Int J Software Informatics, Volume 5, Issue 4 (2011), pp. 579–593
International Journal of Software and Informatics, ISSN 1673-7288

E-mail: ijsi@iscas.ac.cn
http://www.ijsi.org
Tel: +86-10-62661040

On Strings with Trivial Kolmogorov Complexity

George Barmpalias

(State Key Laboratory of Computer Science, Institute of Software, the Chinese Academy of Sciences, Beijing 100190, China)

Abstract The Kolmogorov complexity of a string is the length of the shortest program that generates it. A binary string is said to have trivial Kolmogorov complexity if its complexity is at most the complexity of its length. Intuitively, such strings carry no more information than the information that is inevitably coded into their length (which is the same as the information coded into a sequence of 0s of the same length). We study the set of these trivial sequences from a computational perspective, and with respect to plain and prefix-free Kolmogorov complexity. This work parallels the well known study of the set of nonrandom strings (which was initiated by Kolmogorov and developed by Kummer, Muchnik, Stephan, Allender and others) and points to several directions for further research.
Key words: Kolmogorov complexity; K-trivial; C-trivial; strings; completness; truth table degrees

Barmpalias G. On strings with trivial Kolmogorov complexity. ***Int J Software Informatics*****, Vol.5, No.4 (2011): 579–593. http://www.ijsi.org/1673-7288/5/i112.htm**

1 Introduction

In order to measure the information that is coded into binary strings, Kolmogorov[9] provided a formal framework which is based on the theory of computability and Turing machines. Given a Turing machine M that operates on binary strings, the Kolmogorov complexity of a string σ relative to M is the length of the shortest string (program) τ such that $M(\tau) = \sigma$ (in words, τ is a *description* of σ). If there is no such string τ, then this complexity of σ is infinite. We denote the Kolmogorov complexity of σ relative to M by $C_M(\sigma)$. The existence of optimal universal Turing machines allows for a theory of Kolmogorov complexity that does not depend on the underlying Turing machine M in any essential way.

Definition 1.1 (Optimal machines, see Ref. [11]). An optimal universal machine U is a Turing machine with the following property: every Turing machine

Barmpalias was supported by a research fund for international young scientists No.611501-10168 and an *International Young Scientist Fellowship* number 2010-Y2GB03 from the Chinese Academy of Sciences. Partial support was also obtained by the *Grand project: Network Algorithms and Digital Information* of the Institute of Software, Chinese Academy of Sciences.
The author would like to thank Martijn Baartse, Adam Day and Frank Stephan for discussing this work and reading early drafts of this paper. The peripheral results presented in Section 5 are joint with Tom Sterkenburg and a detailed presentation of them can be found in his thesis Ref. [18, Chapter 2].
Corresponding author: George Barmpalias, Email: barmpalias@gmail.com
Received 2011-02-23; Revised 2011-05-03; Accepted 2011-05-15.

M can be associated with a constant c such that for all σ, if $M(\sigma)$ is defined then $M(\sigma) = U(\tau)$ for some string τ such that $|\tau| \leq |\sigma| + c$.

In other words, an optimal universal machine can simulate any other machine with only a constant overhead. Let U be the machine that acts on input $0^e1\sigma$ by simulating the machine with code e on input σ and outputs the result of this computation. Clearly the Kolmogorov complexity of strings with respect to any other Turing machine M can only be smaller than the one with respect to U by at most a fixed constant (corresponding to its code). Hence without loss of generality we may fix U as the underlying machine and denote the corresponding complexity by C. A variation of this approach is obtained when we restrict our considerations to prefix-free Turing machines, i.e. with prefix-free domain (no string in it is an extension of another). The above considerations allow for the definition of prefix-free complexity K which is based on an underlying *optimal universal prefix-free machine*. The latter is defined as in Definition 1.1 but restricted to prefix-free machines.

Kolmogorov[9] called a string σ random if $C(\sigma) \geq |\sigma|$ and showed that the set of non-random strings is simple (i.e. computably enumerable and its complement does not contain any infinite computably enumerable sets). Intuitively, a string is random if it contains a lot of information. The study of the computational properties of the set of non-random strings continued with the work of Kummer, Muchnik, Stephan, Allender and others. Kummer[10] used a non-uniform argument in order to show that it is truth table complete. Muchnik and Positselsky[13] studied the set of non-random strings with respect to prefix-free Kolmogorov complexity and showed it may be truth table incomplete if a certain underlying optimal universal prefix-free machine is chosen. On the other hand Allender, Buhrman, and Koucký[13] showed that the latter can be truth table complete under a suitably chosen underlying optimal universal prefix-free machine, and studied the same question with respect to resource-bounded versions of Kolmogorov complexity. Stephan[17] studied the set of nonrandom strings (with respect to C and K) further, in the context of the lattice of computably enumerable sets and the Ershov hierarchy of n-c.e. sets.

In this paper we are interested in the collection of strings that are very far from being random. These strings can be described as easily as a sequence of 0s of the same length. In other words, in terms of Kolmogorov complexity such a string is as simple as the one that we obtain if we switch all of its digits to 0. This notion can be formalized in terms of plain or prefix-free complexity as follows.

Definition 1.2 (K_e and C_e trivial strings). A string σ is called K_e-trivial if $K(\sigma) \leq K(|\sigma|) + e$. Similarly, a string is called C_e-trivial if $C(\sigma) \leq C(|\sigma|) + e$.

