RESOLUTE SEQUENCES IN INITIAL SEGMENT COMPLEXITY

GEORGE BARMPALIAS AND ROD G. DOWNEY

Abstract. We study infinite sequences whose initial segment complexity is invariant under effective insertions of blocks of zeros in-between their digits. Surprisingly, such *resolute* sequences may have nontrivial initial segment complexity. In fact, we show that they occur in many well known classes from computability theory, e.g. in every jump class and every high degree. Moreover there are degrees which consist entirely of resolute sequences, while there are degrees which do not contain any. Finally we establish connections with the contiguous c.e. degrees, the ultracompressible sequences, the anti-complex sequences thus demonstrating that this class is an interesting superclass of the sequences with trivial initial segment complexity.

1. Introduction

Given an infinite random binary sequence X we may reduce its initial segment complexity by inserting blocks of zeros between its original digits. Even a single zero in-between every other digit of X will reduce its complexity dramatically. But what if X is not random? Can we always alter the complexity of its initial segments by 'spreading out' its digits in an effective manner? Clearly if X has trivial initial segment complexity, the simplification of its initial segments will not result in a 'measurable' reduction of their complexity. Surprisingly, there are nontrivial sequences X whose initial segment complexity is invariant under such effective 'block inserting' operations. Intuitively, these sequences have the property that

> it is very hard to locate bits of significant information in their initial segments.

In this article we exhibit such examples in a variety of classes from computability theory and study this proper superclass of the family of sequences with trivial initial segment complexity. In particular, we establish connections with a number of notions from computability and Kolmogorov complexity like the jump hierarchy, the contiguous degrees, the ultracompressible sets of [LL99], the facile sets of [Nie09, Section 8.2] and the anti-complex sets of [FGSW12].

Last revision: 25 August 2012.

Key words and phrases. Kolmogorov complexity, Computably enumerable sets, trivial reals.

This research was partially done whilst the authors were visiting fellows at the Isaac Newton Institute for the Mathematical Sciences, Cambridge U.K., in the programme 'Semantics & Syntax'. Barmpalias was supported by the *Research fund for international young scientists* number 611501-10168 from the National Natural Science Foundation of China, and an *International Young Scientist Fellowship* number 2010-Y2GB03 from the Chinese Academy of Sciences; partial support was also received from the project *Network Algorithms and Digital Information* number ISCAS2010-01 from the Institute of Software, Chinese Academy of Sciences.

1.1. **Formal expressions of resoluteness.** We measure the complexity of binary strings σ via the plain Kolmogorov complexity $C(\sigma)$ prefix-free Kolmogorov complexity $K(\sigma)$; this is the length of the shortest program that produces σ in an underlying plain or prefix-free machine respectively. For background on Kolmogorov complexity we refer to [DH10]. Let $X \leq_K Y$ denote $\exists \forall n\ K(X\restriction_n) \leq K(Y\restriction_n) + c$ and similarly for the plain complexity. These preorders induce equivalence relations $\equiv_K$, $\equiv_C$ and corresponding degree structures that are known as the K-degrees and the C-degrees respecively. Intuitively, two sequences in the same degree have the same initial segment complexity.

The operation of inserting 0s between various digits of a given sequence is equivalent to shifting the bits of the sequence at various places and filling in the gaps with 0s. Let us refer to increasing functions $f : \mathbb{N} \to \mathbb{N}$ as *shifts*. If we view an infinite binary sequence X as a set of natural numbers, then the result of such a shift operation may be expressed as the image of X under f.

Definition 1.1 (Shifts)**.** *An increasing function $f : \mathbb{N} \to \mathbb{N}$ is called shift. For each set Z we let $Z_f = \{f(n) \mid n \in Z\}$. A shift is called trivial if $\forall n\ (f(n) < n + c)$ for some constant c.*

Invariance under shift operations with respect to the plain and the prefix-free complexity can be defined as follows.

Definition 1.2 (Invariance)**.** *A set Z is K-invariant under f if $Z \equiv_K Z_f$ and is C-invariant under f if $Z \equiv_C Z_f$.*

For every sequence Z and every computable shift f we have $Z_f \leq_K Z$, so the application of a computable shift on a sequence may only reduce its initial segment complexity. Note that if a shift f is trivial then for every sequence X the sequence Z_f is (modulo finitely many bits) merely the result of shifting the bits of Z by a fixed number of places. Trivial shifts are not very interesting from our point of view as they preserve most notions of complexity on all sequences.

Definition 1.3 (K-resolute sequences)**.** *An infinite sequence Z is called K-resolute if $Z \equiv_K Z_f$ for all computable shifts f. The C-resolute sequences are defined analogously.*

This definition is arguably a faithful formalization of the property that we discussed earlier, i.e. the ability of a sequence to preserve its initial segment complexity despite any computable insertion of blocks of 0s in-between its digits. This is an expression of 'resoluteness' of a sequence, i.e. the *inability to locate significant amounts of information in its initial segments*. There are other, similar ways to express this informal concept. For example, consider property (1.1).

$$\text{For all computable shifts } f,\ \exists c \forall n\ K(Z\restriction_{f(n)}) \leq K(Z\restriction_n) + c. \tag{1.1}$$

This also expresses a form of 'resoluteness' of a sequence. Moreover it is not hard to see that K-resolute sequences meet (1.1). Indeed, since there exists a constant c such that $\forall n\ |K(Z_f\restriction_{f(n)}) - K(Z\restriction_n)| < c$, for any computable shift f,

$$\text{if } Z \equiv_K X_f, \text{ then } \exists c \forall n\ |K(Z\restriction_{f(n)}) - K(Z\restriction_n)| < c. \tag{1.2}$$

We say that a set is *weakly K-resolute* if it meets condition (1.1).

Yet another form of 'resoluteness' may be expressed in terms of conditional complexity, as in (1.3). Here an *order* is a nondecreasing and unbounded function.

$$\text{For all computable orders } g,\ \exists c \forall n\ K(Z\restriction_n \mid n) \leq K(Z\restriction_{g(n)} \mid n) + c. \tag{1.3}$$

Moreover each of the above notions has a version with respect to plain complexity.

As interesting as it may be, we will not be concerned with the technical question about the relationship between the above resoluteness notions. Instead, we focus on the notion of Definition 1.3 and note that our main results also hold for the two other notions (also with respect to plain complexity).

In the following, *we use the term 'resolute' to refer collectively to any of the above three formal variations on this concept and the versions with respect to plain complexity*, while 'K-resolute' is reserved for the notion of Definition 1.3. A degree is K-resolute if it contains a K-resolute set.

1.2. **Resoluteness and complexity.** Intuitively, sequences with 'consistently high complexity' cannot be resolute. On the other hand, sequences with trivial complexity are resolute. We give an overview of the relationship between complexity and resoluteness in more precise terms. In our context, trivial sequences are the K-trivial sequences, i.e. the sequences X such that $\exists c \forall n K(X \upharpoonright_n) \leq K(n) + c$. It was shown in [Nie05] that this class is downward closed under Turing reducibility. Hence, if X is K-trivial and f is a computable shift then $X_f \equiv_K X$. In other words, K-trivial sequences are K-resolute.

