 3.4 implies that the original run of P^e_e cannot return. By induction we see that there must be some $k \leq e$ such that some run of P^e_k never returns but, if $k > 0$, every run of P^e_{k-1} which is called by that run of P^e_k does return (we can call that a *golden run* of P^e_k).

By Lemma 3.2, there is a last run of P^e_{k-1} which is called by the golden run of P^e_k, and returns with witness x at some stage s_k. (If $k = 0$ then s_0 is the stage at which the golden run of P^e_0 is called, and x is the witness which is chosen.) The golden run of P^e_k goes on to define $\Psi^A(x) = s_k$ and waits forever for s_k to show up in S^e_x. At the end, we have $\Psi^A(x) = s_k$ so $\langle S^e_x \rangle_x$ is not a trace of Ψ^A, which means that R_e is met. $\square$

Lemma 3.6. *The enumeration $\langle A_s \rangle$ obeys c.*

Proof. For every e and every $k = 1, 2, \dots, e$, at most n_k^e runs of P_k^e ever return, and each time one such run returns, it enumerates into A a number whose cost at the time is bounded by δ_k^e. Hence the total amount spent by all the runs of P_k^e is

$$n_k^e\, \delta_k^e = 2^{-e}.$$

It follows that

$$\sum_{s<\omega} c(x,s)[\![x \text{ least such that } A_s(x) \neq A_{s+1}(x)]\!] \leq \sum_e e2^{-e}$$

which is finite. □

Corollary 3.7 (Ng [14])**.** *For every order function h there is an order function $\widetilde{h}$ such that there is a c.e. set which is h jump-traceable but is not $\widetilde{h}$ jump-traceable. Hence, there is no order function h such that the strongly jump-traceable degrees coincide with the h jump-traceable degrees.*

Proof. Because of the proximity between the tracing of all A-partial computable functions and tracing J^A, it is sufficient to show that for every order function h there is a c.e. set A which is not strongly jump-traceable, but such that J^A has a c.e. trace bounded by h.

Given an order function h, by Proposition 2.2, let c be a benign cost function such that for every c.e. set A which obeys c, J^A has a c.e. trace bounded by h. By Theorem 1.3, there is a c.e. set A which obeys c and is not strongly jump-traceable. □

4. The diamond class for being LR-hard

In this section we prove Theorem 1.4: every strongly jump-traceable c.e. set is computable in every LR-hard random set. The class LRH is Σ_3^0 by the equivalence of LR and LK-reducibility (see [17, 8.5.12]). Nonetheless, it is somewhat hard to work with. We actually show that $\mathsf{SJT}_{c.e.}$ is a subclass of a $\mathcal{H}^\Diamond$, where $\mathcal{H}$ is a class which contains LRH and is nicer than LRH. The class $\mathcal{H}$ we use is the class of $\emptyset'$*-tracing* sets.

Definition 4.1. A set X is $\emptyset'$*-tracing* if there is some order function h such that every Δ_2^0 function f has an X-c.e. trace which is bounded by h.

Importantly, this definition is not a true relativisation of the notion of c.e. traceability. If it were, we would say that $\emptyset'$ is c.e. traceable *relative to* X if there is some X-computable non-decreasing, unbounded function h such that every $f \leq_T \emptyset' \oplus X$ has an X-c.e. trace bounded by h. Full relativisation of c.e. traceability, and in fact of many other notions, does not yield useful concepts, at least not as useful of partial relativisation as in Definition 4.1. In this paper, we choose to use the preposition "by" to denote partial relativisation as in Definition 4.1: a set X is $\emptyset'$-tracing iff $\emptyset'$ is c.e. traceable by X. We remark that the preposition "by" was also used by other authors to denote full relativisation, but so have "in" and "over".

Proposition 4.2. *Every set $X \in \mathsf{LRH}$ is $\emptyset'$-tracing.*

Proof. We say that X is *JT-hard* if there is an order function h such that $J^{\emptyset'}$ has an X-c.e. trace bounded by h ($\emptyset'$ is jump-traceable by X). Simpson [22], relying on work of Kjos-Hanssen, Miller and Solomon, showed that every $X \in \mathit{LRH}$ is JT-hard. Clearly, every JT-hard set is $\emptyset'$-tracing. $\square$

Now, the plan is to find some Δ^0_2 function f and a benign cost function c such that if A obeys c and X is a random set which traces f, then $A \leq_T X$. Let h be an order function. We can then fix a *universal oracle trace* for h: a uniformly c.e. sequence $\langle V_n \rangle$ of operators, such that for every oracle $X \in 2^\omega$, $\langle V_n^X \rangle$ is an X-c.e. trace bounded by h, such that every function f which has an X-c.e. trace bounded by $\sqrt{h}$ is almost traced by $\langle V_n^X \rangle$.

