CALCULUS OF COST FUNCTIONS

ANDRÉ NIES

Abstract. Cost functions provide a framework for constructions of sets Turing below the halting problem that are close to computable. We carry out a systematic study of cost functions. We relate their algebraic properties to their expressive strength. We show that the class of additive cost functions describes the K-trivial sets. We prove a cost function basis theorem, and give a general construction for building computably enumerable sets that are close to being Turing complete.

Contents

1991 *Mathematics Subject Classification.* Primary: 03F60; Secondary: 03D30.
Key words and phrases. computability, randomness, lowness, cost functions.
Research partially supported by the Marsden Fund of New Zealand, grant no. 08-UOA-187, and by the Hausdorff Institute of Mathematics, Bonn.

1. Introduction

In the time period from 1986 to 2003, several constructions of computably enumerable (c.e.) sets appeared. They turned out to be closely related.

(a) Given a Martin-Löf random (ML-random for short) Δ^0_2 set Y, Kučera [15] built a c.e. incomputable set $A \le_T Y$. His construction is interesting because in the case that $Y <_T \emptyset'$, it provides a c.e. set A such that $\emptyset <_T A <_T \emptyset'$, without using injury to requirements as in the traditional proofs. ($\emptyset'$ denotes the halting problem.)

(b) Kučera and Terwijn [17] built a c.e. incomputable set A that is low for ML-randomness: every ML-random set is already ML-random relative to A.

(c) A is called K-trivial if $K(A \upharpoonright_n) \le K(n) + O(1)$, where K denotes prefix-free descriptive string complexity. This means that the initial segment complexity of A grows as slowly as that of a computable set. Downey et al. [8] gave a very short construction (almost a "definition") of a c.e., but incomputable K-trivial set.

The sets in (a) and (b) enjoy a so-called lowness property, which says that the set is very close to computable. Such properties can be classified according to various paradigms introduced in [23, 10]. The set in (a) obeys the *Turing-below-many* paradigm which says that A is close to being computable because it is easy for an oracle set to compute it. A frequent alternative is the *weak-as-an-oracle* paradigm: A is weak in a specific sense when used as an oracle set in a Turing machine computation. An example is the oracle set in (b), which is so weak that it useless as an extra computational device when testing for ML-randomness. On the other hand, K-triviality in (c) is a property stating that the set is far from random: by the Schnorr-Levin Theorem, for a random set Z the initial segment complexity grows fast in that $K(Z \upharpoonright_n) \ge n - O(1)$. For background on the properties in (a)-(c) see [7] and [22, Ch. 5].[1]

A central point for starting our investigations is the fact that the constructions in (a)–(c) look very similar. In hindsight this is not surprising: the classes of sets implicit in (a)-(c) coincide! Let us discuss why.

(b) coincides with (c): Nies [21], with some assistance by Hirschfeldt, showed that lowness for ML-randomness is the same as K-triviality. For this he introduced a method now known as the "golden run".

(a) coincides with (b): The construction in (a) is only interesting if $Y \not\ge_T \emptyset'$. Hirschfeldt, Nies and Stephan [13] proved that if A is a c.e. set such that $A \le_T Y$ for some ML-random set $Y \not\ge_T \emptyset'$, then A is K-trivial, confirming the intuition sets of the type built by Kučera are close to computable. They asked whether, conversely, for every K-trivial set A there is a ML-random set $Y \ge_T A$ with $Y \not\ge_T \emptyset'$. This question became known as the ML-covering problem. Recently the question was solved in the affirmative by combining the work of seven authors in two separate papers. In fact, there is a single

[1] We note that the result (c) has a complicated history. Solovay [25] built a Δ^0_2 incomputable set A that is K-trivial. Constructing a c.e. example of such a set was attempted in various sources such as [4], and unpublished work of Kummer.

ML-random Δ^0_2 set $Y \not\geq_T \emptyset'$ that is Turing above all the K-trivials. A summary is given in [1].

The common idea for these constructions is to ensure lowness of A dynamically, by restricting the overall manner in which numbers can be enumerated into A. This third lowness paradigm has been called *inertness* in [23]: a set A is close to computable because it is computably approximable with a small number of changes.

The idea is implemented as follows. The enumeration of a number x into A at a stage s bears a cost $\mathbf{c}(x, s)$, a non-negative rational that can be computed from x and s. We have to enumerate A in such a way that the sum of all costs is finite. A construction of this type will be called a *cost function construction*.

If we enumerate at a stage more than one number into A, only the cost for enumerating the least number is charged. So, we can reduce cost by enumerating A in "chunks".

1.1. **Background on cost functions.** The general theory of cost functions began in [22, Section 5.3]. It was further developed in [11, 10, 6]. We use the language of [22, Section 5.3] which already allows for the constructions of Δ^0_2 sets. The language is enriched by some notation from [6]. We will see that most examples of cost functions are based on randomness-related concepts.

Definition 1.1. A *cost function* is a computable function

$$\mathbf{c} : \mathbb{N} \times \mathbb{N} \to \{x \in \mathbb{Q} : \ x \geq 0\}.$$

Recall that a *computable approximation* is a computable sequence of finite sets $\langle A_s \rangle_{s\in\mathbb{N}}$ such that $\lim_s A_s(x)$ exists for each x.

Definition 1.2. (i). Given a computable approximation $\langle A_s \rangle_{s\in\mathbb{N}}$ and a cost function $\mathbf{c}$, for $s > 0$ we let

$$\mathbf{c}_s(A_s) = \mathbf{c}(x, s) \text{ where } x < s \ \& \ x \text{ is least s.t. } A_{s-1}(x) \neq A_s(x);$$

if there is no such x we let $\mathbf{c}_s(A_s) = 0$. This is the cost of changing A_{s-1} to A_s. We let

$$\mathbf{c}\langle A_s \rangle_{s\in\mathbb{N}} = \sum_{s>0} \mathbf{c}_s(A_s)$$

be the total cost of all the A-changes. We will often write $\mathbf{c}\langle A_s \rangle$ as a shorthand for $\mathbf{c}\langle A_s \rangle_{s\in\mathbb{N}}$.

(ii) We say that $\langle A_s \rangle_{s\in\mathbb{N}}$ *obeys* $\mathbf{c}$ if $\mathbf{c}\langle A_s \rangle$ is finite. We denote this by

$$\langle A_s \rangle \models \mathbf{c}.$$

(iii) We say that a Δ^0_2 set A *obeys* $\mathbf{c}$, and write $A \models \mathbf{c}$, if some computable approximation of A obeys $\mathbf{c}$.

A cost function $\mathbf{c}$ acts like a global restraint, which is successful if the condition $\mathbf{c}\langle A_s \rangle < \infty$ holds. Kučera's construction mentioned in (a) above needs to be recast in order to be viewed as a cost-function construction [11, 22]. In contrast, (b) and (c) can be directly seen as cost function constructions. In each of (a)–(c) above, one defines a cost function $\mathbf{c}$ such that any set A obeying $\mathbf{c}$ has the lowness property in question. For, if

$A \models \mathbf{c}$, then one can enumerate an auxiliary object that has in some sense a bounded weight.

In (a), this object is a Solovay test that accumulates the errors in an attempted computation of A with oracle Y. Since Y passes this test, Y computes A.

In (b), one is given a $\Sigma^0_1(A)$ class $\mathcal{V} \subseteq 2^\omega$ such that the uniform measure $\lambda\mathcal{V}$ is less than 1, and the complement of $\mathcal{V}$ consists only of ML-randoms. Using that A obeys $\mathbf{c}$, one builds a Σ^0_1 class $\mathcal{S} \subseteq 2^\omega$ containing $\mathcal{V}$ such that still $\lambda\mathcal{S} < 1$. This implies that A is low for ML-randomness.

In (c) one builds a bounded request set (i.e., Kraft-Chaitin set) which shows that A is K-trivial.

The cost function in (b) is adaptive in the sense that $\mathbf{c}(x, s)$ depends on A_{s-1}. In contrast, the cost functions in (a) and (c) can be defined in advance, independently of the computable approximation of the set A that is built.

The main existence theorem, which we recall as Theorem 2.7 below, states that for any cost function $\mathbf{c}$ with the limit condition $\lim_x \liminf_s \mathbf{c}(x, s) = 0$, there is an incomputable c.e. set A obeying $\mathbf{c}$. The cost functions in (a)-(c) all have the limit condition. Thus, by the existence theorem, there is an incomputable c.e. set A with the required lowness property.

Besides providing a unifying picture of these constructions, cost functions have many other applications. We discuss some of them.

Weak 2-randomness is a notion stronger than ML-randomness: a set Z is weakly 2-random if Z is in no Π^0_2 null class. In 2006, Hirschfeldt and Miller gave a characterization of this notion: a ML-random is weakly 2-random if and only if it forms a minimal pair with $\emptyset'$. The implication from left to right is straightforward. The converse direction relies on a cost function related to the one for Kučera's result (a) above. (For detail see e.g. [22, Thm. 5.3.6].) Their result can be seen as an instance of the randomness enhancement principle [23]: the ML-random sets get more random as they lose computational complexity.

The author [21] proved that the single cost function $\mathbf{c}_{\mathcal{K}}$ introduced in [8] (see Subsection 2.3 below) characterises the K-trivials. As a corollary, he showed that every K-trivial set A is truth-table below a c.e. K-trivial D. The proof of this corollary uses the general framework of change sets spelled out in Proposition 2.14 below. While this is still the only known proof yielding $A \leq_{\mathrm{tt}} D$, Bienvenu et al. [2] have recently given an alternative proof using Solovay functions in order to obtain the weaker reduction $A \leq_T D$.

In model theory, one asks whether a class of structures can be described by a first order theory. Analogously, we ask whether an ideal of the Turing degrees below $\mathbf{0}'$ is given by obedience to all cost functions of an appropriate type. For instance, the K-trivials are axiomatized by $\mathbf{c}_{\mathcal{K}}$.

Call a cost function $\mathbf{c}$ *benign* if from n one can compute a bound on the number of disjoint intervals $[x, s)$ such that $\mathbf{c}(x, s) \geq 2^{-n}$. Figueira et al. [9] introduced the property of being strongly jump traceable (s.j.t.), which is an extreme lowness property of an oracle A, even stronger than being low for K. Roughly speaking, A is s.j.t. if the jump $J^A(x)$ is in T_x whenever it is defined, where $\langle T_x \rangle$ is a uniformly c.e. sequence of sets such that any given order function bounds the size of almost all the T_x. Greenberg and Nies [11] showed that the class of benign cost functions axiomatizes the c.e. strongly jump traceable sets.

Greenberg et al. [10] used cost functions to show that each strongly jump-traceable c.e. set is Turing below each ω-c.e. ML-random set. As a main result, they also obtained the converse. In fact they showed that any set that is below each superlow ML-random set is s.j.t.

The question remained whether a general s.j.t. set is Turing below each ω-c.e. ML-random set. Diamondstone et al. [6] showed that each s.j.t. set A is Turing below a c.e., s.j.t. set D. To do so, as a main technical result they provided a benign cost function $\mathbf{c}$ such that each set A obeying $\mathbf{c}$ is Turing below a c.e. set D which obeys every cost function that A obeys. In particular, if A is s.j.t., then $A \models \mathbf{c}$, so the c.e. cover D exists and is also s.j.t. by the above-mentioned result of Greenberg and Nies [11]. This gives an affirmative answer to the question. Note that this answer is analogous to the result [1] that every K-trivial is below an incomplete random.

1.2. **Overview of our results.** The main purpose of the paper is a systematic study of cost functions and the sets obeying them. We are guided by the above-mentioned analogy from first-order model theory: cost functions are like sentences, sets are like models, and obedience is like satisfaction. So far this analogy has been developed only for cost functions that are monotonic (that is, non-increasing in the first component, non-decreasing in the stage component). In Section 3 we show that the conjunction of two monotonic cost functions is given by their sum, and implication $\mathbf{c} \to \mathbf{d}$ is equivalent to $\underline{\mathbf{d}} = O(\underline{\mathbf{c}})$ where $\underline{\mathbf{c}}(x) = \sup_s \mathbf{c}(x, s)$ is the limit function.

In Section 4 we show that a natural class of cost functions introduced in Nies [23] characterizes the K-trivial sets: a cost function $\mathbf{c}$ is additive if $\mathbf{c}(x, y) + \mathbf{c}(y, z) = \mathbf{c}(x, z)$ for all $x < y < z$. We show that such a cost function is given by an enumeration of a left-c.e. real, and that implication corresponds to Solovay reducibility on left-c.e. reals. Additive cost functions have been used prominently in the solution of the ML-covering problem [1]. The fact that a given K-trivial A obeys every additive cost function is used to show that $A \leq_T Y$ for the Turing incomplete ML-random set constructed by Day and Miller [5].

Section 5 contains some more applications of cost functions to the study of computational lowness and K-triviality. For instance, strengthening the result in [10] mentioned above, we show that each c.e., s.j.t. set is below any complex ω-c.e. set Y, namely, a set Y such that there is an order function g with $g(n) \leq^+ K(Y \restriction_n)$ for each n. In addition, the use of the reduction is bounded by the identity. Thus, the full ML-randomness assumed in [10] was too strong a hypothesis. We also discuss the relationship of cost functions and a weakening of K-triviality.

In the remaining part of the paper we obtain two existence theorems. Section 6 shows that given an arbitrary monotonic cost function $\mathbf{c}$, any nonempty Π^0_1 class contains a Δ^0_2 set Y that is so low that each c.e. set $A \leq_T Y$ obeys $\mathbf{c}$. In Section 7 we relativize a cost function $\mathbf{c}$ to an oracle set Z, and show that there is a c.e. set D such that $\emptyset'$ obeys $\mathbf{c}^D$ relative to D. This much harder "dual" cost function construction can be used to build incomplete c.e. sets that are very close to computing $\emptyset'$. For instance, if $\mathbf{c}$ is the cost function $\mathbf{c}_{\mathcal{K}}$ for K-triviality, then D is LR-complete.

2. Basics

We provide formal background, basic facts and examples relating to the discussion above. We introduce classes of cost functions: monotonic, and proper cost functions. We formally define the limit condition, and give a proof of the existence theorem.

2.1. **Some easy facts on cost functions.**

Definition 2.1. We say that a cost function $\mathbf{c}$ is *nonincreasing in the main argument* if

$$\forall x, s\, [\mathbf{c}(x+1, s) \leq \mathbf{c}(x, s)].$$

We say that $\mathbf{c}$ is *nondecreasing in the stage* if $\mathbf{c}(x, s) = 0$ for $x > s$ and

$$\forall x, s\, [\mathbf{c}(x, s) \leq \mathbf{c}(x, s+1)].$$

If $\mathbf{c}$ has both properties we say that $\mathbf{c}$ is *monotonic*. This means that the cost $\mathbf{c}(x, s)$ does not decrease when we enlarge the interval $[x, s]$.