On the other end of the spectrum, a sequence X is called *random* if there exists a constant c such that $\forall n\ K(X \upharpoonright_n) \geq n - c$. It is clear that random sequences are not K-resolute. In fact, much more is true. A set is called *complex* if $\forall n\ K(X \upharpoonright_{f(n)}) \geq n$ for some computable function f. This definition is from [KHMS06, KHMS11] where it was shown to be equivalent to the condition that a diagonally noncomputable function is weak truth table reducible to X. Clearly complex sets are not weakly K-resolute (i.e. they do not meet (1.1)). It follows that complex sets are not K-resolute. Similar considerations apply to the C-resolute sets.

In turns out that K-resolute sets have very low initial segment complexity, but not necessarily trivial complexity. A class of sequences of 'ultra-low' initial segment complexity was introduced in [LL99]. We say that X is *ultracompressible* if for all computable orders h, there exists c such that $K(X \upharpoonright_n) \leq K(n) + h(n)$ for all sufficiently large n. A related class of sequences of low complexity was introduced in [FGSW12]. A set X is *anti-complex* if for all computable orders f we have $C(A \upharpoonright_{f(n)}) \leq n$ for all but finitely many n. It is not hard to see that in this definition it does not matter if we use prefix-free complexity instead of plain complexity. Also, it is not hard to see that *every ultracompressible set is anti-complex.*

The proof of the following observation uses two notions from computability theory. A set X is called *superlow* if the jump X' of X is truth-table reducible to the halting problem $\emptyset'$. Also, a degree $\mathbf{a}$ is called *array computable* if there exists a function that can be computed from the halting problem with computable use of this oracle, which dominates all $\mathbf{a}$-computable functions.

Proposition 1.4. *Every K-resolute set is ultracompressible (hence, anti-complex). The converse is not true, even for c.e. sets.*

Proof. Let X be a K-resolute set. In order to show that it is ultracompressible, let g be a computable order. Without loss of generality we may assume that g is onto. Let f be a computable increasing function such that $g(f(n)) = n^2$ for all n. Then there exists some constant c such that $K(X_f \upharpoonright_n) \leq K(n) + 2\sqrt{g(n)} + c$ for all n, since $X_f \upharpoonright_n$ has at most $\sqrt{g(n)}$ nonzero bits. Since X is K-resolute,

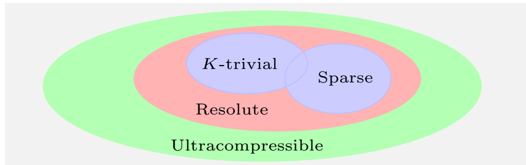

Figure 1. Classes of sequences of low initial segment complexity.

$\exists d \forall n,\ K(X\upharpoonright_n) \le K(n) + 2\sqrt{g(n)} + d$. Since $\lim_n (g(n) - 2\sqrt{g(n)}) = \infty$ this implies that $K(X\upharpoonright_n) \le K(n) + g(n)$ for almost all n. Hence X is ultracompressible.

For the second clause we note that by [Nie09, Theorem 8.2.29], every set with array computable c.e. degree is ultracompressible. Also, by [FGSW12, Theorem 1.3] (and the fact that the array computable c.e. degrees are exactly the c.e. traceable degrees) every set with array computable c.e. degree is anti-complex. On the other hand, by [Nie09, Exercise 8.2.10], every superlow set is array computable. Hence it suffices to construct a superlow c.e. set which is not K-resolute. This is entirely similar to the typical construction of a superlow c.e. set which is not K-trivial (e.g. see [Nie09, Exercise 5.2.10]) where K-triviality is replaced by (1.1). We leave this argument as an exercise for the motivated reader, as it does not present any novel features. □

A variation of ultracompressible sets was introduced in [Nie09, Section 8.2] in terms of conditional complexity. A sequence X is called *facile* if for each order h and all sufficiently large n we have $K(X\upharpoonright_n \mid n) \le h(n)$. It is not hard to see that every facile set is ultracompressible.

Proposition 1.5. *All sequences that meet resoluteness condition* (1.3) *are facile, but the converse does not hold (even for c.e. sets).*

The first clause of this proposition is straightforward while the proof of the second clause is entirely analogous to the argument in the proof of Proposition 1.4.

The analogues of Propositions 1.4 and 1.5 with respect to plain complexity also hold (with similar proofs). We illustrate some of the above observations In Figure 1, where one may interpret 'resolute' with any of the three notions of resoluteness that we considered (i.e. K-resoluteness or one of (1.1), (1.3) and the plain complexity versions of these notions). Note that in the case of (1.3), one may also replace 'ultracompressible' with 'facile' since the latter property is guaranteed by (1.3). Sparse sets will be defined in Section 2.

2. Resoluteness and sparseness

Intuitively, any information in a resolute set is coded in a very sparse way. In other words, a block of high complexity in a sequence may be used in order to reduce its initial segment complexity significantly, by 'spreading out' the bits of this block. In this section we formulate a notion of sparseness that is sufficient to guarantee resoluteness, and flexible enough to provide examples in many classes from computability theory. A concrete motivation for this notion as a tool for the study of resoluteness is the following observation. By direct coding on the values of the iterations of a given computable shift f we show that there are many sequences whose initial segment complexity is invariant under the application of f.

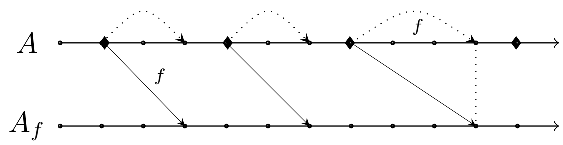

Figure 2. Construction of sparse and resolute sets.

Proposition 2.1. *Let f be a computable shift. Every many-one degree contains a set X such that $X \equiv_K X_f$.*

Proof. Let Y be a set and $g(n) = f^n(0)$. Define $X = \{g(n) \mid n \in Y\}$ so that $Y \equiv_m X$. It remains to show that $K_M(X \restriction_n) \leq K(X_f \restriction_n) + 1$ for a prefix-free machine M and all n. Let $F = \{f^i(0) \mid i \in \mathbb{N}\}$ and given $n > 0$ let $t_0, \dots t_k$ be the first $k+1$ members of F that are less than n. Then all bits of $X \restriction_n$ are 0 except perhaps $t_i, i \leq k$. Moreover $X(t_i) = X_f(t_{i+1})$ for all $i < k$. Hence in order to describe $X \restriction_n$ we just need a description of $X_f \restriction_n$ and the value of $X(t_k)$. This shows that there is a machine M such that $\forall n\ (K_M(X \restriction_n) \leq K(X_f \restriction_n) + 1)$. □

The bits of X of Proposition 2.1 that carry some information ('significant bits') are far apart with respect to f. In particular, the image of the position of each such bit under f is the position of the next significant bit. In fact, for the equivalence $X \equiv_K X_f$ it suffices that the image under f of each significant bit is at most as large as the position of the next significant bit. This property is illustrated in Figure 2 (where 'diamonds' indicate the significant bits and 'dots' indicate the rest of the bits) and is the motivation for Definition 2.2. Given an increasing function g and two sets (viewed as infinite binary sequences) X, Y we denote by $X \otimes_g Y$ the set that is obtained by replacing the $g(i)$th bit of X with the ith bit of Y, for each i.