Given h, and consequently $\langle V_n \rangle$, we are interested in functions f such that for all n, the measure of

$$\left\{ Y \,:\, f(n) \in V_n^Y \right\}$$

is at most 2^{-n}. For the rest of this section, we will call such functions f *rarely traced for h*. Namely, for few oracles Y is $\langle V_n^Y \rangle$ a trace for f.

Lemma 4.3. *Suppose h is an order function and $f \leq_T \emptyset'$ is rarely traced for h. Then there is a cost function c such that $A \leq_T Y$ for every set A which obeys c, and every random set Y such that $\langle V_n^Y \rangle$ almost traces f. If f is also ω-c.e. then c is benign.*

Lemma 4.4. *For every order function h, there is an ω-c.e. function f which is rarely traced for h.*

Proof of Theorem 1.4, given Lemmas 4.3 and 4.4. Let A be a strongly jump-traceable c.e. set, and let Y be an LR-hard random set. By Proposition 4.2, Y is $\emptyset'$-tracing. Let $\widetilde{h}$ be an order function such that every Δ^0_2 function has a Y-c.e. trace bounded by $\widetilde{h}$. Let $h = \widetilde{h}^2$. Then letting $\langle V_n \rangle$ be the universal oracle trace for h, we know that $\langle V_n^Y \rangle$ almost traces every Δ^0_2 function.

By Lemma 4.4, there is an ω-c.e. function f which is rarely traced for h. By Lemma 4.3, there is a benign cost function c such that if A is a c.e. set which obeys c, then $A \leq_T Y$. By the main Theorem 1.2, A obeys c. $\square$

Before we continue, we show the following:

Proposition 4.5. *The ideal $\mathsf{LRH}^\Diamond$ properly contains $\mathsf{SJT}_{c.e.}$.*

We do not know if $\mathsf{LRH}^\Diamond$ coincides with the ideal of K-trivial sets.

Proof. The proof above of Theorem 1.4 shows that $\mathsf{SJT}_{c.e.} \subseteq (\emptyset'\text{-}\mathit{tracing})^\Diamond$. By Proposition 4.2 we have $(\emptyset'\text{-}\mathit{tracing})^\Diamond \subseteq \mathsf{LRH}^\Diamond$. It now suffices to show that the inclusion of $\mathsf{SJT}_{c.e.}$ in $(\emptyset'\text{-}\mathit{tracing})^\Diamond$ is proper.

For definiteness let $h(n) = n$. Choose an ω-c.e. function f as in Lemma 4.4 that is rarely traced for h. Obtain a benign cost function c as in Lemma 4.3. By Theorem 1.3 there is some c.e. set A which obeys c but is not strongly jump-traceable. If A obeys c then $A \leq_T Y$ for any random Y such that $\langle V_n^Y \rangle$ traces f. Hence $A \in (\emptyset'\text{-}\mathit{tracing})^\Diamond$. $\square$

We turn to prove Lemmas 4.3 and 4.4. The proof of 4.4 uses ideas of Hirschfeldt involving the following measure theoretic analog of the pigeon hole principle.

Fact 4.6. Let $\varepsilon > 0$ and $k < \omega$. Let $\mathcal{B}$ be a collection of measurable subsets of 2^ω which has size greater than k/ε, such that every $B \in \mathcal{B}$ has measure at least ε. Then there is some $\mathcal{C} \subseteq \mathcal{B}$ of size $k+1$ such that

$$\bigcap \mathcal{C}$$

is non empty (indeed, is not null).

Proof. [17, Ex. 1.9.15] Suppose that the intersection of any $k+1$ sets in $\mathcal{B}$ is null. Let

$$f = \sum_{B \in \mathcal{B}} 1_B,$$

where 1_B is the indicator function of B. The assumption implies that on a co-null set, $f(x) \leq k$, so $\int f(x)\,dx \leq k$. On the other hand, for every $B \in \mathcal{B}$, $\int 1_B(x)\,dx \geq \varepsilon$ and so

$$\int f(x)\,dx = \sum_{B \in \mathcal{B}} \int 1_B(x)\,dx > \frac{k}{\varepsilon}\varepsilon = k,$$

which is a contradiction. □

Proof of Lemma 4.4. Let h be an order function and let $\langle V_n \rangle$ be the associated universal oracle trace. We define an increasing approximation $\langle f_s \rangle$ for the function f, starting with $f_0(n) = 0$ for all n.