Fact 2.2. *Suppose $A \models \mathbf{c}$. Then for each $\epsilon > 0$ there is a computable approximation $\langle A_s \rangle_{s\in\mathbb{N}}$ of A such that $\mathbf{c}\langle A_s \rangle_{s\in\mathbb{N}} < \epsilon$.* □

Proof. Suppose $\langle \widehat{A}_s \rangle_{s\in\mathbb{N}} \models \mathbf{c}$. Given x_0 consider the modified computable approximation $\langle A_s \rangle_{s\in\mathbb{N}}$ of A that always outputs the final value $A(x)$ for each $x \leq x_0$. That is, $A_s(x) = A(x)$ for $x \leq x_0$, and $A_s(x) = \widehat{A}_s(x)$ for $x > x_0$. Choosing x_0 sufficiently large, we can ensure $\mathbf{c}\langle A \rangle_s < \epsilon$. □

Definition 2.3. Suppose that a cost function $\mathbf{c}(x, t)$ is non-increasing in the main argument x. We say that $\mathbf{c}$ is *proper* if $\forall x\, \exists t\, \mathbf{c}(x, t) > 0$.

If a cost function that is non-increasing in the main argument is not proper, then every Δ^0_2 set obeys $\mathbf{c}$. Usually we will henceforth assume that a cost function $\mathbf{c}$ is proper. Here is an example how being proper helps.

Fact 2.4. *Suppose that $\mathbf{c}$ is a proper cost function and $S = \mathbf{c}\langle A_s \rangle < \infty$ is a computable real. Then A is computable.*

Proof. Given an input x, compute a stage t such that $\delta = \mathbf{c}(x, t) > 0$ and $S - \mathbf{c}\langle A_s \rangle_{s\leq t} < \delta$. Then $A(x) = A_t(x)$. □

A *computable enumeration* is a computable approximation $\langle B_s \rangle_{s\in\mathbb{N}}$ such that $B_s \subseteq B_{s+1}$ for each s.

Fact 2.5. *Suppose $\mathbf{c}$ is a monotonic cost function and $A \models \mathbf{c}$ for a c.e. set A. Then there is a computable* enumeration *$\langle \widetilde{A}_s \rangle$ that obeys $\mathbf{c}$.*

Proof. Suppose $\langle A_s \rangle \models \mathbf{c}$ for a computable approximation $\langle A_s \rangle$ of A. Let $\langle B_t \rangle$ be a computable enumeration of A. Define $\langle \widetilde{A}_s \rangle$ as follows. Let $\widetilde{A}_0(x) = 0$; for $s > 0$ let $\widetilde{A}_s(x) = \widetilde{A}_{s-1}(x)$ if $\widetilde{A}_{s-1}(x) = 1$; otherwise let $\widetilde{A}_s(x) = A_t(x)$ where $t \geq s$ is least such that $A_t(x) = B_t(x)$.

Clearly $\langle \widetilde{A}_s \rangle$ is a computable enumeration of A. If $\widetilde{A}_s(x) \neq \widetilde{A}_{s-1}(x)$ then $A_{s-1}(x) = 0$ and $A_s(x) = 1$. Therefore $\mathbf{c}\langle \widetilde{A}_s \rangle \leq \mathbf{c}\langle A_s \rangle < \infty$. □

2.2. **The limit condition and the existence theorem.**

For a cost function $\mathbf{c}$, let

$$\underline{\mathbf{c}}(x) = \liminf_s \mathbf{c}(x, s). \tag{1}$$

Definition 2.6. *We say that a cost function* $\mathbf{c}$ *satisfies the limit condition if* $\lim_x \underline{\mathbf{c}}(x) = 0$. *That is, for each* e*, for almost every* x *we have*

$$\exists^\infty s\, [\mathbf{c}(x, s) \leq 2^{-e}].$$

In previous works such as [22], the limit condition was defined in terms of $\sup_s \mathbf{c}(x, s)$, rather than $\liminf_s \mathbf{c}(x, s)$. The cost functions previously considered were usually nondecreasing in the stage component, in which case $\sup_s \mathbf{c}(x, s) = \liminf_s \mathbf{c}(x, s)$ and hence the two versions of the limit condition are equivalent. Note that the limit condition is a Π^0_3 condition on cost functions that are nondecreasing in the stage, and Π^0_4 in general.

The basic existence theorem says that a cost function with the limit condition has a c.e., incomputable model. This was proved by various authors for particular cost functions. The following version of the proof appeared in [8] for the particular cost function $\mathbf{c}_{\mathcal{K}}$ defined in Subsection 2.3 below, and then in full generality in [22, Thm 5.3.10].

Theorem 2.7. *Let* $\mathbf{c}$ *be a cost function with the limit condition.*

(i) *There is a simple set* A *such that* $A \models \mathbf{c}$. *Moreover,* A *can be obtained uniformly in (a computable index for)* $\mathbf{c}$.

(ii) *If* $\mathbf{c}$ *is nondecreasing in the stage component, then we can make* A *promptly simple.*

Proof. (i) We meet the usual simplicity requirements

$$S_e\colon\ \#W_e = \infty \ \Rightarrow\ W_e \cap A \neq \emptyset.$$

To do so, we define a computable enumeration $\langle A_s \rangle_{s \in \mathbb{N}}$ as follows. Let $A_0 = \emptyset$. At stage $s > 0$, for each $e < s$, if S_e has not been met so far and there is $x \geq 2e$ such that $x \in W_{e,s}$ and $\mathbf{c}(x, s) \leq 2^{-e}$, put x into A_s. Declare S_e met.

To see that $\langle A_s \rangle_{s \in \mathbb{N}}$ obeys $\mathbf{c}$, note that at most one number is put into A for the sake of each requirement. Thus $\mathbf{c}\langle A_s \rangle \leq \sum_e 2^{-e} = 2$.

If W_e is infinite, then there is an $x \geq 2e$ and $s > x$ such that $x \in W_{e,s}$ and $\mathbf{c}(x, s) \leq 2^{-e}$, because $\mathbf{c}$ satisfies the limit condition. So we meet S_e. Clearly the construction of A is uniform in an index for the computable function $\mathbf{c}$.

(ii) Now we meet the prompt simplicity requirements

$$PS_e\colon\ \#W_e = \infty \ \Rightarrow\ \exists s\, \exists x\, [x \in W_{e,s} - W_{e,s-1} \;\&\; x \in A_s].$$

Let $A_0 = \emptyset$. At stage $s > 0$, for each $e < s$, if PS_e has not been met so far and there is $x \geq 2e$ such that $x \in W_{e,s} - W_{e,s-1}$ and $\mathbf{c}(x, s) \leq 2^{-e}$, put x into A_s. Declare PS_e met.

If W_e is infinite, there is an $x \geq 2e$ in W_e such that $\mathbf{c}(x, s) \leq 2^{-e}$ for all $s > x$, because $\mathbf{c}$ satisfies the limit condition and is nondecreasing in the stage component. We enumerate such an x into A at the stage $s > x$ when x appears in W_e, if PS_e has not been met yet by stage s. Thus A is promptly simple. ∎

Theorem 2.7(i) was strengthened in [22, Thm 5.3.22]. As before let $\mathbf{c}$ be a cost function with the limit condition. Then for each low c.e. set B, there is a c.e. set A obeying $\mathbf{c}$ such that $A \not\leq_T B$. The proof of [22, Thm 5.3.22] is for the case of the stronger version of the limit condition $\lim_x \sup_s \mathbf{c}(x,s) = 0$, but in fact works for the version given above.

The assumption that B be c.e. is necessary: there is a low set Turing above all the K-trivial sets by [16], and the K-trivial sets can be characterized as the sets obeying the cost function $\mathbf{c}_{\mathcal{K}}$ of Subsection 2.3 below.

The following fact implies the converse of Theorem 2.7 in the monotonic case.

Fact 2.8. *Let $\mathbf{c}$ be a monotonic cost function. If a computable approximation $\langle A_s \rangle_{s\in\mathbb{N}}$ of an incomputable set A obeys $\mathbf{c}$, then $\mathbf{c}$ satisfies the limit condition.*

Proof. Suppose the limit condition fails for e. There is s_0 such that

$$\textstyle\sum_{s\geq s_0} \sum_{x<s} \mathbf{c}_s(A_s) \leq 2^{-e}.$$

To compute A, on input n compute $s > \max(s_0, n)$ such that $\mathbf{c}(n,s) > 2^{-e}$. Then $A_s(n) = A(n)$. ☐

Convention 2.9. For a *monotonic* cost function $\mathbf{c}$, we may forthwith assume that $\underline{\mathbf{c}}(x) < \infty$ for each x. For, firstly, if $\forall x\, [\underline{\mathbf{c}}(x) = \infty]$, then $A \models \mathbf{c}$ implies that A is computable. Thus, we may assume there is x_0 such that $\underline{\mathbf{c}}(x)$ is finite for all $x \geq x_0$ since $\underline{\mathbf{c}}(x)$ is nonincreasing. Secondly, changing values $\mathbf{c}(x,s)$ for the finitely many $x < x_0$ does not alter the class of sets A obeying $\mathbf{c}$. So fix some rational $q > \mathbf{c}(x_0)$ and, for $x < x_0$ redefine $\mathbf{c}(x,s) = q$ for all s.

2.3. The cost function for K-triviality.

Let $K_s(x) = \min\{|\sigma| \colon \mathbb{U}_s(\sigma) = x\}$ be the value of prefix-free descriptive string complexity of x at stage s. We use the conventions $K_s(x) = \infty$ for $x \geq s$ and $2^{-\infty} = 0$. Let

$$\mathbf{c}_{\mathcal{K}}(x,s) = \sum_{w=x+1}^{s} 2^{-K_s(w)}. \tag{2}$$

Sometimes $\mathbf{c}_{\mathcal{K}}$ is called the *standard cost function*, mainly because it was the first example of a cost function that received attention. Clearly, $\mathbf{c}_{\mathcal{K}}$ is monotonic. Note that $\underline{\mathbf{c}_{\mathcal{K}}}(x) = \sum_{w>x} 2^{-K(w)}$. Hence $\mathbf{c}_{\mathcal{K}}$ satisfies the limit condition: given $e \in \mathbb{N}$, since $\sum_w 2^{-K(w)} \leq 1$, there is an x_0 such that

$$\textstyle\sum_{w\geq x_0} 2^{-K(w)} \leq 2^{-e}.$$

Therefore $\underline{\mathbf{c}_{\mathcal{K}}}(x) \leq 2^{-e}$ for all $x \geq x_0$.

The following example illustrates that in Definition 1.2, obeying $\mathbf{c}_{\mathcal{K}}$, say, strongly depends on the chosen enumeration. Clearly, if we enumerate $A = \mathbb{N}$ by putting in x at stage x, then the total cost of changes is zero.

Proposition 2.10. *There is a computable enumeration $\langle A_s \rangle_{s\in\mathbb{N}}$ of $\mathbb{N}$ in the order $0, 1, 2, \ldots$ (i.e., each A_s is an initial segment of $\mathbb{N}$) such that $\langle A_s \rangle_{s\in\mathbb{N}}$ does not obey $\mathbf{c}_{\mathcal{K}}$.*

Proof. Since $K(2^j) \leq^+ 2\log j$, there is an increasing computable function f and a number j_0 such that $\forall j \geq j_0\, K_{f(j)}(2^j) \leq j-1$. Enumerate the set $A = \mathbb{N}$ in order, but so slowly that for each $j \geq j_0$ the elements of $(2^{j-1}, 2^j]$ are enumerated only after stage $f(j)$, one by one. Each such enumeration costs at least $2^{-(j-1)}$, so the cost for each interval $(2^{j-1}, 2^j]$ is 1. $\square$

Intuitively speaking, an infinite c.e. set A can obey the cost function $\mathbf{c}_{\mathcal{K}}$ only because during an enumeration of x at stage s one merely pays the current cost $\mathbf{c}_{\mathcal{K}}(x,s)$, not the limit cost $\underline{\mathbf{c}}_{\mathcal{K}}(x)$.

Fact 2.11. *If a c.e. set A is infinite, then $\sum_{x \in A} \underline{\mathbf{c}}_{\mathcal{K}}(x) = \infty$.*

Proof. Let f be a 1-1 computable function with range A. Let L be the bounded request set $\{\langle r, \max_{i \leq 2^{r+1}} f(i)\rangle \colon r \in \mathbb{N}\}$. Let M be a machine for L according to the Machine Existence Theorem, also known as the Kraft-Chaitin Theorem. See e.g. [22, Ch. 2] for background. $\square$

In [21] (also see [22, Ch. 5]) it is shown that A is K-trivial iff $A \models \mathbf{c}_{\mathcal{K}}$. So far, the class of K-trivial sets has been the only known natural class that is characterized by a single cost function. However, recent work with Greenberg and Miller suggests that for a c.e. set A, being below both halves Z_0, Z_1 of some Martin-Löf-random $Z = Z_0 \oplus Z_1$ is equivalent to obeying the cost function $\mathbf{c}(x,s) = \sqrt{\Omega_s - \Omega_x}$.

2.4. **Basic properties of the class of sets obeying a cost function.** In this subsection, unless otherwise stated, cost functions will be monotonic. Recall from Definition 2.3 that a cost function $\mathbf{c}$ is called *proper* if $\forall x\, \exists t\, \mathbf{c}(x,t) > 0$. We investigate the class of models of a proper cost function $\mathbf{c}$. We also assume Convention 2.9 that $\underline{\mathbf{c}}(x) < \infty$ for each x.

The first two results together show that $A \models \mathbf{c}$ implies that A is weak truth-table below a c.e. set C such that $C \models \mathbf{c}$. Recall that a Δ^0_2 set A is called ω-c.e. if there is a computable approximation $\langle A_s \rangle$ such that the number of changes $\#\{s \colon A_s(x) \neq A_{s-1}(x)\}$ is computably bounded in x; equivalently, $A \leq_{\mathrm{wtt}} \emptyset'$ (see [22, 1.4.3]).

Fact 2.12. *Suppose that $\mathbf{c}$ is a proper monotonic cost function. Let $A \models \mathbf{c}$. Then A is ω-c.e.*

Proof. Suppose $\langle A_s \rangle \models \mathbf{c}$. Let g be the computable function given by $g(x) = \mu t.\, \mathbf{c}(x,t) > 0$. Let $\widehat{A}_s(x) = A_{g(x)}(x)$ for $s < g(x)$, and $\widehat{A}_s(x) = A_s(x)$ otherwise. Then the number of times $\widehat{A}_s(x)$ can change is bounded by $\mathbf{c}\langle A_s \rangle / \mathbf{c}(x, g(x))$. $\square$

Let V_e denote the e-th ω-c.e. set (see [22, pg. 20]).

