MARTIN-LÖF REDUCIBILITY AND COST FUNCTIONS

NOAM GREENBERG, JOSEPH S. MILLER, ANDRÉ NIES, AND DAN TURETSKY

Abstract. Martin-Löf (ML)-reducibility compares the complexity of K-trivial sets of natural numbers by examining the Martin-Löf random sequences that compute them. One says that a K-trivial set A is ML-reducible to a K-trivial set B if every ML-random computing B also computes A. We show that every K-trivial set is computable from a c.e. set of the same ML-degree. We investigate the interplay between ML-reducibility and cost functions, which are used to both measure the number of changes in a computable approximation, and the type of null sets intended to capture ML-random sequences. We show that for every cost function there is a c.e. set that is ML-complete among the sets obeying it. We characterise the K-trivial sets computable from a fragment of the left-c.e. random real Ω given by a computable set of bit positions. This leads to a new characterisation of strong jump-traceability.

Contents

1. Introduction 1
2. Some formal definitions and facts 6
3. Inherent enumerability of the K-trivials up to $\equiv_{\mathrm{ML}}$ 8
4. For each cost function there is an ML-complete set 10
5. Each K-trivial set is ML-complete for a cost function 12
6. K-trivial sets Turing below fragments of Ω 15
7. Feeble sets for cost functions, and the structure of ML-degrees 19
8. Proof of Lemma 6.9 22
9. Fragments of Ω and strong jump-traceability 25
References 27

1. Introduction

Martin-Löf (ML) randomness and K-triviality are antipodal properties of sets of natural numbers. Nonetheless, sets of the two kinds interact in interesting ways via Turing reducibility. For instance, combining the results of [3, 6], a c.e. set is K-trivial if and only if it is computable from a Turing incomplete ML-random set; see [2].

Our purpose is to study the relative complexity of the K-trivial sets via a pre-ordering coarser than Turing reducibility that is given by this interaction: $A \leqslant_{\mathrm{ML}} B$

Date: February 10, 2022.

2010 *Mathematics Subject Classification.* Primary 03D32; Secondary 03D30, 68Q30.

Greenberg and Nies were supported by the Marsden Fund of New Zealand. Greenberg was also supported by a Rutherford Discovery Fellowship from the Royal Society of New Zealand. This research was started during a retreat at the Research Centre Coromandel.

if every ML-random computing B also computes A. This preordering, called ML-reducibility, was introduced in [3] by Bienvenu, Kučera, and three of the authors of the present paper. They showed that there is an ML-complete K-trivial (which they called "smart"). While the K-trivials appear somewhat amorphous under Turing reducibility, we will show that an interesting structure emerges when they are viewed through the lens of this preordering. Research in this direction was carried out first by three of the authors of the present paper in [14]. They described a dense linear hierarchy of natural principal ideals in the ML-degrees of the K-trivials. Our first result, Theorem 3.1, shows that each K-trivial is ML-equivalent to a c.e. K-trivial, so the structure we find on the K-trivials is fully given by c.e. witnesses.

Some background on ML-randomness and K-triviality. Sets of natural numbers (often simply referred to as *sets*) will be identified with infinite bit sequences. ML-randomness is central among the notions of randomness given by algorithmic tests. A set $Z \subseteq \omega$ is ML-random (sometimes just called "random" in this paper) if $Z \notin \bigcap_m G_m$ for any sequence of uniformly Σ^0_1 sets in Cantor space such that the Lebesgue measure of G_m is at most 2^{-m}. A sequence $\langle G_m \rangle$ of this kind is called an ML-test.

Chaitin's Ω, the halting probability of a universal prefix-free machine, is an example of an ML-random sequence. Note that Ω is Turing equivalent to the halting problem. There are various ways to build a low ML-random sequence. One way is to use the low basis theorem together with the fact that there is a universal ML-test. Another is to take Ω_R, the bits of Ω with location in an infinite, co-infinite computable set R; to see that Ω_R is low one can e.g. combine [25, Prop. 3.4.10], due to [23], with van Lambalgen's Theorem [29].

Let $K(x)$ denote the prefix-free descriptive complexity of a string x. One says that a set A is K-trivial if there is a constant b such that $K(A \upharpoonright n) \leqslant K(n) + b$ for each n. (Here the "n" in $K(n)$ is interpreted as a string, for instance the string obtained by writing n in binary.) Note that $K(n)$ is, up to a constant not depending on n, the lowest complexity possible for a string of length n. So K-trivial sets have minimal initial segment complexity, again up to a constant. The Levin–Schnorr theorem states that a sequence Z is ML-random iff $K(Z \upharpoonright n) \geqslant n - d$ for some d only depending on Z. As $K(n) \leqslant 2 \log n + O(1)$, the notion of K-triviality is indeed antipodal to ML-randomness: K-trivial by definition means far from random.

Eighteen or so characterisations of the K-trivials are known presently, most of them saying that the set is in some sense close to computable. For instance, A is K-trivial iff A is low for ML-randomness, in the sense that each ML-random is ML-random relative to A [24]. For more recent ones, A is K-trivial iff for each ML-random Y, the symmetric difference $Y \triangle A$ is ML-random [22]; A is K-trivial iff for all Y such that Ω is Y-random, Ω is $Y \oplus A$-random [13].

Despite these characterisations, and the detailed knowledge of the class of K-trivials they appeared to convey, paradoxically, not much progress had been made on the internal structure of the class since the early papers, such as [18, 24]. It was known that the K-trivials are downward closed under Turing reducibility, and that they determine an ideal in the Turing degrees that is contained in the superlow degrees, generated by its c.e. members, and has no greatest degree (i.e., it is nonprincipal). In hindsight it appears that Turing reducibility was the wrong preordering to analyse the internal structure. We will use the coarser ML-reducibility to amend this lack of information.

We briefly review some results that led to the formulation of ML-reducibility. The Kučera–Gacs Theorem [11, 19] states that every set A is Turing below some ML-random set Z. If A is Δ^0_2 then one can take Ω as Z. What can a Turing incomplete ML-random set compute? Kučera [20] showed that each Δ^0_2 ML-random Z is Turing above a noncomputable c.e. set. Hirschfeldt, Nies, and Stephan [18] proved that if Z is Turing incomplete, then Z is necessarily ML-random relative to any such c.e. set A it computes. By definition, this means that A is a *basis for ML-randomness*, which implies that it is K-trivial [18]. Since being K-trivial also means close to computable [24], this shows that an incomplete ML-random can only compute c.e. sets that are close to computable. In the other direction, the works [2, 3, 6] mentioned in the first paragraph show that there is, in fact, a single incomplete Δ^0_2 ML-random Turing above all the K-trivials.

Complexity classes of K-trivials. The K-trivials are closed downward under $\leqslant_{\mathrm{ML}}$ by Day and Miller [6]. By an *ML-complexity class* in the K-trivials we mean a subclass that determines an ML-ideal, namely, it is closed downward under $\leqslant_{\mathrm{ML}}$, and closed under the join operator $\oplus$. Several works from 2012 on can be viewed as studies of such classes. For example, a set is strongly jump-traceable (see the beginning of Section 9 for the definition) if and only if it is computable from all ML-random sequences that are ω-c.a., that is, weak truth table below the halting problem [12]. This implies that the strongly jump-traceables form an ML-ideal.

Given an ML-complexity class $\mathcal{C}$ of K-trivials, one can ask the following two general questions:

(a) Can one describe the class only referring to properties of its members, rather than to randoms that compute them?
(b) Is some set ML-complete for the class? In other words, is the ML-ideal principal?

As an example of a description in (a) consider the original definition [9] of strong jump traceability of a set A, which (as the name indicates) refers to a way to tightly approximate the values of J^A, the Turing jump function of A.

A general way to formulate a condition of lowness among the K-trivials is by restricting the changes of computable approximations. Recall that each K-trivial A is Δ^0_2 and hence has a computable approximation $\langle A_s \rangle$. A *cost function* is a computable function

$$\mathbf{c}\colon \mathbb{N} \times \mathbb{N} \to \{r \in \mathbb{R} \,:\, r \geqslant 0\},$$

which is typically chosen to be nondecreasing in s, nonincreasing in x, and to satisfy the *limit condition*, namely, the asymptotic cost $\mathbf{c}(x) = \sup_s \mathbf{c}(x,s)$ approaches 0 as x increases. The idea is that at stage s, the least x such that $A_s(x)$ changes incurs the cost $\mathbf{c}(x,s)$, which subsumes the cost of changes at larger numbers at the same stage; a set A *obeys* a cost function $\mathbf{c}$ if it has a computable approximation that is sufficiently "inert" in that it incurs a finite total cost. (This means more than that there are few changes; it also means that changes need to be carried out in "blocks", which saves costs because only the change at the least number is counted.) It is a basic fact that each cost function that satisfies the limit condition is obeyed by a noncomputable c.e. set. Building on previous results, Nies [27, Thm. 4.3] showed that obedience to the cost function $\mathbf{c}_\Omega(x,s) = \Omega_s - \Omega_x$ characterizes the K-trivials.

We will show that an affirmative answer to (a) via obedience to a cost function implies an affirmative answer to (b). Given a cost function $\mathbf{c}$ that is at least as

strong as $\mathbf{c}_\Omega$, we show as an easy corollary to our second main result, Theorem 4.3, that *some c.e. set A obeys $\mathbf{c}$ and is ML-complete among the sets obeying $\mathbf{c}$.* So if a cost function $\mathbf{c}$ describes a complexity class, then that class has an ML-complete member. In particular, this holds for the class of all K-trivials, which therefore determines a principal ML-ideal, a result that was first obtained in [3] using similar methods. (At present this appears to be essentially the only known way to show a complexity class in the K-trivials has an ML-complete member.) Given a low c.e. set B, some c.e. set $A \nleqslant_{\mathrm{T}} B$ obeys $\mathbf{c}$ by [25, Thm. 5.3.22]. So the sets obeying $\mathbf{c}$ (being K-trivial, and hence low) never form a Turing principal ideal. This lends support to our thesis that, compared to Turing reducibility, the coarser ML-reducibility leads to a more satisfying complexity theory of the K-trivials.

The class of half-bases is a further example of a complexity class that can be described by a cost function. It yields one level of the dense hierarchy of ML-ideals described in [14] which we alluded to earlier on. One says that a set A is a *half-base* if there is an ML-random Y such that $A \leqslant_{\mathrm{T}} Y_0, Y_1$, where Y_0 consists of the bits in the even, and Y_1 of the bits in the odd positions. Note that each half-base is a basis for ML-randomness by van Lambalgen's theorem, and hence K-trivial by [18]. By [14, Thm. 1.1.], one can require that $Y = \Omega$; furthermore, by [14, Thm. 1.3.] the class of half-bases can be described by the cost function $\mathbf{c}_{\Omega,1/2}(x,s) = \sqrt{\mathbf{c}_\Omega(x,s)}$. Larger cost functions are harder to obey, in a sense made precise in [27, Thm. 3.4]. So $\mathbf{c}_{\Omega,1/2}$ describes a proper subclass of the K-trivials. The properness of the inclusion of the class of half bases in the K-trivials is a result first obtained in [3, Thm. 1.3]. By our method, there is an ML-complete half-base.

By definition, a set A is ML-complete for a class $\mathcal{C}$ of K-trivials if it has the least class of ML-randoms computing it among the members of $\mathcal{C}$. We build a c.e. set A that is ML-complete for the sets obeying $\mathbf{c}$ by showing that ML-randoms computing A cannot be very random, in the sense that they fail a generalised type of ML test called a $\mathbf{c}$-*test*, where the convergence of the measure of G_m to 0 is bounded by $O(\mathbf{c}(m))$, rather than 2^{-m} (recall here that $\mathbf{c}(m) = \sup_s \mathbf{c}(m,s)$). We then use the important, if easy Proposition 2.6 below: any set B obeying $\mathbf{c}$ is Turing below any ML-random failing such a test.

Our Theorem 4.3 extends the basic fact that every cost function $\mathbf{c}$ is obeyed by a noncomputable c.e. set. The c.e. set A we build obeys $\mathbf{c}$, but "only just", in the sense that the collection of ML-randoms computing A is as small as possible. We call such a set A *smart for* $\mathbf{c}$, continuing the terminology of [3] for $\mathbf{c}_\Omega$; the theorem states that each cost function has a smart set. Then we verify, as an easy consequence of this existence of a smart set, that smartness for $\mathbf{c}$ coincides with ML-completeness for $\mathbf{c}$.

The previous works [3, 14] and, in particular, the present paper show that far from being an obstacle, the fact that K-trivials are close to computable can be advantageous for the study of their relative computational complexity. Tools can be applied that would not work for computationally more complex Δ^0_2 sets. The main idea described above is to differentiate between K-trivials via the incomplete randoms that compute them. In contrast, a c.e. set that is not K-trivial only allows randoms above the halting problem to compute it by [18]; such c.e. sets all have the same ML-degree.

The K-trivials Turing below a fragment Ω_R of Ω. Bit sequences derived in some way from Chaitin's Ω often play a special role in the algorithmic theory of randomness.

For an infinite computable set R, we have defined above the sequences Ω_R of bits of Ω with a location in R. Section 6 introduces a cost function $\mathbf{c}_{\Omega,R}$ that describes the class of K-trivials computable from Ω_R. As a main result of this paper, we show in Theorem 6.6 that this cost function essentially only depends on the function $n \to |R \cap n|$ that gives the number of elements of R less than n, taken up to an additive constant. For instance, if R is the set of even numbers and S the odd numbers, then the cost functions corresponding to R and S are equivalent as far as obedience goes, and hence the K-trivials below Ω_R coincide with the K-trivials below Ω_S. This can be extended to k/n bases, for $1 \leqslant k < n$, in the sense of [14], where one takes as R any union of k sets of the form $n\mathbb{N} + r$, $0 \leqslant r < n$. Let $B_{k/n}$ be a smart set for the corresponding cost function. Via the results in [14] these sets determine a chain in the ML-degrees of K-trivials that is isomorphic to $(0,1)_{\mathbb{Q}}$. For detail see the discussion around Theorem 6.1.

As an application of Theorem 6.6, in Theorem 7.1 we show that the K-trivials computable from Ω_R are exactly those that obey $\mathbf{c}_{\Omega,R}$, as promised. Given a cost function $\mathbf{c}$, an ML-random set Y failing a $\mathbf{c}$-test will be called *feeble for* $\mathbf{c}$ if the only K-trivials it computes are the ones that obey $\mathbf{c}$. This notion is dual to smartness for K-trivials. In this language, we show that Ω_R is feeble for $\mathbf{c}_{\Omega,R}$.

Structure of the $\leqslant_{\mathrm{ML}}$-degrees of K-trivials. By its definition, ML-reducibility is a weakening of Turing reducibility. The least degree consists of the computable sets. The usual join operation $\oplus$ induces a least upper bound in the ML-degrees.

Dual to Theorem 4.3, we will show in Proposition 5.1 that each K-trivial A is smart for some cost function $\mathbf{c}_{(A)}$ that can be obtained uniformly from A. This yields further degree theoretical information in Corollary 5.3. Firstly, there are no minimal pairs in the ML-degrees of K-trivials. Secondly, no ML-degree of a noncomputable K-trivial contains a maximal Turing degree; in particular, it contains an infinitely ascending chain of Turing degrees. We do not know whether the ML-degree of a noncomputable K-trivial can contain a minimal Turing degree.