The idea is to keep, for each n, the measure of

$$\left\{Y \,:\, f_s(n) \in V^Y_{n,s}\right\} \tag{1}$$

not greater than 2^{-n}. So if we see at stage s that the measure of

$$\left\{Y \,:\, f_{s-1}(n) \in V^Y_{n,s}\right\}$$

is exceeding 2^{-n}, then we redefine $f_s(n) = s$, to keep the measure of the set (1) bounded by 2^{-n} at stage s.

To show that f is ω-c.e. (and well-defined), we see that for each n, the value $f_s(n)$ changes at most $2^n h(n)$ many times. For if $p = f_{s-1}(n)$ and $f_s(n) = s$, then the set

$$B^n_p = \left\{Y \,:\, p \in V^Y_n\right\}$$

has measure at least 2^{-n}. The intersection of B^n_p for more than $h(n)$ many such p's is empty, because $\left|V^Y_n\right| \leq h(n)$ for every Y. Fact 4.6 implies that there can be no more than $h(n)/2^{-n}$ many m's which are discarded as values for $f_s(n)$. □

For an alternative shorter write-up of the following proof of Lemma 4.3 see [17, Lemma 8.5.19]; the present write-up gives more intuition on how to obtain the Turing reduction from A to Y. An argument similar to the present one recurs in the somewhat simpler setting of Proposition 5.1 below.

Proof of Lemma 4.3. Let h be an order function, $\langle V_n \rangle$ the associated universal oracle trace, and let $\langle f_s(n) \rangle$ be a computable approximation of a function f which is rarely traced for h. By speeding up the approximation of f, we may assume that at every stage s, the measure of

$$\left\{Y \,:\, f_s(n) \in V^Y_{n,s}\right\}$$

is bounded by 2^{-n}.

We first explain the main ideas behind the proof. Suppose that Y is a random set and that $\langle V_n^Y \rangle$ almost traces f, and that A is a set with a computable approximation $\langle A_s \rangle$ obeying the cost function c that we will define. We want to show that Y computes A; suppose that at stage s, for every n, we have committed that $Y \upharpoonright u(n)$ computes $A_s \upharpoonright k(n)$, where $u(n)$ is a use marker and $k(n)$ is a target marker. The main challenge, of course, is to ensure the correctness of the computation once $A \upharpoonright k(n)$ changes. The mechanism for doing that is enumerating a sequence of c.e. open classes $\langle U_n \rangle$, where $Y \in U_n$ will roughly imply that $Y \upharpoonright u(n)$ has committed to compute a wrong initial segment of A. We will ensure that the sets $\langle U_n \rangle$ together add up to a Solovay test, in the sense that $\sum_n \mu(U_n)$ is finite; we would then know that Y can be in at most finitely many U_n's, and so (from some point) must compute A correctly.

The challenge then moves to make the sum $\sum_n \mu(U_n)$ finite. This is where we utilise the assumption that $\langle V_n^Y \rangle$ almost traces f. A naïve strategy for limiting the damage is setting $k(n) = n$, waiting for $f_s(n) \in V_{n,s}^Y$, with some use $u(n)$ and then mapping $Y \upharpoonright u(n)$ to $A_s \upharpoonright n$. Since the measure of the set of Z's which trace $f_s(n)$ in V_n^Z is at most 2^{-n}, it would seem that this ensures that the measure of U_n too is bounded by 2^{-n}. This, however, does not take into account the fact that $f_t(n)$ may change after stage t; following the naïve strategy would have us enumerate a weight of 2^{-n} for each possible value of $f_t(n)$, and this does not have a finite total. The problem becomes acute when the following sequence of events occurs: we have three approximations for $f(n)$, at stage s_1, s_2 and s_3. At stages t_1 and t_3 ($s_1 < t_1 < s_2 < s_3 < t_3$), the current value of $f(n)$ is traced in V_n^Y and so Y computes both $A_{t_1} \upharpoonright n$ and $A_{t_3} \upharpoonright n$. However, a number below n enters A between stages s_2 and s_3, so $A_{t_1} \upharpoonright n \neq A_{t_3} \upharpoonright n$. However, $f_{s_2}(n)$ does not appear in V_n^Y, so the change in A between stage s_2 and s_3 does not allow us to enumerate Y into U_n.