Whenever we write $K(n)$ for $n \in \mathbb{N}$ we identify n with the string 0^n (the particular encoding of numbers into strings is not significant). The intuitive notion of trivial strings that we discussed above corresponds to the special case $e = 0$ in Definition 1.2. However since the theory of Kolmogorov complexity is always dependent on an underlying constant corresponding to a particular optimal universal system of descriptions, it is necessary to allow the formal definition to depend on a similar constant. Let us use the terms 'K-trivial and C-trivial' if the choice of the underlying constant e of Definition 1.2 is not essential, and even talk about the collection of 'trivial' strings when the nature of the underlying system of descriptions (plain or prefix-free) is also not important in the particular context. The facts that we prove about the sets of K-trivial and C-trivial strings do not depend on the underlying constant e, as long as the constant is chosen to be above a certain value. However, as it happens with the case of the collection of the nonrandom strings, some properties of the collection of trivial strings may depend on the choice of the underlying optimal universal systems of descriptions.

In Section 2 we examine the collections of trivial strings in the light of the simplicity and immunity notions from classical computability theory. We show that there is no simple set of trivial strings. Notice that Kolmogorov[9] followed a similar approach to the study of the collection of nonrandom strings, where he showed that it forms a simple set. Despite this analogy, the real reason that we follow this path of investigation is the basic question of whether the trivial strings can be effectively enumerated. Although intuitively this does not seem likely, a direct attack to this question via a straightforward diagonalization fails. However, as it turns out, the approach that we initiate in Section 3 leads to the negative answer of this basic question in Section 3.

In Section 3 we show that if $e \in \mathbb{N}$ is chosen large enough, the set of K_e-trivial strings intersects every infinite computably enumerable set of strings. Moreover, the same holds for the C-trivial strings. This result, combined with the work of Section 2, shows that when $e \in \mathbb{N}$ is chosen sufficiently large the collections of K_e-trivial and C_e-trivial strings are not computably enumerable. The results in Sections 2 and 3 also show that the many-one degree of these sets of strings (again, for suitably large constant e) is incomparable with the many-one degree of the halting problem. In particular, they are many-one incomplete. Continuing this degree theoretic analysis of the sets of trivial strings, in Section 4 we show that (for suitably large $e \in \mathbb{N}$) the set of K_e-trivial stings is in the same weak truth table degree as the halting problem. Moreover, this also holds for the set C_e-trivial strings. We note that this also holds for the set of non-random strings. An natural question here is whether these collections are also truth table complete. In the case of the set of nonrandom strings, this question turned out to be quite interesting. As we discussed above, in the case of plain complexity the set of nonrandom strings is truth table complete (independently of the underlying optimal universal machine). However in the case of prefix-free complexity it can be truth table complete or incomplete, depending on the underlying optimal universal system of descriptions.

In Section 4 we start a similar analysis to determine whether the collection of trivial strings is truth table complete or not. First, we show that in the case of prefix-free complexity it can be made complete or incomplete, if suitable underlying systems of descriptions are chosen. For the case of plain complexity we show that it is complete, again subject to a suitable choice of the underlying Turing machine. However the question remains as to whether it is truth table complete with respect to any choice of underlying optimal universal machine. In other words, does Kummer's theorem hold for the case of the trivial strings? Kummer's original argument does not translate to this case. We leave this interesting question open.

In Section 4 we study subsets of $\mathbb{N}$ (viewed as infinite binary sequences) in the light of trivial strings. A set $X \subseteq \mathbb{N}$ is called K-trivial if there is some $e \in \mathbb{N}$ such that all of its initial segments are K_e-trivial. An analogous definition applies in the case of plain Kolmogorov complexity. We show that if X is computably enumerable and not K-trivial, then it is the disjoint union of two computably enumerable sets that are not K-trivial. A similar result holds for the case of C-triviality. Actually, we prove a stronger result that can be stated in terms of the so-called C and K reducibilities (denoted $\leq_C$ and $\leq_K$ respectively) that were proposed in Ref. [7] as measures of relative randomness.

Definition 1.3 ($\leq_C$ and $\leq_K$, see Ref. [7]). Given $X, Y \subset \mathbb{N}$ we say that $X \leq_C Y$ if there is some $c \in \mathbb{N}$ such that $C(X \restriction_n) \leq C(Y \restriction_n)$ for all $n \in \mathbb{N}$. Similarly, we say that $X \leq_K Y$ if there is some $c \in \mathbb{N}$ such that $K(X \restriction_n) \leq K(Y \restriction_n)$ for all $n \in \mathbb{N}$.

We show that any computably enumerable set A is the disjoint union of two c.e. (short for 'computably enumerable') sets A_0, A_1 such that $A_i <_K A$ and $A_i \not\leq_K A_{1-i}$ for $i = 0, 1$. This theorem also holds for $\leq_C$ and can be seen as an analogue of the well known Sacks splitting theorem (e.g. see Ref. [15, Theorem 3.1]) for randomness reducibilities. Finally in Section 6 we conclude with suggestions for further research on the collection of trivial strings and a number of open questions.