As a further, more powerful application of Theorem 6.6 we will show in Theorem 7.4 that every countable partial ordering is embeddable into the ML-degrees of K-trivial sets. Also, based on a method of Kučera [20], we obtain a pair of incomparable degrees below each non-zero K-trivial ML-degree. In fact, we show that for each noncomputable c.e. K-trivial D, there are c.e. sets $A, B \leqslant_{\mathrm{T}} D$ such that $A \mid_{\mathrm{ML}} B$.

Only basic facts are known on ML-reducibility outside the K-trivials. Since each Turing degree above $\varnothing'$ contains a ML-random, for sets above $\varnothing'$ the two reducibilities coincide. More generally, for PA-complete sets A, B, we have

$$A \leqslant_{ML} B \Leftrightarrow A \oplus \varnothing' \leqslant_T B \oplus \varnothing'.$$

(We thank the referee for pointing this out.) For the direction from left to right, one uses the result of Stephan [28] that the only PA-complete ML-randoms are the ones above $\varnothing'$.

In the final section, Section 9, we connect obedience of the cost functions $\mathbf{c}_{\Omega,R}$ to strong jump traceability, the aforementioned very strong lowness notion. While the original definition [9] was combinatorial, Hirschfeldt et al. [12] showed that a set is strongly jump traceable iff it is Turing below each ω-c.a. ML-random Y. We prove that A is strongly jump traceable iff it obeys all the cost functions $\mathbf{c}_{\Omega,R}$ for infinite computable R. In particular, it suffices to let the ω-c.a. sets Y as above range over the fragments Ω_R.

Remark 1.1. The correspondence between cost functions and ML-degrees is incomplete. Firstly, a cost function $\mathbf{c}$ determines an ML-degree, that of the sets which are smart (equivalently, ML-complete) for $\mathbf{c}$. However, not every set in that degree obeys $\mathbf{c}$. Secondly, every K-trivial set A is smart for some cost function $\mathbf{c}_{(A)}$. However, this cost function is not determined by the ML-degree of A; in fact, in Theorem 5.4 we construct an example of a set A such that even the set that results from A by removing the first bit does not obey $\mathbf{c}_{(A)}$.

Ideally, we could characterise ML-reducibility on K-trivials in terms of which cost functions they obey. This would give a satisfying positive answer to the following question, which remains open:

Question 1.2. Is the relation $\leqslant_{\mathrm{ML}}$ on the K-trivial sets arithmetical?

In fact, a weaker question remains open: whether ML-completeness among the K-trivals is arithmetical. Another question that remains unsettled is whether the ML-degrees of K-trivials are dense. Given that Question 1.2 remains open, it is hard to envisage a requirement-based construction showing density, as this would need some sort of effective listing of "ML-reduction procedures". Cost function-based methods appear to be insufficient here.

2. Some formal definitions and facts

In this section, for easy reference, we provide formal definitions of some of the notions discussed above. We discuss some technical detail and basic connections that will be important for the rest of the paper.

Definition 2.1 ([3])**.** For sets A and B, we write $A \leqslant_{\mathrm{ML}} B$ if $B \leqslant_{\mathrm{T}} Y$ implies $A \leqslant_{\mathrm{T}} Y$ for every ML-random sequence Y.

Cost functions were introduced in [25, Section 5.3] and developed further in [15, 27].

Definition 2.2. A *cost function* is a computable function

$$\mathbf{c}\colon \mathbb{N} \times \mathbb{N} \to \{r \in \mathbb{R} \,:\, r \geqslant 0\}.$$

We only consider *monotonic* cost functions (satisfying $\mathbf{c}(x,s) \leqslant \mathbf{c}(x,s+1)$ and $\mathbf{c}(x,s) \geqslant \mathbf{c}(x+1,s)$) that have the *limit condition*: for all x, $\underline{\mathbf{c}}(x) = \lim_s \mathbf{c}(x,s)$ exists, and $\lim_x \underline{\mathbf{c}}(x) = 0$. Further, we assume that $\mathbf{c}(x,s) = 0$ when $x \geqslant s$.

The original purpose of cost functions was to quantify the number of changes required in a computable approximation of a Δ^0_2 set A: $\mathbf{c}(x,s)$ is the cost of changing at stage s our guess about the value of $A(x)$. Monotonicity means that the cost of a change increases with time, and that changing the value at a smaller number is more costly. Formally:

Definition 2.3 ([25])**.** Let $\langle A_s \rangle$ be a computable approximation of a Δ^0_2 set A, and let $\mathbf{c}$ be a cost function. The *total* $\mathbf{c}$*-cost* of the approximation is

$$\mathbf{c}\langle A_s \rangle = \sum_{s \in \omega} \{\mathbf{c}(x,s) \,:\, x \text{ is least such that } A_{s-1}(x) \neq A_s(x)\}\,.$$

We say that a Δ^0_2 set A *obeys* $\mathbf{c}$ if the total $\mathbf{c}$-cost of *some* computable approximation of A is finite. We write $A \models \mathbf{c}$. For cost functions $\mathbf{c}$ and $\mathbf{d}$, we write $\mathbf{c} \to \mathbf{d}$ if $A \models \mathbf{c}$ implies $A \models \mathbf{d}$ for each Δ^0_2 set A. By [27, Thm. 3.4], this is equivalent to $\underline{\mathbf{d}} \leqslant^{\times} \underline{\mathbf{c}}$.

The basic existence theorem for cost functions, e.g., described in [27, Thm. 2.7(i)], says that if a cost function $\mathbf{c}$ has the limit condition, then some non-computable

c.e. set obeys A. As mentioned, an important example of a cost function is $\mathbf{c}_\Omega(x,s) = \Omega_s - \Omega_x$, where $\langle \Omega_s \rangle$ is an increasing sequence of rational numbers converging to a left-c.e., ML-random real Ω. A set obeys this cost function if and only if it is K-trivial ([27, Thm. 4.3], which modified a result for a related cost function in [24]).

Definition 2.4. Let $\mathbf{c}$ be a cost function and let A be a Δ^0_2 set. We say that A is *ML-complete for* $\mathbf{c}$ if $A \models \mathbf{c}$, and $\forall B\, [B \models \mathbf{c} \Rightarrow B \leqslant_{\mathrm{ML}} A]$.

Note that the implication arrow only goes from left to right; it is not true in general that the class of sets obeying a cost function is well behaved (see Remark 1.1).

We next add some formal detail to our discussion of Theorem 4.3 above, that each cost function has a smart c.e. set. As mentioned, cost functions can also be used to introduce randomness notions between weak 2-randomness and ML-randomness.

Definition 2.5 ([3], Def. 2.13)**.** Let $\mathbf{c}$ be a cost function. A sequence $\langle V_n \rangle$ of uniformly c.e. open sets such that $V_n \supseteq V_{n+1}$ for each n is a $\mathbf{c}$-*bounded test* (or $\mathbf{c}$-test for short) if $\mu(V_n) \leqslant^\times \underline{\mathbf{c}}(n)$ for all n.

We say that such a test *captures* a set Y if $Y \in \bigcap_n V_n$. We also say that such a Y *fails* the test. A sequence is $\mathbf{c}$-*random* if it fails no $\mathbf{c}$-test.

The fundamental connection between our two uses of a cost function is the following:

Proposition 2.6 ([3], Prop. 4.2)**.** *If $A \models \mathbf{c}$ and Y is an ML-random sequence captured by a $\mathbf{c}$-bounded test, then $A \leqslant_{\mathrm{T}} Y$.*

To sketch the proof, say Y is covered by the $\mathbf{c}$-test $\langle V_n \rangle$. define a functional Γ by letting $\Gamma^X(n) = A_s(n)$ if X goes into V_n at stage s. An A-change threatens to invalidate these definitions, so we build a Solovay test; if $A(n)$ is the least change at stage s, put $V_{n,s}$ into the test. Being ML-random, Y is only in finitely many components of the Solovay test, so $\Gamma^Y(n) = A(n)$ for sufficiently large n.

As mentioned, Kučera showed that every Δ^0_2 ML-random sequence is Turing above a non-computable c.e. set. Hirschfeldt and Miller in unpublished work dating from 2006 strengthened this: below any Σ^0_3 null class of randoms there is a non-computable c.e. set. Relying on Proposition 2.6, these proofs can be framed in the language of cost functions; see [15] and [25, 5.3.15], respectively.

The existence of an ML-complete K-trivial was shown in [3]. Given that K-trivials are the sets obeying $\mathbf{c}_\Omega$, and the ML-randoms computing all K-trivials are the ones that fail some $\mathbf{c}_\Omega$-test, ML–completeness for K-trivials coincides with being smart for $\mathbf{c}_\Omega$ in the sense of the next definition:

Definition 2.7. Let $\mathbf{c}$ be a cost function and A be a K-trivial set. We say that A is *smart for* $\mathbf{c}$ if A obeys $\mathbf{c}$ and for each ML-random set Y,

$$Y \text{ is captured by a } \mathbf{c}\text{-bounded test} \Leftrightarrow A \leqslant_{\mathrm{T}} Y.$$

Informally, A is as complex as possible for obeying $\mathbf{c}$, in the sense that the only random sets Y above A are the ones that have to be there because of Proposition 2.6. To summarise the discussion in the introduction, in Theorem 4.3, we show that there is a smart set for any cost function $\mathbf{c}$ such that obedience to $\mathbf{c}$ implies K-triviality. In Corollary 4.4, we use this to show that if $\mathbf{c} \to \mathbf{c}_\Omega$ then A is smart for $\mathbf{c}$ iff A is ML-complete for $\mathbf{c}$. Dual to Theorem 4.3, in Proposition 5.1, we prove that any K-trivial set is smart for some cost function $\mathbf{c}_{(A)}$; when A is c.e., this will be the strongest cost function obeyed by A.

3. Inherent enumerability of the K-trivials up to $\equiv_{\mathrm{ML}}$

In this section, we considerably strengthen the result [24] that every K-trivial is computable from a c.e. K-trivial: we show that the c.e. K-trivial can be taken to have the same ML-degree. This is a powerful tool. It is usually easier to prove results for the c.e. K-trivials; extra work is needed to lift results to the general case. Theorem 3.1 simplifies this process in many cases. Indeed, we use it in both Section 5 and Section 8 for this purpose.

Theorem 3.1. *For every K-trivial set A, there is a (K-trivial) c.e. set D such that $D \geqslant_{\mathrm{T}} A$ and $D \equiv_{\mathrm{ML}} A$.*

Note that the K-triviality of D is free: every K-trivial is Turing below an incomplete ML-random sequence [2, 6], and every c.e. set below an incomplete random is K-trivial. So by virtue of being ML-equivalent to A, the c.e. set D must be K-trivial.

Theorem 3.1 follows from a fact of independent interest. Intuitively, the fact states that each K-trivial A has a computable approximation that converges faster than any computation of A from a random.

Lemma 3.2. *For every K-trivial set A, there is a computable approximation $\langle A_s \rangle$ of A such that for every ML-random X and Turing functional Φ with $A = \Phi^X$ the following holds. For sufficiently large n, if $A \upharpoonright n \preccurlyeq \Phi_s^X$, then $A_t \upharpoonright n = A \upharpoonright n$ for every $t \geqslant s$.*

Proof of Theorem 3.1. Assuming that the lemma holds, we argue that a random computing A must also compute a modulus for A; this modulus will have a c.e. degree. Let $\langle A_s \rangle$ be the approximation from the lemma. Let D be the change-set for this approximation: $(n,k) \in D$ if and only if there is a sequence of stages $s_0 < \cdots < s_k$ with $A_{s_i}(n) \neq A_{s_{i+1}}(n)$ for all $i < k$. D is clearly c.e., and it can compute $A(n)$ by searching for the least k with $(n,k) \notin D$ and considering the parity of k and the value of $A_0(n)$.

Suppose that $A = \Phi^X$ for some random X. By the lemma, there is N such that for all $n \geqslant N$, the approximation converges to $A \upharpoonright n$ faster than Φ^X does. Thus X can compute $D(n,k)$ by waiting until a stage t with $A \upharpoonright n \preccurlyeq \Phi_t^X$ and then only searching for sequences of stages $s_0 < \cdots < s_k$ such that $s_k \leqslant t$. For $n < N$, we can arrange that our computation knows $D(n,k)$ by table lookup. □

For the rest of the paper, we fix a Turing functional Υ that is universal in the sense that $\Upsilon^{0^e 1 \hat{\ } X} = \Phi_e^X$ for each X and e. We assume that for every e, for all sufficiently large n, for all s and X, $\Phi_{e,s}(X;n)\downarrow \Rightarrow \Upsilon_s(0^e 1 \hat{\ } X; n)\downarrow$.

Proof of Lemma 3.2. It suffices to prove the lemma for the functional Υ. Let us first give a brief explanation of the proof. We will use the "Main Lemma" derived from the golden run construction [25, 5.5.1]. The Main Lemma says that if we design a left-c.e. oracle discrete measure on ω, (equivalently, an adaptive additive cost function, or a prefix-free oracle machine), then there is a computable approximation $\langle A_s \rangle$ of A such that the total of all weights that are believed at some stage of the construction and later are shown to be false is finite. Roughly, we would like, at stage s, to put the weight $\mu(\Upsilon_s^{-1}[A_s \upharpoonright n])$ on the string $A_s \upharpoonright n$, where μ is Lebesgue measure on Cantor space, and

$$\Upsilon_s^{-1}[\sigma] = \{X \in 2^\omega \,:\, \Upsilon_s^X \succcurlyeq \sigma\}.$$

Such an approximation for A will be as required: we can put a Solovay test on the reals that compute A too early, and thus random oracles will only converge and agree with $A_s \upharpoonright n$ after it has settled.

The problem is that this definition does not give a discrete measure: there is no reason to believe that $\sum_n \mu(\Upsilon^{-1}[A \upharpoonright n])$ is finite. What we notice, though, is that if an oracle X gives us a correct version of A too early, then this version $A_s \upharpoonright n = A \upharpoonright n$ will later change to $A_t \upharpoonright n \neq A \upharpoonright n$, but after that will need to change back. We can thus put the weight not on $A \upharpoonright n$ but on an incorrect version $A_t \upharpoonright n$. And this is guaranteed to give a measure: the collection of strings of the form $(A \upharpoonright n)\hat{}(1-A(n))$ that disagree with A only on the last bit is pairwise incomparable, and so the preimages under Υ of these strings are pairwise disjoint.

We provide the formal details. For $\sigma \in 2^{<\omega}$ with $\sigma \neq \langle\rangle$, define $\hat{\sigma}$ to be the binary string of the same length which disagrees with σ on the final bit, but agrees on all other bits. For example, if $\sigma = 001011$, then $\hat{\sigma} = 001010$. For $\sigma \in 2^{<\omega}$ with $\sigma \neq \langle\rangle$, for brevity, define

$$\mathcal{U}_\sigma = \left\{X \ : \ \Upsilon^X \geqslant \hat{\sigma}\right\}.$$

To avoid the need of repeatedly dealing with $\langle\rangle$ separately, define $\mathcal{U}_{\langle\rangle} = \varnothing$. Note that $(\mathcal{U}_\sigma)_{\sigma \in 2^{<\omega}}$ are uniformly Σ^0_1-classes. Also, for $\sigma \not\preceq \rho$, $\mathcal{U}_\sigma$ and $\mathcal{U}_\rho$ are disjoint.