The solution is to share the responsibility down the ladder. First, instead of just waiting for $f_s(n)$ to appear in $V_{n,s}^Y$, we wait for $f_s(m)$, for all $m \le n$, to appear in $V_{m,s}^Y$ (or, if $\langle V_n^Y \rangle$ only traces f from some m^* onwards, all $m \in [m^*, n]$). Then, at any stage at which $f_t(m)$ changes, we let $Y \upharpoonright u(m)$ be responsible for computing $A \upharpoonright n$; that is, we set $k(m) \ge n$. This will correspond to setting $c(n,t) \ge 2^{-m}$. Then, if $A \upharpoonright n$ changes, we look at the least m such that $k(m) \ge n$. This is the least m such that the approximation for $f(m)$ has changed since stage s. It follows that the approximation for $f(m-1)$ has not changed since stage s, and so the current value of $f(m-1)$ is in $V_{m-1,s}^Y$. This allows us to enumerate Y into U_m, and thus record the change in $A \upharpoonright n$.

We can now give the formal details. First, we define the cost function c. For all $x < \omega$, let $c(x,0) = 2^{-x}$. For any stage $t > 0$, let y be the least number such that $f_t(y) \neq f_{t-1}(y)$. We let, for all $x < t$,

$$c(x,t) = \max\{c(x,t-1), 2^{-y}\},$$

(and leave $c(x,t)$ unchanged for all $x \ge t$). A short examination will reveal that c is monotone and satisfies the limit condition. If $\langle f_s \rangle$ is an ω-c.e. approximation – say the mind-change function is bounded by a computable function g – then c is benign: suppose that $\mathcal{I}$ is a set of pairwise disjoint intervals of natural numbers such that for all $[x,s) \in \mathcal{I}$ we have $c(x,t) \ge 2^{-n}$. If $[x,t) \in \mathcal{I}$ and $x > n$, then

$f_r(m) \neq f_{r-1}(m)$ for some $m \leq n$ and $r \in [x, t)$. Hence

$$|\mathcal{I}| \leq (n+1) + \sum_{m \leq n} g(m),$$

which is a computable bound.

Let $\langle A_t \rangle$ be a computable approximation of a set A which obeys c. We introduce some notation: for any stage $t \geq n$, let $s_n(t)$ be the least stage $s \leq t$ such that for all $r \in [s, t]$, $f_r(n) = f_s(n)$. By changing $f_n(n)$, we may assume that for all t,

$$f_{s_n(t)}(n) \neq f_{s_n(t)-1}(n),$$

so $s_n(t) \geq n$ for all $t \geq n$.

At stage t, let n be the least such that

$$A_t \restriction s_n(t) \neq A_{t-1} \restriction s_n(t).$$

For that n, enumerate all the sets Y such that

$$f_t(n-1) \in V^Y_{n-1,t}$$

into U_n.

We note that by our assumption that the approximation for f is a "rarely traced" one, the measure of the collection of sets which are enumerated into U_n at a given stage is at most $2^{-(n-1)}$. Let n_t be the unique number n such that sets are enumerated into U_n at stage t (let $n_t = \infty$ is there is no such n). So

$$\sum_n \mu(U_n) \leq 2 \sum_t 2^{-n_t}.$$

Secondly, if sets are enumerated into U_n at a stage t, then there is some $x < s_n(t)$ such that $A_t(x) \neq A_{t-1}(x)$, and such that $c(x, t) \geq 2^{-n}$. Let $m_t = -\log_2 c(x, s)$, where x is least such that $A_t(x) \neq A_{t-1}(x)$. Then $m_t \leq n_t$ for all t, so

$$\sum_t 2^{-n_t} \leq \sum_t 2^{-m_t}.$$

The assumption that $\langle A_s \rangle$ obeys c means that the sum on the right is finite. Hence $\sum_n \mu(U_n)$ is finite as well.

By standard randomness arguments (for example, thinking of the union of the U_n's as a Solovay test), if Y is random, then $Y \notin U_n$ for almost all n. Let Y be a random set, and suppose that $\langle V^Y_n \rangle$ almost traces f. To complete the proof, we need to show that $A \leq_T Y$. Let n^* be large enough so that:

- $Y \notin U_n$ for all $n > n^*$, and
- $f(n) \in V^Y_n$ for all $n \geq n^*$.

Let s^* be a stage late enough so that

$$f_s \restriction n^* + 1 = f \restriction n^* + 1$$

for all $s \geq s^*$, and such that no sets are enumerated into any U_m for $m \leq n^*$ after stage s^*.

Let $n > n^*$. We claim that:

> If $t > s^*$ and for all $k \in [n^*, n)$ we have $f_t(k) \in V^Y_{k,t}$,
> then $A_t \restriction s_n(t) = A \restriction s_n(t)$.