Fact 2.13. *For each cost function $\mathbf{c}$, the index set $\{e \colon V_e \models \mathbf{c}\}$ is Σ^0_3.*

Proof. Let D_n denote the n-th finite set of numbers. We may view the i-th partial computable function Φ_i as a (possibly partial) computable approximation $\langle A_t \rangle$ by letting $A_t \simeq D_{\Phi_i(t)}$ (the symbol $\simeq$ indicates that 'undefined' is a possible value). Saying that Φ_i is total and a computable approximation of V_e is a Π^0_2 condition of i and e. Given that Φ_i is total, the condition that $\langle A_t \rangle \models \mathbf{c}$ is Σ^0_2. $\square$

The *change set* (see [22, 1.4.2]) for a computable approximation $\langle A_s \rangle_{s\in\mathbb{N}}$ of a Δ^0_2 set A is a c.e. set $C \geq_T A$ defined as follows: if $s > 0$ and $A_{s-1}(x) \neq A_s(x)$ we put $\langle x, i\rangle$ into C_s, where i is least such that $\langle x, i\rangle \notin C_{s-1}$. If A is ω-c.e. via this approximation then $C \geq_{tt} A$. The change set can be used to prove the implication of the Shoenfield Limit Lemma that $A \in \Delta^0_2$ implies $A \leq_T \emptyset'$; moreover, if A is ω-c.e., then $A \leq_{\mathrm{wtt}} \emptyset'$.

Proposition 2.14 ([22], Section 5.3)**.** *Let the cost function* $\mathbf{c}$ *be non-increasing in the first component. If a computable approximation* $\langle A_s \rangle_{s\in\mathbb{N}}$ *of a set* A *obeys* $\mathbf{c}$*, then its change set* C *obeys* $\mathbf{c}$ *as well.*

Proof. Since $x < \langle x, i\rangle$ for each x, i, we have

$$C_{s-1}(x) \neq C_s(x) \rightarrow A_{s-1} \upharpoonright_x \neq A_s \upharpoonright_x$$

for each x, s. Then, since $\mathbf{c}(x, s)$ is non-increasing in x, we have $\mathbf{c}\langle C_s\rangle \leq \mathbf{c}\langle A_s\rangle < \infty$. □

This yields a limitation on the expressiveness of cost functions. Recall that A is superlow if $A' \leq_{\mathrm{tt}} \emptyset'$.

Corollary 2.15. *There is no cost function* $\mathbf{c}$ *monotonic in the first component such that* $A \models \mathbf{c}$ *iff* A *is superlow.*

Proof. Otherwise, for each superlow set A there is a c.e. superlow set $C \geq_T A$. This is clearly not the case: for instance A could be ML-random, and hence of diagonally non-computable degree, so that any c.e. set $C \geq_T A$ is Turing complete. □

For $X \subseteq \mathbb{N}$ let $2X$ denote $\{2x \colon x \in X\}$. Recall that $A \oplus B = 2A \cup (2B+1)$. We now show that the class of sets obeying $\mathbf{c}$ is closed under $\oplus$ and closed downward under a restricted form of weak truth-table reducibility.

Clearly, $E \models \mathbf{c}$ & $F \models \mathbf{c}$ implies $E \cup F \models \mathbf{c}$.

Proposition 2.16. *Let the cost function* $\mathbf{c}$ *be monotonic in the first component. Then* $A \models \mathbf{c}$ & $B \models \mathbf{c}$ *implies* $A \oplus B \models \mathbf{c}$.

Proof. Let $\langle A_s\rangle$ by a computable appoximation of A. By the monotonicity of $\mathbf{c}$ we have $\mathbf{c}\langle A_s\rangle \geq \mathbf{c}(2A_s)$. Hence $2A \models \mathbf{c}$. Similarly, $2B + 1 \models \mathbf{c}$. Thus $A \oplus B \models \mathbf{c}$. □

Recall that there are superlow c.e. sets A_0, A_1 such that $A_0 \oplus A_1$ is Turing complete (see [22, 6.1.4]). Thus the foregoing result yields a a stronger form of Cor. 2.15: no cost function characterizes superlowness within the c.e. sets.

3. Look-ahead arguments

This core section of the paper introduces an important type of argument. Suppose we want to construct a computable approximation of a set A that obeys a given monotonic cost function. If we can anticipate that $A(x)$ needs to be changed in the future, we try to change it as early as possible, because earlier changes are cheaper. Such an argument will be called a *look-ahead argument*. (Also see the remark before Fact 2.11.) The main application of this method is to characterize logical properties of cost functions algebraically.

3.1. **Downward closure under $\leq_{ibT}$.** Recall that $A \leq_{ibT} B$ if $A \leq_{\mathrm{wtt}} B$ with use function bounded by the identity. We now show that the class of models of $\mathbf{c}$ is downward closed under $\leq_{ibT}$.

Proposition 3.1. *Let $\mathbf{c}$ be a monotonic cost function. Suppose that $B \models \mathbf{c}$ and $A = \Gamma^B$ via a Turing reduction Γ such that each oracle query on an input x is at most x. Then $A \models \mathbf{c}$.*

Proof. Suppose $B \models \mathbf{c}$ via a computable approximation $\langle B_s \rangle_{s\in\mathbb{N}}$. We define a computable increasing sequence of stages $\langle s(i) \rangle_{i\in\mathbb{N}}$ by $s(0) = 0$ and

$$s(i+1) = \mu s > s(i)\, [\Gamma^B \upharpoonright_{s(i)} [s] \downarrow].$$

In other words, $s(i+1)$ is the least stage s greater than $s(i)$ such that at stage s, $\Gamma^B(n)$ is defined for each $n < s(i)$. We will define $A_{s(k)}(x)$ for each $k \in \mathbb{N}$. Thereafter we let $A_s(x) = A_{s(k)}(x)$ where k is maximal such that $s(k) \leq s$.

Suppose $s(i) \leq x < s(i+1)$. For $k < i$ let $A_{s(k)}(x) = v$, where $v = \Gamma^B(x)[s(i+2)]$. For $k \geq i$, let $A_{s(k)}(x) = \Gamma^B(x)[s(k+2)]$. (Note that these values are defined. Taking the $\Gamma^B(x)$ value at the large stage $s(k+2)$ represents the look-ahead.)

Clearly $\lim_s A_s(x) = A(x)$. We show that $\mathbf{c}\langle A_s \rangle \leq \mathbf{c}\langle B_t \rangle$. Suppose that x is least such that $A_{s(k)}(x) \neq A_{s(k)-1}(x)$. By the use bound on the reduction procedure Γ, there is $y \leq x$ such that $B_t(y) \neq B_{t-1}(y)$ for some t, $s(k+1) < t \leq s(k+2)$. Then $\mathbf{c}(x, s(k)) \leq \mathbf{c}(y,t)$ by monotonicity of $\mathbf{c}$. Therefore $\langle A_s \rangle \models \mathbf{c}$. □

3.2. **Conjunction of cost functions.** In the remainder of this section we characterize conjunction and implication of monotonic cost functions algebraically. Firstly, we show that a set A is a model of $\mathbf{c}$ and $\mathbf{d}$ if and only if A is a model of $\mathbf{c} + \mathbf{d}$. Then we show that $\mathbf{c}$ implies $\mathbf{d}$ if and only if $\underline{\mathbf{d}} = O(\underline{\mathbf{c}})$.

Theorem 3.2. *Let $\mathbf{c}, \mathbf{d}$ be monotonic cost functions. Then*

$$A \models \mathbf{c} \;\&\; A \models \mathbf{d} \;\Leftrightarrow\; A \models \mathbf{c} + \mathbf{d}.$$

Proof. $\Leftarrow$: This implication is trivial.
$\Rightarrow$: We carry out a look-ahead argument of the type introduced in the proof of Proposition 3.1. Suppose that $\langle E_s \rangle_{s\in\mathbb{N}}$ and $\langle F_s \rangle_{s\in\mathbb{N}}$ are computable approximations of a set A such that $\langle E_s \rangle \models \mathbf{c}$ and $\langle F_s \rangle \models \mathbf{d}$. We may assume that $E_s(x) = F_s(x) = 0$ for $s < x$ because changing $E(x)$, say, to 1 at stage x will not increase the cost as $\mathbf{c}(x,s) = 0$ for $x > s$. We define a computable increasing sequence of stages $\langle s(i) \rangle_{i\in\mathbb{N}}$ by letting $s(0) = 0$ and

$$s(i+1) = \mu s > s(i)\, [E_s \upharpoonright_{s(i)} = F_s \upharpoonright_{s(i)}].$$

We define $A_{s(k)}(x)$ for each $k \in \mathbb{N}$. Thereafter we let $A_s(x) = A_{s(k)}(x)$ where k is maximal such that $s(k) \leq s$.

Suppose $s(i) \leq x < s(i+1)$. Let $A_{s(k)}(x) = 0$ for $k < i$. To define $A_{s(k)}(x)$ for $k \geq i$, let $j(x)$ be the least $j \geq i$ such that $v = E_{s(j+1)}(x) = F_{s(j+1)}(x)$.

$$A_{s(k)}(x) = \begin{cases} v & \text{if } i \leq k \leq j(x) \\ E_{s(k+1)}(x) = F_{s(k+1)}(x) & \text{if } k > j(x). \end{cases}$$

Clearly $\lim_s A_s(x) = A(x)$. To show $(\mathbf{c}+\mathbf{d})\langle A_s\rangle < \infty$, suppose that $A_{s(k)}(x) \neq A_{s(k)-1}(x)$. The only possible cost in the case $i \le k \le j(x)$ is at stage $s(i)$ when $v = 1$. Such a cost is bounded by 2^{-x}. XX Now consider a cost in the case $k > j(x)$. There is a least y such that $E_t(y) \neq E_{t-1}(y)$ for some t, $s(k) < t \le s(k+1)$. Then $y \le x$, whence $\mathbf{c}(x, s(k)) \le \mathbf{c}(y,t)$ by the monotonicity of $\mathbf{c}$. Similarly, using $\langle F_s\rangle$ one can bound the cost of changes due to $\mathbf{d}$. Therefore $(\mathbf{c}+\mathbf{d})\langle A_s\rangle \le 4 + \mathbf{c}\langle E_s\rangle + \mathbf{d}\langle F_s\rangle < \infty$. $\square$

3.3. **Implication between cost functions.**

Definition 3.3. 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.

If a cost function $\mathbf{c}$ is monotonic in the stage component, then $\underline{\mathbf{c}}(x) = \sup_s \mathbf{c}(x,s)$. By Remark 2.9 we may assume $\underline{\mathbf{c}}(x)$ is finite for each x. We will show $\mathbf{c} \to \mathbf{d}$ is equivalent to $\underline{\mathbf{d}}(x) = O(\underline{\mathbf{c}}(x))$. In particular, whether or not $A \models \mathbf{c}$ only depends on the limit function $\underline{\mathbf{c}}$.

Theorem 3.4. *Let $\mathbf{c}, \mathbf{d}$ be cost functions that are monotonic in the stage component. Suppose $\mathbf{c}$ satisfies the limit condition in Definition 2.6. Then*

$$\mathbf{c} \to \mathbf{d} \;\Leftrightarrow\; \exists N\, \forall x \left[N\underline{\mathbf{c}}(x) > \underline{\mathbf{d}}(x)\right].$$

Proof. $\Leftarrow$: We carry out yet another look-ahead argument. We define a computable increasing sequence of stages $s(0) < s(1) < \ldots$ by $s(0) = 0$ and

$$s(i+1) = \mu s > s(i).\forall x < s(i) \left[N\mathbf{c}(x,s) > \mathbf{d}(x,s)\right].$$

Suppose A is a Δ^0_2 set with a computable approximation $\langle A_s\rangle \models \mathbf{c}$. We show that $\langle \widetilde{A}_t\rangle \models \mathbf{d}$ for some computable approximation $\langle \widetilde{A}_t\rangle$ of A. As usual, we define $\widetilde{A}_{s(k)}(x)$ for each $k \in \mathbb{N}$. We then let $\widetilde{A}_s(x) = \widetilde{A}_{s(k)}(x)$ where k is maximal such that $s(k) \le s$.

Suppose $s(i) \le x < s(i+1)$. If $k < i+1$ let $\widetilde{A}_{s(k)}(x) = A_{s(i+2)}(x)$. If $k \ge i+1$ let $\widetilde{A}_{s(k)}(x) = A_{s(k+1)}(x)$.

Given k, suppose that x is least such that $\widetilde{A}_{s(k)}(x) \neq \widetilde{A}_{s(k)-1}(x)$. Let i be the number such that $s(i) \le x < s(i+1)$. Then $k \ge i+1$. We have $A_t(x) \neq A_{t-1}(x)$ for some t such that $s(k) < t \le s(k+1)$. Since $x < s(i+1) \le s(k)$, by the monotonicity hypothesis this implies $N\mathbf{c}(x,t) \ge N\mathbf{c}(x,s(k)) > \mathbf{d}(x,s(k))$. So $\mathbf{d}\langle\widetilde{A}_s\rangle \le N \cdot \mathbf{c}\langle A_s\rangle < \infty$. Hence $A \models \mathbf{d}$.

$\Rightarrow$: Recall from the proof of Fact 2.13 that we view the e-th partial computable function Φ_e as a (possibly partial) computable approximation $\langle B_t\rangle$, where $B_t \simeq D_{\Phi_e(t)}$.

Suppose that $\exists N\, \forall x \left[N\underline{\mathbf{c}}(x) > \underline{\mathbf{d}}(x)\right]$ fails. We build a set $A \models \mathbf{c}$ such that for no computable approximation Φ_e of A we have $\mathbf{d}\,\Phi_e \le 1$. This suffices for the theorem by Fact 2.2. We meet the requirements

$$R_e\colon\ \Phi_e \text{ is total and approximates } A \;\Rightarrow\; \Phi_e \not\models \mathbf{d}.$$

The idea is to change $A(x)$ for some fixed x at sufficiently many stages s with $N\mathbf{c}(x,s) < \mathbf{d}(x,s)$, where N is an appropriate large constant. After each change we wait for recovery from the side of Φ_e. In this way our $\mathbf{c}$-cost of changes to A remains bounded, while the opponent's $\mathbf{d}$-cost of changes to Φ_e exceeds 1.

For a stage s, we let $\mathsf{init}_s(e) \le s$ be the largest stage such that R_e has been initialized at that stage (or 0 if there is no such stage). Waiting for recovery is implemented as follows. We say that s is *e-expansionary* if $s = \mathsf{init}_s(e)$, or $s > \mathsf{init}_s(e)$ and, where u is the greatest e-expansionary stage less than s,

$$\exists t \in [u, s)\, [\Phi_{e,s}(t) \downarrow \ \& \ \Phi_{e,s}(t) \upharpoonright_u = A_u \upharpoonright_u].$$

The strategy for R_e can only change $A(x)$ at an e-expansionary stage u such that $x < u$. In this case it preserves $A_u \upharpoonright_u$ until the next e-expansionary stage. Then, Φ_e also has to change its mind on x: we have

$$x \in \Phi_e(u-1) \leftrightarrow x \notin \Phi_e(t) \text{ for some } t \in [u, s).$$

We measure the progress of R_e at stages s via a quantity $\alpha_s(e)$. When R_e is initialized at stage s, we set $\alpha_s(e)$ to 0. If R_e changes $A(x)$ at stage s, we increase $\alpha_s(e)$ by $\mathbf{c}(x, s)$. R_e is declared satisfied when $\alpha_s(e)$ exceeds 2^{-b-e}, where b is the number of times R_e has been initialized.