Definition 2.2 (Sparse sets)**.** *Given an increasing function g we say that a set A is g-sparse if $A = E \otimes_f X$ for a computable set E, a computable function f with $g(f(i)) < f(i+1)$ for all i, and some set X. A set B is called sparse if it is g-sparse for all computable increasing functions g.*

Traditional notions of sparseness are based on the feature that the 1s have (in a certain sense) 'low density' in the initial segment of a sequence. An example here is the various immunity notions from classical computability theory (immune, hyperimmune, hyperhyperimmune etc.). These notions are closer to the special case of Definition 2.2 with $E = \emptyset$. Definition 2.2 also involves a notion of domination. For example, if we require $E = \emptyset$ in the definition then we only get sets A with $A' \geq_T \emptyset''$. Such sets compute a function that dominates all computable functions. By considering sparseness modulo computable sets (i.e. allowing E to be a computable set that depends on the choice of g) we obtain a much richer class, as we demonstrate in the following sections. For example, in Section 3 we show that sparse sets occur in all jump classes.

We show that sparseness indeed guarantees resoluteness. In order to do this, we need two technical observations.

Lemma 2.3. *Let $f, g : \mathbb{N} \to \mathbb{N}$ be computable increasing functions with the property that $g(n+1) > f(g(n))$ for all but finitely many n. If E is a computable set then $E \otimes_g X \equiv_K (E \otimes_g X)_f$ for all sets X.*

Proof. It suffices to find a prefix-free machine M such that

$$\forall n\ K_M\big((E \otimes_g X) \restriction_n\big) \leq K\big((E \otimes_g X)_f \restriction_n\big) + 1$$

By the choice of f, g, for each sufficiently large number n we have $|g(\mathbb{N}) \restriction_n| \leq |f(g(\mathbb{N})) \restriction_n| + 1$. Hence for each sufficiently large n, in order to describe the first n bits of $E \otimes_g X$ we only need a description of the first n bits of $(E \otimes_g X)_f$ and at most one extra bit. It follows that the machine M with the desired property exists. □

The following lemma can be proved similarly.

Lemma 2.4. *Let g be an order and let f be a computable increasing function such that $f(n) \leq g(f(n+1))$ for all but finitely many n. If $Z = E \oplus_f X$ where E is a computable set and X is any set, then $\exists c \forall n\ K(Z \restriction_n \mid n) \leq K(Z \restriction_{g(n)} \mid n) + c$.*

If a set A is sparse, then given any any computable order h we have $A = E \otimes_f X$ for some computable set E, some set X and some computable increasing function f such that $f(i) < h(f(i+1))$. Indeed, consider the increasing function $h_* : n \mapsto \min\{i\ :\ \max\{n, h(n-1)\} < h(i)\}$. Since A is sparse, it can be written as $E \otimes_f X$ for some computable set E, some set X and some computable increasing function f such that $h_*(f(i)) < f(i+1)$. In particular, $f(i) < h(f(i+1))$ for all i.

The following is a direct consequence of the above discussion, Definitions 1.3, 2.2, and Lemmas 2.3, 2.4. Moreover it holds in terms of plain complexity by the same arguments.

Corollary 2.5. *Every sparse set Z is K-resolute and meets* (1.3).

We can show that there are Turing complete sparse (hence resolute) sets. Since (by [DHNS03]) complete sets are not K-trivial, these are the first (and most easily produced) nontrivial examples of resolute sets. Without additional effort, we take a step further and show the following stronger statement. A degree $\mathbf{a}$ is called high if $\mathbf{a}' \geq \mathbf{0}''$.

Theorem 2.6. *Every high c.e. degree contains a sparse (hence resolute) c.e. set.*

Proof. Let $\mathbf{a}$ be a high c.e. degree and let A be a c.e. set in $\mathbf{a}$. By a simple variation of Martin's characterization of high c.e. degrees in terms of dominating functions in [Mar66], $\mathbf{a}$ computes a sequence (x_i) with computable approximations $\lim_s x_i[s] = x_i$ satisfying the following properties for each i, s:

(a) $f(x_i) < x_{i+1}$ for all computable shifts f and all but finitely many i;
(b) $x_i[s] < x_{i+1}[s]$ and $x_i[s] \leq x_i[s+1]$;
(c) if $x_n[s] \neq x_n[s+1]$ then $x_n[s+1] > s+1$;
(d) $x_i[s] \in \mathbb{N}^{[i]}$.

Consider the set D consisting of the numbers n such that $n = x_i[s]$ for some i, s such that $x_i \neq x_i[s]$. Clearly D is c.e. and is computable from A. Moreover it is easy to see that it computes A, so that it has degree $\mathbf{a}$. It remains to show that D is sparse. We define a computable set E and a computable function g such that $D = E \otimes_g X$ for some set X and $f(g(n)) < g(n+1)$ for all n. Let i_0 be a number such that $f(x_i) < x_{i+1}$ for all $i \geq i_0$ and let s_0 be a stage such that $x_i[s] = x_i$ for all $i \leq i_0$ and $s \geq s_0$. Define $g(0) = x_{i_0}$ and let $g(i+1) = x_{i+1}[s]$ for the least stage $s \geq s_0$ such that $x_{i+1}[s] > f(g(i))$ and $s > g(x_{i+1}[s])$. In this case we say that $g(i+1)$ was defined at stage s. According to the hypothesis about (x_i), the function g is total and computable. Moreover $f(g(n)) < g(n+1)$ for all n. The set E is defined recursively as follows. To compute $E \upharpoonright_n$ find the least j, s_j such that $g(j)$ is defined at stage s_j and $g(j) > n$. For each $t < n$, if $t \not\in g(\mathbb{N})$ and $t = x_k[s]$ for some $k < j$ and $s \leq s_i$ let $E(t) = 1$; otherwise let $E(t) = 0$. According to the properties of (x_i), after stage s_i there will be no additional positions $< n$ occupied by approximations to the values of (x_i). Hence the sets E, D agree on the positions in $\mathbb{N} - g(\mathbb{N})$. Hence $D = E \otimes_g X$ for some set X. □

The proof of Theorem 2.6 can be modified to a construction of a hyperimmune Π^0_1 sparse set. Curiously enough, such sets have to be high.

Proposition 2.7. *Every sparse hyperimmune set is high.*

Proof. Let A be a sparse set which is not high. Let f be an A-computable function such that for each i there exist at least $2i$ numbers in $A \upharpoonright_{f(i)}$. Since A is not high there exists a computable function h such that $h(i) > f(i)$ for infinitely many i. Since A is sparse, we may choose a computable set E and a computable function g such that $g(i) > h(i)$ for all i and $A = E \oplus_g X$ for some set X. From this presentation it follows that A is not hyperimmune. □

It is, perhaps, not surprising that there are K-resolute sets that are not sparse. By applying Proposition 2.7 to a K-trivial hyperimmune set we get the following (since K-trivial sets are not high [Nie05]).

Corollary 2.8. *There exists a K-trivial set which is not sparse.*

At this point we have justified the diagram in Figure 1. We note that all of the depicted classes are *meager*, in the sense of Baire category. Indeed, every weakly 2-generic set has effective packing dimension 1 so it is not ultracompressible.

Another way to obtain sparse and resolute sets is to use basis theorems on effectively closed sets. For this purpose, we need the following fact.