For our argument, we will require a computable approximation to A that obeys an adaptive cost function—a cost function where the cost at a given stage depends on the approximation up to that stage. For $\langle A_t \rangle$, a computable approximation of A, define

$$\mathbf{c}^{\langle A_t \rangle}(n, s) = \mu(\mathcal{U}_{A \upharpoonright n+1}[s]).$$

Claim 3.2.1. There is a computable approximation $\langle A_t \rangle$ to A that obeys $\mathbf{c}$. That is, if n_s is least with $A_s(n_s) \neq A_{s+1}(n_s)$, then $\sum_s \mathbf{c}^{\langle A_t \rangle}(n_s, s) < \infty$.

Proof. Uniformly in σ and s, let $C_{\sigma,s} \subset 2^{<\omega}$ be a finite anti-chain that generates $\mathcal{U}_{\sigma,s}$, with $C_{\sigma,s} \subseteq C_{\sigma,s+1}$. Define an oracle machine M with $M^\sigma_s(\pi)\downarrow$ for $\pi \in C_{\sigma,s}$. Since $\mathcal{U}_\sigma$ and $\mathcal{U}_\rho$ are disjoint for $\sigma \not\preceq \rho$, M is prefix-free.

Note that for $s > n$ and any computable approximation $\langle A_t \rangle$,

$$\sum_\pi 2^{-|\pi|} \left[\!\left[M^A(\pi)[s]\downarrow \ \& \ \text{use}\, M^A(\pi)[s] = n+1 \right]\!\right] \geqslant \mathbf{c}^{\langle A_t \rangle}(n, s).$$

Fix a computable approximation $\langle \dot{A}_q \rangle$ to A. By the Main Lemma [25, 5.5.1] derived from the golden run construction, there is a computable sequence $q(0) < q(1) < \cdots$ with $q(0) \geqslant 1$, such that if we define m_s to be least with $\dot{A}_{q(s)} \neq \dot{A}_{q(s+1)}$, then

$$\sum_s \sum_\pi 2^{-|\pi|} \left[\!\left[M^{\dot{A}}(\pi)[q(s)]\downarrow \ \& \ m_s < \text{use}\, M^{\dot{A}}(\pi)[q(s)] \leqslant q(s-1) \right]\!\right] < \infty.$$

Now let $A_t = \dot{A}_{q(t)}$, so $n_s = m_s$. Since $s \leqslant q(s-1)$, if $n_s < s$ then the inner summation above is at least $\mathbf{c}^{\langle A_t \rangle}(n_s, s)$. So $\sum_s \mathbf{c}^{\langle A_t \rangle}(n_s, s) < \infty$, as desired. $\square_{3.2.1}$

Now, let $\langle B_s \rangle$ be a uniformly computable sequence of finite anti-chains with

$$\bigcup_{\tau \in B_s} [\tau] = \mathcal{U}_{A \upharpoonright n_s+1}[s],$$

and define $S = \bigcup_s B_s$. Note that

$$\begin{aligned}\sum_{\tau\in S} 2^{-|\tau|} &\leqslant \sum_s \sum_{\tau\in B_s} 2^{-|\tau|}\\ &= \sum_s \mu(\mathcal{U}_{A\upharpoonright n_s+1}[s])\\ &= \sum_s \mathbf{c}^{\langle A_t\rangle}(n_s, s) < \infty.\end{aligned}$$

Thus S is a Solovay test.

Let X be random and suppose that $A = \Phi_e^X$. Then $Y = 0^e 1\hat{\ } X$ is random and $A = \Upsilon^Y$. Since Y is not captured by S, we can fix an s_0 such that no B_s with $s \geqslant s_0$ contains an initial segment of Y. Fix N such that for all $n \geqslant N$, if $A \upharpoonright n \preccurlyeq \Upsilon_s^Y$, then $s \geqslant s_0$.

Claim 3.2.2. For all $n \geqslant N$, if $A \upharpoonright n \preccurlyeq \Upsilon_s^Y$, then $A_t \upharpoonright n = A \upharpoonright n$ for every $t \geqslant s$.

Proof. Suppose $n \geqslant N$ were a counterexample. Let s be such that $A \upharpoonright n \preccurlyeq \Upsilon_s^Y$ and $t \geqslant s$ be such that $A_t \upharpoonright n \neq A \upharpoonright n$ and $A_{t+1} \upharpoonright n = A \upharpoonright n$. Note that definitionally, $n_t < n$. Since $A_t \upharpoonright n_t = A_{t+1} \upharpoonright n_t$, we know that $A_t \upharpoonright n_t = A \upharpoonright n_t \prec \Upsilon_s^Y$ and $A_t \upharpoonright n_t + 1 \neq A \upharpoonright n_t + 1$. So $(\widehat{A_t \upharpoonright n_t + 1}) = A_{t+1} \upharpoonright n_t + 1$, and $Y \in \mathcal{U}_{A_t \upharpoonright n_t+1}$. By assumption, Y has already entered this Σ^0_1-class by stage s. Since $t \geqslant s$, B_t contains an initial segment of Y, contrary to our choice of N and s_0. $\square_{3.2.2}$

Since for sufficiently large n, convergence of $\Phi_{e,s}^X$ up to n implies convergence of Υ_s^Y up to n, the lemma follows. $\square_{3.2}$

4. For each cost function there is an ML-complete set

In this section, we prove the existence of a smart set as in Definition 2.7 for each cost function $\mathbf{c}$ that implies K-triviality. This yields in Corollary 4.4 the equivalence of smartness for $\mathbf{c}$, and ML-completeness for the class of sets obeying $\mathbf{c}$, and thereby the existence of an ML-complete for the class.

For cost functions $\mathbf{c}$ and $\mathbf{d}$, one writes $\mathbf{c} \to \mathbf{d}$ if $A \models \mathbf{c}$ implies $A \models \mathbf{d}$ for every Δ^0_2 set A. By [27, Thm. 3.4], this is equivalent to $\underline{\mathbf{c}} \geqslant^\times \underline{\mathbf{d}}$, that is, $\underline{\mathbf{c}}$ multiplicatively dominates $\underline{\mathbf{d}}$ (we may assume $\underline{\mathbf{c}}(x) > 0$ for every x). Recall that a set obeys $\mathbf{c}_\Omega$ if and only if it is K-trivial. So all sets obeying a cost function $\mathbf{c}$ are K-trivial if and only if $\mathbf{c} \to \mathbf{c}_\Omega$.

We start with two simple lemmas.

Lemma 4.1. *Suppose that aY fails a $\mathbf{c}$-bounded test $\bigcap_n V_n$, where $a \in \{0, 1\}$. Then Y fails a $\mathbf{c}$-bounded test.*

Proof. We may suppose $a = 0$ and $X \in V_n$ implies $X(0) = 0$. Then $\mu(T[V_n]) = 2\mu(V_n)$, where T is the usual shift operator on Cantor space, and so $\langle T[V_n]\rangle$ is also a $\mathbf{c}$-bounded test. Clearly Y fails it. $\square$

We recall [27] that an *additive* cost function is a cost function of the form $\mathbf{c}_\alpha(n, s) = \alpha_s - \alpha_n$, where $\langle \alpha_s\rangle$ is an increasing approximation of a left-c.e. real α. So $\underline{\mathbf{c}}_\alpha(n) = \alpha - \alpha_n$. Since Ω is Solovay complete among the left-c.e. reals, every time we see an increase in α, we can cause a proportional and later increase in Ω. Thus:

Lemma 4.2. *If $\mathbf{c}_\alpha$ is an additive cost function, then $\mathbf{c}_\Omega \to \mathbf{c}_\alpha$.*

In particular, $2^{-n} \leqslant^{\times} \underline{\mathbf{c}}_{\Omega}(n)$.

Theorem 4.3.
Given a cost function $\mathbf{c}$ *such that* $\mathbf{c} \to \mathbf{c}_{\Omega}$*, one can uniformly obtain a c.e. set* A *which is smart for* $\mathbf{c}$*.*

Proof. Recall that Υ is a "universal" Turing functional in the sense that $\Upsilon^{0^e 1 \hat{} X} = \Phi_e^X$ for all X and e. We build A and a $\mathbf{c}$-test $\langle \mathcal{U}_k \rangle$ capturing any ML-random Y such that $A = \Upsilon^Y$. This suffices for the theorem by Lemma 4.1. The tension in this construction is between trying to capture all reals computing A, and keeping the measure of $\mathcal{U}_n$ bounded by (a multiple of) $\underline{\mathbf{c}}(n)$. The idea is for us to move A in case we see that too many oracles compute it. This needs to be done judiciously; we must ensure that A obeys $\mathbf{c}$. The basic idea, as in [3], is to charge the cost of changing A to the increase in the measure of the *error set*, the set of oracles that have already been proven to be incorrect about A. Since $\mathbf{c} \to \mathbf{c}_{\Omega}$, the increase in the error set is bounded by $\mathbf{c}$, and so we can catch our tail.

We proceed to the details. It will be clear from the proof that the construction is uniform in $\mathbf{c}$.

We apply the usual language for strings: if $\sigma <_L \tau$ we say that σ lies to the left of τ, and τ to the right of σ. By delaying computations from appearing in Υ during the construction, we may assume that for all Y and s, Υ_s^Y does not lie to the right of A_s. We build a global "error set":

$$\mathcal{E}_s = \left\{ Y \,:\, \Upsilon_s^Y \text{ lies to the left of } A_s \right\}.$$

An enumeration of a number into A causes A to move to the right, and so potentially adds elements $\mathcal{E}$; no elements can ever leave $\mathcal{E}$. The basic idea, again, is that we enumerate a number x into A only when the cost $\mathbf{c}(x, s)$ is smaller than the amount by which the measure of $\mathcal{E}$ will be increased.

We will ensure that at every stage s,

$$(\diamond) \qquad \mu(\mathcal{U}_{k,s}) \leqslant \mathbf{c}(k, s) + \mu(\mathcal{E}_{s+1} - \mathcal{E}_k).$$

By Lemma 4.2, $\mu(\mathcal{E} - \mathcal{E}_k) = \underline{\mathbf{c}}_{\mu(\mathcal{E})}(k) \leqslant^{\times} \underline{\mathbf{c}}_{\Omega}(k)$, so as $\underline{\mathbf{c}}_{\Omega}(k) \leqslant^{\times} \underline{\mathbf{c}}(k)$, the test $\langle \mathcal{U}_k \rangle$ is indeed a $\mathbf{c}$-test. We reserve the interval $I_k = [2^k, 2^{k+1})$ for ensuring $(\diamond)$.

The construction of the $\mathbf{c}$-test $\langle \mathcal{U}_k \rangle$ and the c.e. set A is as follows. At stage $s > k$ we let

$$\mathcal{V}_{k,s} = \left\{ Y \,:\, \Upsilon_s^Y \prec A_s \ \&\ \Upsilon_s^Y \upharpoonright 2^{k+1} \text{ is defined} \right\};$$

and

$$\mathcal{U}_{k,s} = \bigcup_{t \in [k,s]} \mathcal{V}_{k,t}.$$

As $\mathcal{V}_{k,s} \supseteq \mathcal{V}_{k+1,s}$, we have $\mathcal{U}_{k,s} \supseteq \mathcal{U}_{k+1,s}$ so $\langle \mathcal{U}_k \rangle$ is nested. Note that $\mathcal{V}_{k,s}$ is disjoint from $\mathcal{E}_k$, for every s, hence $\mathcal{U}_{k,s}$ is disjoint from $\mathcal{E}_k$.

Let $s > k$. We let $x_s = x_s(k) = \min(I_k - A_s)$. If $(\diamond)$ threatens to fail at s, namely $\mu(\mathcal{U}_{k,s}) > \mathbf{c}(k, s) + \mu(\mathcal{E}_s - \mathcal{E}_k)$, we enumerate $x_s(k)$ into A_{s+1}. This causes $\mathcal{U}_{k,s}$ to go into $\mathcal{E}_{s+1}$. Since $\mathcal{U}_{k,s}$ is disjoint from $\mathcal{E}_k$, it follows in this case that $\mu(\mathcal{U}_{k,s}) \leqslant \mu(\mathcal{E}_{s+1} - \mathcal{E}_k)$, and so $(\diamond)$ holds at stage s. (Now the cycle can repeat: during stages $t \geqslant s = 1$, the class $\mathcal{U}_k$ is allowed to add measure up to $\mathbf{c}(k, t)$ without any action necessary. If the measure added exceeds $\mathbf{c}(k, t)$ another enumeration into A will be needed.)

First we verify that x_s always exists, that is, we enumerate at most 2^k times for $\mathcal{U}_k$. By Lemma 4.2, we may assume that $\mathbf{c}(x,s) \geqslant 2^{-x}$ for $x < s$. (To be clear, here we are using the fact that if $\underline{\mathbf{c}} =^\times \underline{\mathbf{d}}$, then the same sets obey $\mathbf{c}$ and $\mathbf{d}$.) If we enumerate $x_s(k)$ into A_{s+1}, then $\mu(\mathcal{U}_{k,s}) > 2^{-k} + \mu(\mathcal{E}_s - \mathcal{E}_k)$. Since $\mathcal{U}_{k,s} \cap \mathcal{E}_k = \varnothing$, and $\mathcal{U}_{k,s} \subseteq \mathcal{E}_{s+1}$, it follows that $\mu(\mathcal{E}_{s+1} - \mathcal{E}_s) > 2^{-k}$. Since $\mu(\mathcal{E}) \leqslant 1$, this can happen at most 2^k times.

Recall that for all Y and s, Υ_s^Y does not lie to the right of A_s. Hence, if $A = \Upsilon^Z$ then $Z \in \bigcap_k \mathcal{U}_k$. It remains to verify that $A \models \mathbf{c}$. If we enumerate $x_s(k)$ into A_{s+1}, then

$$\mu(\mathcal{U}_{k,s} - \mathcal{E}_s) = \mu(\mathcal{U}_{k,s} - (\mathcal{E}_s - \mathcal{E}_k)) \geqslant \mu(\mathcal{U}_{k,s}) - \mu(\mathcal{E}_s - \mathcal{E}_k) > \mathbf{c}(k,s) \geqslant \mathbf{c}(x,s).$$

Since $\mathcal{U}_{k,s} - \mathcal{E}_s \subseteq \mathcal{E}_{s+1} - \mathcal{E}_s$, we see that $\mathbf{c}(x,s) < \mu(\mathcal{E}_{s+1} - \mathcal{E}_s)$. This implies that the total cost of the enumeration of A is at most $\mu(\mathcal{E}) \leqslant 1$. □

ML-completeness for a cost function was defined in 2.4.

Corollary 4.4. *Suppose that $\mathbf{c}$ is a cost function such that $\mathbf{c} \to \mathbf{c}_\Omega$. Let A be a Δ^0_2 set. Then A is smart for $\mathbf{c} \Leftrightarrow A$ is ML-complete for $\mathbf{c}$.*

Proof. ($\Rightarrow$) Suppose that $A \leqslant_{\mathrm{T}} Y$ for ML-random Y. Then some $\mathbf{c}$-bounded test captures Y. If $B \models \mathbf{c}$, then $B \leqslant_{\mathrm{T}} Y$ by the basic fact, Proposition 2.6.
($\Leftarrow$) Let $\widetilde{A}$ be smart for $\mathbf{c}$. If $A \leqslant_{\mathrm{T}} Y$ for ML-random Y, then $\widetilde{A} \leqslant_{\mathrm{T}} Y$, so Y is captured by a $\mathbf{c}$-bounded test. □

In particular, the ML-degree of a smart set A for $\mathbf{c}$ is uniquely determined by $\mathbf{c}$. In contrast, for each low c.e. set A, there is a c.e. set $B \nleqslant_{\mathrm{T}} A$ such that $B \models \mathbf{c}$ [25, 5.3.22]. If A is smart for $\mathbf{c}$, then $A \oplus B$ is also smart for $\mathbf{c}$. As every K-trivial is low, the Turing degree of a set A that is smart for $\mathbf{c}$ is never uniquely determined by $\mathbf{c}$.