Construction of $\langle A_s \rangle$ and $\langle \alpha_s \rangle$. Let $A_0 = \emptyset$. Let $\alpha_0(e) = 0$ for each e.
Stage $s > 0$. Let e be least such that s is e-expansionary and $\alpha_{s-1}(e) \le 2^{-b-e}$ where b is the number of times R_e has been initialized so far. If e exists do the following.

Let x be least such that $\mathsf{init}_s(e) \le x < s$, $\mathbf{c}(x, s) < 2^{-b-e}$ and

$$2^{b+e}\mathbf{c}(x, s) < \mathbf{d}(x, s).$$

If x exists let $A_s(x) = 1 - A_{s-1}(x)$. Also let $A_s(y) = 0$ for $x < y < s$. Let $\alpha_s(e) = \alpha_{s-1}(e) + \mathbf{c}(x, s)$. Initialize the requirements R_i for $i > e$ and let $\alpha_s(i) = 0$. (This preserves $A_s \upharpoonright_s$ unless R_e itself is later initialized.) We say that R_e *acts.*

Verification. If s is a stage such that R_e has been initialized for b times, then $\alpha_s(e) \le 2^{-b-e+1}$. Hence the total cost of changes of A due to R_e is at most $\sum_b 2^{-b-e+1} = 2^{-e+2}$. Therefore $\langle A_s \rangle \models \mathbf{c}$.

We show that *each R_e only acts finitely often, and is met.* Inductively, $\mathsf{init}_s(e)$ assumes a final value s_0. Let b be the number of times R_e has been initialized by stage s_0.

Since the condition $\exists N \,\forall x\, [N\underline{\mathbf{c}}(x) > \underline{\mathbf{d}}(x)]$ fails, there is $x \ge s_0$ such that for some $s_1 \ge x$, we have $\forall s \ge s_1\, [2^{b+e}\mathbf{c}(x, s) < \mathbf{d}(x, s)]$. Furthermore, since $\mathbf{c}$ satisfies the limit condition, we may suppose that $\underline{\mathbf{c}}(x) < 2^{-b-e}$. Choose x least.

If Φ_e is a computable approximation of A, there are infinitely many e-expansionary stages $s \ge s_1$. For each such s, we can choose this x at stage s in the construction. So we can add at least $\mathbf{c}(x, s_1)$ to $\alpha(e)$. Therefore $\alpha_t(e)$ exceeds the bound 2^{-b-e} for some stage $t \ge s_1$, whence R_e stops acting at t. Furthermore, since $\mathbf{d}$ is monotonic in the second component and by the initialization due to R_e, between stages s_0 and t we have caused $\mathbf{d}\,\Phi_e$ to increase by at least $2^{b+e}\alpha_t(e) > 1$. Hence R_e is met. $\square$

The foregoing proof uses in an essential way the ability to change $A(x)$, for the same x, for a multiple number of times. If we restrict implication to c.e. sets, the implication from left to right in Theorem 3.4 fails. For a trivial example, let $\mathbf{c}(x,s) = 4^{-x}$ and $\mathbf{d}(x,s) = 2^{-x}$. Then each c.e. set obeys $\mathbf{d}$, so $\mathbf{c} \to \mathbf{d}$ for c.e. sets. However, we do not have $\mathbf{d}(x) = O(\mathbf{c}(x))$.

We mention that Melnikov and Nies (unpublished, 2010) have obtained a sufficient algebraic condition for the non-implication of cost functions via a c.e. set. Informally speaking, the condition $\mathbf{d}(x) = O(\mathbf{c}(x))$ fails "badly".

Proposition 3.5. *Let* $\mathbf{c}$ *and* $\mathbf{d}$ *be monotonic cost functions satisfying the limit condition such that* $\sum_{x\in\mathbb{N}} \underline{\mathbf{d}}(x) = \infty$ *and, for each* $N > 0$,

$$\sum \underline{\mathbf{d}}(x)[\![N\underline{\mathbf{c}}(x) > \underline{\mathbf{d}}(x)]\!] < \infty.$$

Then there exists a c.e. set A *that obeys* $\mathbf{c}$*, but not* $\mathbf{d}$.

The hope is that some variant of this will yield an algebraic criterion for cost function implication restricted to the c.e. sets.

4. Additive cost functions

We discuss a class of very simple cost functions introduced in [23]. We show that a Δ^0_2 set obeys all of them if and only if it is K-trivial. There is a universal cost function of this kind, namely $\mathbf{c}(x,s) = \Omega_s - \Omega_x$. Recall Convention 2.9 that $\underline{\mathbf{c}}(x) < \infty$ for each cost function $\mathbf{c}$.

Definition 4.1 ([23])**.** *We say that a cost function* $\mathbf{c}$ *is* additive *if* $\mathbf{c}(x,s) = 0$ *for* $x > s$*, and for each* $x < y < z$ *we have*

$$\mathbf{c}(x,y) + \mathbf{c}(y,z) = \mathbf{c}(x,z).$$

Additive cost functions correspond to nondecreasing effective sequences $\langle\beta_s\rangle_{s\in\mathbb{N}}$ of non-negative rationals, that is, to effective approximations of left-c.e. reals β. Given such an approximation $\langle\beta\rangle = \langle\beta_s\rangle_{s\in\mathbb{N}}$, let for $x \le s$

$$\mathbf{c}_{\langle\beta\rangle}(x,s) = \beta_s - \beta_x.$$

Conversely, given an additive cost function $\mathbf{c}$, let $\beta_s = \mathbf{c}(0,s)$. Clearly the two effective transformations are inverses of each other.

4.1. K-triviality and the cost function $\mathbf{c}_{\langle\Omega\rangle}$.

The standard cost function $\mathbf{c}_{\mathcal{K}}$ introduced in (2) is *not* additive. We certainly have $\mathbf{c}_{\mathcal{K}}(x,y)+\mathbf{c}_{\mathcal{K}}(y,z) \le \mathbf{c}_{\mathcal{K}}(x,z)$, but by stage z there could be a shorter description of, say, $x+1$ than at stage y, so that the inequality may be proper. On the other hand, let g be a computable function such that $\sum_w 2^{-g(w)} < \infty$; this implies that $K(x) \le^+ g(x)$. The "analog" of $\mathbf{c}_{\mathcal{K}}$ when we write $g(x)$ instead of $K_s(x)$, namely $\mathbf{c}_g(x,s) = \sum_{w=x+1}^{s} 2^{-g(w)}$ is an additive cost function.

Also, $\mathbf{c}_{\mathcal{K}}$ is dominated by an additive cost function $\mathbf{c}_{\langle\Omega\rangle}$ we introduce next. Let $\mathbb{U}$ be the standard universal prefix-free machine (see e.g. [22, Ch. 2]). Let $\langle\Omega\rangle$ denote the computable approximation of Ω given by $\Omega_s = \lambda\,\mathrm{dom}(\mathbb{U}_s)$. (That is, Ω_s is the Lebesgue measure of the domain of the universal prefix-free machine at stage s.)

Fact 4.2. *For each* $x \le s$*, we have* $\mathbf{c}_{\mathcal{K}}(x,s) \le \mathbf{c}_{\langle\Omega\rangle}(x,s) = \Omega_s - \Omega_x$.

Proof. Fix x. We prove the statement by induction on $s \ge x$. For $s = x$ we have $\mathbf{c}_{\mathcal{K}}(x,s) = 0$. Now

$$\mathbf{c}_{\mathcal{K}}(x,s+1) - \mathbf{c}_{\mathcal{K}}(x,s) = \textstyle\sum_{w=x+1}^{s+1} 2^{-K_{s+1}(w)} - \sum_{w=x+1}^{s} 2^{-K_s(w)} \le \Omega_{s+1} - \Omega_s,$$

because the difference is due to convergence at stage s of new $\mathbb{U}$-computations. $\square$

Theorem 4.3. *Let A be Δ^0_2. Then the following are equivalent.*

(i) *A is K-trivial.*
(ii) *A obeys each additive cost function.*
(iii) *A obeys $\mathbf{c}_{\langle\Omega\rangle}$, where $\Omega_s = \lambda dom(\mathbb{U}_s)$.*

Proof. (ii) $\to$ (iii) is immediate, and (iii) $\to$ (i) follows from Fact 4.2. It remains to show (i)$\to$(ii).

Fix some computable approximation $\langle A_s\rangle_{s\in\mathbb{N}}$ of A. Let $\mathbf{c}$ be an additive cost function. We may suppose that $\underline{\mathbf{c}}(0) \le 1$.

For $w > 0$ let $r_w \in \mathbb{N} \cup \infty$ be least such that $2^{-r_w} \le c(w-1, w)$ (where $2^{-\infty} = 0$). Then $\sum_w 2^{-r_w} \le 1$. Hence by the Machine Existence Theorem we have $K(w) \le^+ r_w$ for each w. This implies $2^{-r_w} = O(2^{-K(w)})$, so $\sum_{w>x} 2^{-r_w} = O(\underline{\mathbf{c}_{\mathcal{K}}}(x))$ and hence $\underline{\mathbf{c}}(x) = \sum_{w>x} c(w-1, w) = O(\underline{\mathbf{c}_{\mathcal{K}}}(x))$. Thus $\mathbf{c}_{\mathcal{K}} \to \mathbf{c}$ by Theorem 3.4, whence the K-trivial set A obeys $\mathbf{c}$. (See [3] for a proof not relying on Theorem 3.4.) $\square$

Because of Theorem 3.4, we have $\mathbf{c}_{\langle\Omega\rangle} \leftrightarrow \mathbf{c}_{\mathcal{K}}$. That is,

$$\Omega - \Omega_x \sim \textstyle\sum_{w=x+1}^{\infty} 2^{-K(w)}.$$

This can easily be seen directly: for instance, $\mathbf{c}_{\mathcal{K}} \le \mathbf{c}_{\langle\Omega\rangle}$ by Fact 4.2.

4.2. **Solovay reducibility.** Let $\mathbb{Q}_2$ denote the dyadic rationals, and let the variable q range over $\mathbb{Q}_2$. Recall Solovay reducibility on left-c.e. reals: $\beta \le_S \alpha$ iff there is a partial computable $\phi\colon \mathbb{Q}_2 \cap [0, \alpha) \to \mathbb{Q}_2 \cap [0, \beta)$ and $N \in \mathbb{N}$ such that

$$\forall q < \alpha\big[\beta - \phi(q) < N(\alpha - q)\big].$$

Informally, it is easier to approximate β from the left, than α. See e.g. [22, 3.2.8] for background.

We will show that reverse implication of additive cost functions corresponds to Solovay reducibility on the corresponding left-c.e. reals. Given a left-c.e. real γ, we let the variable $\langle\gamma\rangle$ range over the nondecreasing effective sequences of rationals converging to γ.

Proposition 4.4. *Let α, β be left-c.e. reals. The following are equivalent.*

(i) *$\beta \le_S \alpha$*
(ii) *$\forall\langle\alpha\rangle\exists\langle\beta\rangle\,[c_{\langle\alpha\rangle} \to c_{\langle\beta\rangle}]$*
(iii) *$\exists\langle\alpha\rangle\exists\langle\beta\rangle\,[c_{\langle\alpha\rangle} \to c_{\langle\beta\rangle}]$.*

Proof. (i) $\to$ (ii). Given an effective sequence $\langle\alpha\rangle$, by the definition of $\le_S$ there is an effective sequence $\langle\beta\rangle$ such that $\beta - \beta_x = O(\alpha - \alpha_x)$ for each x. Thus $\underline{\mathbf{c}}_{\langle\beta\rangle} = O(\underline{\mathbf{c}}_{\langle\alpha\rangle})$. Hence $\mathbf{c}_{\langle\alpha\rangle} \to \mathbf{c}_{\langle\beta\rangle}$ by Theorem 3.4.

(iii) $\to$ (i). Suppose we are given $\langle\alpha\rangle$ and $\langle\beta\rangle$ such that $\underline{\mathbf{c}}_{\langle\beta\rangle} = O(\underline{\mathbf{c}}_{\langle\alpha\rangle})$. Define a partial computable function ϕ by $\phi(q) = \beta_x$ if $\alpha_{x-1} \le q < \alpha_x$. Then $\beta \le_S \alpha$ via ϕ. $\square$

4.3. **The strength of an additive cost function.** Firstly, we make some remarks related to Proposition 4.4. For instance, it implies that an additive cost function can be weaker than $\mathbf{c}_{\langle\Omega\rangle}$ without being obeyed by all the Δ^0_2 sets.

Proposition 4.5. *There are additive cost functions $\mathbf{c}, \mathbf{d}$ such that $\mathbf{c}_{\langle\Omega\rangle} \to \mathbf{c}$, $\mathbf{c}_{\langle\Omega\rangle} \to \mathbf{d}$ and $\mathbf{c}, \mathbf{d}$ are incomparable under the implication of cost functions.*

Proof. Let $\mathbf{c}, \mathbf{d}$ be cost functions corresponding to enumerations of Turing (and hence Solovay) incomparable left-c.e. reals. Now apply Prop. 4.4. ☐

Clearly, if β is a computable real then any c.e. set obeys $\mathbf{c}_{\langle\beta\rangle}$. The intuition we garner from Prop. 4.4 is that a more complex left-c.e. real β means that the sets $A \models \mathbf{c}_{\langle\beta\rangle}$ become less complex, and conversely. We give a little more evidence for this principle: if β is non-computable, we show that a set $A \models \mathbf{c}_{\langle\beta\rangle}$ cannot be weak truth-table complete. However, we also build a non-computable β and a c.e. Turing complete set that obeys $\mathbf{c}_{\langle\beta\rangle}$

Proposition 4.6. *Suppose β is a non-computable left-c.e. real and $A \models \mathbf{c}_{\langle\beta\rangle}$. Then A is not weak truth-table complete.*

Proof. Assume for a contradiction that A is weak truth-table complete. We can fix a computable approximation $\langle A_s\rangle$ of A such that $\mathbf{c}_{\langle\beta\rangle}\langle A_s\rangle \le 1$. We build a c.e. set B. By the recursion theorem we can suppose we have a weak truth-table reduction Γ with computable use bound g such that $B = \Gamma^A$. We build B so that $\beta - \beta_{g(2^{e+1})} \le 2^{-e}$, which implies that β is computable.

Let $I_e = [2^e, 2^{e+1})$. If ever a stage s appears such that $\beta_s - \beta_{g(2^{e+1})} \le 2^{-e}$, then we start enumerating into $B \cap I_e$ sufficiently slowly so that $A \upharpoonright_{g(2^{e+1})}$ must change 2^e times. To do so, each time we enumerate into B, we wait for a recovery of $B = \Gamma^A$ up to $2^{(e+1)}$. The A-changes we enforce yield a total cost > 1 for a contradiction. ☐

Proposition 4.7. *There is a non-computable left-c.e. real β and a c.e. set $A \models \mathbf{c}_{\langle\beta\rangle}$ such that A is Turing complete.*

Proof. We build a Turing reduction Γ such that $\emptyset' = \Gamma(A)$. Let $\gamma_{k,s} + 1$ be the use of the computation $\Gamma^{\emptyset'}(k)[s]$. We view γ_k as a movable marker as usual. The initial value is $\gamma_{k,0} = k$. Throughout the construction we maintain the invariant

$$\beta_s - \beta_{\gamma_{k,s}} \le 2^{-k}.$$

Let $\langle\phi_e\rangle$ be the usual effective list of partial computable functions. By convention, at each stage at most one computation $\phi_e(k)$ converges newly. To make β non-computable, it suffices to meet the requirements

$$R_k\colon\ \phi_k(k)\downarrow\ \Rightarrow\ \beta - \beta_{\phi_k(k)} \ge 2^{-k}.$$

Strategy for R_k. If $\phi_k(k)$ converges newly at stage s, do the following.