Theorem 2.9. *There exists a non-empty Π^0_1 class with no computable paths which consists entirely of sparse sequences.*

Proof. We will define a partial computable function φ with binary values such that the set of all total extensions of it, is the required Π^0_1 class. Fix a computable double sequence $(x_n[s])$ with $x_0[s] = 0$ and such that (a)-(d) of Theorem 2.6 hold. Let (φ_e) be an effective sequence of all partial computable functions. We assume the standard convention that if $\varphi_e(n)[s] \downarrow$ then e, n are less than s. The construction of φ is as follows: at stage s, if $\varphi_e(x_e)[s] \downarrow$ for some $e < s$ and $\varphi(x_e[s])$ is undefined, then define $\varphi(x_e[s]) = 1 - \varphi_e(x_e)[s]$. Moreover for each $i < s$, if $i \neq x_e[s]$ for all $e < s$ and $\varphi(i)$ is undefined, define $\varphi(i) = 0$.

For the verification, first note that since each $x_e[s]$ reaches a limit as $s \to \infty$ (and each time it is redefined it takes a value on which φ is currently undefined) the Π^0_1 class of total extensions of φ is perfect. Indeed, there are infinitely many $e \in \mathbb{N}$ for which φ_e is the empty function, and for these numbers e the function φ will be undefined on $x_e := \lim_s x_e[s]$. Second, there are no computable extensions of φ. Indeed, given $e \in \mathbb{N}$, if φ_e is total then by the construction we have $\varphi(x_e) \neq \varphi_e(x_e)$ where $x_e := \lim_s x_e[s]$. Finally, we show that every extension of φ is sparse. Let g be an increasing computable function. Then by the properties of $(x_e[s])$ there exists some e_0 such that $f(x_e) < x_{e+1}$ for all $e > e_0$. We may define a computable set E as follows. Let $E \upharpoonright_{x_{e_0}} = \varphi \upharpoonright_{x_{e_0}}$ and let s_0 be a stage where $x_{e_0}[s]$ has reached a limit.

Also let $y_0 = x_{e_0}$. At step $e > e_0$ find a stage $s > s_0$ such that $f(x_i)[s] < x_{i+1}[s]$ for each $i \le e$ and $x_{e+1}[s] < s$. Then define the bits of E in the interval $(y_{e-1}, x_e[s]]$ to be the bits of φ in the same interval, except where φ is (currently) undefined in which case we choose value 0. Also let $y_e = x_e[s]$.

For each n, positions between y_n and y_{n+1} in E include the values of some codes $x_e[s]$, where s is the stage found in step $n+1$ of the construction of E. By the construction, if (z_i) is the sequence of these positions we have $g(z_j) < z_{j+1}$ for each j. Therefore any extension A of φ can be written as $E \otimes_f X$ where $f(i) = z_i$ and X is some set (giving the bits of A on positions z_i). □

A set is called *computably dominated* if every function computable from it is dominated by a computable function. The strong version of the computably dominated basis theorem (see [Nie09, Theorem 1.8.44]) says that every Π^0_1 class without computable paths has a perfect subclass of computably dominated sets. By applying this basis theorem to the class of Theorem 2.9 we get more examples of sparse sets.

Corollary 2.10. *There are uncountably many computably dominated sparse sets.*

In particular, there are uncountably many resolute sets.

3. Jump inversion with K-resolute sequences

A set A is called superlow if $A' \equiv_{tt} \emptyset'$. Curiously enough, it is more involved to produce a non-trivial sparse c.e. set which does not realize the highest jump, than it is to produce one that does. A possible heuristic explanation for this is that the the notion of sparseness involves some type of domination (which is characteristic to high sets). Similar remarks apply to the construction of K-resolute sets (compare with the straightforward constructions of Section 2 that produce high sets).

Theorem 3.1. *There exists a superlow sparse c.e. set A which is not K-trivial.*

Proof. We use a priority tree construction to construct a c.e. set A with the required properties. Let (Φ_e) be an effective enumeration of all Turing functionals and let (φ_e) be an effective enumeration of all strictly increasing partial computable functions. Without loss of generality we may assume that, for all e, i, s, if $\varphi_e(i+1)[s] \downarrow$ then $\varphi_e(i)[s] \downarrow$. Moreover let $*$ denote concatenation of strings. In order to ensure that A is sparse it suffices to satisfy the following conditions.

$$R_e: \ \varphi_e \text{ is total} \Rightarrow \exists E, X, g, \ (A = E \otimes_g X \ \text{ and } \ \forall i, \ \varphi_e(g(i)) < g(i+1))$$

where E ranges over all computable sets, X ranges over all sets and g ranges over all computable functions. In order to ensure that A is not K-trivial it suffices to construct a prefix-free machine N (as usual, by enumerating a Kraft-Chaitin set of requests) such that the following conditions are satisfied.

$$P_e: \ \exists n \ K(A \restriction_n) > K_N(n) + e.$$

We will also ensure that $A' \equiv_{tt} \emptyset'$ by ensuring that the values of A' can be approximated with a computable modulus of convergence. The priority tree is the full binary tree where each level e is associated with requirements R_e, P_e. In particular, each node of length e has two branches with labels 1,0 that correspond to a guess about whether φ_e is total or not. In addition, each such node (based on the guesses about the totality of φ_i, $i < e$) will work toward the satisfaction of P_e. The construction will proceed in stages $s+1$, where a path δ_s of length s will be defined through the tree. Given a node α, we say that stage s is an α-stage if $\alpha \subset \delta_s$. Define $\ell_\alpha(s)$ to be $\max\{i \mid \forall j < i\ \varphi_{|\alpha|}(j)[s]\downarrow\}$ if s is an α-stage and 0 otherwise. A stage s is called α-expansionary if $\ell_\alpha(s) > \ell_\alpha(t)$ for all $t < s$.

Let $\alpha \to n_\alpha$ be a one-one function from the nodes of the tree to $\mathbb{N}$ such that the sum of 2^{-n_α} for all nodes α is at most $1/2$. We fix the priority list $P_0, R_0, P_1, \dots$. Each node carries a strategy P_α for $P_{|\alpha|}$ and a strategy R_α for $R_{|\alpha|}$. Injury of a strategy P_α means the initialization of it and all of its parameters. There will be no injury of the R_α strategies. Strategy P_α may be injured either because δ_s moves to the left of it, or because $\Phi_i^A(i)[s]$ becomes defined at some stage s for some $i < |\alpha|$ (with appropriately large use). At stage $s+1$ the quota for weight of the N-requests that node α may enumerate is $2^{-t_\alpha[s]}$, where $t_\alpha[s] = n_\alpha + u_\alpha[s] + |\alpha|$ and $u_\alpha[s]$ is the number of times that P_α has been injured in the stages up to s. A number is called *large* at some stage of the construction if it is larger than the value of every parameter of the construction up to that stage.

Strategy R_α will define a computable sequence (q_i^α) of potential 'codes' such that $\phi_e(q_i^\alpha) < q_{i+1}^\alpha$ for all i such that $q_{i+1}^\alpha\downarrow$. If $\varphi_{|\alpha|}$ is partial, the sequence (q_i^α) will be finite. These 'codes' will be chosen inductively as a subsequence of (q_i^β), where β is the largest initial segment of α with $\beta * 1 \subseteq \alpha$ (if such segment does not exist, codes are chosen as a subsequence of the identity sequence). For each α we let $p_i^\alpha = q_i^\beta$ where β is as above, and $p_i^\alpha = i$ if such β does not exist.