For suppose, for contradiction, that there is a stage $u > t$ such that $A_u \upharpoonright s \neq A_{u-1} \upharpoonright s$, where $s = s_n(t)$. Let m be the least such that $A_u \upharpoonright s_m(u) \neq A_{u-1} \upharpoonright s_m(u)$. Since $u > s^*$, we have $m > n^*$; but also $m \leq n$ because $s_n(u) \geq s_n(t) = s$.

Now we claim that $f_u(m-1) \in V^Y_{m-1,u}$. For by assumption, $f_t(m-1) \in V^Y_{m-1,t}$, the latter set is a subset of $V^Y_{m-1,u}$, and if $f_u(m-1) \neq f_t(m-1)$ then $s_{m-1}(u) > t \geq s$, which would contradict the minimality of u. Hence Y gets enumerated into U_m at stage u, which is a contradiction.

Now as $s_n(t) \geq n$ for all n, we certainly have $A \leq_T Y$. □

5. The classes $(\omega\text{-c.e.})^\Diamond$ and $\mathit{Superlow}^\Diamond$

We show that $\mathit{SJT}_{c.e.} \subseteq (\omega\text{-c.e.})^\Diamond$ (Theorem 1.5) and that $\mathit{Superlow}^\Diamond$ is properly contained in the K-trivial degrees (Theorem 5.3). Of course, $\mathit{Superlow} \subseteq \omega\text{-c.e.}$ and so $(\omega\text{-c.e.})^\Diamond \subseteq \mathit{Superlow}^\Diamond$.

Theorem 1.5 is an immediate consequence of the main Theorem 1.2 and the following proposition. For an alternative write-up of the proof see [17, Fact 5.3.13].

Proposition 5.1. *Let Y be a Δ^0_2 random set. Then there is a cost function c with the limit condition such that every set which obeys c is computable from Y. If, further, Y is ω-c.e., then c is benign.*

Proof. This is similar to the proof of Lemma 4.3. Let $\langle Y_s \rangle$ be a computable approximation for Y. The idea is to let $Y \upharpoonright n-1$ compute $A \upharpoonright s$ if $Y_s \upharpoonright n \neq Y_{s-1} \upharpoonright n$. Let $c(x, 0) = 2^{-x}$. For $t > 0$, if $n = n_t$ is the least such that $Y_t \upharpoonright n \neq Y_{t-1} \upharpoonright n$, then we let, for all $x < t$,

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

Again it is easy to verify that c is monotone, and satisfies the limit condition. Suppose now that the number of stages s such that $Y_s(m) \neq Y_{s-1}(m)$ is bounded by $g(m)$, where g is a computable function. If $\mathcal{I}$ is a set of pairwise disjoint intervals of natural numbers such that for all $[x, s) \in \mathcal{I}$ we have $c(x, s) \geq 2^{-n}$, then for all $[x, s) \in \mathcal{I}$ such that $x > n$, there is some $t \in (x, s]$ such that $Y_t \upharpoonright n \neq Y_{t-1} \upharpoonright n$. Hence

$$|\mathcal{I}| \leq (n+1) + \sum_{m<n} g(m).$$

Thus, if Y is ω-c.e. via the computable approximation $\langle Y_s \rangle$, then c is benign.

Now suppose that $\langle A_s \rangle$ is a computable approximation of a set A which obeys c. For all n and $t \geq n$, let $s_n(t)$ be the least stage $s \leq t$ such that for all $r \in [s, t]$, $Y_r \upharpoonright n = Y_t \upharpoonright n$. Again, without loss of generality, $s_n(t) \geq n$ for all $t \geq n$. At a stage t, if n is least such that $A_t \upharpoonright s_n(t) \neq A_{t-1} \upharpoonright s_n(t)$, then we enumerate $Y_t \upharpoonright n-1$ into a Solovay test G. The fact that $\langle A_s \rangle$ obeys c implies that indeed, the sum $\sum_{\sigma \in G} 2^{-|\sigma|}$ is finite, with an argument which mirrors the one in the proof of Lemma 4.3.

Since Y is random, only finitely many initial segments of Y are enumerated into G; suppose that the last one is enumerated at some stage s^*. We now claim that if $t > s^*$ and

$$Y \upharpoonright n-1 = Y_t \upharpoonright n-1,$$

then

$$A \upharpoonright s_n(t) = A_t \upharpoonright s_n(t).$$

Indeed, we show that the approximation $A_s \upharpoonright s_n(t)$ cannot change after stage t. Assume otherwise. Let $u > t$ be a stage at which $A_u \upharpoonright s_n(t) \neq A_{u-1} \upharpoonright s_n(t)$. let m be the least number such that $A_u \upharpoonright s_m(u) \neq A_{u-1} \upharpoonright s_m(u)$. Since s_n is non-decreasing, we have $s_n(u) \geq s_n(t)$, so $A_u \upharpoonright s_n(u) \neq A_{u-1} \upharpoonright s_n(u)$; so by the minimality of m, we have $m \leq n$.