1. Enumerate $\gamma_{k,s}$ into A. (This incurs a cost of at most 2^{-k}.)
2. Let $\beta_s = \beta_{s-1} + 2^{-k}$.
3. Redefine γ_i $(i \ge k)$ to large values in an increasing fashion.

In the construction, we run the strategies for the R_k. If k enters $\emptyset'$ at stage s, we enumerate $\gamma_{k,s}$ into A.

Clearly each R_k acts at most once, and is met. Therefore β is non-computable. The markers γ_k reach a limit. Therefore $\emptyset' = \Gamma(A)$. Finally, we maintain the stage invariant, which implies that the total cost of enumerating A is at most 4. $\square$

As pointed out by Turetsky, it can be verified that β is in fact Turing complete.

Next, we note that if we have two computable approximations from the left of the same real, we obtain additive cost functions with very similar classes of models.

Proposition 4.8. *Let $\langle\alpha\rangle, \langle\beta\rangle$ be left-c.e. approximations of the same real. Suppose that $A \models \mathbf{c}_{\langle\alpha\rangle}$. Then there is $B \equiv_m A$ such that $B \models \mathbf{c}_{\langle\beta\rangle}$. If A is c.e., then B can be chosen c.e. as well.*

Proof. Firstly, suppose that A is c.e. By Fact 2.5 choose a computable enumeration $\langle A_s\rangle \models \mathbf{c}_{\langle\alpha\rangle}$.

By the hypothesis on the sequences $\langle\alpha\rangle$ and $\langle\beta\rangle$, there is a computable sequence of stages $s_0 < s_1 < \ldots$ such that $|\alpha_{s_i} - \beta_{s_i}| \leq 2^{-i}$. Let f be a strictly increasing computable function such that $\alpha_x \leq \beta_{f(x)}$ for each x.

To define B, if x enters A at stage s, let i be greatest such that $s_i \leq s$. If $f(x) \leq s_i$ put $f(x)$ into B at stage s_i.

Clearly

$$\alpha_s - \alpha_x \geq \alpha_{s_i} - \alpha_x \geq \alpha_{s_i} - \beta_{f(x)} \geq \beta_{s_i} - \beta_{f(x)} - 2^{-i}.$$

So $\mathbf{c}_{\langle\beta\rangle}\langle B_s\rangle \leq \mathbf{c}_{\langle\alpha\rangle}\langle A_s\rangle + \sum_i 2^{-i}$.

Let R be the computable subset of A consisting of those x that are enumerated early, namely x enters A at a stage s and $f(x) > s_i$ where i is greatest such that $s_i \leq s$. Clearly $B = f(A - R)$. Hence $B \equiv_m A$.

The argument can be adapted to the case that A is Δ^0_2. Given a computable approximation $\langle A_s\rangle$ obeying $\mathbf{c}_{\langle\alpha\rangle}$, let t be the least s_i such that $s_i \geq f(x)$. For $s \leq t$ let $B_s(f(x)) = A_t(x)$. For $s > t$ let $B_s(f(x)) = A_{s_i}(x)$ where $s_i \leq s < s_{i+1}$. $\square$

5. Randomness, lowness, and K-triviality

Benign cost functions were briefly discussed in the introduction.

Definition 5.1 ([11])**.** A monotonic cost function $\mathbf{c}$ is called *benign* if there is a computable function g such that for all k,

$$x_0 < x_1 < \ldots < x_k \ \& \ \forall i < k\, [\mathbf{c}(x_i, x_{i+1}) \geq 2^{-n}] \text{ implies } k \leq g(n).$$

Clearly such a cost function satisfies the limit condition. Indeed, $\mathbf{c}$ satisfies the limit condition if and only if the above holds for some $g \leq_T \emptyset'$. For example, the cost function $\mathbf{c}_{\mathcal{K}}$ is benign via $g(n) = 2^n$. Each additive cost function is benign where $g(n) = O(2^n)$. For more detail see [11] or [22, Section 8.5].

For definitions and background on the extreme lowness property called strong jump traceability, see [11, 10] or [22, Ch. 8]. We will use the main result in [11] already quoted in the introduction: a c.e. set A is strongly jump traceable iff A obeys each benign cost function.

5.1. **A cost function implying strong jump traceability.** The following type of cost functions first appeared in [11] and [22, Section 5.3]. Let $Z \in \Delta^0_2$ be ML-random. Fix a computable approximation $\langle Z_s \rangle$ of Z and let $\mathbf{c}_Z$ (or, more accurately, $\mathbf{c}_{\langle Z_s \rangle}$) be the cost function defined as follows. Let $\mathbf{c}_Z(x,s) = 2^{-x}$ for each $x \geq s$; if $x < s$, and $e < x$ is least such that $Z_{s-1}(e) \neq Z_s(e)$, we let

$$c_Z(x,s) = \max(c_Z(x,s-1), 2^{-e}). \tag{3}$$

Then $A \models \mathbf{c}_Z$ implies $A \leq_T Z$ by the aforementioned result from [11], which is proved like its variant above.

A Demuth test is a sequence of c.e. open sets $(S_m)_{m \in \mathbb{N}}$ such that

- $\forall m \, \lambda S_m \leq 2^{-m}$, and there is a function f such that S_m is the Σ^0_1 class $[W_{f(m)}]^{\prec}$;
- $f(m) = \lim_s g(m,s)$ for a computable function g such that the size of the set $\{s \colon\ g(m,s) \neq g(m,s-1)\}$ is bounded by a computable function $h(m)$.

A set Z passes the test if $Z \not\in S_m$ for almost every m. We say that Z is Demuth random if Z passes each Demuth test. For background on Demuth randomness see [22, pg. 141].

Proposition 5.2. *Suppose Y is a Demuth random Δ^0_2 set and $A \models c_Y$. Then $A \leq_T Z$ for each ω-c.e. ML-random set Z.*

In particular, A is strongly jump traceable by [10].

Proof. Let $G^s_e = [Y_t \upharpoonright_e]$ where $t \leq s$ is greatest such that $Z_t(e) \neq Z_{t-1}(e)$. Let $G_e = \lim_s G^s_e$. (Thus, we only update G_e when $Z(e)$ changes.) Then $(G_e)_{e \in \mathbb{N}}$ is a Demuth test. Since Y passes this test, there is e_0 such that

$$\forall e \geq e_0 \, \forall t \, [Z_t(e) \neq Z_{t-1}(e) \to \exists s > t \ Y_{s-1} \upharpoonright_e \neq Y_s \upharpoonright_e].$$

We use this fact to define a computable approximation $(\widehat{Z}_u)$ of Z as follows: let $\widehat{Z}_u(e) = Z(e)$ for $e \leq e_0$; for $e > e_0$ let $\widehat{Z}_u(e) = Z_s(e)$ where $s \leq u$ is greatest such that $Y_{s-1} \upharpoonright_e \neq Y_s \upharpoonright_e$.

Note that $\mathbf{c}_{\widehat{Z}}(x,s) \leq c_Y(x,s)$ for all x, s. Hence $A \models \mathbf{c}_{\widehat{Z}}$ and therefore $A \leq_T Z$. $\square$

Recall that some Demuth random set is Δ^0_2. Kučera and Nies [18] in their main result strengthened the foregoing proposition in the case of a c.e. sets A: if $A \leq_T Y$ for some Demuth random set Y, then A is strongly jump traceable. Greenberg and Turetsky [12] obtained the converse of this result: every c.e. strongly jump traceable is below a Demuth random.

Remark 5.3. For each Δ^0_2 set Y we have $\mathbf{c}_Y(x) = 2^{-F(x)}$ where F is the Δ^0_2 function such that

$$F(x) = \min\{e \colon\ \exists s > x \, Y_s(e) \neq Y_{s-1}(e)\}.$$

Thus F can be viewed as a modulus function in the sense of [24].

For a computable approximation Φ define the cost function $\mathbf{c}_\Phi$ as in (3). The following (together with Rmk. 5.3) implies that any computable approximation Φ of a ML-random Turing incomplete set changes late at small numbers, because the convergence of Ω_s to Ω is slow.

Corollary 5.4. *Let $Y <_T \emptyset'$ be a ML-random set. Let Φ be any computable approximation of Y. Then $\mathbf{c}_\Phi \to \mathbf{c}_{\mathcal{K}}$ and therefore $O(c_\Phi(x)) = \mathbf{c}_{\langle\Omega\rangle}(x)$.*

Proof. If $A \models \mathbf{c}_\Phi$ then $C \models \mathbf{c}_\Phi$ where $C \geq_T A$ is the change set of the given approximation of A as in Prop. 2.14. By [13] (also see [22, 5.1.23]), C and therefore A is K-trivial. Hence $A \models \mathbf{c}_{\langle\Omega\rangle}$. □

5.2. **Strongly jump traceable sets and d.n.c. functions.** Recall that we write $X \leq_{ibT} Y$ if $X \leq_T Y$ with use function bounded by the identity. When building prefix-free machines, we use the terminology of [22, Section 2.3] such as Machine Existence Theorem (also called the Kraft-Chaitin Theorem), bounded request set etc.

Theorem 5.5. *Suppose an ω-c.e. set Y is diagonally noncomputable via a function that is weak truth-table below Y. Let A be a strongly jump traceable c.e. set. Then $A \leq_{ibT} Y$.*

Proof. By [14] (also see [22, 4.1.10]) there is an order function h such that $2h(n) \leq^+ K(Y \upharpoonright_n)$ for each n. The argument of the present proof goes back to Kučera's injury free solution to Post's problem (see [22, Section 4.2]). Our proof is phrased in the language of cost functions, extending the similar result in [11] where Y is ML-random (equivalently, the condition above holds with $h(n) = \lfloor n/2 \rfloor + 1$.

Let $\langle Y_s \rangle$ be a computable approximation via which Y is ω-c.e. To help with building a reduction procedure for $A \leq_{ibT} Y$, via the Machine Existence Theorem we give prefix-free descriptions of initial segments $Y_s \upharpoonright_e$. On input x, if at a stage $s > x$, e is least such that $Y(e)$ has changed between stages x and s, then we still hope that $Y_s \upharpoonright_e$ is the final version of $Y \upharpoonright_e$. So whenever $A(x)$ changes at such a stage s, we give a description of $Y_s \upharpoonright_e$ of length $h(e)$. By hypothesis A is strongly jump traceable, and hence obeys each benign cost function. We define an appropriate benign cost function $\mathbf{c}$ so that a set A that obeys $\mathbf{c}$ changes little enough that we can provide all the descriptions needed.

To ensure that $A \leq_{ibT} Y$, we define a computation $\Gamma(Y \upharpoonright_x)$ with output $A(x)$ at the least stage $t \geq x$ such that $Y_t \upharpoonright_x$ has the final value. If Y satisfies the hypotheses of the theorem, $A(x)$ cannot change at any stage $s > t$ (for almost all x), for otherwise $Y \upharpoonright_e$ would receive a description of length $h(e) + O(1)$, where e is least such that $Y(e)$ has changed between x and s.

We give the details. Firstly we give a definition of a cost function $\mathbf{c}$ which generalizes the definition in (3). Let $\mathbf{c}(x, s) = 0$ for each $x \geq s$. If $x < s$, and $e < x$ is least such that $Y_{s-1}(e) \neq Y_s(e)$, let

$$\mathbf{c}(x, s) = \max(\mathbf{c}(x, s-1), 2^{-h(e)}). \tag{4}$$

Since Y is ω-c.e., $\mathbf{c}$ is benign. Thus each strongly jump traceable c.e. set obeys $\mathbf{c}$ by the main result in [11]. So it suffices to show that $A \models \mathbf{c}$ implies

$A \leq_{ibT} Y$ for any set A. Suppose that $\mathbf{c}\langle A_s\rangle \leq 2^u$. Enumerate a bounded request set L as follows. When $A_{s-1}(x) \neq A_s(x)$ and e is least such that $e = x$ or $Y_{t-1}(e) \neq Y_t(e)$ for some $t \in [x, s)$, put the request $\langle u + h(e), Y_s \upharpoonright_e\rangle$ into L. Then L is indeed a bounded request set.

Let d be a coding constant for L (see [22, Section 2.3]). Choose e_0 such that $h(e) + u + d < 2h(e)$ for each $e \geq e_0$. Choose $s_0 \geq e_0$ such that $Y \upharpoonright_{e_0}$ is stable from stage s_0 on.

To show $A \leq_{ibT} Y$, given an input $x \geq s_0$, using Y as an oracle, compute $t > x$ such that $Y_t \upharpoonright_x = Y \upharpoonright_x$. We claim that $A(x) = A_t(x)$. Otherwise $A_s(x) \neq A_{s-1}(x)$ for some $s > t$. Let $e \leq x$ be the largest number such that $Y_r \upharpoonright_e = Y_t \upharpoonright_e$ for all r with $t < r \leq s$. If $e < x$ then $Y(e)$ changes in the interval $(t, s]$ of stages. Hence, by the choice of $t \geq s_0$, we cause $K(y) < 2h(e)$ where $y = Y_t \upharpoonright_e = Y \upharpoonright_e$, contradiction. $\square$

Example 5.6. For each order function h and constant d, the class

$$P_{h,d} = \{Y :\ \forall n\, 2h(n) \leq K(Y \upharpoonright_n) + d\}$$

is Π^0_1. Thus, by the foregoing proof, each strongly jump traceable c.e. set is ibT below each ω-c.e. member of $P_{h,d}$.

We discuss the foregoing Theorem 5.5, and relate it to results in [10, 11].

1. In [10, Thm 2.9] it is shown that given a non-empty Π^0_1 class P, each jump traceable set A Turing below each superlow member of P is already strongly jump traceable. In particular this applies to superlow c.e. sets A, since such sets are jump traceable [20]. For many non-empty Π^0_1 classes such a set is in fact computable. For instance, it could be a class where any two distinct members form a minimal pair. In contrast, the nonempty among the Π^0_1 classes $P = P_{h,d}$ are examples where being below each superlow (or ω-c.e.) member characterizes strong jump traceability for c.e. sets.

2. Each superlow set A is weak truth-table below *some* superlow set Y as in the hypothesis of Theorem 5.5. For let P be the class of $\{0, 1\}$-valued d.n.c. functions. By [22, 1.8.41] there is a set $Z \in P$ such that $(Z \oplus A)' \leq_{\mathrm{tt}} A'$. Now let $Y = Z \oplus A$. This contrasts with the case of ML-random covers: if a c.e. set A is not K-trivial, then each ML-random set Turing above A is already Turing above $\emptyset'$ by [13]. Thus, in the case of ibT reductions, Theorem 5.5 applies to more oracle sets Y than [11, Prop. 5.2].