5. Each K-trivial set is ML-complete for a cost function

Given a K-trivial set A, we will define a cost function $\mathbf{c}_{(A)}$ with $A \models \mathbf{c}_{(A)}$ such that every random computing A is captured by a $\mathbf{c}_{(A)}$ test. In other words, we build $\mathbf{c}_{(A)}$ in such a way that A is smart for $\mathbf{c}_{(A)}$. Furthermore, in case that A is c.e., $\mathbf{c}_{(A)}$ is the strongest cost function that A obeys, in the sense that if $A \models \mathbf{c}$, then $\mathbf{c}_{(A)} \to \mathbf{c}$. In the introduction we mentioned applications of this to the structure of ML-degree of K-trivials, such as showing that there is no minimal pair.

We will provide an example showing that $\mathbf{c}_{(A)}$ may not behave in an overly nice way. We build a c.e. K-trivial A such that the class of sets obeying $\mathbf{c}_{(A)}$ is not closed downward under $\leqslant_{\mathrm{T}}$. In fact, in our example $T(A) \not\models \mathbf{c}_{(A)}$, where $T(A)$ is the shift of A, obtained by deleting the first bit.

As before, Υ denotes a universal Turing functional. Let A be K-trivial. The idea for defining $\mathbf{c}_{(A)}$ is as follows. Suppose first that A is c.e., and let $\langle A_s \rangle$ be an effective enumeration of A that obeys $\mathbf{c}_\Omega$ [27]. We want to define $\mathbf{c}_{(A)}$ so that we can capture by a $\mathbf{c}_{(A)}$-bounded test all the reals Z such that $\Upsilon^Z = A$. The natural test is $\mathcal{U}_k = \bigcup_{s \geqslant k} \Upsilon_s^{-1}[A_s \upharpoonright k]$. So we define $\underline{\mathbf{c}}_A(k) = \mu(\mathcal{U}_k)$. Why does A obey this cost function? Since the approximation is left-c.e., A does not have to pay for the measure of the oracles that compute $A \upharpoonright k$ correctly: these only appear after $A \upharpoonright k$ has settled, and after that, all changes to A are beyond k. So A only needs to pay for the measure of those reals Z that compute an incorrect version $A_s \upharpoonright k$. This

price is bounded by the increase of the measure of the error set: those oracles that compute some string to the left of A. Thus the total A-cost is the same as the total A-cost of an additive cost function, and hence bounded by the total $\mathbf{c}_\Omega$-cost of this enumeration; but this was chosen to be finite.

When A is not c.e., we use a c.e. intermediary. Let us give the details of the definition. By Theorem 3.1, fix a c.e. set $C \equiv_{\mathrm{ML}} A$ that computes A; let Ψ be a Turing functional such that $A = \Psi^C$. Fix an enumeration $\langle C_s \rangle$ of C and an approximation $\langle A_s \rangle$ of A that witnesses $A \models \mathbf{c}_\Omega$. By speeding up both Ψ and our approximations, we may assume that $A_s \upharpoonright s \preccurlyeq \Psi_s^{C_s}$ for every s. To unify our construction with the earlier discussion, we assume that if A is c.e., then $C = A$ and Ψ is the natural reduction with identity bounded use.

Similarly to what we did above, we let

$$\mathcal{E}_s = \left\{ Y \,:\, \Upsilon_s^Y \text{ lies to the left of } C_s \right\}$$

and

$$\mathcal{V}_{x,s} = \left\{ Y \,:\, \Upsilon_s^Y \prec C_s \ \&\ A_s \upharpoonright x+1 \preccurlyeq \Psi_s^{\Upsilon_s^Y} \right\};$$

we then let

$$\mathbf{c}_{(A)}(x,s) = \mu \left(\bigcup_{x<t\leqslant s} \mathcal{V}_{x,t} \right).$$

Note that $\mathbf{c}_{(A)}$ is monotonic, as $\mathcal{V}_{x,t} \supseteq \mathcal{V}_{x+1,t}$. It satisfies the limit condition if A is non-computable: certainly for all x, $\underline{\mathbf{c}}_A(x) \leqslant 1$. If $A \upharpoonright k$ has stabilised by stage s, then $\underline{\mathbf{c}}_A(s) \leqslant \mu \left\{ Y \,:\, A \upharpoonright k \preccurlyeq \Psi^{\Upsilon^Y} \right\}$. Hence $\lim_x \underline{\mathbf{c}}_A(x) \leqslant \mu \left\{ Y \,:\, A = \Psi^{\Upsilon^Y} \right\}$; if A is non-computable, this is 0.

Proposition 5.1. *A is smart for $\mathbf{c}_{(A)}$.*

Proof. First we show that A obeys $\mathbf{c}_{(A)}$. In fact, the fixed approximation $\langle A_s \rangle$ witnesses this. Define an increasing approximation of the left-c.e. "error real" by

$$\varepsilon_s = \mu \left(\mathcal{E}_{s+1} \right).$$

Suppose that $A_s(x) \neq A_{s+1}(x)$. For each $t \in (x,s]$ and every $Y \in \mathcal{V}_{x,t}$, Υ_t^Y lies to the left of C_{s+1}, and so $Y \in \mathcal{E}_{s+1}$; on the other hand $Y \notin \mathcal{E}_t$ and so $Y \notin \mathcal{E}_{x+1}$. It follows that

$$\mathbf{c}_{(A)}(x,s) \leqslant \mu \left(\mathcal{E}_{s+1} \smallsetminus \mathcal{E}_{x+1} \right) = \mathbf{c}_\varepsilon(x,s).$$

By Lemma 4.2, $\mathbf{c}_\Omega \to \mathbf{c}_\varepsilon$, and so

$$\mathbf{c}_{(A)}\langle A_s \rangle \leqslant \mathbf{c}_\varepsilon \langle A_s \rangle \leqslant^\times \mathbf{c}_\Omega \langle A_s \rangle,$$

and we assumed that the latter is finite.

Next we show that every random real that computes A is captured by some $\mathbf{c}_{(A)}$-bounded test. Since $C \leqslant_{\mathrm{ML}} A$, every such real computes C. By Lemma 4.1, it suffices to build a $\mathbf{c}_{(A)}$-test capturing any random Y such that $C = \Upsilon^Y$. The desired test is the test $\mathcal{U}_k = \bigcup_{s>k} \mathcal{V}_{k,s}$ defined above. (Again, we assume that we delay computations, so for all Y and s, Υ_s^Y does not lie to the right of C_s.) $\square$

As promised, in the case that A is c.e., $\mathbf{c}_{(A)}$ is the strongest cost function that A obeys. In particular, $\mathbf{c}_{(A)} \to \mathbf{c}_\Omega$.

Proposition 5.2. *Suppose that A is c.e. For any cost function $\mathbf{c}$ such that $A \models \mathbf{c}$, we have $\mathbf{c}_{(A)} \to \mathbf{c}$.*

Proof. After multiplying $\mathbf{c}$ by a constant, we may assume that $\underline{\mathbf{c}}(0) < 1/2$. Fix a computable speed-up f such that $\mathbf{c}\langle A_{f(s)}\rangle < 1/2$ (again, see [27]). Define a Turing functional Γ such that at every stage t,

$$\mu\left(\{Y : A_{f(t)} \upharpoonright x+1 \prec \Gamma_t^Y\} - \mathcal{E}_{\Gamma,t}\right) = \mathbf{c}(x,t),$$

where $\mathcal{E}_{\Gamma,t} = \{Y : \Gamma_t^Y \text{ lies to the left of } A_{f(t)}\}$. By a simple argument $\mu(\mathcal{E}_{\Gamma,t}) \leqslant \mathbf{c}\langle A_{f(s)}\rangle < 1/2$ for every t, so this construction may proceed.

Fix e with $\Phi_e = \Gamma$. Then

$$\begin{aligned}
\underline{\mathbf{c}}_A(x) &= \mu\left(\bigcup_{x<t} \mathcal{V}_{x,t}\right) \\
&= \mu\left(\bigcup_{x<t} \left\{Y : A_t \upharpoonright x+1 \preccurlyeq \Upsilon_t^Y\right\}\right) \\
&\geqslant \mu\left\{Y : A \upharpoonright x+1 \preccurlyeq \Upsilon^Y\right\} \\
&\geqslant 2^{-(e+1)} \cdot \mu\left\{Y : A \upharpoonright x+1 \preccurlyeq \Upsilon^{0^e 1^\frown Y}\right\} \\
&\geqslant 2^{-(e+1)} \underline{\mathbf{c}}(x).
\end{aligned}$$

□

We provide the promised applications to the ML-degrees.

Corollary 5.3. *(a) There is no minimal pair in the ML-degrees of K-trivials.*
(b) The ML-degree of a noncomputable K-trivial never contains a maximal Turing degree.

Proof. (a) Given noncomputable K-trivials A, B, let D be a noncomputable set obeying the cost function $\mathbf{c}_{(A)} + \mathbf{c}_{(B)}$. Then $D \leqslant_{\mathrm{ML}} A, B$ by Proposition 2.6.

(b) Suppose A is in the ML-degree. By Theorem 3.1 we may assume that A is c.e. Some c.e. K-trivial $B \nleqslant_{\mathrm{T}} A$ obeys $\mathbf{c}_{(A)}$ by [25, 5.3.22]; then $A \oplus B \equiv_{\mathrm{ML}} A$. □

Recall that $T(A)$ is the shift of A, which is obtained by deleting the first bit.

Theorem 5.4. *For every cost function $\mathbf{d}$ there is a cost function $\mathbf{c} \geqslant \mathbf{d}$ and a c.e. set A such that $A \models \mathbf{c}$ and $T(A) \not\models \mathbf{c}$.*

Since $\mathbf{c}_{(A)} \to \mathbf{c}$, this shows that $T(A) \not\models \mathbf{c}_{(A)}$. Thus $\mathbf{c}_{(T(A))} \not\to \mathbf{c}_{(A)}$. In contrast, for each ML-random Y,

$$Y \text{ fails some } \mathbf{c}_{(A)} \text{ test} \Leftrightarrow Y \geqslant_{\mathrm{T}} A \Leftrightarrow Y \geqslant_{\mathrm{T}} T(A) \Leftrightarrow Y \text{ fails some } \mathbf{c}_{(T(A))}\text{-test}$$

(see Definition 2.5 for $\mathbf{c}$-tests). So we have a pair of inequivalent cost functions that determine the same randomness notion.

Proof. The main idea is to enumerate the set A and the cost function $\mathbf{c}$ so that it has "sudden drops": numbers x with $\underline{\mathbf{c}}(x)$ much smaller than $\underline{\mathbf{c}}(x-1)$.

Let $\langle B_t^0\rangle, \langle B_t^1\rangle, \ldots$ be a listing of all (possibly partial) computable enumerations. In particular, let $\langle D_n\rangle$ be an effective listing of the finite sets, and let $B_{t+1}^e = B_t^e \cup D_{\varphi_e(t+1)}$, where defined.

At a stage s, we may declare $\mathbf{c}(s-1,s) \geqslant \alpha$ for some dyadic rational α, which by monotonicity entails that $\mathbf{c}(y,t) \geqslant \alpha$ for each $y < s$ and $t \geqslant s$. At the end of stage s, we will define $\mathbf{c}(x,s)$ for every $x < s$ to be the least value consistent with all of our declarations and also with $\mathbf{c}(x,s) \geqslant \mathbf{d}(x,s)$.

We must meet the global requirements that $\mathbf{c}$ has the limit condition and that $A \models \mathbf{c}$. We must also meet the requirements

$$R_e\colon\ T(A) = \bigcup_t B^e_t \Rightarrow \mathbf{c}\langle B^e_t\rangle \geqslant 1.$$

The strategy for R_e seeks to find an x and an s where $\mathbf{c}(x-1,s)$ is large and $x-1 \notin B^e_s$. Then it enumerates x into A and waits until it sees a $t > s$ with $x-1 \in B^e_t$. This will increase $\mathbf{c}\langle B^e_t\rangle$ by at least $\mathbf{c}(x-1,s)$. Then the strategy seeks to repeat the process with a new x, continuing until $\mathbf{c}\langle B^e_t\rangle \geqslant 1$.

To ensure that $\mathbf{c}$ has the limit condition, we will give R_e a bound α_e beyond which it is not allowed to increase $\mathbf{c}$. This bound will also ensure that R_e does not interfere with $R_{e'}$ for $e' < e$. To ensure that $A \models \mathbf{c}$, we will not allow R_e to cause enumerations with total cost exceeding 2^{-e}. Other than a discussion of α_e, our full strategy for R_e is:

(1) Let s be the current stage. Declare $\mathbf{c}(s-1,s) \geqslant \alpha_e$.
(2) At stage $s+1$, declare $\mathbf{c}(s,s+1) \geqslant 2^{-e}\cdot\alpha_e$.
(3) Wait for a stage $u > s$ when one of the following happens:
  (a) If $\mathbf{c}(s,u) > 2^{-e}\cdot\alpha_e$, return to Step (1).
  (b) If s is enumerated into A, return to Step (1).
  (c) If B^e_s converges with $s-1 \notin B^e_s$, enumerate s into A and proceed to Step (4).
(4) Wait until B^e_r converges for some $r > s$ with $s-1 \in B^e_r$.
(5) If $\mathbf{c}\langle B^e_t\rangle^r_{t=0} \geqslant 1$, terminate the strategy. Otherwise, return to Step (1).

Note that case (3a) might occur because of the actions of some other strategy, or might instead occur because of $\mathbf{c}(s,u) \geqslant \mathbf{d}(s,u)$. The latter can occur only finitely many times, because $\mathbf{d}$ satisfies the limit condition.

Note also that if we reach Step (5), then $s-1 \notin B^e_s$, $s-1 \in B^e_r$, and $\mathbf{c}(s-1,s) \geqslant \alpha_e$, so $\mathbf{c}\langle B^e_t\rangle^r_{t=0} - \mathbf{c}\langle B^e_t\rangle^s_{t=0} \geqslant \alpha_e$. Thus we will reach Step (5) at most $1/\alpha_e$ times before meeting the requirement and terminating the strategy. Each enumeration has a cost of $2^{-e}\cdot\alpha_e$ by construction, and so the total cost of enumerations by this strategy is at most 2^{-e}.

If the strategy waits forever at Step (3), then either $\langle B^e_t\rangle$ is partial, or $s \notin A$ but $s-1 \in \bigcup_t B^e_t$, meaning we satisfy R_e by negating the hypothesis. It thus remains only to show that we do not return to Step (1) via case (3a) or (3b) infinitely many times.

We wish to ensure that no $R_{e'}$-strategy for $e' > e$ can increase $\mathbf{c}(s,u)$ beyond $2^{-e}\cdot\alpha_e$. So we define $\alpha_0 = 1$, $\alpha_{e+1} = 2^{-e}\cdot\alpha_e$. Now case (3a) cannot be caused by the action of any $R_{e'}$-strategy for $e' > e$. Nor can case (3b), because of our action at Step (2). It is then a simple induction that no strategy returns to Step (1) more than finitely many times. □

6. K-trivial sets Turing below fragments of Ω

Previous research. We begin by discussing in some detail a theorem that motivated the present results. For a set $Z \subseteq \mathbb{N}$ thought of as a bit sequence and an infinite set $R \subseteq \mathbb{N}$, we denote by Z_R the sequence obtained by erasing the bits of Z in locations outside of R. If $1 \leqslant j \leqslant n$, then the jth *n-column* of Z is $Z_{(j-1+n\mathbb{N})}$. A set A is a *k/n-base* if it is computable from the join of any k of the n-columns of some random sequence X, in all possible ways.