At stage u, we enumerate the string $Y_u \upharpoonright m-1$ into G. Since $u > t > s^*$, $Y_u \upharpoonright m-1$ cannot be an initial segment of Y. The minimality of m implies that $A_u \upharpoonright s_{m-1}(u) = A_{u-1} \upharpoonright s_{m-1}(u)$, so by the assumption on u, we have $s_{m-1}(u) < s_n(t) \leq t$; so

$$Y_u \upharpoonright m-1 = Y_t \upharpoonright m-1 \subseteq Y_t \upharpoonright n-1.$$

But the assumption is that $Y_t \upharpoonright n-1$ is an initial segment of Y. This is a contradiction. □

In fact, we can improve Proposition 5.1 to show that the use of the reduction of A to Y can grow as slowly as we like. Already we see that the proof gives us $A \leq_{\mathrm{wtt}} Y$, indeed $A \leq_{\mathrm{ibT}} Y$: A is reducible to Y with the use of the reduction bounded by the identity function. In fact, we can have the use grow as slowly as we like.

Proposition 5.2. *If A is a strongly jump-traceable c.e. set and Y is an ω-c.e. random set, then for every order function h there is a Turing reduction of A to Y whose associated use function is bounded by h.*

In the terminology of [8], $A \leq_{T(tu)} Y$: A is reducible to Y with *tiny use*.

Proof. We prove the following equivalent statement: if p is a strictly increasing recursive function, then there is a Turing functional Φ such that $\Phi(Y) = A$ and such that for all n and all $i < p(n)$, the computation $\Phi(Y \upharpoonright n; i)$ is defined.

We modify the proof of Proposition 5.1. Define, for all x,

$$c(x, 0) = \max\left\{2^{-n} \,:\, p(n) \leq x\right\}$$

and for $s > 0$, if n is the least such that $Y_s \upharpoonright n \neq Y_{s-1} \upharpoonright n$, let, for all $x < p(s)$,

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

The argument in the proof of Proposition 5.1 shows that if the number of stages s such that $Y_s(m) \neq Y_{s-1}(m)$ is bounded by a computable function g, and $\mathcal{I}$ is a set of pairwise disjoint intervals of numbers such that for all $[x, s) \in \mathcal{I}$, $c(x, s) \geq 2^{-n}$, then

$$|\mathcal{I}| \leq p(n) + 1 + \sum_{m \leq n} g(m),$$

so c is benign.

Say that $\langle A_s \rangle$ obeys c. Again we let, for all n and $t \geq p(n)$, $s_n(t)$ be the least stage $s \leq t$ such that for all $r \in [s, t]$ we have $Y_r \upharpoonright n = Y_t \upharpoonright n$; by manipulating the approximation $\langle Y_s \rangle$, we may assume that for all n, for all $t \geq p(n)$, we have $s_n(t) \geq p(n)$. The rest of the proof now follows the proof of Proposition 5.1 verbatim, to show that for almost all t, if $Y \upharpoonright n-1 = Y_t \upharpoonright n-1$, then $A \upharpoonright s_n(t) = A_t \upharpoonright s_n(t)$. This gives us a Turing functional Φ such that $\Phi(Y) = A$ and such that for all n all $i < p(n)$, the computation $\Phi(Y \upharpoonright n; i)$ is defined, as required. □

We turn to prove:

Theorem 5.3. *There is a K-trivial degree which is not in $\mathsf{Superlow}^{\Diamond}$.*

Theorem 5.3 follows from Proposition 5.4, applied to any Π^0_1 class which only contains random sets.

Proposition 5.4. *If $\mathcal{P}$ is a non-empty Π^0_1 class, then there is some superlow $Y \in \mathcal{P}$ and a K-trivial c.e. set D such that D is not computable from Y.*

Proof. This is an elaboration on the (super)low basis theorem which states that $\mathcal{P}$ has a superlow member. At stage s, we define a sequence $\langle \mathcal{P}_{0,s}, \mathcal{P}_{1,s}, \dots, \mathcal{P}_{2s,s}\rangle$ as follows.