Strategy for P_α.

(1) Pick a *large* number m_α.
(2) Let D_α be a set of 2^{t_α} terms of $(p^\alpha_{\langle\alpha,2i+1\rangle})$ with $i > m_\alpha$.
(3) Let $r_\alpha = \max D_\alpha + 1$ and enumerate an N-description of r_α of length $t_\alpha - |\alpha|$.
(4) Wait until $K(A\restriction_{r_\alpha})[s] \le K_N(r_\alpha)[s] + |\alpha|$.
(5) Enumerate $\max(D_\alpha - A[s])$ into A and go to step 4.

Note that the loop between steps 4 and 5 can only be repeated at most $2^{t_\alpha} - 1$ times. Hence $K(A\restriction_{r_\alpha}) > K_N(r_\alpha) + |\alpha|$. We say that node P_α *requires attention* at stage $s+1$ if $\alpha \subseteq \delta_s$ and the strategy for α is ready to perform the next step. In other words, in the following cases:

(a) m_α is undefined;
(b) $m_\alpha\downarrow$ but D_α is undefined, and there are 2^{t_α} terms of (p_i^α) as required in step (2) of the strategy for P_α;
(c) the strategy is in step 4 and $K(A\restriction_{q_\alpha})[s] \le K_N(n)[s] + e$.

The strategy R_α operates at α-expansionary stages and defines (q_i^α). Note that since $\varphi_{|\alpha|}$ is increasing, we also have $q_i^\alpha \ge i$ for all I such that q_i^α is defined.

Strategy for R_α.

(1) Let q_0^α be $p^\alpha_{\langle\alpha,0\rangle}$.
(2) Let j be the largest number such that $q_j^\alpha\downarrow$ and define q_{j+1}^α to be the least $p^\alpha_{\langle\alpha,2i\rangle}$ which is greater than $\varphi_{|\alpha|}(q_j^\alpha)$.

We say that R_α *requires attention* at stage $s+1$ if $\alpha * 1 \subseteq \delta_s$ and the strategy is ready to perform the next step. In other words, if q_0^α is undefined or $\varphi_{|\alpha|}(q_j^\alpha)$ is defined for some j but q_{j+1}^α is undefined. The construction includes an injury of the strategies from the implicit lowness requirement. Injury of α means injury P_α.

Construction. At stage $s+1$ define a path δ_s of length s inductively, starting from the root and from each node α choosing branch 1 if s is an α-expansionary stage and 0 otherwise. Injure all nodes to the right of δ_s. For each $e < s$ such that $\Phi_e^A(e)[s]\downarrow$ with use u_e and each α such that $|\alpha| \geq e$, $m_\alpha[s] < u_e$ injure P_α. For each $\alpha \subset \delta_s$ for which R_α requires attention, execute the next step of R_α. If some P_α with $\alpha \subset \delta_s$ requires attention, pick the least such α, execute the next step of its strategy.

Verification. We first verify that $A' \leq_{tt} \emptyset'$. Each P_α may only be injured by computations $\Phi_e^A(e)[s]\downarrow$ for $e \leq |\alpha|$. Every injury of α initiates another round of the strategy of α. At any round, the maximum amount of enumerations into A that α may perform is bounded by the current value of t_α. On the other hand, the current value of t_α may be computed by the number of injuries that α has endured. It follows that there is a computable bound on the number of times that each computation $\Phi_e^A(e)[s]\downarrow$ may be 'disturbed' (by enumeration into A below the current use of the computation). Hence A' may be computably approximated with a computable modulus of convergence, which shows that $A' \leq_{tt} \emptyset'$.

Let δ be the leftmost path such that for all n we have $\delta\restriction_n \subset \delta_s$ at infinitely many stages s. By induction we show the following for each $\alpha \subset \delta$:

(i) P_α is injured or requires attention finitely many times;
(ii) if $\alpha * 1 \subset \delta$ then (q_i^α) is total.

Let α be a node and suppose that these clauses hold for all $\beta \subset \alpha$. According to the above discussion, beyond a certain stage the computations $\Phi_e^A(e)$ will either converge permanently or diverge permanently. Therefore the lowness requirements will stop injuring P_α. On the other hand (by the induction hypothesis) beyond a certain stage the strategies P_γ for $\gamma \subset \alpha$ and γ to the left of α will stop requiring attention. Therefore they will cease enumerating numbers in to A and P_α will stop being injured. After such a stage, P_α will stop requiring attention before it has completed 2^{t_α} enumerations into A. This completes the induction step for (i).

For (ii) note that by the induction hypothesis the sequence (p_i^α) is total. We may assume that $\alpha * 1 \subset \delta$ (otherwise (ii) holds trivially). Then there will be infinitely many α-expansionary stages and, by the construction, (q_i^α) will be totally defined. This completes the inductive proof of (i), (ii).

Finally we show that A meets all requirements P_e, R_e. Let α be the unique node on δ of length e. For P_e, let s_0 be a stage after which P_α is not injured. By properties (i)-(iii) that we established it follows that (p_i^α) is total. Moreover the terms $p^\alpha_{\langle\alpha,2i+1\rangle}$ may only be enumerated into A by P_α. Hence the strategy P_α will complete the preliminary steps (1)-(3) and will enter the loop (4)-(5) thus ensuring (as explained in the remark following the description of this strategy) that $K(A\restriction_{r_\alpha}) > K_N(r_\alpha) + |\alpha|$ for a certain number r_α. It follows that P_e is met.

For R_e, if $\alpha * 0 \subseteq \delta$ then φ_e is partial and R_e is met. Otherwise $\alpha * 1 \subseteq \delta$ and by (iii), (q_i^α) is total. Note that when a term q_i^α is defined, the strategies to the right of $\alpha * 1$ will not enumerate into A any numbers $\leq q_i^\alpha$. The same holds for the strategies $\gamma \subseteq \alpha$ or those that lie to the left of $\alpha * 1$. Moreover the only numbers enumerated in A by the strategies extending $\alpha * 1$ are terms of (q_i^α). It follows that α defines a computable set E and the function $g(i) = q_i^\alpha$ such that $A = E \otimes_g X$ for some set X. Moreover by the definition of (q_i^α) we have $\varphi_e(g(i)) < g(i+1)$ for all i. Hence R_e is met. $\square$

In some respect, the construction in the proof of Theorem 3.1 resembles the construction of a maximal set. However, the lowness requirements apparently make the use of some type of a tree argument necessary.

Corollary 3.2. *There exists a superlow K-resolute c.e. set A which is not K-trivial.*

A general jump inversion theorem for sparse sets is easy to obtain since we have already constructed a perfect Π_1^0 class of sparse sets with no computable paths.

Theorem 3.3 (Jump inversion with sparse sets)**.** *Every jump class contains a sparse set. In particular, every degree above $\mathbf{0}'$ contains the jump of a sparse set.*

Proof. By [JS72] given a degree $\mathbf{a} \geq \mathbf{0'}$ and a Π_1^0 class P with no computable members, there exists $X \in P$ such that X' is a member of $\mathbf{a}$. Therefore the theorem is a consequence of Theorem 2.9. $\square$

The jump inversion for c.e. sparse sets involves a modification of the argument that we used in the proof of Theorem 3.1.