For a computable real p such that $0 < p \leqslant 1$, let $\mathbf{c}_{\Omega,p}(x,s) = (\Omega_s - \Omega_x)^p$. As mentioned in the introduction, three of the authors of the present paper proved:

Theorem 6.1 ([14]). *The following are equivalent for a set A and $1 \leqslant k < n$:*

(1) *A is a k/n-base.*
(2) *A is a k/n-base witnessed by Ω, i.e., it is computable from the join of* any *k of the n-columns of Ω.*
(3) *A obeys $\mathbf{c}_{\Omega,k/n}$.*

Hence the p-bases are characterised by cost functions. Theorem 4.3 implies that, for every rational $p \in (0,1)$, there is a smart p-base: a greatest ML-degree of p-bases. If $p < q$, then every p-base is also a q-base, as $\mathbf{c}_{\Omega,p} \geqslant^{\times} \mathbf{c}_{\Omega,q}$. However there also is a q-base that is not a p-base. Thus, the smart p-bases form a dense chain of ML-degrees.

Using Theorem 6.6 below, we can add fourth equivalence to Theorem 6.1, one that appears to be significantly weaker than (2):

(4) *A is K-trivial and is computable from the join of* some *choice of k of the n-columns of Ω.*

In other words: if a K-trivial is computable from *some* k/n-fragment of Ω, then it is computable from *any* k/n-fragment of Ω. Recall that any c.e. set computable from a Turing incomplete random set is K-trivial [18]. Since every k/n-fragment of Ω is incomplete, we obtain:

Corollary 6.2. *If X and Y are both k/n-fragments of Ω for $k < n$, then X and Y compute the same c.e. sets.*

In particular, if a c.e. set is computable from one half of Ω, it is also computable from the other half.

The cost functions $\mathbf{c}_R$. We now turn to the general analysis of the question which K-trivials are computed by fragments of Ω. For $n \geqslant 1$ and $T \subseteq \{1,2,\dots,n\}$, let $R(T,n) = \bigcup_{j\in T} j - 1 + n\mathbb{N}$. So $\Omega_{R(T,n)}$ is the join of the n-columns of Ω indexed by T (up to a simple computable permutation, depending on how we take the join).

Let R be an infinite computable set. The first question is how to generalise the cost function $\mathbf{c}_{\Omega,k/n}$ to a cost function $\mathbf{c}_{\Omega,R}$. A basic step in the analysis of k/n-bases was the observation that if $T \subseteq \{1,2,\dots,n\}$ has size k, then $\Omega_{R(T,n)}$ is captured by a $\mathbf{c}_{\Omega,k/n}$-test; this gave the implication (3)→(2) of Theorem 6.1. We would like to capture the bits of Ω_R that are given by $\Omega \upharpoonright n$ by the nth component of a $\underline{\mathbf{c}}_{\Omega,R}$-test. So perhaps the first guess would be to define $\underline{\mathbf{c}}_{\Omega,R}(n) = (\Omega - \Omega_n)^{|R\cap n|/n}$. It turns out that this is not quite right; it works if $R = R(T,n)$, but that is misleading because in that case the density of initial segments of R is more or less constant k/n. What would work is $\underline{\mathbf{c}}_{\Omega,R}(n) = (\Omega - \Omega_n)^{|R\cap k(n)|/k(n)}$, where $\Omega - \Omega_n \in (2^{-k(n)-1}, 2^{-k(n)}]$. However, it is not clear that this cost function will be monotonic if the density of R varies. We get around this technical complication by using a "discrete" version.

For $n < \omega$, let

$$k(n) = \lfloor -\log_2(\Omega - \Omega_n) \rfloor,$$

so $2^{-k(n)-1} < \Omega - \Omega_n \leqslant 2^{-k(n)}$. Define $k_s(n)$ similarly, replacing Ω by Ω_s. Note that $k(n) \leqslant n$ for all but finitely many n (otherwise, Ω would not be random).

Definition 6.3. For an infinite computable $R \subseteq \omega$, define

$$\mathbf{c}_{\Omega,R}(n,s) = 2^{-|R\cap k_s(n)|}.$$

The cost function $\mathbf{c}_{\Omega,R}$ is monotonic: $k_{s+1}(n) \leqslant k_s(n)$ and $k_s(n+1) \geqslant k_s(n)$. It also satisfies the limit condition: $\underline{\mathbf{c}}_{\Omega,R}(n) = 2^{-|R\cap k(n)|}$ is finite and, since $\lim_n k(n) = \infty$ and R is infinite, $\lim_n \underline{\mathbf{c}}_{\Omega,R}(n) = 0$. Finally, we note that $\mathbf{c}_{\Omega,R}(n,s) \cdot \mathbf{c}_{\Omega,R^{\complement}}(n,s) = 2^{-k_s(n)} =^{\times} \Omega_s - \Omega_n$, where $R^{\complement}$ is the complement of R.

Remark 6.4. For any infinite R, note that $\underline{\mathbf{c}}_{\Omega,R}(n) =^{\times} (\Omega - \Omega_n)^{|R\cap k(n)|/k(n)}$. For,

$$\left(2^{-k(n)-1}\right)^{|R\cap k(n)|/k(n)} < (\Omega - \Omega_n)^{|R\cap k(n)|/k(n)} \leqslant \left(2^{-k(n)}\right)^{|R\cap k(n)|/k(n)}$$
$$2^{-|R\cap k(n)|} \cdot 2^{-|R\cap k(n)|/k(n)} < (\Omega - \Omega_n)^{|R\cap k(n)|/k(n)} \leqslant 2^{-|R\cap k(n)|}$$
$$\frac{1}{2} \cdot \underline{\mathbf{c}}_{\Omega,R}(n) < (\Omega - \Omega_n)^{|R\cap k(n)|/k(n)} \leqslant \underline{\mathbf{c}}_{\Omega,R}(n).$$

In particular, if $T \subseteq \{1,2,\ldots,n\}$ has size k, then $\underline{\mathbf{c}}_{\Omega,R(T,n)} =^{\times} \underline{\mathbf{c}}_{\Omega,k/n}$.

Proposition 6.5. *Ω_R is captured by a $\mathbf{c}_{\Omega,R}$-test.*

Proof. We use the idea from the proof of [14, Prop. 2.9] that each p-Oberwolfach test is covered by a $\mathbf{c}_{\Omega,p}$-tests.

For a string $\sigma \in 2^{<\omega}$ of length t, let σ_R denote the string of length $|R \cap t|$ given by the bits of σ with location in R. Let $G_\sigma = \{X \in 2^\omega : \sigma_R \prec X\}$. Clearly $\mu(G_\sigma) = 2^{-|R\cap t|}$.

Let $U_n = \bigcup_{s\geqslant n} G_{\Omega_s \upharpoonright n}$. Then $\Omega_R \in \bigcap_n U_n$. Note also that $U_{n+1} \subseteq U_n$. Since $\Omega - \Omega_n \leqslant 2^{-k(n)}$, the set $\{\Omega_s \upharpoonright k(n) : s \geqslant n\}$ contains at most two strings. But $U_n \subseteq \bigcup_{s\geqslant n} G_{\Omega_s \upharpoonright k(n)}$, so

$$\mu(U_n) \leqslant 2 \cdot 2^{-|R\cap k(n)|} = 2 \cdot \underline{\mathbf{c}}_{\Omega,R}(n). \qquad \square$$

We proceed to the main theorem of this section. The equivalence (i)↔(iii) provides a simple combinatorial description of when a fragment Ω_S computes no more K-trivials than another fragment Ω_R: for each number m, the size of S below m exceeds the size of R below m by at most a constant.

Theorem 6.6. *The following are equivalent for infinite computable sets R and S:*

(i) *Each K-trivial computed by Ω_S is also computed by Ω_R.*
(ii) *Ω_R is captured by a $\mathbf{c}_{\Omega,S}$-test.*
(iii) $|S \cap m| \leqslant^{+} |R \cap m|$.
(iv) $\mathbf{c}_{\Omega,S} \to \mathbf{c}_{\Omega,R}$.

The promised application (from the beginning of the section) of the main result follows easily: if $T, T' \subseteq \{1,2,\ldots,n\}$ have size k, then $|R(T,n) \cap m| =^{+} |R(T',n) \cap m|$. Therefore, any k-trivial computable from $\Omega_{R(T,n)}$ is computable from every k/n-fragment of Ω, hence is a k/n-base.

Remark 6.7 (The dual of ML-reducibility). The relative complexity of fragments of Ω can be understood in terms of the dual of ML-reducibility. For general ML-random sequences Y and Z, we write $Y \leqslant_{\mathrm{ML}*} Z$ if for every K-trivial set A, if $A \leqslant_{\mathrm{T}} Y$ then $A \leqslant_{\mathrm{T}} Z$.

Again Turing reducibility implies ML*-reducibility. The top degree consists of those randoms that compute all K-trivial sets; these are the randoms that fail some $\mathbf{c}_\Omega$-test (i.e., the non-*Oberwolfach randoms* [3]). Of course, these include all the Turing complete randoms. The bottom degree consists of the weakly 2-random sequences, the randoms that compute no K-trivial sets.

Using this language, (i) in Theorem 6.6 states that $\Omega_S \leqslant_{\mathrm{ML}*} \Omega_R$. The equivalence (i)↔(iii) in 6.6 provides a complete characterisation of ML*-reducibility between fragments of Ω by a simple combinatorial condition on the underlying computable sets. The intuition is that as R gets thinner, Ω_R gets computationally weaker (in the coarse sense of ML*). The randomness enhancement principle says that among ML-random sets, being computationally weaker is equivalent to being more random (see [26] and the discussion there). By this principle, Ω_R also gets more random as R gets thinner.

Due to the relative length of the proof, we state and prove the implication (ii)→(iii) of Theorem 6.6 separately.

Proposition 6.8.
Let R and S be infinite computable sets such that $|S \cap m| \nleqslant^+ |R \cap m|$, i.e., the function $m \mapsto |S \cap m| - |R \cap m|$ is unbounded. Then Ω_R is $\mathbf{c}_{\Omega,S}$-random.

Proof. Suppose for a contradiction that Ω_R can be captured by a $\mathbf{c}_{\Omega,S}$-test. Then using Proposition 6.5 there is a nested test $\langle \mathcal{U}_n \rangle$ capturing the sequence $Z = \Omega_R \oplus \Omega_{R^{\complement}}$ with $\mu(\mathcal{U}_n) \leqslant (\underline{\mathbf{c}}_{\Omega,S}(n)) \cdot (\underline{\mathbf{c}}_{\Omega,R^{\complement}}(n))$. We show that this implies that Z is not ML-random. To do this, we show how to uniformly enumerate an open set $\mathcal{V}$ of small measure that contains Z.

Note that $|S \cap m| - |R \cap m| = |S \cap m| + |R^{\complement} \cap m| - m$. Given a rational $\varepsilon > 0$, we can thus effectively find a k with $|S \cap k| + |R^{\complement} \cap k| - k > 1 - \log \varepsilon$.

Define a location $n_s > k$ recursively at stages $s \geqslant k$. Recall that $k_s(n) = \lfloor -\log_2(\Omega_s - \Omega_n) \rfloor$. For $s = k$, or if $k_{s+1}(n_s) < k$, we let n_{s+1} be the least $n \geqslant k$ such that $k_{s+1}(n) > k$. Otherwise, we let $n_{s+1} = n_s$.

There are at most 2^{k+1} stages s at which $n_s \neq n_{s-1}$. For let $s < t$ be two such stages, then $\Omega_s - \Omega_{n_s} \leqslant 2^{-(k+1)}$. But $\Omega_t - \Omega_{n_s} > 2^{-k}$, so $\Omega_t - \Omega_s > 2^{-(k+1)}$.

Let $\mathcal{V} = \bigcup_{s>k} \mathcal{U}_{n_s,s}$. For each stage $s \geqslant k$, we have $k_s(n_s) \geqslant k$, so

$$\mu(\mathcal{U}_{n_s,s}) \leqslant (\mathbf{c}_{\Omega,S}(n_s,s)) \cdot (\mathbf{c}_{\Omega,R^{\complement}}(n_s,s)) = 2^{-|S \cap k_s(n_s)|} \cdot 2^{-|R^{\complement} \cap k_s(n_s)|} \leqslant 2^{-|S \cap k|} \cdot 2^{-|R^{\complement} \cap k|} = 2^{-(|S \cap k| + |R^{\complement} \cap k|)} < 2^{\log \varepsilon - k - 1} = 2^{-k-1} \cdot \varepsilon.$$

Therefore,

$$\mu(\mathcal{V}) \leqslant 2^{k+1} \cdot 2^{-k-1} \cdot \varepsilon = \varepsilon. \qquad \square$$

The hardest implication is (iv)→(i). To prove it, we rely on a lemma of interest on its own. Informally, the lemma says that if $X \oplus Y$ is ML-random, but X is not too random in the sense that X fails a $\mathbf{c}_{\Omega,R^{\complement}}$-test for a co-infinite computable set R, then any K-trivial Turing below the "other side" Y obeys the complementary cost function $\mathbf{c}_{\Omega,R}$. For example, let $X = \Omega_{R^{\complement}}$ and $Y = \Omega_R$, so that in addition Y fails a $\mathbf{c}_{\Omega,R}$-test. In this case, any K-trivial set A obeying $\mathbf{c}_{\Omega,R}$ is below Y; the lemma says that these are the only K-trivials below Y. (See Theorem 7.1.)

Lemma 6.9. *Let $R \subseteq \omega$ be computable and co-infinite. Suppose that $X \oplus Y$ is ML-random, and that X is captured by a $\mathbf{c}_{\Omega,R^{\complement}}$-test. Suppose that A is K-trivial, and that $A \leqslant_{\mathrm{T}} Y$. Then A obeys $\mathbf{c}_{\Omega,R}$.*

The proof will be the content of Section 8.

Remark 6.10. While $\mathbf{c}_{\Omega,R}$ was only defined for infinite R, the definition can be interpreted for finite R, in which case the cost function does not satisfy the limit

condition, and the sets obeying it will be the computable ones. Lemma 6.9 holds for R finite or co-finite as well. The case $|R^{\complement}| < \infty$ tells us nothing: the hypothesis that X fails a $\mathbf{c}_{\Omega,R^{\complement}}$-test is trivial, since there is such a test that captures the entire interval; meanwhile, $\mathbf{c}_{\Omega,R} =^{\times} \mathbf{c}_{\Omega}$, so the conclusion $A \models \mathbf{c}_{\Omega,R}$ is simply a restatement of the fact that A is K-trivial.

The case $|R| < \infty$ is a weaker version of a known result: the assumption that X fails a $\mathbf{c}_{\Omega,R^{\complement}}$-test tells us that X is not $\mathbf{c}_{\Omega}$-random, and thus that X is LR-hard [3, Thm. 1.5]. Since Y is X-random, Y is 2-random, and so the only K-trivials that Y computes are the computable sets.