The reader will notice that a number of results in this paper about the sets of K_e-trivial and C_e-trivial strings hold when e is chosen 'sufficiently large'. For these results, it is possible to find certain underlying optimal universal machines and certain (finitely many) $e \in \mathbb{N}$ with respect to which they do not hold. For example, consider Corrollary 3.2 which says that (given some underlying optimal universal machine) for sufficiently large $e \in \mathbb{N}$ the corresponding set of K_e-trivial strings are not computably enumerable. We exhibit a particular optimal universal machine with respect to which the above statement does not hold for $e = 0$.[1)] Given any optimal universal machine U define machine V by $V(11\tau) = U(\tau)$ and $V(0\tau) = 0^{|U(\tau)|}$. Then

$$\{\tau \mid K_V(\tau) \leqslant K_V(|\tau|)\} = \{0^n \mid n \in \mathbb{N}\} \tag{1.1}$$

Indeed, by definition we have $K_V(0^n) \leqslant K_V(n)$ for each $n \in \mathbb{N}$. On the other hand, if $\sigma \neq 0^{|\sigma|}$ then $K_V(\sigma) = K_U(\sigma) + 2$. But by the second clause of the definition of V we have that $K_V(0^{|\sigma|}) \leq 1 + K_U(\sigma)$. Hence $K_V(0^{|\sigma|}) < K_V(\sigma)$ which shows that σ is not K_0-trivial with respect to V. This proves (1.1), which in turn shows that the set of K_0-trivial strings with respect to V is computable. Similar counterexamples can be found for other results in this paper that require 'sufficiently large $e \in \mathbb{N}$'.

For background on algorithmic randomness we refer the reader to Ref. [11] and Ref. [14].

2 Simplicity and Immunity

Clearly every string is K_e-trivial for some $e \in \mathbb{N}$. What we are interested in here is the complexity of *the set* of K_e-trivial strings. The same applies to C-triviality. Recall that a set of strings is called immune if it does not contain any infinite c.e. set of strings. It is called *simple* if it is c.e. and it intersects all infinite c.e. sets of strings (i.e. its complement is immune).

Proposition 2.1. For each $e \in \mathbb{N}$, the sets of K_e-trivial and C_e-trivial strings are not immune.

Proposition 2.1 follows from the fact that the infinite set of strings $0^n, n \in \mathbb{N}$ is computable and each of its members is K_e-trivial and C_e-trivial for all $e \in \mathbb{N}$.

---

1) This nice example was provided to us by Adam Day.

Theorem 2.2. Let $e \in \mathbb{N}$. There is no simple set of C_e-trivial strings.
Proof: If A is a simple set of strings, then

$$\text{For each } n \in \mathbb{N} \text{ there exists } t \in \mathbb{N} \text{ such that there are at least } n \text{ strings in } A \text{ of length } t. \tag{2.1}$$

Indeed, otherwise if m be the largest number such that for infinitely many n there are m strings of length n in A the set

$$\{n \mid \text{there are exactly } m \text{ strings of length } n \text{ in } A\}$$

is c.e. and infinite. Hence if 2.1 did not hold, we would be able to enumerate an infinite c.e. set of strings in the complement of A. This would contradict the fact that A is simple.

It remains to show that given a simple set of strings A, there exists $\sigma \in A$ such that $C(\sigma) > C(|\sigma|) + e$.

We construct a Turing machine M and by the recursion theorem we may use an index d of it in its definition. The machine M simply searches for some $n \in \mathbb{N}$ and a stage s such that 2^{d+e+1} strings of length n are in $A[s]$. By (2.1) this search halts. Then it describes n with a string of length 1.

By the standard encoding of the underlying optimal universal machine that we have chosen we have $C(i) \leq C_M(i) + d$ for all $i \in \mathbb{N}$. Moreover by the definition of M we have $C_M(n) = 1$, where n is a number such that there are 2^{d+e+1} strings of length n in A. For each C_e-trivial string σ we have $C(\sigma) \leq C_M(|\sigma|) + e + d$ hence if σ is of length n then $C(\sigma) \leq 1 + e + d$. This means that there are less than 2^{1+e+d} many C_e-trivial strings of length n. Hence A contains a string which is not C_e-trivial. □

An important fact in descriptive string complexity (e.g. see Ref. [14, Lemma 5.2.21]) is that for each machine N there is some constant d such that

$$|\{\sigma \mid N(\sigma) = n \;\wedge\; |\sigma| \leq C(n) + b\}| \leq b^2 \cdot 2^{b+d} \tag{2.2}$$

for all $b \in \mathbb{N}$ and strings σ. Let U be the underlying optimal universal machine and consider the machine $N(\sigma) = |U(\sigma)|$. Given $n \in \mathbb{N}$ all descriptions of strings of length n are N-descriptions of n.

Given $n \in \mathbb{N}$, the number of strings of length n such that $C(\sigma) \leq C(n) + b$ is at most the number of the descriptions of length $\leq C(n) + b$ that describe strings of length n. Therefore it is at most the number of N-descriptions of n that have length $\leq C(n) + b$. Hence by (2.2) there is some constant c such that for all $b \in \mathbb{N}$

$$|\{\tau \mid C(\tau) \leq C(|\tau|) + b\}| \leq b^2 \cdot 2^{b+c}. \tag{2.3}$$

The proof of Theorem 2.2 becomes shorter if we use (2.3) instead of constructing machine M. We follow this route in the proof of the following analogous theorem for prefix-free complexity. In particular we use the so-called coding theorem (see e.g. Ref. [14, Theorems 2.2.25 and 2.2.26]) which implies that there is a constant d such that for all $b, n \in \mathbb{N}$ the cardinality of the set $\{\sigma \mid |\sigma| = n \;\wedge\; K(\sigma) \leq K(n) + b\}$ is less than 2^{b+d}.

Theorem 2.3. Let $e \in \mathbb{N}$. There is no simple set of K_e-trivial strings.