Theorem 3.4 (Jump inversion with sparse c.e. sets)**.** *For every Σ_2^0 set $S \geq_T \emptyset'$ there exists a sparce c.e. set A such that $A' \equiv_T S$.*

Proof. The argument here is similar to the one in the proof of Theorem 3.1, so we use the same notation and terminology. Requirements R_e remain the same and the tree of strategies is also the same. We also need to satisfy $S \equiv_T A'$. The coding of S into A' will be achieved via the standard 'thickness' requirements, only that the codes that are used need to be chosen from the ones produced by the R_e strategies. Let D be a c.e. set such that if $e \in S$ then $D^{[e]} = \mathbb{N}\restriction_n$ for some n, and if $e \notin S$ then $D^{[e]} = \mathbb{N}^{[e]}$. We may fix an enumeration of D such that at each stage s, if $\langle e, n\rangle \in D^{[e]}[s]$ then $\langle e, i\rangle \in D^{[e]}[s]$ for all $i < n$. In the following construction, 'injury' of a node α is merely a way to say that δ_{2s+1} moved to the left of α. We may assume that if Φ_e^A is undefined then $\Phi_e^A[s]$ is undefined for infinitely many s.

Construction. At stage $2s+1$ define a path δ_{2s+1} of length s inductively, starting from the root and from each node α choosing branch 1 if s is an α-expansionary stage and 0 otherwise. For each $\alpha \subset \delta_{2s+1}$ for which R_α requires attention, execute the next step of R_α and injure all nodes that lie to the right of δ_{2s+1}. At stage $2s+2$, for each α with $|\alpha| < s$ enumerate into A all $p^\alpha_{\langle \alpha, 2i+1\rangle}$ which are defined with $i \in D^{[|\alpha|]}[2s+2]$ and are larger than all uses of any computations $\Phi_i^A(j)[2s+1]\downarrow$ with $j < e$ and larger than the last stage where α was injured.

Verification. We first verify that $A' \leq_T S$. Since S computes $\emptyset'$, it also computes D. In order to decide if $\Phi_e^A(e)\downarrow$ we first compute a stage s_0 at which $D^{[i]}[s_0] = D^{[i]}$ for all $i < e$ such that $D^{[i]}$ is finite. Moreover inductively, we may compute whether $\Phi_i^A(i)\downarrow$ for each $i < e$, and a stage $s_1 > s_0$ such that $\Phi_i^A(j)[s]\downarrow$ for all $s \geq s_1$ and each $j < e$ such that $\Phi_j^A(j)\downarrow$. Let v_e be the maximum use of the oracle A in these computations. Next, we ask if there is a stage $2s+2 > s_1$ such that $\Phi_e^A(e)[2s+2]$ is defined with some use u_e and for all $p^\alpha_{\langle \alpha, 2i+1\rangle}[s]$ with $|\alpha| < e$, $D^{[|\alpha|]} = \mathbb{N}^{[|\alpha|]}$,

$p^{\alpha}_{\langle \alpha,2i+1\rangle}[s] > v_e$ and $p^{\alpha}_{\langle \alpha,2i+1\rangle}[s] \le u_e$ we have $i \in D^{[|\alpha|]}[2s+2]$ or the last stage where α was injured is larger than u_e. If such a stage does not exist, then clearly $\Phi^A_e(e)$ is undefined. Otherwise, the construction will preserve the computation, hence $\Phi^A_e(e)$ is defined. Hence S computes A'.

Next, we show that $S \le_T A'$. In order to decide if $e \in S$ we first find a stage s_0 at which all computations $\Phi^A_i(i)$, $i < e$ that eventually converge, actually converge at s_0 with correct A use. Moreover let u_e be the maximum of these uses. Then we search for some x such that one of the following holds :

(a) for all α with $|\alpha| = e$, all $i > x$ and all $s > x$ either $p^{\alpha}_{\langle \alpha,2i+1\rangle}[s]$ is undefined, or it is $\le u_e$ or it is a member of $A[s]$;
(b) for all α with $|\alpha| = e$, all $i > x$ and all x either $p^{\alpha}_{\langle \alpha,2i+1\rangle}[x]$ is undefined, or it is $\le u_e$ or it is not a member of $A[s]$.

If $e \in S$ we show that (a) holds for some x. Indeed, in this case $D^{[|\alpha|]} = \mathbb{N}^{[|\alpha|]}$ and all defined terms $p^{\alpha}_{\langle \alpha,2i+1\rangle}[s]$ which are not prohibited by the convergence of $\Phi^A_i(i)$, $i < e$ will eventually enter A according to the construction. By a standard use of 'true stages' in the enumeration of A (i.e. stages s where for the least number n entering A we have $A[s] \upharpoonright_n$ is a prefix of A) we get that all of these terms that are larger than u_e will be permitted to enter A at some point of the construction.

If $e \notin S$ we show that (b) holds for some x. Indeed, in this case $D^{[|\alpha|]}$ is finite for all α with $|\alpha| = e$. Let x be larger than all the elements of this set. Then none of the codes $p^{\alpha}_{\langle \alpha,2i+1\rangle}$ with $i > x$ that are defined may enter A.

Furthermore, it is not possible that both (a), (b) occur for some x. Indeed, let δ be the leftmost path such that $\delta \subset \delta_{2s+1}$ for infinitely many s. If $\alpha = \delta \upharpoonright_{|\alpha|}$ then (p^{α}_i) is total, which shows that at least one of (a), (b) must fail (for sufficiently large x). Since the search for an x such that (a) or (b) hold is computable in A', this gives a computation of whether $e \in S$ from A'.

We conclude with a proof that A meets each R_e requirement. Let α be the unique node on δ of length e and let s_0 be a stage such that δ_s is to the right of α or an extension of it, for all $s \ge s_0$. If $\alpha * 0 \subseteq \delta$ then φ_e is partial and R_e is met. Otherwise $\alpha * 1 \subseteq \delta$ and (q^{α}_i) is total. Note that when a term q^{α}_i is defined at some stage $s > s_0$, no numbers p^{β}_i for β to the right of $\alpha * 1$ will enumerated into A after s, unless they are larger than s. The same holds for the nodes β which lie to the left of $\alpha * 1$. Moreover the only numbers enumerated in A by the strategies extending $\alpha * 1$ are terms of (q^{α}_i). Finally the finitely many nodes γ that prefix α enumerate computable (possibly infinite) sets of codes $p^{\gamma}_{\langle \gamma,2i+1\rangle}$ into A. It follows that α can define a computable set E and the function $g(i) = q^{\alpha}_i$ such that $A = E \otimes_g X$ for some set X. Moreover by the definition of (q^{α}_i) we have $\varphi_e(g(i)) < g(i+1)$ for all i. Hence R_e is met. □

4. Completely resolute and resolute-free degrees

We are interested in two extremes, namely the degrees which do not contain resolute sets and the degrees that consist entirely of resolute sets. A degree is called *completely K-resolute* if every set in it is K-resolute. Similar definitions apply to the other notions of resoluteness that we have considered. Note that every K-trivial degree is completely K-resolute, so we will be interested in nontrivial examples of such degrees. A degree is called *resolute-free* if it does not contain any resolute set (with respect to any of the definitions of resoluteness that we have discussed).