Proof of Theorem 6.6, assuming Lemma 6.9. (i)→(ii) Let A be smart for $\mathbf{c}_{\Omega,S}$ by Theorem 4.3. Thus $A \models \mathbf{c}_{\Omega,S}$. Note that Ω_S is captured by a $\mathbf{c}_{\Omega,S}$-test by Proposition 6.5. By Proposition 2.6, $A \leqslant_{\mathrm{T}} \Omega_S$, and so $A \leqslant_{\mathrm{T}} \Omega_R$. By the definition of smartness for cost functions, Ω_R is captured by a $\mathbf{c}_{\Omega,S}$-test.

(ii)→(iii) This is the contrapositive of Proposition 6.8.

(iii)→(iv) Fix b such that $|S \cap m| \leqslant |R \cap m| + b$ for all m. Then $|S \cap k(n)| \leqslant |R \cap k(n)| + b$ for all n, meaning that $\underline{\mathbf{c}}_{\Omega,R} \leqslant 2^b \underline{\mathbf{c}}_{\Omega,S}$, and so $\mathbf{c}_{\Omega,S} \to \mathbf{c}_{\Omega,R}$.

(iv)→(i) Suppose $\mathbf{c}_{\Omega,S} \to \mathbf{c}_{\Omega,R}$, or equivalently $\underline{\mathbf{c}}_{\Omega,R} \leqslant^{\times} \underline{\mathbf{c}}_{\Omega,S}$. Since

$$\underline{\mathbf{c}}_{\Omega,R} \cdot \underline{\mathbf{c}}_{\Omega,R^{\complement}} =^{\times} \underline{\mathbf{c}}_{\Omega} =^{\times} \underline{\mathbf{c}}_{\Omega,S} \cdot \underline{\mathbf{c}}_{\Omega,S^{\complement}},$$

it follows that $\underline{\mathbf{c}}_{\Omega,S^{\complement}} \leqslant^{\times} \underline{\mathbf{c}}_{\Omega,R^{\complement}}$. By Proposition 6.5, $X := \Omega_{S^{\complement}}$ is captured by a $\mathbf{c}_{\Omega,S^{\complement}}$-test, which hence is a $\mathbf{c}_{\Omega,R^{\complement}}$-test. By Lemma 6.9, for any K-trivial $A \leqslant_{\mathrm{T}} \Omega_S := Y$, $A \models \mathbf{c}_{\Omega,R}$. Thus $A \leqslant_{\mathrm{T}} \Omega_R$ by Proposition 2.6. $\square_{6.6}$

7. Feeble sets for cost functions, and the structure of ML-degrees

We discuss some ramifications of Theorem 6.6, and also Lemma 6.9, which is a main technical ingredient to the proof of the theorem.

A notion that is dual to smartness for a cost function. We will call an ML-random failing a $\mathbf{c}$-test *feeble for* $\mathbf{c}$ if the only K-trivials it computes are the ones that obey $\mathbf{c}$. Recall that a K-trivial set obeying a cost function $\mathbf{c}$ is smart for $\mathbf{c}$ if only the ML-random sets that fail a $\mathbf{c}$-test compute it (Def. 2.7). Feebleness for $\mathbf{c}$ is dual to smartness for $\mathbf{c}$: in each case, by Proposition 2.6 the definition says that the collection of sets of the "opposite" type that are Turing comparable to the given set is as small as possible.

As a consequence of Lemma 6.9, we obtain a natural characterisation of the K-trivial sets that are Turing below Ω_R for some infinite computable set R.

Theorem 7.1. *If R is an infinite computable set, then Ω_R is feeble for $\mathbf{c}_{\Omega,R}$. (So the K-trivials computable from Ω_R are exactly those that obey $\mathbf{c}_{\Omega,R}$.)*

Proof. By Proposition 6.5, Ω_R fails a $\mathbf{c}_{\Omega,R}$-test.

Suppose $A \leqslant_{\mathrm{T}} \Omega_R := Y$ and A is K-trivial. The sequence $\Omega_R \oplus \Omega_{R^{\complement}}$ is ML-random and $X := \Omega_{R^{\complement}}$ fails a $\mathbf{c}_{\Omega,R^{\complement}}$-test, so $A \models \mathbf{c}_{\Omega,R}$ by Lemma 6.9. $\square$

We make two observations that follow from the definitions of smartness and feebleness via Proposition 2.6. They tell us that cost functions that admit feeble sequences are special. The first observation implies that if $\mathbf{c}$ has a feeble random sequence, then the collection of sets that obey $\mathbf{c}$ determines a principal ideal of ML-degrees.

Proposition 7.2.
Let $\mathbf{c}$ *be a cost function such that* $\mathbf{c} \to \mathbf{c}_\Omega$*. Suppose that* Z *is feeble for* $\mathbf{c}$ *and that* B *is smart for* $\mathbf{c}$*. Then the following are equivalent for a* K*-trivial set* A*:*

(1) $A \models \mathbf{c}$*;*
(2) $A \leqslant_{\mathrm{T}} Z$*;*
(3) $A \leqslant_{\mathrm{ML}} B$*.*

Thus, for example, no random can be feeble for the cost function $\mathbf{c}_{(A)}$ for the set A built in the proof of Theorem 5.4.

The next observation says informally that for cost functions admitting feeble sequences, the following are all equivalent: cost function implication, ML-reducibility between corresponding smart sets, and ML* reducibility between corresponding feeble sets in the sense of Remark 6.7.

Proposition 7.3. *Let* $\mathbf{c}$ *and* $\mathbf{d}$ *be cost functions such that* $\mathbf{c} \to \mathbf{c}_\Omega$ *and* $\mathbf{d} \to \mathbf{c}_\Omega$*. Suppose that* $Z_{\mathbf{c}}$ *and* $Z_{\mathbf{d}}$ *are feeble for* $\mathbf{c}$ *and* $\mathbf{d}$ *(respectively), and that* $B_{\mathbf{c}}$ *and* $B_{\mathbf{d}}$ *are smart for* $\mathbf{c}$ *and* $\mathbf{d}$ *(respectively). The following are equivalent:*

(1) $\mathbf{c} \to \mathbf{d}$*;*
(2) $B_{\mathbf{c}} \leqslant_{\mathrm{ML}} B_{\mathbf{d}}$*;*
(3) *Each* K*-trivial Turing below* $Z_{\mathbf{c}}$ *is Turing below* $Z_{\mathbf{d}}$*, i.e.* $Z_{\mathbf{c}} \leqslant_{\mathrm{ML}*} Z_{\mathbf{d}}$*.*

Proof. We obtain from Proposition 7.2 that (2) is equivalent to (3):

$$B_{\mathbf{c}} \leqslant_{\mathrm{ML}} B_{\mathbf{d}} \text{ iff } \forall K\text{-triv. } A\,[A \leqslant_{\mathrm{ML}} B_{\mathbf{c}} \to A \leqslant_{\mathrm{ML}} B_{\mathbf{d}}] \text{ iff } Z_{\mathbf{c}} \leqslant_{\mathrm{ML}*} Z_{\mathbf{d}}.$$

For (1)→(2), recall from Section 2 that (1) is equivalent to $\underline{\mathbf{d}} \leqslant^{\times} \underline{\mathbf{c}}$, and hence implies that every $\mathbf{d}$-test is a $\mathbf{c}$-test. If $B_{\mathbf{d}} \leqslant_{\mathrm{T}} Y$ for ML-random Y then Y fails a $\mathbf{c}$-test, and hence $B_{\mathbf{c}} \leqslant_{\mathrm{T}} Y$.

For (2)→(1), recall that $B_{\mathbf{c}}$ is ML-complete for $\mathbf{c}$. So if A obeys $\mathbf{c}$ then $A \leqslant_{\mathrm{ML}} B_{\mathbf{d}}$. By Proposition 7.2 applied to $\mathbf{d}$, this implies that A obeys $\mathbf{d}$. □

The structure of the ML-degrees of K*-trivials.* In Corollary 5.3, we provided two facts on the ML-degrees. Both were based on Proposition 5.1 that each K-trivial A is ML-complete for a cost functions $\mathbf{c}_{(A)}$. Here we obtain a further structural result using the tools developed above (in this section and the preceding section). In contrast to what the results in [14] suggested, our result shows that the partial order of ML-degrees is far from linear.

Theorem 7.4.
There is an infinite antichain of ML-degrees of uniformly c.e. K*-trivial sets. Furthermore, every countable partial ordering is embeddable into the ML-degrees of* K*-trivial sets, and via a computable embedding if the given ordering is computable.*

Proof. We fix a uniformly computable partition of $\mathbb{N}$ into countably many sets R_n such that the upper density of each R_n is 1 (greater than 1/2 would do). For a computable set $F \subseteq \mathbb{N}$, let $R(F) = \bigcup_{n \in F} R_n$. By Theorem 4.3 let B_F be the c.e. K-trivial set, uniformly obtained from the cost function, that is smart for $\mathbf{c}_{\Omega, R(F)}$. The required uniformly c.e. antichain is $\langle B_{\{n\}} \rangle$. The map $F \mapsto B_F$ is an embedding of the partial ordering of computable sets under inclusion into the ML-degrees. This suffices, because one can embed the countable atomless Boolean algebra into the algebra of computable sets under inclusion using its representation as the interval algebra of the countable dense linear order $[0, 1)_{\mathbb{Q}}$.

To see that the map is a partial order embedding, first suppose that $F \subseteq G$. Then $R(F) \subseteq R(G)$, and so $\mathbf{c}_{\Omega,R(F)} \geqslant \mathbf{c}_{\Omega,R(G)}$; by Proposition 7.3, $B_F \leqslant_{\mathrm{ML}} B_G$.

On the other hand, if $F \nsubseteq G$, take some $n \in F \smallsetminus G$; so $R_n \subseteq R(F)$ but $R_n \cap R(G) = \varnothing$. The fact that the upper density of R_n is 1 implies that $|R(F) \cap m| \nleqslant^+ |R(G) \cap m|$. By Theorem 6.6, this implies that $\mathbf{c}_{\Omega,R(F)} \not\to \mathbf{c}_{\Omega,R(G)}$. Hence $B_F \nleqslant_{\mathrm{ML}} B_G$ by Theorem 7.1 and Proposition 7.3. □

We obtain a related structural result about the ML-degrees without using the tools developed above. We give an alternative construction of incomparable ML-degrees, and use it to prove downward density.

Theorem 7.5. *For every non-computable c.e. set D, there are c.e. sets $A, B \leqslant_{\mathrm{T}} D$ such that $A \mid_{\mathrm{ML}} B$.*

Proof. We extend Kučera's injury-free proof [20] of the Friedberg–Muchnik theorem, as presented in [25, Section 4.2]. The theorem states that there are Turing incomparable c.e. sets A, B. Two versions of Kučera's proof are given there; the first relies on $\{0,1\}$-valued d.n.c. functions as in [25, Cor. 4.2.3], the second on ML-randomness as in [25, Cor. 4.2.5]. The second version actually shows that there are ML-random Δ^0_2 sets Y, Z such that $A \leqslant_{\mathrm{T}} Y$, $B \leqslant_{\mathrm{T}} Z$, $A \nleqslant_{\mathrm{T}} Z$, and $B \nleqslant_{\mathrm{T}} Y$. Therefore $A \mid_{\mathrm{ML}} B$ as witnessed by Y, Z.

To ensure that $A, B \leqslant_{\mathrm{T}} D$, all we need to do is modify [25, Cor. 4.2.5]:

Lemma 7.6. *There is a computable function r such that for each e, if $Y = \Phi_e^{\varnothing'}$ is total and ML-random, then $A = W_{r(e)} \leqslant_{\mathrm{wtt}} Y$, $A \leqslant_{\mathrm{T}} D$, and A is non-computable.*

To see this, we use the cost function version of Kucera's result as presented in [15] and [25, 5.3.13]. Given an ML-random Δ^0_2 set Y, one defines a cost function c_Y such that if $A \models c_Y$, then $A \leqslant_{\mathrm{wtt}} Y$. The cost function c_Y emulates a given computable approximation of Y, and is therefore obtained uniformly from an e such that $Y = \Phi_e^{\varnothing'}$. The construction of a non-computable c.e. set A obeying a given cost function with the limit condition [27, Thm. 2.7(i)] is compatible with simple permitting, so we can ensure that $A \leqslant_{\mathrm{T}} D$. It is also uniform in the cost function (when D is fixed). So we obtain the c.e. set A uniformly in e, as required. □

Remark 7.7. The following may be relevant towards Question 1.2 above. We distinguish $\leqslant_{\mathrm{ML}}$ from certain variants that are clearly arithmetical. Fix a notation η for an infinite computable ordinal. Intuitively, a Δ^0_2 set is η-c.a. if it has a computable approximation where the number of changes is bounded by counting downward in the canonical computable well-order given by η. See e.g. [12, Def. 7.1] for the formal definition of η-c.a. sets and more background.

Restricting the ML-randoms in the definition 2.1 of $\leqslant_{\mathrm{ML}}$ to the η-c.a. sets yields a reducibility strictly weaker than $\leqslant_{\mathrm{ML}}$. For, the η-c.a. sets form a Σ^0_3 class, so there is a noncomputable c.e. set D below all the η-c.a. ML-randoms. Now by Theorem 7.5, let $A, B \leqslant_{\mathrm{T}} D$ be c.e. sets such that $A \mid_{\mathrm{ML}} B$. Then A and B are equivalent in the sense of the reducibility based on η-c.a. sets. Note that by the proof of Theorem 7.5, in fact A and B are incomparable for the weaker variant of ML-reducibility based on ML-random Δ^0_2 sets.

8. Proof of Lemma 6.9

Recall that for $\mathcal{A} \subseteq 2^\omega$ and $Z \in 2^\omega$, the Lebesgue (binary) lower density $\varrho_2(\mathcal{A}|Z)$ of $\mathcal{A}$ at Z is $\liminf_n \mu(\mathcal{A}|Z \upharpoonright n)$, where $\mu(\mathcal{A}|\sigma) = \mu(\mathcal{A} \cap [\sigma])/\mu([\sigma])$ is the conditional probability of $\mathcal{A}$ given $[\sigma]$. Notice that $\varrho_2(2^\omega \times \mathcal{A}|X \oplus Z) = \varrho_2(\mathcal{A}|Z)$ for any bit sequence X.

A difference test is one of the form $\langle \mathcal{U}_n \cap \mathcal{P} \rangle$, where the open sets $\mathcal{U}_n$ are uniformly Σ^0_1 and nested, $\mathcal{P}$ is Π^0_1, and $\mu(\mathcal{U}_n \cap \mathcal{P}) \leqslant 2^{-n}$. Franklin and Ng [10] proved that an ML-random sequence Z is difference random (i.e., passes all difference tests) if and only if Z is Turing incomplete.

A bit sequence Z is a *positive density point* if the lower density $\underline{\varrho_2}(\mathcal{P}|Z)$ is positive for any Π^0_1 class $\mathcal{P}$ that contains Z (If Z is ML-random, then it makes no difference whether one takes the binary, or the full density defined in the setting of the unit interval.)

We also require a result from [4] due to Bienvenu, Hölzl and two of the authors of the present paper. It implies that an ML-random is difference random if and only if it is a positive density point. Furthermore, the failure of these properties will be witnessed on the same Π^0_1 classes, an observation we will use below.

Fact 8.1 ([4], Lemma 3.3)**.** *Suppose that $\mathcal{Q}$ is a Π^0_1-class that contains an ML-random sequence Z. Then Z fails a difference test of the form $\langle \mathcal{V}_n \cap \mathcal{Q} \rangle$ iff $\mathcal{Q}$ has lower density 0 at Z.*

The purpose of this section is to prove Lemma 6.9, which we recall here. We will first give a proof in the case that A is c.e.