Proof: Suppose that A is a simple set of strings. In the proof of Theorem 2.2 it was shown that (2.1) holds. By the coding theorem there is some constant c such that for each $n \in \mathbb{N}$ there are at most c strings of length n which are K_e-trivial. Hence A cannot consist entirely of K_e-trivial strings. □

The above results will be used in Section 3 in order to establish that the set of K_e-trivial strings and the set of C_e-trivial strings are not computably enumerable, for suitably large $e \in \mathbb{N}$.

3 Computable Enumerability

We wish to exhibit some $e \in \mathbb{N}$ such that the set of K_e-trivial strings and the set of C_e-trivial strings intersect every infinite c.e. set of strings. For this reason, we need the following simple fact about the complexities K and C.

If $V = \{t_s \mid s \in \mathbb{N}\}$ is a computable 1-1 enumeration of an infinite c.e. set then there are infinitely many $s \in \mathbb{N}$ such that $C(t_s) < C(t_s)[s]$. The same holds for K in place of C. (3.1)

To see why (3.1) holds, construct a Turing machine M as follows. By the recursion theorem we can use an index d of M in its definition. For each $i \in \mathbb{N}$ find some $s > i$ such that $C(t_s)[s] > d+i+1$ and describe t_s with a string of length $i+1$. Let this chosen s be s_i. For each i there is at most one description of length $i+1$, hence machine M as prescribed above exists. By the standard encoding of the chosen underlying universal machine, $C(t_{s_i}) \le d + C_M(t_{s_i}) = d + i + 1$, hence $C(t_{s_i}) < C(t_{s_i})[s_i]$ for each $i \in \mathbb{N}$. The same argument shows (3.1) for the prefix free complexity.

The following result shows that the complements of the sets of C_e and K_e-trivial strings are immune. It will be combined with Theorems 2.2 and 2.3 in order to show that the sets of C_e and K_e-trivial strings are not computably enumerable.

Theorem 3.1. There is $e \in \mathbb{N}$ such that every infinite c.e. set of strings contains a C_e-trivial and a K_e-trivial string.

Proof: Let (W_n) be an effective enumeration of all c.e. sets of strings. For the part of the theorem that refers to plain complexity, it suffices to define a Turing machine M such that

$$R_n: \ W_n \text{ is infinite} \Rightarrow \ \exists \sigma \in W_n \ [C_M(\sigma) \le C(|\sigma|)]$$

for each $n \in \mathbb{N}$. Indeed, by the standard encoding of the underlying universal machine we have $C(\tau) \le C_M(\tau) + e$ for an index e of M and all strings τ. Hence each infinite c.e. set will contain a string σ such that $C(\sigma) \le C(|\sigma|) + e$. Similarly, for the prefix-free complexity it suffices to define a prefix-free machine M such that the modified R_n, $n \in \mathbb{N}$ with K in place of C are satisfied. The following argument will produce such a machine M simply by shuffling the descriptions that are used by the underlying universal machine (i.e. mapping the same descriptions to possibly different strings). Therefore exactly the same argument applies to both C and K. Without loss of generality we state the argument in terms of C.

In order to meet a single condition R_n one would only have to wait for a string σ to appear in W_n and then ensure that $C_M(\sigma) \le C(|\sigma|)$ by letting M describe σ with the descriptions that the underlying universal machine gives to $|\sigma|$. However when we try to satisfy all R_n we meet the following problem. When we decide to act on some $\sigma \in W_n$, other strategies may have already acted on other strings of length $|\sigma|$. If we ignore this issue, we may easily end up with M not having enough short descriptions for certain strings. In fact, Theorem 2.3 shows that such a standard 'simple set construction' strategy is bound to fail.

The solution is to employ a more sophisticated finite injury argument using (3.1). Consider the strategies in a priority list $R_0, R_1, \dots$ We order the strings first by length and then lexicographically. Also, we use the term 'universal description' to refer to the descriptions issued by the underlying optimal universal machine. At each stage each strategy may hold a string σ. At each stage, each strategy holds at most one string. Moreover at each stage and for each $n \in \mathbb{N}$ there is at most one strategy that holds a string of length n. A strategy may get hold of a string σ such that some τ of the same length is currently held by a lower priority strategy. In this case the latter strategy no longer holds τ. Consequently, if σ is held by some strategy R_j, the only reason why R_j may lose σ is if at some later stage a higher priority strategy gets hold of some string of length $|\sigma|$.

Strategy R_n requires attention at stage $s+1$ if

- it does not hold a string and
- there is some $\sigma \in W_n[s+1]$ such that $C(|\sigma|)[s+1] < C(|\sigma|)[s]$ and none of $R_i, i < n$ holds a string of length $|\sigma|$.

In this case, if σ is the least string with this property we say that R_n requires attention via σ.

Construction. At stage $s+1$ if R_i, $i < s$ is the highest priority that requires attention via some string σ, let it get hold of σ. If some R_j held some τ with $|\tau| = |\sigma|$ at stage s, it loses it at this stage. For each R_j, $j < s$ which holds some σ, if $C(|\sigma|)[s+1] < C(|\sigma|)[s]$ use the new universal description of $|\sigma|$ as an M-description of σ.