- Let $\mathcal{P}_{0,s} = \mathcal{P}$.
- Given $\mathcal{P}_{2e,s}$, if we see at stage s that $J^X(e)\!\downarrow$ for all $X \in \mathcal{P}_{2e,s}$, then we let $\mathcal{P}_{2e+1,s} = \mathcal{P}_{2e,s}$; otherwise, we let
$$\mathcal{P}_{2e+1,s} = \{X \in \mathcal{P}_{2e,s} \,:\, J^X(e)\!\uparrow\}.$$
- Given $\mathcal{P}_{2e+1,s}$, we try to meet the requirement R_e which states that $\Phi_e(Y) \neq D$. Each such requirement will choose a witness x_e. If we see, at stage s, that for all $X \in \mathcal{P}_{2e+1,s}$ we have $\Phi_e(X, x_e)\!\downarrow = 0$, then we let $\mathcal{P}_{2e+2,s} = \mathcal{P}_{2e+1,s}$ (and enumerate x_e into D if not done so already). Otherwise, we let
$$\mathcal{P}_{2e+2,s} = \{X \in \mathcal{P}_{2e+1,s} \,:\, \Phi_e(X, x_e)\!\uparrow \;\vee\; \Phi_e(X, x_e)\!\downarrow = 1\}.$$

At the beginning of the next stage, witnesses for the requirements R_e are updated: if $\mathcal{P}_{2e+1,s} \neq \mathcal{P}_{2e+1,s-1}$, then we pick a fresh witness for R_e. At stage s, let $i_s(e)$ be the number of stages s at which $\mathcal{P}_{2e+1,s} \neq \mathcal{P}_{2e+1,s-1}$. If $c_{\mathcal{K}}(x_e, s+1) > 2^{-(e+i_s(e))}$, then we pick a new, fresh witness for R_e. Here again $c_{\mathcal{K}}$ is the standard cost function for K-triviality.

Now by induction, we can show that the Π^0_1 classes $\mathcal{P}_k$ reach a limit, and indeed there is a computable bound on the number of changes of each $\mathcal{P}_k$. As usual, after $\mathcal{P}_{2e,s}$ has stabilised, $\mathcal{P}_{2e+1,s}$ changes at most once. So if there are at most n_{2e} versions of $\mathcal{P}_{2e,s}$, then there are at most $n_{2e+1} = 2n_{2e}$ versions of $\mathcal{P}_{2e+1,s}$.

As long as $\mathcal{P}_{2e+1,s}$ does not change, x_e can change at most $2^{e+i_s(e)}$ many times (as $c_{\mathcal{K}}$ is benign with bound 2^n). Beyond those changes, $\mathcal{P}_{2e+2,s}$ can change at most once before it is injured by a change in $\mathcal{P}_{2e+1,s}$. So there are at most
$$n_{2e+2} = \sum_{i \le n_{2e+1}} \left(2^{e+i} + 1\right)$$
changes in $\mathcal{P}_{2e+2,s}$. The map $k \mapsto n_k$ is computable.

It follows that $\bigcap \mathcal{P}_k$ is a singleton $\{Y\}$, that Y is superlow, and that Y does not compute D. Since D obeys $c_{\mathcal{K}}$, it is K-trivial. $\square$

We finish with an application to c.e. degree theory already discussed in the introduction. We first need a partial relativisation of the superlow basis theorem.

Proposition 5.5. *Let $\mathcal{P}$ be a non-empty Π^0_1 class, and let $B \in 2^\omega$. Then there is some $Y \in \mathcal{P}$ such that*
$$(Y \oplus B)' \le_{\mathrm{tt}} B'.$$

Proof. For a string τ, let

$$Q_\tau = \left\{ Y \in P : \forall e < |\tau| \, \left[\tau(e) = 0 \to J^{Y \oplus B}(e)\uparrow \right] \right\}.$$

Note that Q_τ is a $\Pi^0_1(B)$ class, uniformly in τ. Emptiness of such a class is a $\Sigma^0_1(B)$ condition, so there is a computable function g such that

$$Q_\tau = \emptyset \leftrightarrow g(\tau) \in B'.$$

Thus, there is a Turing functional Ψ such that Ψ^X is total for each oracle X, and $\Psi(B', e) = \tau_e$, where τ_e is the leftmost string τ of length $e+1$ such that Q_τ is non-empty. Let $Y \in \bigcap_e Q_{\tau_e}$. Then

$$e \in (Y \oplus B)' \leftrightarrow \tau_e(e) = 1,$$

so $(Y \oplus B)' \leq_{\mathrm{tt}} B'$. □

In the following we strengthen the result of Ng [15] that there is a non-computable almost superdeep degree.