It turns out that the existence of such degrees is very related to two observations between bounded Turing reductions (i.e. weak truth table reductions) and resolute sets. Note that if f is a computable shift and $A \equiv_K B$ then $A_f \equiv_K B_f$.

Proposition 4.1. *If A is K-resolute and $A \equiv_K B$ then B is also K-resolute. The same holds for 'weakly K-resolute' in place of 'K-resolute'.*

Proof. First, we show the case for K-resolute sequences. Let f be a computable shift. Under the assumptions there are constants c_i such that for all n,

$$K(B\restriction_n) \le K(A\restriction_n) + c_0 \le K(A_f\restriction_n) + c_1 \le K(B_f\restriction_n) + c_3.$$

Hence $B \equiv_K B_f$. Since f was chosen arbitrarily, B is also K-resolute.

Second, for the case of weakly K-resolute sequences let f be a computable shift. Under the assumptions there are constants d_i such that for all n,

$$K(B\restriction_{f(n)}) \le K(A\restriction_{f(n)}) + d_0 \le K(A\restriction_n) + d_1 \le K(B\restriction_n) + d_3.$$

Since f was chosen arbitrarily, B is also weakly K-resolute. $\square$

Proposition 4.2. *If B is weakly K-resolute and $A \le_{wtt} B$ then $A \le_K B$.*

Proof. Since $A \le_{\mathrm{wtt}} B$ there is a computable increasing function f such that $\exists d \forall n\ (K(A\restriction_n) \le K(B\restriction_{f(n)}) + d)$. Since B is weakly K-resolute, there exists a constant c such that $\forall n\ (K(B\restriction_{f(n)})) \le K(B\restriction_n) + c)$. Hence $A \le_K B$. $\square$

These observations point to the fact that in order to produce a degree which does not contain any resolute sets it suffices to produce a set that is not resolute and its Turing degree 'collapses' to (i.e. contains) a single weak truth table degree. Similarly, in order to produce a degree which consists entirely of resolute sets it suffices to produce a resolute set whose Turing degree 'collapses' to (i.e. contains) a single weak truth table degree.

Proposition 4.3. *Every computably dominated K-resolute degree is completely K-resolute. Moreover the same holds for weakly K-resolute in place of K-resolute.*

By the application of the uncountable version of the computably dominated basis theorem for Π^0_1 classes (e.g. see [Nie09, Theorem 1.8.44]) on the class of Theorem 2.9, along with Corollary 2.5 we have the following consequence.

Corollary 4.4. *There exist uncountably many completely K-resolute degrees.*

In particular, since there are only countably many K-trivial degrees, there are completely K-resolute degrees which are not K-trivial. Similar results hold for the resolute-free degrees.

Corollary 4.5. *There exist uncountably many resolute-free degrees.*

Indeed, there are uncountably many 1-random computably dominated degrees.

Corollary 4.6. *Every (weakly) K-resolute sequence computable from a Martin-Löf random computably dominated degree is computable.*

Proof. By Demuth [Dem88] (also see [DH10, Theorem 8.6.1] for a neat proof) every noncomputable set that is truth-table reducible to a Martin-Löf random set is Turing equivalent to a Martin-Löf random set. On the other hand, there is a Martin-Löf random set of computably dominated degree, and computably dominated degrees consist of a single truth-table degree. Hence the statement is a consequence of Propositions 4.1 and 4.2. □

We are interested in c.e. examples of completely K-resolute and resolute-free degrees. In [FGSW12, Theorem 4.3] it was shown that there exists a c.e. degree which contains no anti-complex sets. Since every ultracompressible set is anti-complex, this degree does not contain any ultracompressible or (by Proposition 1.4) resolute sets.

Theorem 4.7. *There exists a resolute-free c.e. degree.*

Finally, we wish to produce nontrivial examples of completely K-resolute c.e. degrees. A c.e. degree is called *contiguous* if all the c.e. sets in it are weak truth table equivalent. The existence of nontrivial contiguous degrees was first shown and exploited in [LS75]. A degree is called *strongly contiguous* if all the sets in it are weak truth table equivalent; in other words, it consists of a single weak truth table degree. The existence of strongly contiguous c.e. degrees was shown in [Dow87].

Theorem 4.8. *Every strongly contiguous c.e. degree is completely K-resolute.*

Proof. We say that a c.e. degree $\mathbf{a}$ is 'wtt-bottomed' if the c.e. weak truth table degrees inside $\mathbf{a}$ have a least element. It suffices to show that every c.e. set in the least weak truth table degree inside a wtt-bottomed degree is K-resolute. Indeed, strongly contiguous c.e. degrees are clearly wtt-bottomed so the result would follow from Proposition 4.1.

Assume that $\mathbf{a}$ is a 'wtt-bottomed' c.e. degree and A is a c.e. set in the least weak truth table c.e. degree inside $\mathbf{a}$. We show that A is K-resolute. Given a computable shift f we wish to construct a prefix-free machine M such that

$$\forall n\ K_M(A\restriction_n) \le K(A_f\restriction_n). \tag{4.1}$$

Without loss of generality we may assume that the weight of the underlying universal prefix-free machine U (i.e. the sum of all $2^{-|\sigma|}$ such that $U(\sigma)\downarrow$) is $< 2^{-1}$. In order to define M, we construct an auxiliary c.e. set B such that $B \equiv_T A$. Let $A[s]$ be the enumeration of A with respect to a standard enumeration of all c.e. sets (and Turing functionals). The enumeration of B will be defined in these stages via a standard system of movable markers $\delta(n)[s]$ When we say 'move $\delta(n)$' at stage $s+1$ of the construction we mean

- enumerate $\delta(n)[s]$ into B;
- let $\delta(n)[s+1] = \langle n, s+1\rangle$;
- let $\delta(i)[s+1] = \langle n, s+1\rangle$ for each $i \in (n, s]$.

Let $\delta(0)[0] = 0$. We may assume that any number that enters A at stage s is strictly less than s. Let $g(n) = \max\{i \mid f(i) \le n\}$. Note that $g(n) \le n$ for all n.

Enumeration of B. At stage $s+1$ define $\delta(s+1)[s+1] = s+1$. If n be the least number that enters A, move $\delta(g(n))$. If there is no such n, do nothing more.

Note that the enumeration of B is well defined. In order to show that $A \equiv_T B$, note that for each m

(4.2) $\delta(m)$ only moves if the approximation to $A \upharpoonright_{f(m+1)}$ changes.

Hence each $\delta(m)$ reaches a limit. Moreover, since $g(n) \leq n$ for all n, every time that the approximation to $A \upharpoonright_n$ changes, the approximation to $B \upharpoonright_{\delta(n)}$ also changes. Also when $\delta(n)$ moves, its current value is enumerated in B. Hence $A \leq_T B$. On the other hand, by (4.2) and the fact that $\delta(m)[s] \geq m$ for all s it follows that the approximation to $B \upharpoonright_n$ does not change unless the approximation to $A \upharpoonright_{f(n+1)}$ changes. This shows that $B \leq_T A$. Hence $A \equiv_T B$.