Verification. First, notice that M operates simply by using the descriptions issued by the underlying universal machine. Since this shuffling is effective, M is indeed a Turing machine. It remains to show that each R_n, $n \in \mathbb{N}$ is met. We use induction to show that each R_n, $n \in \mathbb{N}$ is met and requires attention only finitely often. Suppose that this holds for all $i < n$ and after stage s_0 no R_i, $i < n$ requires attention. If R_n gets hold of a string after stage s_0, or already holds one at stage s_0 clearly it never requires attention after stage s_0. On the other hand if it never gets hold of a string at stages $s \geq s_0$ it also does not require attention after stage s_0 (otherwise it would get hold of a string). If W_n is not infinite, R_n is met. Otherwise if we consider the infinite c.e. set of the lengths of the strings in W_n, it follows from (3.1) that R_n will require attention and get hold of a string σ at some stage $s \geq s_0$ (if it does not already have one at s_0). By the induction hypothesis it will hold σ at all latter stages. Then the construction ensures that $C_M(\sigma) \leq C(|\sigma|)$. Hence R_n is met. □

Since the complexity functions K and C are ω-c.e. (i.e. the limit of a computable function with a computable bound on the number of changes that occur before the limit is reached), the same holds for the sets of C_e-trivial and K_e-trivial strings. The following result shows however that these sets are not 2-c.e., i.e. the difference of two c.e. sets.

Corollary 3.2. For sufficiently large $e \in \mathbb{N}$, the sets of C_e-trivial and K_e-trivial strings are not c.e and not the difference of two c.e. sets.

Proof: Fix e as large as the constant of Theorem 2.1. If the set of C_e-trivial strings was c.e., by Theorem 3.1 it would be simple. But this contradicts Theorem 2.2. A similar argument applies to K_e-trivial strings, using Theorem 2.3. Finally if the set of C_e or K_e-trivial strings was the difference of two c.e. sets then it would either be c.e. or its complement would have an infinite c.e. subset. This would contradict the same theorems. $\square$

As we discussed above, the sets of K_e-trivial and C_e-trivial strings are ω-c.e. for all $e \in \mathbb{N}$ (i.e. they can be computably approximated with a computable bound on the number of changes in the approximation). We do not know if Corollary 3.2 can be extended to all levels of the Ershov hierarchy of the n-c.e. sets. Frank Stephan (personal communication) suggested that the following result from Ref. [3] may possibly be used in order to achieve this extension.

> For every underlying optimal universal machine U, there is a constant a such that the Kolmogorov complexity C (as a function from strings to $\mathbb{N}$) is f-c.e. (for a computable function $f : \mathbb{N} \to \mathbb{N}$) if and only if $f(n) \geq n/a$ for almost all n. (3.2)

Here we say that the Kolmogorov function $C : 2^{<\omega} \to \mathbb{N}$ is f-c.e. for some computable function $f : \mathbb{N} \to \mathbb{N}$ if it has a computable approximation during which the value of C on each string σ changes at most $f(|\sigma|)$ many times.

4 Truth Table Completeness

Recall that a set is c.e. iff it is m-reducible to the halting problem. Also, if the halting problem is m-reducible to a set, then this set has an infinite c.e. set in its complement. Hence the following is a direct consequence of Theorem 3.1 and Corollary 3.2.

Corollary 4.1. For sufficiently large $e \in \mathbb{N}$ the m-degree of the sets of C_e and K_e-trivial strings is incomparable to the m-degree of the halting problem.

We wish to know how the collection of trivial strings is classified with respect to other reducibilities. For example, does it have the same Turing degree as the halting problem? In the following we give a positive answer. In fact, we show that it has the same weak truth table degree as the halting problem. Recall that a set X is weak truth table to a set Y (often denoted by $\leq_{wtt}$) if every digit $X(n)$ of it can be computed via an algorithm which has access to a computably bounded (dependent on n) segment of Y. It is not hard to see that the complexity functions K and C are ω-c.e. (i.e. limits of computable functions with computably bounded number of changes in the approximation). Since a function is truth table reducible to the halting set if and only if it is ω-c.e., we have the following.

Proposition 4.2. For every $e \in \mathbb{N}$ the set of K_e-trivial and the set of C_e-trivial strings are truth table reducible to the halting set.

In other words, they can be computed via a truth table which refers to the halting set. The following theorem shows that, in fact, the collection of trivial strings has the same weak truth table degree as the halting set.

Theorem 4.3. For sufficiently large e, the sets of K_e-trivial and C_e-trivial strings are (uniformly) weak truth table complete (i.e. $\emptyset'$ is wtt-reducible to them through a single reduction).

Proof: We give the proof for prefix-free complexity, as the proof for plain complexity is entirely similar. Since there is a constant d such that $K(n) \leq 2\log n + d$ for all $n \in \mathbb{N}$, there is some n_0 such that $K(n) < 3\log n$ for all $n \geq n_0$. We are going to enumerate a c.e. set V containing at most one string of each length. Then by Ref. [5, Lemma 2.6] there is a constant c such that $K(\sigma) \leq K(|\sigma|) + c$ for all $\sigma \in V$.

Let $d_{e,n}$ be a 1-1 computable double sequence of powers of 2 such that $d_{e,n} > n_0$ and $2^{d_{e,n}-e} > d_{e,n}^3$. By the latter property we have

$$2^{d_{e,n}} \cdot 2^{-(e+3\log d_{e,n})} > 1. \tag{4.1}$$

At stage s, if n is enumerated in $\emptyset'$ for each $e \leq s$ we do the following: if $K(d_{e,n})[s] < 3\log d_{e,n}$ we enumerate into V the least string of length $d_{e,n}$ such that $K(\sigma) \geq 3\log d_{e,n} + e$ (by the choice of $d_{e,n}$, in particular (4.1), such a string exists). Notice that by the choice of $d_{e,n}$ we have $K(d_{e,n}) < 3\log d_{e,n}$ for all $e, n \in \mathbb{N}$. Moreover, V contains at most one string of each length.