Corollary 5.6. *Let the c.e. degree* $\mathbf{a}$ *be strongly jump-traceable. Then for every superlow degree* $\mathbf{b}$*, the degree* $\mathbf{a} \vee \mathbf{b}$ *is also superlow.*

Proof. Let A be a strongly jump-traceable c.e. set, and let B be a superlow set. By Proposition 5.5 applied to a Π^0_1 class of random sets, there is a random set Y such that

$$Y' \leq_{\mathrm{tt}} (Y \oplus B)' \leq_{\mathrm{tt}} B' \leq_{\mathrm{tt}} \emptyset',$$

so Y is superlow. By Theorem 1.5, we have $A \leq_T Y$, so

$$(A \oplus B)' \leq_{\mathrm{tt}} (Y \oplus B)' \leq_{\mathrm{tt}} \emptyset',$$

hence $A \oplus B$ is superlow as well. □

Corollary 5.7 (Diamondstone [5])**.** *There is a promptly simple c.e. degree which does not join to* $\emptyset'$ *with any superlow c.e. set.*

Proof. In [7], the authors show that there is a strongly jump-traceable c.e. set which is promptly simple. By Corollary 5.6, such a set is almost superdeep. Certainly, a degree that is almost superdeep does not join to $\emptyset'$ with any superlow c.e. set. □

References

[1] Klaus Ambos-Spies, Carl G. Jockusch, Jr., Richard A. Shore, and Robert I. Soare. An algebraic decomposition of the recursively enumerable degrees and the coincidence of several degree classes with the promptly simple degrees. *Trans. Amer. Math. Soc.*, 281(1):109–128, 1984.

[2] Richard Beigel, Harry Buhrman, Peter Fejer, Lance Fortnow, Piotr Grabowski, Luc Longpre, Andrej Muchnik, Frank Stephan, and Leen Torenvliet. Enumerations of the Kolmogorov function. *J. Symbolic Logic*, 71(2):501–528, 2006.

[3] Peter Cholak, Rod Downey, and Noam Greenberg. Strong jump-traceabilty I: The computably enumerable case. *Adv. Math.*, 217(5):2045–2074, 2008.

[4] Peter Cholak, Marcia J. Groszek, and Theodore A. Slaman. An almost deep degree. *J. Symbolic Logic*, 66(2):881–901, 2001.

[5] David Diamondstone. Promptness does not imply superlow cuppability. *J. Symbolic Logic*, 74(4):1264–1272, 2009.

[6] Rod G. Downey, Denis R. Hirschfeldt, André Nies, and Frank Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[7] Santiago Figueira, André Nies, and Frank Stephan. Lowness properties and approximations of the jump. *Ann. Pure Appl. Logic*, 152(1-3):51–66, 2008.
[8] Johanna N.Y. Franklin, Noam Greenberg, Frank Stephan and Guohua Wu. Reducibilities with tiny use. In preparation.
[9] N. Greenberg, D. Hirschfeldt, and A. Nies. Characterizing the strongly jump traceable sets via randomness. To appear.
[10] Denis R. Hirschfeldt, André Nies, and Frank Stephan. Using random sets as oracles. *J. Lond. Math. Soc. (2)*, 75(3):610–622, 2007.
[11] Shamil Ishmukhametov. Weak recursive degrees and a problem of Spector. In *Recursion theory and complexity (Kazan, 1997)*, volume 2 of *de Gruyter Ser. Log. Appl.*, pages 81–87. de Gruyter, Berlin, 1999.
[12] Antonín Kučera and Sebastiaan A. Terwijn. Lowness for the class of random sets. *J. Symbolic Logic*, 64(4):1396–1402, 1999.
[13] Joseph S. Miller and André Nies. Randomness and computability: open questions. *Bull. Symbolic Logic*, 12(3):390–410, 2006.
[14] Keng Meng Ng. On strongly jump traceable reals. *Ann. Pure Appl. Logic*, 154(1):51–69, 2008.
[15] Keng Meng Ng. Almost superdeep degrees. In preparation.
[16] Keng Meng Ng. Beyond strong jump-traceablity. *Proc. Lon. Math. Soc.*, To appear.
[17] André Nies. Computability and randomness. Oxford University Press, 2009.
[18] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.
[19] André Nies. Reals which compute little. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 261–275. Assoc. Symbol. Logic, 2006.
[20] A. Nies. Calculus of cost functions. To appear.
[21] A. Nies. Subclasses of the c.e. strongly jump traceable sets. To appear.
[22] Stephen G. Simpson. Almost everywhere domination and superhighness. *Math. Log. Q.*, 53(4-5):462–482, 2007.
[23] Stephen G. Simpson. Mass problems and almost everywhere domination. *Math. Log. Q.*, 53(4-5):483–492, 2007.
[24] Sebastiaan A. Terwijn and Domenico Zambella. Computational randomness and lowness. *J. Symbolic Logic*, 66(3):1199–1205, 2001.