3. Greenberg and Nies [11, Prop. 5.2] have shown that for each order function p, each strongly jump traceable c.e. set is Turing below below each ω-c.e. ML-random set, via a reduction with use bounded by p. We could also strengthen Theorem 5.5 to yield such a "p-bounded" Turing reduction.

5.3. **A proper implication between cost functions.** In this subsection we study a weakening of K-triviality using the monotonic cost function

$$\mathbf{c}_{\max}(x, s) = \max\{2^{-K_s(w)} :\ x < w \leq s\}.$$

Note that $\mathbf{c}_{\max}$ satisfies the limit condition, because

$$\underline{\mathbf{c}}_{\max}(x) = \max\{2^{-K(w)} :\ x < w\}.$$

Clearly $\mathbf{c}_{\max}(x,s) \le \mathbf{c}_{\mathcal{K}}(x,s)$, whence $\mathbf{c}_{\mathcal{K}} \to \mathbf{c}_{\max}$. We will show that this implication of cost functions is proper. Thus, some set obeys $\mathbf{c}_{\max}$ that is not K-trivial.

Firstly, we investigate sets obeying $\mathbf{c}_{\max}$. For a string α, let $g(\alpha)$ be the longest prefix of α that ends in 1, and $g(\alpha) = \emptyset$ if there is no such prefix.

Definition 5.7. We say that a set A is *weakly K-trivial* if

$$\forall n\,[K(g(A\restriction_n)) \le^+ K(n)].$$

Clearly, every K-trivial set is weakly K-trivial. By the following, every *c.e.* weakly K-trivial set is already K-trivial.

Fact 5.8. *If A is weakly K-trivial and not h-immune, then A is K-trivial.*

Proof. By the second hypothesis, there is an increasing computable function p such that $[p(n), p(n+1)) \cap A \neq \emptyset$ for each n. Then

$$K(A\restriction_{p(n)}) \le^+ K(g(A\restriction_{p(n+1)})) \le^+ K(p(n+1)) \le^+ K(p(n)).$$

This implies that A is K-trivial by [22, Ex. 5.2.9]. $\square$

We say that a computable approximation $\langle A_s\rangle_{s\in\mathbb{N}}$ is *erasing* if for each x and each $s > 0$, $A_s(x) \neq A_{s-1}(x)$ implies $A_s(y) = 0$ for each y such that $x < y \le s$. For instance, the computable approximation built in the proof of the implication "$\Rightarrow$" of Theorem 3.4 is erasing by the construction.

Proposition 5.9. *Suppose $\langle A_s\rangle_{s\in\mathbb{N}}$ is an erasing computable approximation of a set A, and $\langle A_s\rangle \models \mathbf{c}_{\max}$. Then A is weakly K-trivial.*

Proof. This is a modification of the usual proof that every set A obeying $\mathbf{c}_{\mathcal{K}}$ is K-trivial (see, for instance, [22, Thm. 5.3.10]).

To show that A is weakly K-trivial, one builds a bounded request set W. When at stage $s > 0$ we have $r = K_s(n) < K_{s-1}(n)$, we put the request $\langle r+1, g(A\restriction_n)\rangle$ into W. When $A_s(x) \neq A_{s-1}(x)$, let r be the number such that $\mathbf{c}_{\max}(x,s) = 2^{-r}$, and put the request $\langle r+1, g(A\restriction_{x+1})\rangle$ into W.

Since the computable approximation $\langle A_s\rangle_{s\in\mathbb{N}}$ obeys $\mathbf{c}_{\max}$, the set W is indeed a bounded request set; since $\langle A_s\rangle_{s\in\mathbb{N}}$ is erasing, this bounded request set shows that A is weakly K-trivial. $\square$

We now prove that $\mathbf{c}_{\max} \not\to \mathbf{c}_{\mathcal{K}}$. We do so via proving a reformulation that is of interest by itself.

Theorem 5.10. *For every $b \in \mathbb{N}$ there is an x such that $\underline{\mathbf{c}}_{\mathcal{K}}(x) \ge 2^b \underline{\mathbf{c}}_{\max}(x)$. In other words,*

$$\textstyle\sum\{2^{-K(w)}\colon\, x < w\} \ge 2^b \max\{2^{-K(w)}\colon\, x < w\}.$$

By Theorem 3.4, the statement of the foregoing Theorem is equivalent to $\mathbf{c}_{\max} \not\to \mathbf{c}_{\mathcal{K}}$. Thus, as remarked above, some set A obeys $\mathbf{c}_{\max}$ via an erasing computable approximation, and does not obey $\mathbf{c}_{\mathcal{K}}$. By Proposition 5.9 we obtain a separation.

Corollary 5.11. *Some weakly K-trivial set fails to be K-trivial.*

Melnikov and Nies [19, Prop. 3.7] have given an alternative proof of the preceding result by constructing a weakly K-trivial set that is Turing complete.

Proof of Theorem 5.10. Assume that there is $b \in \mathbb{N}$ such that

$$\forall x\, [\underline{\mathbf{c}}_{\mathcal{K}}(x) < 2^b \underline{\mathbf{c}}_{\max}(x)].$$

To obtain a contradiction, the idea is that $\mathbf{c}_{\mathcal{K}}(x,s)$, which is defined as a sum, can be made large in many small bits; in contrast, $\mathbf{c}_{\max}(x,s)$, which depends on the value $2^{-K_s(w)}$ for a single w, cannot.

We will define a sequence $0 = x_0 < x_1 < \ldots < x_N$ for a certain number N. When x_v has been defined for $v < N$, for a certain stage $t > x_v$ we cause $\mathbf{c}_{\mathcal{K}}(x_v, t)$ to exceed a fixed quantity proportional to $1/N$. We wait until the opponent responds at a stage $s > t$ with some $w > x_v$ such that $2^{-K_s(w)}$ corresponding to that quantity. Only then, we define $x_{v+1} = s$. For us, the cost $\mathbf{c}_{\mathcal{K}}(x_i, x_j)$ will accumulate for $i < j$, while the opponent has to provide a new w each time. This means that eventually he will run out of space in the domain of the prefix-free machine giving short descriptions of such w's.

In the formal construction, we will build a bounded request set L with the purpose to cause $\mathbf{c}_{\mathcal{K}}(x,s)$ to be large when it is convenient to us. We may assume by the recursion theorem that the coding constant for L is given in advance (see [22, Remark 2.2.21] for this standard argument). Thus, if we put a request $\langle n, y+1\rangle$ into L at a stage y, there will be a stage $t > y$ such that $K_t(y+1) \le n + d$, and hence $\mathbf{c}_{\mathcal{K}}(x,t) \ge \mathbf{c}_{\mathcal{K}}(x,y) + 2^{-n-d}$.

Let $\boldsymbol{k} = 2^{b+d+1}$. Let $N = 2^{\boldsymbol{k}}$.

Construction of L and a sequence $0 = x_0 < x_1 < \ldots < x_N$ of numbers.

Suppose $v < N$ and x_v has already been defined. Put $\langle \boldsymbol{k}, x_v + 1\rangle$ into L. As remarked above, we may wait for a stage $t > x_v$ such that $\mathbf{c}_{\mathcal{K}}(x_v, t) \ge 2^{-\boldsymbol{k}-d}$. Now, by our assumption, we have $\underline{\mathbf{c}}_{\mathcal{K}}(x_i) < 2^b \underline{\mathbf{c}}_{\max}(x_i)$ for each $i \le v$. Hence we can wait for a stage $s > t$ such that

$$\forall i \le v\, \exists w\, \big[x_i < w \le s \;\&\; \mathbf{c}_{\mathcal{K}}(x_i, s) \le 2^{b-K_s(w)}\big]. \tag{5}$$

Let $x_{v+1} = s$. This ends the construction.

Verification. Note that L is indeed a bounded request set. Clearly we have $\mathbf{c}_{\mathcal{K}}(x_i, x_{i+1}) \ge 2^{-\boldsymbol{k}-d}$ for each $i < N$.

Claim 5.12. *Let $r \le \boldsymbol{k}$. Write $R = 2^r$. Suppose $p + R \le N$. Let $s = x_{p+R}$. Then we have*

$$\sum_{w=x_p+1}^{x_{(p+R)}} \min(2^{-K_s(w)}, 2^{-\boldsymbol{k}-b-d+r}) \ge (r+1)2^{-\boldsymbol{k}-b-d+r-1}. \tag{6}$$

For $r = \boldsymbol{k}$, the right hand side equals $(\boldsymbol{k}+1)2^{-(b+d+1)} > 1$, which is a contradiction because the left hand side is at most $\Omega \le 1$.

We prove the claim by induction on r. To verify the case $r = 0$, note that by (5) there is $w \in (x_p, x_{p+1}]$ such that $\mathbf{c}_{\mathcal{K}}(x_p, x_{p+1}) \le 2^{b-K_s(w)}$. Since $2^{-\boldsymbol{k}-d} \le \mathbf{c}_{\mathcal{K}}(x_p, x_{p+1})$, we obtain

$$2^{-\boldsymbol{k}-b-d} \le 2^{-K_s(w)} \text{ (where } s = x_{p+1}).$$

Thus the left hand side in the inequality (6) is at least $2^{-\boldsymbol{k}-b-d}$, while the right hand side equals $2^{-\boldsymbol{k}-b-d-1}$, and the claim holds for $r = 0$.

In the following, for $i < j \leq N$, we will write $\mathcal{S}(x_i, x_j)$ for a sum of the type occurring in (6) where w ranges from $x_i + 1$ to x_j.

Suppose inductively the claim has been established for $r < \boldsymbol{k}$. To verify the claim for $r + 1$, suppose that $p + 2R \leq N$ where $R = 2^r$ as before. Let $s = x_{p+2R}$. Since $\mathbf{c}_{\mathcal{K}}(x_i, x_{i+1}) \geq 2^{-\boldsymbol{k}-d}$, we have

$$\mathbf{c}_{\mathcal{K}}(x_p, s) \geq 2R2^{-k-d} = 2^{-\boldsymbol{k}-d+r+1}.$$

By (5) this implies that there is w, $x_p < w \leq s$, such that

$$2^{-\boldsymbol{k}-b-d+r+1} \leq 2^{-K_s(w)}. \tag{7}$$

Now, in sums of the form $\mathcal{S}(x_q, x_{q+R})$, because of taking the minimum, the "cut-off" for how much w can contribute is at $2^{-\boldsymbol{k}-b-d+r}$. Hence we have

$$\mathcal{S}(x_p, x_{p+2R}) \geq 2^{-\boldsymbol{k}-b-d+r} + \mathcal{S}(x_p, x_{p+R}) + \mathcal{S}(x_{p+R}, x_{p+2R}).$$

The additional term $2^{-\boldsymbol{k}-b-d+r}$ is due to the fact that w contributes at most $2^{-\boldsymbol{k}-b-d+r}$ to $\mathcal{S}(x_p, x_{p+R}) + \mathcal{S}(x_{p+R}, x_{p+2R})$, but by (7), w contributes $2^{-\boldsymbol{k}-b-d-r+1}$ to $\mathcal{S}(x_p, x_{p+2R})$. By the inductive hypothesis, the right hand side is at least

$$2^{-\boldsymbol{k}-b-d+r} + 2 \cdot (r+1)2^{-\boldsymbol{k}-b-d+r-1} = (r+2)2^{-\boldsymbol{k}-b-d+r},$$

as required. □

6. A cost function-related basis theorem for Π^0_1 classes

The following strengthens [10, Thm 2.6], which relied on the extra assumption that the Π^0_1 class is contained in the ML-randoms.

Theorem 6.1. *Let $\mathcal{P}$ be a nonempty Π^0_1 class, and let $\mathbf{c}$ be a monotonic cost function with the limit condition. Then there is a Δ^0_2 set $Y \in \mathcal{P}$ such that each c.e. set $A \leq_T Y$ obeys $\mathbf{c}$.*

Proof. We may assume that $\mathbf{c}(x, s) \geq 2^{-x}$ for each $x \leq s$, because any c.e. set that obeys $\mathbf{c}$ also obeys the cost function $\mathbf{c}(x, s) + 2^{-x}$.

Let $\langle A_e, \Psi_e \rangle_{e \in \mathbb{N}}$ be an effective listing of all pairs consisting of a c.e. set and a Turing functional. We will define a Δ^0_2 set $Y \in \mathcal{P}$ via a computable approximation $(Y_s)_{s \in \mathbb{N}}$, where Y_s is a binary string of length s. We meet the requirements

$$N_e \colon\ A_e = \Psi_e(Y) \ \Rightarrow\ A_e \text{ obeys } \mathbf{c}.$$

We use a standard tree construction at the $\emptyset''$ level. Nodes on the tree $2^{<\omega}$ represent the strategies. Each node α of length e is a strategy for N_e. At stage s we define an approximation δ_s to the true path. We say that s is an *α-stage* if $\alpha \prec \delta_s$.

Suppose that a strategy α is on the true path. If $\alpha 0$ is on the true path, then strategy α is able to build a computable enumeration of A_e via which A_e obeys $\mathbf{c}$. If $\alpha 1$ is on the true path, the strategy shows that $A_e \neq \Psi_e(Y)$.

Let $\mathcal{P}^{\emptyset}$ be the given class $\mathcal{P}$. A strategy α has as an environment a Π^0_1 class $\mathcal{P}^\alpha$. It defines $\mathcal{P}^{\alpha 0} = \mathcal{P}^\alpha$, but usually let $\mathcal{P}^{\alpha 1}$ be a proper refinement of $\mathcal{P}^\alpha$.

Let $|\alpha| = e$. The length of agreement for e at a stage t is $\min\{y\colon\ A_{e,t}(y) \neq \Psi_{e,t}(Y_t)\}$. We say that an α-stage s is *α-expansionary* if the length of agreement for e at stage s is larger than at u for all previous α-stages u.

Let $w^n_0 = n$, and

$$w^n_{i+1} \simeq \mu v > w^n_i.\, \mathbf{c}(w^n_i, v) \geq 4^{-n}. \tag{8}$$

Since $\mathbf{c}$ satisfies the limit condition, for each n this sequence breaks off.

Let $a = w^n_i$ be such a value. The basic idea is to *certify* $A_{e,s}\restriction_w$, which means to ensure that all $X \succ Y_s\restriction_{n+d}$ on $\mathcal{P}^\alpha$ compute $A_{e,s}\restriction_w$. If $A\restriction_w$ changes later then also $Y\restriction_{n+d}$ has to change. Since $Y\restriction_{n+d}$ can only move to the right (as long as α is not initialized), this type of change for n can only contribute a cost of $4^{-n+1}2^{n+d} = 2^{-n+d+2}$.

By [22, p. 55], from an index $\mathcal{Q}$ for a Π^0_1 class in 2^ω we can obtain a computable sequence $(\mathcal{Q}_s)_{s\in\mathbb{N}}$ of clopen classes such that $\mathcal{Q}_s \supseteq \mathcal{Q}_{s+1}$ and $\mathcal{Q} = \bigcap_s \mathcal{Q}_s$. In the construction below we will have several indices for Π^1_1 classes $\mathcal{Q}$ that change over time. At stage s, as usual by $\mathcal{Q}[s]$ we denote the value of the index at stage s. Thus $(\mathcal{Q}[s])_s$ is the clopen approximation of $\mathcal{Q}[s]$ at stage s.