Lemma 6.9. *Let $R \subseteq \omega$ be computable and co-infinite. Suppose that $X \oplus Y$ is ML-random, and that X is captured by a $\mathbf{c}_{\Omega,R^{\complement}}$-test. Suppose that A is K-trivial, and that $A \leqslant_{\mathrm{T}} Y$. Then A obeys $\mathbf{c}_{\Omega,R}$.*

Proof when A is c.e. Fix a computable enumeration $\langle A_s \rangle$ of A. Fix a $\mathbf{c}_{\Omega,R^{\complement}}$-test $\langle \mathcal{U}_n \rangle$ that X fails, and fix a functional Φ with $A = \Phi^Y$. Let $\mathcal{E}$ be the error set for Φ with respect to A: as before, $\mathcal{E}_s$ is the set of oracles Z such that Φ^Z_s lies to the left of A_s. Let $\mathcal{Q} = 2^\omega \times (2^\omega - \mathcal{E})$ (and $\mathcal{Q}_s = 2^\omega \times (2^\omega - \mathcal{E}_s)$).

We carry out a "ravenous sets" construction on $\mathcal{Q}$; for background see [14, Section 3.1]. Uniformly in $k, n \in \omega$, we enumerate Σ^0_1 open sets $\mathcal{V}^k_n \subset 2^\omega \times 2^\omega$. The *goal* for $\mathcal{V}^k_n \cap \mathcal{Q}$ is $2^{-k}(\Omega_{n+1} - \Omega_n)$; we will ensure that no set ever exceeds its goal. In [18] a set playing a role similar to the one of $\mathcal{V}^k_n$ was called "hungry" if it has not reached its goal. The sets $\mathcal{V}^k_n$ are called "ravenous" here, rather than just "hungry", because we may feed them with oracle strings that later leave $\mathcal{Q}$, in which case they get hungry again.

We will also ensure that $\mathcal{V}^k_n$ is disjoint from $\mathcal{V}^k_m$ for $n \neq m$. The parameter k determines the goal for these ravenous sets; otherwise, the constructions for distinct k are independent. The other property that we ensure is that

$$\mathcal{V}^k_n \cap \mathcal{Q} \subseteq \mathcal{U}_n \times \Psi^{-1}[A \upharpoonright n+1].$$

Construction of the sets $\mathcal{V}^k_n$, for parameter k. At stage 0, we begin with $\mathcal{V}^k_n$ empty for every $n \in \omega$. At every stage s, we call one of the sets $\mathcal{V}^k_n$ "awake", and the others "asleep". We start with $\mathcal{V}^k_0$ awake.

At stage s, if $\mathcal{V}^k_n$ is awake at this stage, then it has not reached its goal, i.e., $\mu(\mathcal{V}^k_{n,s} \cap \mathcal{Q}_s) < 2^{-k}(\Omega_{n+1} - \Omega_n)$, and so we try to feed it. We call a product $[\sigma] \times [\tau]$ of basic clopen sets *palatable* (at stage s) if: it is disjoint from $\mathcal{V}^k_{m,s}$ for

all m; $[\sigma] \subseteq \mathcal{U}_{n,s}$; and $A_s \upharpoonright n+1 \preccurlyeq \Phi^\tau_s$ (in particular, $[\sigma] \times [\tau]$ is covered by $\mathcal{Q}_s$). Note that if $[\sigma'] \times [\tau']$ is contained in a palatable set, it is itself palatable. By standard assumptions on the enumerations of Φ and $\mathcal{U}$, we can effectively obtain a finite antichain of palatable sets covering all palatable sets, and so determine the total measure covered by palatable sets. If this measure is less than the appetite of $\mathcal{V}^k_n$, i.e. less than $2^{-k}(\Omega_{n+1} - \Omega_n) - \mu(\mathcal{V}^k_{n,s} \cap \mathcal{Q}_s)$, we enumerate all the palatable sets into $\mathcal{V}^k_{n,s+1}$ and declare that $\mathcal{V}^k_n$ to be awake at stage $s+1$.

If instead the measure covered by palatable sets at stage s exceeds the appetite of $\mathcal{V}^k_n$, then we choose a finite anti-chain of palatable sets covering measure exactly $2^{-k}(\Omega_{n+1}-\Omega_n)-\mu(\mathcal{V}^k_{n,s}\cap\mathcal{Q}_s)$ in some effective fashion and enumerate this antichain into $\mathcal{V}^k_{n,s+1}$. We then put $\mathcal{V}^k_n$ to sleep and declare $\mathcal{V}^k_m$ to be awake at stage $s+1$, where m is least such that $\mathcal{V}^k_{m,s}$ has not reached *half* its goal: $\mu(\mathcal{V}^k_{m,s} \cap \mathcal{Q}_s) < 2^{-(k+1)}(\Omega_{n+1} - \Omega_n)$. (Such m will always exist, of course, because all but finitely many $\mathcal{V}^k_{m,s}$ will be empty. It is important to note, though, that as we enumerate measure into $\mathcal{V}^k_n$, measure leaves $\mathcal{Q}$, and so a set $\mathcal{V}^k_n$ could be put to sleep but re-awakened later.)

Verification. Since X fails a $\mathbf{c}_{\Omega,R^{\mathfrak{c}}}$ test, it is not 2-random. Since X is ML-random relative to Y, this implies that Y is Turing incomplete. So by the result of Franklin and Ng, Y is difference random. Since $Y \notin \mathcal{E}$, by Fact 8.1 $2^\omega - \mathcal{E}$ has positive density at Y. Hence $\mathcal{Q}$ has positive density at $X \oplus Y$ by the fact mentioned at the beginning of this section. Let $\mathcal{V}^k = \bigcup_n \mathcal{V}^k_n$. Since $\sum_n(\Omega_{n+1} - \Omega_n) = \Omega$, the sequence of sets $\langle \mathcal{V}^k \cap \mathcal{Q} \rangle$ is a difference test. By a second application of Fact 8.1, $X \oplus Y \notin \bigcap_k \mathcal{V}^k \cap \mathcal{Q}$.

Since $X \oplus Y \in \mathcal{Q}$, we can fix some k with $X \oplus Y \notin \mathcal{V}^k_n$ for any $n \in \omega$. In the remainder of the proof, we omit the superscript k. We first show that every set $\mathcal{V}_n$ eventually reaches half its goal.

Claim 6.9.1. For every n, there is a stage t such that for all $s \geqslant t$,

$$\mu(\mathcal{V}_{n,s} \cap \mathcal{Q}_s) \geqslant 2^{-(k+1)}(\Omega_{n+1} - \Omega_n).$$

Proof. Fix n. There is a $\sigma \prec X$ with $[\sigma] \subseteq \mathcal{U}_n$, and there is a $\tau \prec Y$ with $A \upharpoonright n+1 \preccurlyeq \Phi^\tau$. Fix t_0 such that $[\sigma] \subseteq \mathcal{U}_{n,t_0}$, $A_{t_0} \upharpoonright n+1 = A \upharpoonright n+1$ and $A \upharpoonright n+1 \preccurlyeq \Phi^\tau_{t_0}$. As each $\mathcal{V}_{m,s}$ is the union of a finite antichain and does not contain (X, Y), at every $s \geqslant t_0$ at which $\mathcal{V}_n$ is awake, there is some palatable $[\sigma'] \times [\tau']$ with $\sigma \preccurlyeq \sigma' \prec X$ and $\tau \preccurlyeq \tau' \prec Y$. We do not enumerate some neighborhood covering $[\sigma'] \times [\tau']$ into $\mathcal{V}_{n,s+1}$, so by construction, $\mathcal{V}_n$ is asleep at stage $s+1$.

If s_0 is a stage when $\mathcal{V}_n$ goes to sleep and $s_1 > s_0$ is a stage at which $\mathcal{V}_n$ wakes back up, then $\mu(\mathcal{Q}_{s_0} - \mathcal{Q}_{s_1}) > 2^{-(k+1)}(\Omega_{n+1} - \Omega_n)$. Thus $\mathcal{V}_n$ can go to sleep only finitely often. It follows that for every n, there are only finitely many stages at which $\mathcal{V}_n$ is awake. Let t be the last stage at which any $\mathcal{V}_m$ for $m \leqslant n$ went to sleep. Then $\mu(\mathcal{V}_{n,s} \cap \mathcal{Q}_s) \geqslant 2^{-(k+1)}(\Omega_{n+1} - \Omega_n)$ for every $s \geqslant t$. For otherwise, when the current $\mathcal{V}_j$ goes to sleep, either $\mathcal{V}_n$ or $\mathcal{V}_m$ for $m < n$ would wake, contrary to the choice of t. $\square_{6.9.1}$

We now define a pair of computable functions f and g by simultaneous recursion. We begin by setting $f(-1) = -1$. Given $f(s-1)$, we define $f(s) > f(s-1)$ and $g(s)$ to be sufficiently large so that for every $n < s$,

$$\Omega_{f(s)} - \Omega_n \leqslant 2(\Omega_{g(s)} - \Omega_n),$$

and for every $n < g(s)$,

$$\mu(\mathcal{V}_{n,f(s)} \cap \mathcal{Q}_{f(s)}) \geqslant 2^{-(k+1)}(\Omega_{n+1} - \Omega_n).$$

Note such values always exist: if $g(s)$ is such that $\Omega - \Omega_s \leqslant 2(\Omega_{g(s)} - \Omega_s)$, then the first requirement is satisfied for every $f(s)$; then given any $g(s)$, a sufficiently large choice of $f(s)$ will satisfy the second requirement. Thus we can find such a pair of values by exhaustive search, and f and g are total.

Recall the following notation from Section 6:

$$k_s(n) = \lfloor -\log_2(\Omega_s - \Omega_n) \rfloor.$$

Observe that $k_{f(s)}(n) \geqslant k_{g(s)}(n) - 1$ for all $n < s$, and so

$$\mathbf{c}_{\Omega,R^{\complement}}(n, f(s)) \leqslant 2 \cdot \mathbf{c}_{\Omega,R^{\complement}}(n, g(s)).$$

The following claim will complete the proof that A obeys $\mathbf{c}_{\Omega,R}$.

Claim 6.9.2. The total cost $\mathbf{c}_{\Omega,R}\langle A_{f(s+1)}\rangle$ is bounded by 2^{k+3}.

Proof. Fix a stage s, and suppose that n be least such that $n \in A_{f(s+1)} - A_{f(s)}$. We may assume $n < s$. Then for all $m \geqslant n$, $\pi_2[\mathcal{V}_{m,f(s)}] \subseteq \mathcal{E}_{f(s+1)}$, where $\pi_2 \colon 2^\omega \times 2^\omega \to 2^\omega$ is the projection onto the second coordinate. Let

$$\mathcal{S} = \bigcup_{m \geqslant n} \mathcal{V}_{m,f(s)} \cap \mathcal{Q}_{f(s)}.$$

Note that by definition of the $\mathcal{Q}_t$

$$\mu\left(\mathcal{E}_{f(s+1)} - \mathcal{E}_{f(s)}\right) \geqslant \mu(\pi_2[\mathcal{S}]). \tag{6.1}$$

The sets $\mathcal{V}_m$ are disjoint by construction, and

$$\mu(\mathcal{V}_{m,f(s)} \cap \mathcal{Q}_{f(s)}) \geqslant 2^{-(k+1)}(\Omega_{m+1} - \Omega_m)$$

for $m < g(s)$ by choice of $f(s)$, so

$$\mu(\mathcal{S}) \geqslant 2^{-(k+1)}(\Omega_{g(s)} - \Omega_n) > 2^{-(k+1)}2^{-(k_{g(s)}(n)+1)} \geqslant 2^{-(k+2)}\mathbf{c}_\Omega(n, g(s)).$$

On the other hand, $\pi_1[\mathcal{S}] \subseteq \mathcal{U}_{n,f(s)}$, where π_1 is projection onto the first coordinate, and $\mu(\mathcal{U}_{n,f(s)}) \leqslant \mathbf{c}_{\Omega,R^{\complement}}(n, f(s)) \leqslant 2 \cdot \mathbf{c}_{\Omega,R^{\complement}}(n, g(s))$. Since $\mathcal{S} \subseteq \pi_1[\mathcal{S}] \times \pi_2[\mathcal{S}]$,

$$\mu(\mathcal{S}) \leqslant \mu(\pi_1[\mathcal{S}]) \cdot \mu(\pi_2[\mathcal{S}]) \leqslant 2 \cdot \mathbf{c}_{\Omega,R^{\complement}}(n, g(s)) \cdot \mu(\pi_2[\mathcal{S}]),$$

whence, as $\mathbf{c}_\Omega \leqslant \mathbf{c}_{\Omega,R} \cdot \mathbf{c}_{\Omega,R^{\complement}}$,

$$2^{-(k+3)}\mathbf{c}_{\Omega,R}(n, g(s)) \leqslant \mu(\pi_2[\mathcal{S}]).$$

Therefore, by (6.1), $\mu(\mathcal{E}_{f(s+1)} - \mathcal{E}_{f(s)}) \geqslant 2^{-k-3} \cdot \mathbf{c}_{\Omega,R}(n, g(s)) \geqslant 2^{-k-3} \cdot \mathbf{c}_{\Omega,R}(n, s)$.

By definition the total cost is the sum over all stages s of the costs of the least change at that stage. We conclude that $\mathbf{c}_{\Omega,R}\langle A_{f(s+1)}\rangle \leqslant 2^{k+3}\mu(\mathcal{E})$.

$\square_{6.9.2,\ \text{Lem. 6.9 for } A \text{ c.e.}}$

We lift the c.e. case to the general case. Thanks to Theorem 3.1, this is relatively easy, say compared to the approach taken in [14].

Lemma 8.2. *For any infinite computable set R, obedience to $\mathbf{c}_{\Omega,R}$ is downward closed under Turing reducibility.*

Proof. This is similar to [14, Prop. 2.3]. For brevity, let $f(k) = |R \cap k|$. We only use the facts that f is non-decreasing, and that there is $d \in \mathbb{N}$ such that $f(k+1) \leqslant f(k) + d$ for each k. As in Section 6 let $k(n) = \lfloor -\log_2(\Omega - \Omega_n) \rfloor$ and $k_s(n) = \lfloor -\log_2(\Omega_s - \Omega_n) \rfloor$, so that $\lim_s k_s(n) = k(n)$ in a nonincreasing fashion.

Let B be a Δ^0_2 set that obeys $\mathbf{c}_{\Omega,R}$. Let $\langle B_t \rangle$ be a computable approximation of B witnessing that $B \models \mathbf{c}_{\Omega,R}$. Let $A \leqslant_{\mathrm{T}} B$, say $A = \Psi^B$ for some functional Ψ.

Since $\mathbf{c}_{\Omega,R} \to \mathbf{c}_{\Omega}$, B is K-trivial. Let ψ be the use function for the computation $\Psi^B = A$. By Barmpalias and Downey [1, Lem. 2.5], as B is K-trivial and ψ is B-computable, one has $\Omega - \Omega_n \leqslant^{\times} \Omega - \Omega_{\psi(n)}$ and hence $k(n) \geqslant^{+} k(\psi(n))$.

Hence there is $b \in \mathbb{Z}$ such that $k(n) \geqslant b + k(\psi(n))$ for each n. We define an increasing sequence of stages $s(i)$, starting with $s(0) = 0$; $s(i)$ is the least stage $s > s(i-1)$ such that $|\Psi^{B_s}_s| > i$ and for all $n \leqslant i$, $k_{i+1}(n) \geqslant b + k_s(\psi_s(n))$, where $\psi_s(n)$ is the use of the computation $\Psi^{B_s}_s(n)$. We then let $A_i = \Psi^{B_{s(i)}}_{s(i)}$.