By the hypothesis on A there exists a Turing functional Γ with a computable bound on use function γ such that $\Gamma^B = A$. Let (s_i) be an increasing computable sequence of the 'expansionary stages' in the reduction $\Gamma^B = A$, i.e. the stages s where the maximum n_s such that $\Gamma^B \upharpoonright_{n_s} = A \upharpoonright_{n_s}$ at stage s is larger than the corresponding numbers n_t for all $t < s$. Clearly, $n_{s_i} \geq i$ for all i. We may assume that if $\gamma(n)[s]$ is defined then its value is $< s$.

(4.3) Suppose that $A[s_t] \upharpoonright_n \neq A[s_{t+1}] \upharpoonright_n$ for some t and $n \leq t$. Then for all $k > t$, if $A[s_k] \upharpoonright_n \neq A[s_{k+1}] \upharpoonright_n$ we also have have $A[s_k] \upharpoonright_{g(n)} \neq A[s_{k+1}] \upharpoonright_{g(n)}$

Indeed, by stage s_t all $\gamma(i)$, $i \leq t$ are defined. In the interval of stages $[s_t, s_{t+1}]$ some number $m < n$ enters A. Hence in the construction of B (which runs on the same stages) $\delta(g(m))[s_{t+1}]$ is defined and larger than $\gamma(n)$. Hence if $A[s_k] \upharpoonright_{g(n)} = A[s_{k+1}] \upharpoonright_{g(n)}$ for some $k > t$ we also have $A[s_k] \upharpoonright_{g(m)} = A[s_{k+1}] \upharpoonright_{g(m)}$ which means that no number $\leq \gamma(n)$ will be enumerated into B in the interval of stages $(s_k, s_{k+1}]$. Since (s_i) are expansionary stages and $\Gamma^B = A$ it follows that $A[s_k] \upharpoonright_n \neq A[s_{k+1}] \upharpoonright_n$. This concludes the proof of (4.3).

Finally we may use (4.3) in order to construct a prefix-free machine M with the property (4.1). We do this dynamically during the stages (s_i) using a standard Kraft-Chaitin request set. At stage s_i, for each $n < t$ such that $K_M(A \upharpoonright_n)[s_i] > K(A_f \upharpoonright_n)[s_i]$ we enumerate a description of $A \upharpoonright_n$ of length $K(A_f \upharpoonright_n)[s_i]$. It suffices to show that the 'weight' of the requests is bounded by 1. Fix n. By (4.3) each description of the universal machine U of a string of length n (in particular the strings that have been current values of $A_f \upharpoonright_n$) corresponds to at most two M-descriptions (which we enumerate in order to reduce $K_M(A \upharpoonright_n)$). Indeed, (4.3) says that if we enumerate two descriptions of $A \upharpoonright_n$ based on the same U-description of $A_f \upharpoonright_n$ (in fact, same value of $A_f \upharpoonright_n$) then the next description of $A \upharpoonright_n$ will be enumerated based on a new description (and new value) of $A_f \upharpoonright_n$. Since the weight of the domain of the universal prefix-free machine is $< 2^{-1}$, the weight of the request set for M is bounded by 1. □

We note that the proof of Theorem 4.8 is easily adaptable for the other resoluteness notions that we have considered. For example, it holds with respect to plain complexity.

A degree $\mathbf{a}$ is low if $\mathbf{a}' = \mathbf{0}'$. By [ASF88] there exists a strongly contiguous c.e. degree which is not low. Hence we have the following consequence.

Corollary 4.9. *There exists a completely K-resolute c.e. degree which is not low.*

A number of questions regarding the relationship between the Turing degrees and the K-degrees were raised in [MN06] and answered in [MS07]. For example, in [MS07] it was observed that there are uncountably many sets such that all of the sets in their Turing degree are in the same K-degree (i.e. have the same initial segment prefix-free complexity). In particular, the K-trivial c.e. degrees are not the only degrees such that all of their members are in the same K-degree. Here we see this phenomenon inside the c.e. degrees.

Corollary 4.10. *There exists a c.e. Turing degree which is not low, yet all the sets in it have the same initial segment (plain or prefix-free) complexity .*

We note that by [MS07] the complete c.e. degree does not have this property, i.e. it contains sets with different initial segment complexity.

References

[ASF88] K. Ambos-Spies and P. A. Fejer. Degree theoretic splitting properties of recursively enumerable sets. *J. Symbolic Logic*, 53:1110–1137, 1988.

[Dem88] O. Demuth. Remarks on the structure of tt-degrees based on constructive measure theory. *Comment. Math. Univ. Carolin.*, 29(2):233–247, 1988.

[DH10] Rod Downey and Denis Hirshfeldt. *Algorithmic Randomness and Complexity.* Springer, 2010.

[DHNS03] Rod G. Downey, Denis R. Hirschfeldt, André Nies, and Frank Stephan. Trivial reals. In *Proceedings of the 7th and 8th Asian Logic Conferences*, pages 103–131, Singapore, 2003. Singapore Univ. Press.

[Dow87] R. G. Downey. Δ^0_2 degrees and transfer theorems. *Illinois J. Math.*, 31:419–427, 1987.

[FGSW12] Johanna Franklin, Noam Greenberg, Frank Stephan, and Guohua Wu. Anti-complexity, lowness and highness notions, and reducibilities with tiny use. Submitted, 2012.

[JS72] Carl G. Jockusch, Jr. and Robert I. Soare. Π^0_1 classes and degrees of theories. *Trans. Amer. Math. Soc.*, 173:33–56, 1972.

[KHMS06] Bjørn Kjos-Hanssen, Wolfgang Merkle, and Frank Stephan. Kolmogorov complexity and the recursion theorem. In *STACS*, pages 149–161, 2006.

[KHMS11] Bjørn Kjos-Hanssen, Wolfgang Merkle, and Frank Stephan. Kolmogorov complexity and the recursion theorem. *Trans. Amer. Math. Soc.*, 363, 2011.

[LL99] James I. Lathrop and Jack H. Lutz. Recursive computational depth. *Inf. Comput.*, 153(1):139–172, 1999.

[LS75] R. Ladner and L. Sasso. The weak truth-table degrees of the recursively enumerable sets. *Ann. Math. Logic*, 8:429–448, 1975.

[Mar66] Donald A Martin. Classes of recursively enumerable sets and degrees of unsolvability. *Z. Math. Logik Grundlag. Math.*, 12:295–310, 1966.

[MN06] Joseph S. Miller and André Nies. Randomness and computability: open questions. *Bull. Symbolic Logic*, 12(3):390–410, 2006.

[MS07] Wolfgang Merkle and Frank Stephan. On C-degrees, H-degrees and T-degrees. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC 2007), San Diego, USA, 12–16 June 2007*, pages 60–69, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[Nie05] André Nies. Lowness properties and randomness. *Adv. Math.*, 197(1):274–305, 2005.

[Nie09] André Nies. *Computability and Randomness.* Oxford University Press, 2009.

George Barmpalias: State Key Lab of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, P.O. Box 8718, China.

E-mail address: barmpalias@gmail.com

URL: http://www.barmpalias.net

Rod G. Downey: School of Mathematics, Statistics and Operations Research, Victoria University, P.O. Box 600, Wellington, New Zealand.

E-mail address: rod.downey@vuw.ac.nz

URL: http://homepages.ecs.vuw.ac.nz/~downey