Construction of Y.
Stage 0. *Let $\delta_0 = \emptyset$ and $\mathcal{P}^\emptyset = \mathcal{P}$. Let $Y_0 = \emptyset$.*
Stage $s > 0$. Let $\mathcal{P}^\emptyset = \mathcal{P}$.
For each β such that $\delta_{s-1} <_L \beta$ we initialize strategy β. We let Y_s be the leftmost path on the current approximation to $\mathcal{P}^{\delta_{s-1}}$, i.e., the leftmost string y of length $s-1$ such that $[y] \cap (\mathcal{P}^{\delta_{s-1}}[s-1])_s \neq \emptyset$. For each α, n, if $Y_s\restriction_{n+d} \neq Y_{s-1}\restriction_{n+d}$ where $d = \mathsf{init}_s(\alpha)$, then we declare each existing value w^n_i to be (α, n)-*unsatisfied.*
Substage k, $0 \leq k < s$. Suppose we have already defined $\alpha = \delta_s\restriction_k$. Run strategy α (defined below) at stage s, which defines an outcome $r \in \{0, 1\}$ and a Π^0_1 class $\mathcal{P}^{\alpha r}$. Let $\delta_s(k) = r$.

We now describe the strategies α and procedures $\mathcal{S}^\alpha_n$ they call. To initialize a strategy α means to cancel the run of this procedure. Let

$$d = \mathsf{init}_s(\alpha) = |\alpha| + \text{the last stage when } \alpha \text{ was initialized.}$$

Strategy α at an α-stage s.

(a) If no procedure for α is running, call procedure $\mathcal{S}^\alpha_n$ with parameter w, where n is least, and i is chosen least for n, such that $w = w^n_i \leq s$ is not (α, n)-satisfied. Note that n exists because $w^s_0 = s$ and this value is not (α, n)-satisfied at the beginning of stage s. By calling this procedure, we attempt to certify $A_{e,s}\restriction_w$ as discussed above.
(b) While such a procedure $\mathcal{S}^\alpha_n$ is running, give outcome 1. (This procedure will define the current class $\mathcal{P}^{\alpha 1}$.)
(c) If a procedure $\mathcal{S}^\alpha_n$ returns at this stage, goto (d).
(d) If s is α-expansionary, give outcome 0, let $\mathcal{P}^{\alpha 0} = \mathcal{P}^\alpha$, and continue at (a) at the next α-stage. Otherwise, give outcome 1, let $\mathcal{P}^{\alpha 1} = \mathcal{P}^\alpha$, and stay at (d).

Procedure $\mathcal{S}^\alpha_n$ with parameter w at a stage s.
If $n + d \geq s - 1$ let $\mathcal{P}^{\alpha 1} = \mathcal{P}^\alpha$. Otherwise, let

$$\mathcal{Q} = \mathcal{P}^\alpha \cap \{X \succ z\colon\ \Psi^X_e \not\succ A_{e,s}\restriction_w\}, \tag{9}$$

where $z = Y_s \upharpoonright_{n+d}$. (Note that each time $Y \upharpoonright_{n+d}$ or $A_e \upharpoonright_w$ has changed, we update this definition of $\mathcal{Q}$.)

(e) If $\mathcal{Q}_s \neq \emptyset$ let $\mathcal{P}^{\alpha 1} = \mathcal{Q}$. If the definition of $\mathcal{P}^{\alpha 1}$ has changed since the last α-stage, then each β such that $\alpha 1 \preceq \beta$ is initialized.

(f) If $\mathcal{Q}_s = \emptyset$, declare w to be (α, n)-*satisfied* and return. ($A_{e,s} \upharpoonright_w$ is certified as every $X \in \mathcal{P}^\alpha$ extending z computes $A_{e,s} \upharpoonright_w$ via Ψ_e. If $A_e \upharpoonright_w$ changes later, the necessarily $z \not\preceq Y$.)

Claim 6.2. *Suppose a strategy α is no longer initialized after stage s_0. Then for each n, a procedure $\mathcal{S}^\alpha_n$ is only called finitely many times after s_0.*

There are only finitely many values $w = w^n_i$ because $\mathbf{c}$ satisfies the limit condition. Since α is not initialized after s_0, $\mathcal{P}^\alpha$ and $d = \mathsf{init}_s(\alpha)$ do not change. When a run of $\mathcal{S}^\alpha_n$ is called at a stage s, the strategies $\beta \succeq \alpha 1$ are initialized, hence $\mathsf{init}_t(\beta) \geq s > n + d$ for all $t \geq s$. So the string $Y_s \upharpoonright_{n+d}$ is the leftmost string of length $n + d$ on $\mathcal{P}^\alpha$ at stage s. This string has to move to the right between the stages when $\mathcal{S}^\alpha_n$ is called with the same parameter w, because w is declared (α, n)-unsatisfied before $\mathcal{S}^\alpha_n$ is called again with parameter w. Thus, procedure $\mathcal{S}^\alpha_n$ can only be called 2^{n+d} times with parameter w.

Claim 6.3. *$\langle Y_s \rangle_{s \in \mathbb{N}}$ is a computable approximation of a Δ^0_2 set $Y \in \mathcal{P}$.*

Fix $k \in \mathbb{N}$. For a stage s, if $Y_s \upharpoonright_k$ is to the left of $Y_{s-1} \upharpoonright_k$ then there are α, n with $n + \mathsf{init}_s(\alpha) \leq k$ such that $\mathcal{P}^\alpha[s] \neq \mathcal{P}^\alpha[s-1]$ because of the action of a procedure $\mathcal{S}^\alpha_n$ at (e) or (f).

There are only finitely many pairs α, s such that $\mathsf{init}_s(\alpha) \leq k$. Thus by Claim 6.2 there is stage s_0 such that at all stages $s \geq s_0$, for no α and n with $n + \mathsf{init}_s(\alpha) \leq k$, a procedure $\mathcal{S}^\alpha_n$ is called.

While a procedure $\mathcal{S}^\alpha_n$ is running with a parameter w, it changes the definition of $\mathcal{P}^{\alpha 1}$ only if $A_e \upharpoonright_w$ changes ($e = |\alpha|$), so at most w times. Thus there are only finitely many s such that $Y_s \upharpoonright_k \neq Y_{s-1} \upharpoonright_k$.

By the definition of the computable approximation $\langle Y_s \rangle_{s \in \mathbb{N}}$ we have $Y \in \mathcal{P}$. This completes Claim 6.3.

As usual, we define the true path f by $f(k) = \liminf_s \delta_s(k)$. By Claim 6.2 each $\alpha \prec f$ is only initialized finitely often, because each β such that $\beta 1 \prec \alpha$ eventually is stuck with a single run of a procedure $\mathcal{S}^\beta_m$.

Claim 6.4. *If $e = |\alpha|$ and $\alpha 1 \prec f$, then $A_e \neq \Psi^Y_e$.*

Some procedure $\mathcal{S}^\alpha_n$ was called with parameter w, and is eventually stuck at (e) with the final value $A_e \upharpoonright_w$. Hence the definition $\mathcal{Q} = \mathcal{P}^{\alpha 1}$ eventually stabilizes at α-stages s. Since $Y \in \mathcal{Q}$, this implies $A_e \neq \Psi^Y_e$.

Claim 6.5. *If $e = |\alpha|$ and $\alpha 0 \prec f$, then A_e obeys $\mathbf{c}$.*

Let $A = A_e$. We define a computable enumeration $(\widehat{A}_p)_{p \in \mathbb{N}}$ of A via which A obeys $\mathbf{c}$.

Since $\alpha 0 \prec f$, each procedure $\mathcal{S}^\alpha_n$ returns. In particular, since $\mathbf{c}$ has the limit condition and by Claims 6.2 and 6.3, each value $w = w^n_i$ becomes permanently (α, n)-satisfied. Let $d = \mathsf{init}_s(\alpha)$. Let s_0 be the least $\alpha 0$-stage such that $s_0 \geq d$, and let

$s_{p+1} = \mu s \geq s_p + 2\,[s$ is $\alpha 0$-stage &

$$\forall n, i\, (w = w_i^n < s_p \rightarrow w \text{ is } (\alpha, n)\text{-satisfied at } s)].$$

As in similar constructions such as [22], for $p \in \mathbb{N}$ we let

$$\widehat{A}_p = A_{s_{p+2}} \cap [0, p).$$

Consider the situtation that $p > 0$ and $x \le p$ is least such that $\widehat{A}_p(x) \neq \widehat{A}_{p-1}(x)$. We call this situation an *n-change* if n is least such that $x < w_i^n < s_p$ for some i. (Note that $n \le p+1$ because $w_0^{p+1} = p+1$.) Thus (x, s_p) contains no value of the form w_j^{n-1}, whence $\mathbf{c}(x, p) \le \mathbf{c}(x, s_p) \le 4^{-n+1}$. We are done if we can show there are at most 2^{n+d} many n-changes, for in that case the total cost $\mathbf{c}\langle \widehat{A}_p \rangle$ is bounded by $\sum_n 4^{-n+1} 2^{n+d} = O(2^d)$.

Recall that $\mathcal{P}^\alpha$ is stable by stage s_0. Note that $Y \restriction_{n+d}$ can only move to the right after the first run of $\mathcal{S}_n^\alpha$, as observed in the proof of Claim 6.2.

Consider n-changes at stages $p < q$ via parameters $w = w_i^n$ and $w' = w_k^n$ (where possibly $k < i$). Suppose the last run of $\mathcal{S}_n^\alpha$ with parameter w that was started before s_{p+1} has returned at stage $t \le s_{p+2}$, and similarly, the last run of $\mathcal{S}_n^\alpha$ with parameter w' that was started before s_{q+1} has returned at stage t'. Let $z = Y_t \restriction_{n+d}$ and $z' = Y_{t'} \restriction_{n+d}$. We show $z <_L z'$; this implies that there are at most 2^{n+d} many n-changes.

At stage t, by definition of returning at (f) in the run of $\mathcal{S}_n^\alpha$, we have $\mathcal{Q} = \emptyset$. Therefore $\Psi_{e,t}^X \succ A_{e,t} \restriction_w$ for each X on $\mathcal{P}_t^\alpha$ such that $X \succ z$. Now

$$\widehat{A}_p(x) \neq \widehat{A}_{p-1}(x),\ x < w \text{ and } t \le s_{p+1},$$

so $A_{s_{p+2}} \restriction_w \neq A_t \restriction_w$, The stage s_{p+2} is $\alpha 0$-expansionary, and $Y_{s_{p+2}}$ is on $\mathcal{P}_t^\alpha$. Therefore

$$Y_{r-1} \restriction_{n+d} <_L Y_r \restriction_{n+d}$$

for some stage r such that $t < r \le s_{p+2}$. Thus, at stage r, the value w' was declared (α, n)-unsatisfied. Hence a new run of $\mathcal{S}_n^\alpha$ with parameter w' is started after r, which has returned by stage $s_{q+1} \ge s_{p+2}$. Thus $r < t'$. So $z \le_L Y_{r-1} \restriction_{n+d} <_L Y_r \restriction_{n+d} \le_L z'$, whence $z <_L z'$ as required. This concludes Claim 6.5 and the proof. $\square$

7. A dual cost function construction

Given a relativizable cost function $\mathbf{c}$, let $D \rightarrow W^D$ be the c.e. operator given by the cost function construction in Theorem 2.7 relative to the oracle D. By pseudo-jump inversion there is a c.e. set D such that $W^D \oplus D \equiv_T \emptyset'$, which implies $D <_T \emptyset'$. Here, we give a direct construction of a c.e. set $D <_T \emptyset'$ so that the total cost of $\emptyset'$-changes as measured by $\mathbf{c}^D$ is finite. More precisely, there is a D-computable enumeration of $\emptyset'$ obeying $\mathbf{c}^D$.

If $\mathbf{c}$ is sufficiently strong, then the usual cost function construction builds an incomputable c.e. set A that is close to being computable. The dual cost function construction then builds a c.e. set D that is close to being Turing complete.

7.1. **Preliminaries on cost functionals.** Firstly we clarify how to relativize cost functions, and the notion of obedience to a cost function. Secondly we provide some technical details needed for the main construction.

Definition 7.1. (i) A *cost functional* is a Turing functional $\mathbf{c}^Z(x,t)$ such that for each oracle Z, $\mathbf{c}^Z$ either is partial, or is a cost function relative to Z. We say that $\mathbf{c}$ is non-increasing in main argument if this holds for each oracle Z such that $\mathbf{c}^Z$ is total. Similarly, $\mathbf{c}$ is non-decreasing in the stage argument if this holds for each oracle Z such that $\mathbf{c}^Z$ is total. If both properties hold we say that $\mathbf{c}$ is monotonic.
(ii) Suppose $A \le_T Z'$. Let $\langle A_s \rangle$ be a Z-computable approximation of A. We write $\langle A_s \rangle \models^Z \mathbf{c}^Z$ if

$$\mathbf{c}^Z\langle A_s\rangle = \sum_{x,s} \mathbf{c}^Z(x,s) \llbracket x < s \;\&\; \mathbf{c}^Z(x,s)\downarrow \;\&\; x \text{ is least s.t. } A_{s-1}(x) \neq A_s(x) \rrbracket$$

is finite. We write $A \models^Z c^Z$ if $\langle A_s\rangle \models^Z c^Z$ for some Z-computable approximation $\langle A_s \rangle$ of A.

For example, $\mathbf{c}^Z_{\mathcal{K}}(x,s) = \sum_{x<w\le s} 2^{-K^Z_s(w)}$ is a total monotonic cost functional. We have $A \models^Z \mathbf{c}^Z_{\mathcal{K}}$ iff A is K-trivial relative to Z.

We may convert a cost functional $\mathbf{c}$ into a total cost functional $\widetilde{\mathbf{c}}$ such that $\widetilde{\mathbf{c}}^Z(x) = \mathbf{c}^Z(x)$ for each x with $\forall t\, \mathbf{c}^Z(x,t)\downarrow$, and, for each Z, x, t, the computation $\widetilde{\mathbf{c}}^Z(x,t)$ converges in t steps. Let

$$\widetilde{\mathbf{c}}^Z(x,s) = \mathbf{c}^Z(x,t) \text{ where } t \le s \text{ is largest such that } \mathbf{c}^Z(x,t)[s]\downarrow.$$

Clearly, if $\mathbf{c}$ is monotonic in the main/stage argument then so is $\widetilde{\mathbf{c}}$.