We claim that the approximation $\langle A_i \rangle$ witnesses that A obeys $\mathbf{c}_{\Omega,R}$. The reason is that if $A_i(n) \neq A_{i+1}(n)$ and $n \leqslant i$, then the A-cost paid is $2^{-f(k_{i+1}(n))}$, whereas at some stage $t \in (s(i), s(i+1)]$ we see a change in B below $v = \psi_{s(i)}(n)$, showing that the total cost paid by B along this interval of stages is at least $2^{-f(k_t(v))} \geqslant 2^{-f(k_{s(i)}(v))}$. This allows us to bound the A-cost by the assumed property of f, as $k_{i+1}(n) \geqslant b + k_{s(i)}(v)$. □

Note that in fact obedience to $\mathbf{c}_{\Omega,R}$ is downward closed under ML-reducibility (by Theorem 7.1), but this used Lemma 6.9.

Proof of Lemma 6.9 in the general case. Let A be K-trivial and suppose that the hypotheses of the lemma hold. By Theorem 3.1, let $C \geqslant_{\mathrm{T}} A$ be c.e. such that $C \equiv_{\mathrm{ML}} A$. The ML-equivalence implies that $C \leqslant_{\mathrm{T}} Y$; the c.e. case shows that $C \models \mathbf{c}_{\Omega,R}$. By Lemma 8.2, A obeys $\mathbf{c}_{\Omega,R}$ as well. □

9. Fragments of Ω and strong jump-traceability

A cost function $\mathbf{c}$ is *benign* [15] if from a rational $\varepsilon > 0$, we can compute a bound on the length of any sequence $n_1 < s_1 \leqslant n_2 < s_2 \leqslant \cdots \leqslant n_\ell < s_\ell$ such that $\mathbf{c}(n_i, s_i) \geqslant \varepsilon$ for all $i \leqslant \ell$. For example, $\mathbf{c}_\Omega$ is benign, with the bound being $1/\varepsilon$.

By an order function we mean a computable, non-decreasing, and unbounded function. A set A is *strongly jump-traceable* if for every order function h, for every ψ partial computable in A, there is an h-bounded c.e. trace for ψ; that is, there is a sequence $\langle T(n) \rangle$ such that $|T(n)| \leqslant h(n)$, $T(n)$ is uniformly c.e., and $\psi(n) \in T(n)$ for all $n \in \operatorname{dom} \psi$. Figueira et al. [9] introduced this notion, and built a non-computable c.e. of this kind.

Plain (rather than strong) jump traceability is the notion where one existentially quantifies over order functions h. For c.e. sets this is equivalent to superlowness. Universally quantifying over order functions indeed places a very strong restriction on the computational power of the set. For instance, the strongly jump-traceable sets form an ideal in the Turing degrees [5, 7] which is a proper sub-ideal of the K-trivials [8]. A characterisation that will concern us here is that a set is strongly jump-traceable if and only if it obeys all benign cost functions [15, 7]. For more on strong jump-traceability, see the survey article [16].

One can characterise strong jump-traceability using computability from ML-random sequences. There is more than one such characterisation. For example, a set is strongly jump-traceable if and only if it is computable from all superlow

ML-random sequences [12], also if and only if it is computable from all superhigh random sequences [12, 16]. Alternatively, a c.e. set is strongly jump-traceable if and only if it is computable from a Demuth random sequence by combining [17] and [21]; this extends to all K-trivials by Theorem 3.1. As a consequence one has:

Proposition 9.1. *The strongly jump-traceable sets form an ideal in the ML-degrees.*

In [14, 5.3.1], it is observed that every strongly jump-traceable set is a p-base for all $p > 0$. However, this is not a characterisation. The sets that are p-bases for all $p > 0$ are the $1/\omega$-bases, those which are computable from each column from an infinite partition of some random sequence. Equivalently, they are computable from Ω_R for all computable sets R such that $\liminf_n |R \cap n|/n$ is positive. Some such sets are not strongly jump-traceable as pointed out in [14, 5.3.1]. Here we see that we obtain a characterisation of strong jump-traceability if we drop the density condition.

Proposition 9.2. *For any infinite computable set R, $\mathbf{c}_{\Omega,R}$ is benign.*

Proof. Given a rational $\varepsilon > 0$, first, we compute an m with $2^{-|R\cap m|} < \varepsilon$. Let $n_1 < s_1 \leqslant n_2 < s_2 < \cdots \leqslant n_\ell < s_\ell$ be a sequence such that for all $i \leqslant \ell$, $\mathbf{c}_{\Omega,R}(n_i, s_i) > \varepsilon$. This means that $k_{s_i}(n_i) < m$, and so $\Omega_{s_i} - \Omega_{n_i} \geqslant 2^{-m}$. So

$$1 > \Omega > \sum_{i\leqslant\ell} \Omega_{s_i} - \Omega_{n_i} \geqslant \ell \cdot 2^{-m},$$

and thus $\ell < 2^m$. □

Proposition 9.3. *For any benign cost function $\mathbf{c}$, there is an infinite computable set R with $\mathbf{c}_{\Omega,R} \to \mathbf{c}$.*

Proof. Suppose $g(\varepsilon)$ is a computable bound witnessing that $\mathbf{c}$ is benign. We will construct a left-c.e. real $\beta < 1$. Since Ω is Solovay complete, by the recursion theorem, we may assume that we already know a constant $\delta > 0$ and a computable approximation to Ω with $\delta(\beta_s - \beta_n) < \Omega_s - \Omega_n$ for all n and s. Choose a computable sequence $m_0 < m_1 < \cdots$ such that

$$\sum_i \frac{2^{-m_i} \cdot g\left(2^{-(i+1)}\right)}{\delta} < 1.$$

Let $R = \{m_0 < m_1 < \cdots\}$.

Define $\beta_0 = 0$. At stage $s+1$, if $\mathbf{c}(n, s+1) \leqslant \mathbf{c}_{\Omega,R}(n, s)$ for all n, then let $\beta_{s+1} = \beta_s$. Otherwise, let n be least with $\mathbf{c}(n, s+1) > \mathbf{c}_{\Omega,R}(n, s)$. Let $i = \lfloor -\log \mathbf{c}(n, s+1) \rfloor$. Define $\beta_{s+1} = \beta_s + 2^{-m_i}/\delta$. The point of this is to increase Ω: in this case, we have

$$\Omega_{s+1} - \Omega_s > \delta \cdot (\beta_{s+1} - \beta_s) = 2^{-m_i},$$

and so $k_{s+1}(s) \leqslant m_i$. In turn, this implies that $\mathbf{c}_{\Omega,R}(s, s+1) \geqslant 2^{-i}$.

Claim 9.3.1. For all n, $\underline{\mathbf{c}}_{\Omega,R}(n) \geqslant \underline{\mathbf{c}}(n)$.

Proof. We show that $\mathbf{c}(n, s) \leqslant \mathbf{c}_{\Omega,R}(n, s)$ for and s and all $n < s$. Suppose this holds for s; we verify it for $s+1$. Suppose there is $\hat{n} \leqslant s$, chosen least, such that $\mathbf{c}(\hat{n}, s+1) > \mathbf{c}_{\Omega,R}(\hat{n}, s)$ (otherwise there is nothing to do for $s+1$). For all $n < \hat{n}$,

$$\mathbf{c}(n, s+1) \leqslant \mathbf{c}_{\Omega,R}(n, s) \leqslant \mathbf{c}_{\Omega,R}(n, s+1).$$

Let $i = \lfloor -\log \mathbf{c}(\hat{n}, s+1) \rfloor$. For all $n \geqslant \hat{n}$ such that $n \leqslant s$,

$$\mathbf{c}(n, s+1) \leqslant \mathbf{c}(\hat{n}, s+1) \leqslant 2^{-i} \leqslant \mathbf{c}_{\Omega,R}(s, s+1) \leqslant \mathbf{c}_{\Omega,R}(n, s+1),$$

as required. $\square_{9.3.1}$

The proof of the proposition will be complete once we show:

Claim 9.3.2. $\beta < 1$.

Proof. Fix i and let $s_0 < s_1 < s_2 < \cdots$ be the stages s with $\beta_{s+1} - \beta_s = 2^{-m_i}/\delta$. By construction, for every $n \leqslant s_j$, $\mathbf{c}_{\Omega,R}(n, s_j+1) \geqslant 2^{-i}$. Also by construction, there is some $n_j \leqslant s_j$ with $2^{-(i+1)} < \mathbf{c}(n_j, s_j+1) \leqslant 2^{-i}$ and $\mathbf{c}_{\Omega,R}(n_j, s_j) < \mathbf{c}(n_j, s_j+1)$. Thus $n_0 < s_0 + 1 \leqslant n_1 < s_1 + 1 \leqslant \cdots$. It follows that there are at most $g\left(2^{-(i+1)}\right)$ such stages. So

$$\beta = \sum_s \beta_{s+1} - \beta_s \leqslant \sum_i g\left(2^{-(i+1)}\right) \frac{2^{-m_i}}{\delta} < 1,$$

by the choice of m_i. $\square_{9.3.2,\ 9.3}$

It follows that the sets which obey $\mathbf{c}_{\Omega,R}$ for all computable R are precisely the strongly jump-traceable sets. Theorem 7.1 implies the following, which extends the result from [12] that a set is strongly jump-traceable if and only if it is computable from all ω-computably approximable random sequences.

Corollary 9.4. *A (K-trivial) set A is strongly jump-traceable if and only if $A \leqslant_{\mathrm{T}} \Omega_R$ for every infinite computable set R.*

Note that K-triviality is for free here, as such a set is a 1/2-base.

Similarly, we see that a K-trivial set A is strongly jump-traceable if and only if for every infinite computable R, we have $A \leqslant_{\mathrm{ML}} B_R$, where B_R is ML-complete for $\mathbf{c}_{\Omega,R}$. That is, the ML-ideal of strongly jump-traceable sets is the intersection of the infinitely many principal ideals given by the sets B_R. We conjecture that this ideal is not principal.

References

[1] George Barmpalias and Rod G. Downey. Exact pairs for the ideal of the K-trivial sequences in the Turing degrees. *J. Symb. Log.*, 79(3):676–692, 2014.

[2] Laurent Bienvenu, Adam R. Day, Noam Greenberg, Antonín Kučera, Joseph S. Miller, André Nies, and Dan Turetsky. Computing K-trivial sets by incomplete random sets. *Bull. Symb. Log.*, 20(1):80–90, 2014.

[3] Laurent Bienvenu, Noam Greenberg, Antonín Kučera, André Nies, and Dan Turetsky. Coherent randomness tests and computing the K-trivial sets. *J. Eur. Math. Soc. (JEMS)*, 18(4):773–812, 2016.

[4] Laurent Bienvenu, Rupert Hölzl, Joseph S. Miller, and André Nies. Denjoy, Demuth and density. *J. Math. Log.*, 14(1):1450004, 35, 2014.

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

[6] Adam R. Day and Joseph S. Miller. Density, forcing, and the covering problem. *Math. Res. Lett.*, 22(3):719–727, 2015.

[7] David Diamondstone, Noam Greenberg, and Daniel D. Turetsky. Inherent enumerability of strong jump-traceability. *Trans. Amer. Math. Soc.*, 367(3):1771–1796, 2015.

[8] Rod Downey and Noam Greenberg. Strong jump-traceability II: K-triviality. *Israel J. Math.*, 191(2):647–665, 2012.

[9] Santiago Figueira, André Nies, and Frank Stephan. Lowness properties and approximations of the jump. *Ann. Pure Appl. Logic*, 152(1-3):51–66, 2008.

[10] Johanna N. Y. Franklin and Keng Meng Ng. Difference randomness. *Proc. Amer. Math. Soc.*, 139(1):345–360, 2011.
[11] Péter Gács. Every sequence is reducible to a random one. *Inform. and Control*, 70(2-3):186–192, 1986.
[12] Noam Greenberg, Denis R. Hirschfeldt, and André Nies. Characterizing the strongly jump-traceable sets via randomness. *Adv. Math.*, 231(3-4):2252–2293, 2012.
[13] Noam Greenberg, Joseph S. Miller, Benoit Monin, and Daniel Turetsky. Two more characterizations of K-triviality. *Notre Dame J. Form. Log.*, 59(2):189–195, 2018.
[14] Noam Greenberg, Joseph S Miller, and André Nies. Computing from projections of random points. *Journal of Mathematical Logic*, page 1950014, 2019.
[15] Noam Greenberg and André Nies. Benign cost functions and lowness properties. *J. Symbolic Logic*, 76(1):289–312, 2011.
[16] Noam Greenberg and Dan Turetsky. Strong jump-traceability. *Bull. Symb. Log.*, 24(2):147–164, 2018.
[17] Noam Greenberg and Daniel D. Turetsky. Strong jump-traceability and Demuth randomness. *Proc. Lond. Math. Soc. (3)*, 108(3):738–779, 2014.
[18] Denis R. Hirschfeldt, André Nies, and Frank Stephan. Using random sets as oracles. *J. Lond. Math. Soc. (2)*, 75(3):610–622, 2007.
[19] Antonín Kučera. Measure, Π^0_1-classes and complete extensions of PA. In *Recursion theory week (Oberwolfach, 1984)*, volume 1141 of *Lecture Notes in Math.*, pages 245–259. Springer, Berlin, 1985.
[20] Antonin Kučera. An alternative, priority-free, solution to Post's problem. In *Mathematical foundations of computer science, 1986 (Bratislava, 1986)*, volume 233 of *Lecture Notes in Comput. Sci.*, pages 493–500. Springer, Berlin, 1986.
[21] Antonin Kučera and André Nies. Demuth randomness and computational complexity. *Annals of Pure and Applied Logic*, 162:504–513, 2011.
[22] Rutger Kuyper and Joseph S. Miller. Nullifying randomness and genericity using symmetric difference. *Ann. Pure Appl. Logic*, 168(9):1692–1699, 2017.
[23] A. Nies, F. Stephan, and S. Terwijn. Randomness, relativization and Turing degrees. *J. Symbolic Logic*, 70(2):515–535, 2005.
[24] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.
[25] André Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.
[26] André Nies. Interactions of computability and randomness. In *Proceedings of the International Congress of Mathematicians*, pages 30–57. World Scientific, 2010.
[27] André Nies. Calculus of cost functions. In S. Barry Cooper and Mariya I Soskova, editors, *The Incomputable: Journeys Beyond the Turing Barrier*, Theory and Applications of Computability, pages 183–216. Springer, 2017.
[28] Frank Stephan. Marin-Löf random and PA-complete sets. In *Logic Colloquium '02*, volume 27 of *Lect. Notes Log.*, pages 342–348. Assoc. Symbol. Logic, La Jolla, CA, 2006.
[29] Michiel van Lambalgen. The axiomatization of randomness. *J. Symbolic Logic*, 55(3):1143–1167, 1990.

School of Mathematics and Statistics, Victoria University of Wellington, Wellington, New Zealand
E-mail address: greenberg@msor.vuw.az.nz

Department of Mathematics, University of Wisconsin, Madison, WI 53706, USA
E-mail address: jmiller@math.wisc.edu

Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand
E-mail address: andre@cs.auckland.ac.nz

School of Mathematics and Statistics, Victoria University of Wellington, Wellington, New Zealand
E-mail address: dan@msor.vuw.ac.nz