Now fix $e > c$. We show how to (uniformly) compute $\emptyset'$ from the set of K_e-trivial strings. To see if $n \in \emptyset'$, find a stage $s > e, n$ such that $K(d_{e,n})[s] < 3\log d_{e,n}$ and $K(\sigma)[s] < 3\log d_{e,n} + e$ for all K_e-trivial strings of length $d_{e,n}$. We claim that $n \in \emptyset'$ iff $n \in \emptyset'[s]$. Indeed, if n was enumerated in $\emptyset'$ at a later stage s', a string τ of length $d_{e,n}$ such that $K(\tau)[s'] \geq 3\log d_{e,n} + e$ would be enumerated in V. Such a string will not be K_e-trivial by the choice of s. Since $e > c$, all strings in V are K_e-trivial. This is a contradiction. □

In view of Theorems 4.3 and 2.1 it is natural to ask if the sets of K_e-trivial and C_e-trivial strings are (or can be, under a suitable choice of the underlying optimal universal machine) truth table complete. As a matter of fact, the truth table degrees of many sets that are naturally found in algorithmic randomness tend to depend on the underlying universal coding of machines. For example, consider the set of nonrandom strings. In the case of prefix free complexity, Muchnik and Positselsky[13] showed that the set of nonrandom strings has incomplete truth table degree, under a certain choice of the underlying optimal universal prefix-free machine. On the other hand in Ref. [1] it was shown that the same set is truth table equivalent to the halting problem, under a different choice of the underlying optimal universal prefix-free machine. Despite this, Kummer[7] showed that in the case of plain complexity the set of nonrandom strings is always truth table complete. Another example of the ambiguity of the truth table degrees of notions from algorithmic randomness is Chaitin's Ω. In Ref. [8] it was shown that there are two universal optimal prefix-free machines M, N such that the truth table degrees of the respective halting probabilities Ω_N, Ω_M are incomparable. Despite this, it is known (see e.g. Ref. [6]) that the truth table degree of the halting probability of any universal optimal prefix-free machine is strictly below the truth table degree of the halting problem.

In the following we discuss the truth table degrees of the set of K and C trivial strings. Theorem 4.4 is a consequence of the following theorem from Ref. [13].

$$\text{There is a universal optimal prefix-free machine } V \text{ such that the set } O_V = \{\langle \sigma, n\rangle \mid K_V(\sigma) < n\} \text{ is not truth table complete.} \tag{4.2}$$

Theorem 4.4. There exists a universal optimal prefix-free machine V such that for all $e \in \mathbb{N}$ the set of $K_{V,e}$-trivial strings is not truth table complete.

Proof: Fix $e \in \mathbb{N}$ and let V be the machine of (4.2). By the same fact it suffices to show that the set $S_K(e) = \{\sigma \mid K_V(\sigma) \le K_V(|\sigma|) + e\}$ is truth table reducible to O_V. Let $L(e) = \{\langle \sigma, t\rangle \mid K_V(|\sigma|) = t\}$. Clearly, $L(e) \le_{tt} O_V$. Since there is a constant c such that $\forall\sigma,\ K_V(|\sigma|) \le |\sigma| + c$ we have $S_K(e) \le_{tt} O_V \oplus L(e)$. Hence since $L(e) \le_{tt} O_V$ we have $S_K(e) \le_{tt} O_V$. □

The following result is based on an adaptation of an idea that was used in Ref. [1] in order to show that with respect to a certain universal optimal prefix-free machine, the set of K-nonrandom strings is truth table complete.

Theorem 4.5. There exists an optimal universal prefix-free machine V such that for all $e \in \mathbb{N}$ the sets of $K_{V,e}$-trivial strings are (uniformly) truth table complete. The same is true for the plain complexity, regarding the set of $C_{F,e}$-trivial strings with respect to a plain machine F.

Proof: The proof applies uniformly to the prefix-free and plain complexity cases. We elaborate on the case of prefix-free complexity. Given a prefix-free machine M, let

$$T_M(e) = \{\sigma \mid K_M(\sigma) \le K(|\sigma|) + e\}$$

be the set of $K_{M,e}$-trivial strings. In the following we let 2^m (for $m \in \mathbb{N}$) denote the set of strings of length m (apart from its standard meaning as a number). It will be clear from the context whether we regard it as a number or a set of strings. We wish to build an optimal universal prefix-free machine V such that for a certain constant c and each $e \in \mathbb{N}$ and $n > c$

$$n \in \emptyset' \iff |2^{\langle e,n\rangle} \cap T_V(e)| \text{ is odd.} \tag{4.3}$$

This condition implies that $\emptyset' \le_{tt} T_V(e)$. Moreover these truth-table reductions are uniform in e (because the constant c does not depend on e). Given a string σ we let $\overline{\sigma}$ denote the string that is obtained from σ if every 0 is replaced by a 1 and every 1 is replaced by a 0. Let U be a universal optimal prefix-free machine. We define a machine N as follows: if $U(\sigma) = \tau$ we let $N(0\sigma) = \tau$ and $N(1\sigma) = \overline{\tau}$. Clearly N is prefix-free and for all strings σ, $e \in \mathbb{N}$,

$$\sigma \in T_N(e) \iff \overline{\sigma} \in T_N(e), \quad \text{hence } |2^m \cap T_N(e)| \text{ is even for each } m \in \mathbb{N}. \tag{4.4}$$

Without loss of generality we may choose the 1-1 pairing function $\langle e, n\rangle$ such that $\sum_{e,n} 2^{-K_N(\langle e,n\rangle)} < 2^{-3}$ and $\langle e, n\rangle > e + n$ for all $e, n \in \mathbb{N}$.