Suppose that D is c.e. and we compute $\mathbf{c}^D(x,t)$ via hat computations [24, p. 131]: the use of a computation $\mathbf{c}^D(x,t)[s]\downarrow$ is no larger than the least number entering D at stage s. Let N_D be the set of non-deficiency stages; that is, $s \in N_D$ iff there is $x \in D_s - D_{s-1}$ such that $D_s\restriction_x = D\restriction_x$. Any hat computation existing at a non-deficiency stage is final. We have

$$\mathbf{c}^D(x) = \sup_{s\in N_D} \widetilde{\mathbf{c}}^{D_s}(x,s). \tag{10}$$

For, if $\mathbf{c}^D(x,t)[s_0]\downarrow$ with D stable below the use, then $\mathbf{c}^D(x,t) \le \widetilde{\mathbf{c}}^{D_s}(x,s)$ for each $s \in N_D$. Therefore $\mathbf{c}^D(x) \le \sup_{s\in N_D} \widetilde{\mathbf{c}}^{D_s}(x,s)$. For the converse inequality, note that for $s \in N_D$ we have $\widetilde{\mathbf{c}}^{D_s}(x,s) = \mathbf{c}^D(x,t)$ for some $t \le s$ with D stable below the use.

7.2. **The dual existence theorem.**

Theorem 7.2. *Let $\mathbf{c}$ be a total cost functional that is nondecreasing in the stage component and satisfies the limit condition for each oracle. Then there is a Turing incomplete c.e. set D such that $\emptyset' \models^D \mathbf{c}^D$.*

Proof. We define a cost functional $\Gamma^Z(x,s)$ that is nondecreasing in the stage. We will have $\underline{\Gamma}^D(x) = \mathbf{c}^D(x)$ for each x, where $\underline{\Gamma}^D(x) = \lim_t \Gamma^D(x,t)$, and $\emptyset'$ with its given computable enumeration obeys Γ^D. Then $\emptyset' \models^D \mathbf{c}^D$ by the easy direction '$\Leftarrow$' of Theorem 3.4 relativized to D.

Towards $\Gamma^D(x) \ge \mathbf{c}^D(x)$, when we see a computation $\widetilde{\mathbf{c}}^{D_s}(x,s) = \alpha$ we attempt to ensure that $\Gamma^D(x,s) \ge \alpha$. To do so we enumerate relative to D a set G of "wishes" of the form

$$\rho = \langle x, \alpha \rangle^u,$$

where $x \in \mathbb{N}$, α is a nonnegative rational, and $u+1$ is the use. We say that ρ is a *wish about* x. If such a wish is enumerated at a stage t and $D_t \upharpoonright_u$ is stable, then the wish is granted, namely, $\Gamma^D(x,t) \geq \alpha$. The converse inequality $\Gamma^D(x) \leq \mathbf{c}^D(x)$ will hold automatically.

To ensure $D <_T \emptyset'$, we enumerate a set F, and meet the requirements

$$N_e\colon\ F \neq \Phi_e^D.$$

Suppose we have put a wish $\rho = \langle x, \alpha \rangle^u$ into G^D. To keep the total Γ^D-cost of the given computable enumeration of $\emptyset'$ down, when x enters $\emptyset'$ we want to remove ρ from G^D by putting u into D. However, sometimes D is preserved by some N_e. This will generate a *preservation cost.* N_e starts a run at a stage s via some parameter v, and "hopes" that $\emptyset'_s \upharpoonright_v$ is stable. If $\emptyset' \upharpoonright_v$ changes after stage s, then this run of N_e is cancelled. On the other hand, if $x \geq v$ and x enters $\emptyset'$, then the ensuing preservation cost can be afforded. This is so because we choose v such that $\widetilde{c}_s^{D_s}(v,s)$ is small. Since $\widetilde{\mathbf{c}}^D$ has the limit condition, eventually there is a run $N_e(v)$ with such a low-cost v where $\emptyset' \upharpoonright_v$ is stable. Then the diagonalization of N_e will succeed.

Construction of c.e. sets F, D and a D-c.e. set G of wishes.
Stage $s > 0$. We may suppose that there is a unique $n \in \emptyset'_s - \emptyset'_{s-1}$.

1. Canceling N_e's. Cancel all currently active $N_e(v)$ with $v > n$.
2. Removing wishes. For each $\rho = \langle x, \alpha \rangle^u \in G^D[s-1]$ put in at a stage $t < s$, if $\emptyset'_s \upharpoonright_{x+1} \neq \emptyset'_t \upharpoonright_{x+1}$ and ρ is not held by any $N_e(v)$, then put $u-1$ into D_s, thereby removing ρ from G^D.
3. Adding wishes. For each $x < s$ pick a large u (in particular, $u \notin D_s$) and put a wish $\langle x, \alpha \rangle^u$ into G where $\alpha = \widetilde{\mathbf{c}}^{D_s}(x,s)$. The set of queries to the oracle D for this enumeration into G is contained in $[0,r) \cup \{u\}$, where r is the use of $\widetilde{\mathbf{c}}^{D_s}(x,s)$ (which may be much smaller than s). Then, from now on this wish is kept in G^D unless (a) $D \upharpoonright_r$ changes , or (b) u enters D.
4. Activating $N_e(v)$. For each $e < s$ such that N_e is not currently active, see if there is v, $e \leq v \leq n$ such that

- $\widetilde{\mathbf{c}}^{D_s}(v,s) \leq 3^{-e}/2$,
- $v > w$ for each w such that $N_i(w)$ is active for some $i < e$, and
- $\Phi_e^D \upharpoonright_{x+1} = F \upharpoonright_{x+1}$ where $x = \langle e, v, |\emptyset' \cap [0,v)| \rangle$,

If so, choose v least and activate $N_e(v)$. Put x into F. Let N_e *hold* all wishes for some $y \geq v$ that are currently in G^D. Declare that such a wish is no longer held by any $N_i(w)$ for $i \neq e$. (We also say that N_e *takes over* the wish.)

Go to stage s' where s' is larger than any number mentioned so far.

Claim 1. *Each requirement N_e is activated only finitely often, and met. Hence $F \not\leq_T D$.*
Inductively suppose that N_i for $i < e$ is no longer activated after stage t_0. Assume for a contradiction that $F = \Phi_e^D$. Since $\mathbf{c}^D$ satisfies the limit condition, by (10) there is a least v such that $\widetilde{\mathbf{c}}^{D_s}(v,s) \leq 3^{-e}/2$ for infinitely many $s > t_0$. Furthermore, $v > w$ for any w such that some $N_i(w)$, $i < e$, is active at t_0. Once $N_e(v)$ is activated, it can only be canceled by a change

of $\emptyset' \restriction_v$. Then there is a stage $s > t_0$, $\widetilde{\mathbf{c}}^{D_s}(v,s) \leq 3^{-e}/2$, such that $\emptyset' \restriction_v$ is stable at s and $\Phi_e^D \restriction_{x+1} = F \restriction_{x+1}$ where $x = \langle e, v, |\emptyset' \cap [0,v)|\rangle$. If some $N_e(v')$ for $v' \leq v$ is active after (1.) of stage s then it remains active, and N_e is met. Now suppose otherwise.

Since we do not activate $N_e(v)$ in (4) of stage s, some $N_e(w)$ is active for $w > v$. Say it was activated last at a stage $t < s$ via $x = \langle e, w, |\emptyset'_t \cap [0,w]|$. Then $x' = \langle e, v, |\emptyset'_t \cap [0,v)|\rangle$ was available to activate $N_e(v)$ as $x' \leq x$ and hence $\Phi_e^D \restriction_{x'+1} = F \restriction_{x'+1} [t]$. Since w was chosen minimal for e at stage t, we had $\widetilde{\mathbf{c}}^{D_t}(v,t) > 3^{-e}/2$. On the other hand, $\widetilde{\mathbf{c}}^{D_s}(v,s) \leq 3^{-e}/2$, hence $D_t \restriction_t \neq D_s \restriction_t$. When $N_e(w)$ became active at t it tried to preserve $D \restriction_t$ by holding all wishes about some $y \geq w$ that were in $G^D[t]$. Since $N_e(w)$ did not succeed, it was cancelled by a change $\emptyset'_t \restriction_w \neq \emptyset'_s \restriction_w$. Hence $N_e(w)$ is not active at stage s, contradiction. $\diamond$

We now define $\Gamma^Z(x,t)$ for an oracle Z (we are interested only in the case that $Z = D$). Let s be least such that $D_s \restriction_t = Z \restriction_t$. Output the maximum α such that some wish $\langle x, \alpha\rangle^u$ for $u \leq t$ is in $G^D[s]$.

Claim 2. *(i) $\Gamma^D(x,t)$ is nondecreasing in t. (ii) $\forall x\, \underline{\Gamma}^D(x) = \underline{\mathbf{c}}^D(x)$.*
(i). Suppose $t' \geq t$. As above let s be least such that $D_s \restriction_t$ is stable. Let s' be least such that $D_{s'} \restriction_{t'}$ is stable. Then $s' \geq s$, so a wish as in the definition of $\Gamma^D(x,t)$ above is also in $G^D[s']$. Hence $\Gamma^D(x,t') \geq \Gamma^D(x,t)$.
(ii). Given x, to show that $\Gamma^D(x) \geq \mathbf{c}^D(x)$ pick t_0 such that $\emptyset' \restriction_{x+1}$ is stable at t_0. Let $s \in N_D$ and $s > t_0$. At stage s we put a wish $\langle x, \alpha\rangle^u$ into G_D where $\alpha = \widetilde{\mathbf{c}}^{D_s}(x,s)$. This wish is not removed later, so $\Gamma^D(x) \geq \alpha$.

For $\Gamma^D(x) \leq \mathbf{c}^D(x)$, note that for each $s \in N_D$ we have $\widetilde{\mathbf{c}}^{D_s}(x,s) \geq \Gamma^{D_s}(x,s)$ by the removal of a wish in 3(a) of the construction when the reason the wish was there disappears. $\diamond$

Claim 3. *The given computable enumeration of $\emptyset'$ obeys Γ^D.*
First we show by induction on stages s that N_e holds in total at most 3^{-e} at the end of stage s, namely,

$$3^{-e} \geq \sum_x \max\{\alpha \colon\ N_e \text{ holds a wish } \langle x, \alpha\rangle^u\} \tag{11}$$

Note that once $N_e(v)$ is activated and holds some wishes, it will not hold any further wishes later, unless it is cancelled by a change of $\emptyset' \restriction_v$ (in which case the wishes it holds are removed).

We may assume that $N_e(v)$ is activated at (3.) of stage s. Wishes held at stage s by some $N_i(w)$ where $i < e$ will not be taken over by $N_e(v)$ because $w < v$. Now consider wishes held by a $N_i(w)$ where $i > e$. By inductive hypothesis the total of such wishes is at most $\sum_{i>e} 3^{-i} = 3^{-e}/2$ at the beginning of stage s. The activation of $N_e(v)$ adds at most another $3^{-e}/2$ to the sum in (11).

To show $\Gamma^D\langle \emptyset'_s\rangle < \infty$, note that any contribution to this quantity due to n entering $\emptyset'$ at stage s is because a wish $\langle n, \delta\rangle^u$ is eventually held by some $N_e(v)$. The total is at most $\sum_e 3^{-e}$. $\square$

The study of non-monotonic cost function is left to the future. For instance, we conjecture that there are cost functions $\mathbf{c}, \mathbf{d}$ with the limit condition such that for any Δ^0_2 sets A, B,

$$A \models \mathbf{c} \text{ and } B \models \mathbf{d} \Rightarrow A, B \text{ form a minimal pair.}$$

It is not hard to build cost functions $\mathbf{c}, \mathbf{d}$ such that only computable sets obey both of them. This provides some evidence for the conjecture.

References

[1] L. Bienvenu, A. Day, N. Greenberg, A. Kučera, J. Miller, A. Nies, and D. Turetsky. Computing K-trivial sets by incomplete random sets. *Bull. Symb. Logic*, 20:80–90, 2014.

[2] L. Bienvenu, R. Downey, N. Greenberg, W. Merkle, and A. Nies. Kolmogorov complexity and Solovay functions. Submitted, 2014.

[3] L. Bienvenu, N. Greenberg, A. Kučera, A. Nies, and D. Turetsky. Coherent randomness tests and computing the K-trivial sets. Submitted, 2013.

[4] C. Calude and A. Grozea. The Kraft-Chaitin theorem revisited. *J. Univ. Comp. Sc.*, 2:306–310, 1996.

[5] A. R. Day and J. S. Miller. Density, forcing and the covering problem. Submitted, http://arxiv.org/abs/1304.2789, 2013.

[6] D. Diamondstone, N. Greenberg, and D. Turetsky. Inherent enumerability of strong jump-traceability. Submitted, http://arxiv.org/abs/1110.1435, 2012.

[7] R. Downey and D. Hirschfeldt. *Algorithmic randomness and complexity*. Springer-Verlag, Berlin, 2010. 855 pages.

[8] R. Downey, D. Hirschfeldt, A. Nies, and F. Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore University Press.

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

[10] N. Greenberg, D. Hirschfeldt, and A. Nies. Characterizing the strongly jump-traceable sets via randomness. *Adv. Math.*, 231(3-4):2252–2293, 2012.

[11] N. Greenberg and A. Nies. Benign cost functions and lowness properties. *J. Symbolic Logic*, 76:289–312, 2011.

[12] N. Greenberg and D. Turetsky. Strong jump-traceability and Demuth randomnesss. *Proc. Lond. Math. Soc.*, 108:738–779, 2014.

[13] D. Hirschfeldt, A. Nies, and F. Stephan. Using random sets as oracles. *J. Lond. Math. Soc. (2)*, 75(3):610–622, 2007.

[14] B. Kjos-Hanssen, W. Merkle, and F. Stephan. Kolmogorov complexity and the Recursion Theorem. In *STACS 2006*, volume 3884 of *Lecture Notes in Comput. Sci.*, pages 149–161. Springer, Berlin, 2006.

[15] A. Kučera. An alternative, priority-free, solution to Post's problem. In *Mathematical foundations of computer science, 1986 (Bratislava, 1986)*, volume 233 of *Lecture Notes in Comput. Sci.*, pages 493–500. Springer, Berlin, 1986.

[16] A. Kučera and T. Slaman. Low upper bounds of ideals. *J. Symbolic Logic*, 74:517–534, 2009.

[17] A. Kučera and S. Terwijn. Lowness for the class of random sets. *J. Symbolic Logic*, 64:1396–1402, 1999.

[18] A. Kučera and A. Nies. Demuth randomness and computational complexity. *Ann. Pure Appl. Logic*, 162:504–513, 2011.

[19] A. Melnikov and A. Nies. K-triviality in computable metric spaces. *Proc. Amer. Math. Soc.*, 141(8):2885–2899, 2013.

[20] A. Nies. Reals which compute little. In *Logic Colloquium '02*, Lecture Notes in Logic, pages 260–274. Springer–Verlag, 2002.

[21] A. Nies. Lowness properties and randomness. *Adv. in Math.*, 197:274–305, 2005.

[22] A. Nies. *Computability and randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, Oxford, 2009.

[23] A. Nies. Interactions of computability and randomness. In *Proceedings of the International Congress of Mathematicians*, pages 30–57. World Scientific, 2010.

[24] R. I. Soare. *Recursively Enumerable Sets and Degrees*. Perspectives in Mathematical Logic, Omega Series. Springer–Verlag, Heidelberg, 1987.

[25] R. Solovay. Handwritten manuscript related to Chaitin's work. IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 215 pages, 1975.