The following construction defines a new machine V dynamically. Notice that $V(0\sigma) = N(\sigma)$ for all σ, while $V(1\sigma)$ is defined under certain conditions. A string ρ is said to be *acceptable at stage* $s + 1$ if it is incomparable with all strings σ such that $V(1\sigma)[s]\downarrow$. In other words, if the addition of 1ρ to the domain of V preserves the prefix-freeness of V. The construction will choose acceptable strings ρ_i, $i \in \mathbb{N}$ of certain lengths in the course of stages. The existence of these strings will follow from the Kraft-Chatin theorem, once we show that $\sum_i 2^{-|\rho_i|} < 1$ in the verification. We order the strings first by length and then lexicographically.

Construction of V At stage $s+1$ do the following. If s is even and $N(\sigma)[s] = \tau$ for some strings σ, τ of length $< s$, let $V(0\sigma) = \tau$. If s is odd, for each $e, n < s$ with

- $n \in \emptyset'[s]$ and $2^{\langle e,n\rangle} - T_V(e)[s] \neq \emptyset$
- $|2^{\langle e,n\rangle} \cap T_V(e)[s]|$ is even

pick a string $\tau \notin T_V(e)[s]$ of length $\langle e,n\rangle$ and the leftmost acceptable string ρ of length $K_N(\langle e,n\rangle)[s] + e$ and let $V(1\rho) = \tau$.

Verification First we show that the acceptable strings that are requested in the course of the construction exist. By the Kraft-Chaitin theorem it suffices to show that $\sum_i 2^{-|\ell_i|} < 1$, where (ℓ_i) is the sequence of the lengths of the strings that are requested during the construction.

Let us divide the machine V into two parts. The left part V_ℓ is the restriction of V to the strings that are prefixed by 0. The right part V_r is the restriction of V to the strings that are prefixed by 1. Clearly the domains of V_ℓ, V_r are disjoint and $V_\ell(0\sigma) = N(\sigma)$ for all strings σ. Moreover for each $e, n \in \mathbb{N}$,

$$\begin{aligned} &T_V(e) = T_{V_\ell}(e) \cup T_{V_r}(e) \text{ at each stage} \\ &|2^{\langle e,n\rangle} \cap T_{V_\ell}(e)| \text{ is even at each stage} \end{aligned} \tag{4.5}$$

where the second clause follows by (4.4). Each request for an acceptable string is associated with a pair $\langle e,n\rangle$ and a current value k of $K_N(\langle e,n\rangle)$ at the stage where the request is issued. By (4.5) for each $\langle e,n\rangle$, k at most one request is made and this is for a string of length $k+e$. Since $K_N(\langle e,n\rangle)[s]$ is non-increasing in s, the requests associated with $\langle e,n\rangle$ have weight at most $\sum_i 2^{-K_N(\langle e,n\rangle)-e-i}$ which is at most $2 \cdot 2^{-K_N(\langle e,n\rangle)-e}$. So the requests associated with e have weight at most $2^{1-e}\sum_n 2^{-K_N(\langle e,n\rangle)}$ which is at most 2^{-e-2} by the choice of the pairing function. This shows that the total weight of the requests that occur in the construction is at most $\sum_e 2^{-e-2} < 1$.

Hence the construction is sound. By the choice of N and the definition of acceptable strings, the machine V is prefix-free. Moreover, by the definition of V and the encoding $n \to 0^n$ of numbers into strings we have $K_V(n) = K_N(n) + 1$ for each $n \in \mathbb{N}$. It remains to show there is a constant c such that (4.3) holds for all $e \in \mathbb{N}$ and all $n > c$. By the coding theorem (see e.g. Ref. [14, Theorem 2.2.26]) there exists a constant c such that $|2^{\langle e,n\rangle} \cap T_V(e)| < 2^{c+e}$ for each $e, n \in \mathbb{N}$. By the choice of the pairing function we have $e + n < \langle e,n\rangle$ for all $e, n \in \mathbb{N}$. Hence $|2^{\langle e,n\rangle} \cap T_V(e)| < 2^{\langle e,n\rangle}$ for each $e \in \mathbb{N}$ and each $n > c$. Given any $e \in \mathbb{N}$ and $n > c$ we demonstrate (4.3). If $n \notin \emptyset'$ the part V_r of the machine will not enumerate any descriptions for strings of length $\langle e,n\rangle$. Therefore by (4.4) the number $|2^{\langle e,n\rangle} \cap T_V(e)|$ is even and (4.3) holds for the chosen e, n. Now suppose that $n \in \emptyset'$. Since $|2^{\langle e,n\rangle} \cap T_V(e)| < 2^{\langle e,n\rangle}$ at each stage of the construction there will be a string of length $\langle e,n\rangle$ which is not in $T_V(e)$. Since we also have $K_V(n) = K_N(n) + 1$, the construction will ensure (in the odd stages after n appears in $\emptyset'$) that $|2^{\langle e,n\rangle} \cap T_{V_r}(e)| = 1$. Since $|2^{\langle e,n\rangle} \cap T_{V_\ell}(e)|$ is always even we can conclude that $|2^{\langle e,n\rangle} \cap T_V(e)|$ is odd. Hence (4.3) holds for the given e, n.

Notice that instead of using the Kraft-Chaitin theorem, we could let the domain of V_r be the strings 1ρ such that $N(\rho) = 0^n$ for some $n \in \mathbb{N}$. This shows that the above proof applies invariably to the case of plain complexity.

We do not know if the set of C_e-trivial strings is tt-complete with respect to all underlying optimal universal machines and for all sufficiently large e.

5 Splitting Theorems for $\leq C$ and $\leq K$

The archetypical splitting theorem for the computably enumerable sets in a degree structure is the so-called Sacks splitting theorem (e.g. see Ref. [15, Theorem 3.1]). This asserts that each c.e. set of non-zero Turing degree is the disjoint union of two c.e. sets of incomparable degrees which are strictly lower than the degree of the original set. With the growing interest in weak reducibilities as measures of relative randomness, the local and global study of the corresponding degree structures has become very relevant. In Ref. [5] it was observed that Sacks' original argument can be translated into the context of weak reducibilities. This observation was applied to the so-called LR reducibility, which can be used to compare oracles in terms of the power that they have in 'derandomizing' sequences. A necessary condition for this approach of emulating arguments from the theory of Turing degrees is that the reducibility in question is Σ^0_3.

In this section we use this intuition to show that a splitting theorem holds for the reducibilities of relative randomness based on plain and prefix-free complexity. Although the proofs follow Sacks' original ideas, the translation of the argument is not always trivial. This was already observed in Ref. [12] where the fact that $\oplus$ is not a join operator in the structure of the LR degrees meant that Sacks' argument (as this is presented in Ref. [15, Theorem 3.1]) gave a weaker version of the splitting theorem for this structure. In Ref. [2, Footnote 6] it was observed that a modified argument gave the full analogue of the splitting theorem for the LR degrees. A detailed presentation of this modified argument can be found in Ref. [18, Chapter 2]. In this section we use an analogue of this argument, along with special properties of the reducibilities $\leq_C, \leq_K$, in order to show the splitting theorem in this context. The work in this section is joint with Tom Sterkenburg, and his thesis Ref. [18, Chapter 2] contains a more detailed presentation of it.

In the following we let $A_0|_C A_1$ mean that $A_0 \not\leq_C A_1$ and $A_1 \not\leq_C A_0$. Similar notation applies to other reducibilities. First, we need the following lemma that gives useful information about the nature of $\leq_C$ and $\leq_K$.

Lemma 5.1. Suppose that a set A is the disjoint union of two c.e. sets A_0, A_1. If $A_0|_K A_1$ then $A_0, A_1 <_K A$. Similarly, if $A_0|_C A_1$ then $A_0, A_1 <_C A$.

Proof: Observe that if A is the disjoint union of two c.e. sets A_0, A_1 then A_0, A_1 are identity bounded Turing reducible to A (i.e. Turing reducible via a reduction which uses only the first n bits of A on the argument n). Indeed, to determine if x is in (say) A_0, we can check if $x \in A$. If so, we know it is in one of the disjoint parts A_0 and A_1; and we can computably enumerate both of them until x appears in one. As a consequence, $A_i \upharpoonright_n$ can be described using $A \upharpoonright n$. Hence the initial segment complexity of the former is no more than that of the latter, up to a constant. In other words, $A_i \leq_K A$. Moreover, if A_0 and A_1 are K-incomparable then $A \not\leq_K A_i$ because otherwise we would have $A_{1-i} \leq_K A_i$.

It follows in the same way from $A_0, A_1 \leq_{ibT} A$ (where $\leq_{ibT}$ denotes the identity bounded Turing reducibility) that $A_0, A_1 \leq_C A$. So if $A_0|_C A_1$ then $A \not\leq_C A_0, A_1$ just as in the prefix-free case.
We are now ready to prove the splitting theorem for $\leq_K$. As we indicated above, the construction is not a straightforward adaptation of the argument behind the splitting theorem of Sacks for the Turing degrees. On the other hand, one can use the ideas below to prove Sack's theorem for the Turing degrees without any assumption about $\oplus$ being a least upper bound operator in the Turing degrees.

Theorem 5.2. Let A be a c.e. set such that $A >_K \emptyset$. Then A is the disjoint union of two c.e. sets A_0, A_1 such that $A_0|_K A_1$ and $A_0, A_1 <_K A$.

Proof: In the course of enumerating the elements of A into A_0 and A_1 we satisfy the following requirement for $e \in \mathbb{N}$ and $i = 0, 1$.

$$R_{\langle e,i\rangle} : \exists n \left[K(A_{1-i}\restriction_n) > K(A_i\restriction_n) + e\right].$$

Thus we ensure that $A_0 \not\leq_K A_1$ and $A_1 \not\leq_K A_0$. By Lemma 5.1 we also get $A_0, A_1 <_K A$. Define the *length of agreement* of R_e at stage s by

$$l(e,i)[s] = \text{the greatest } n \leq s \text{ such that } K(A_{1-i}\restriction_n)[s] \leq K(A_i\restriction_n)[s] + e$$

and let the restraint of R_e at stage s be given by

$$r(e,i)[s] = \max_{t \leq s}\{l(e,i)[t], e\}.$$

Notice that by definition the restraint is non-decreasing in the stages s. Let $A_i[0] = \emptyset$ for $i = 0, 1$ and without loss of generality assume that at each stage exactly one element is enumerated in A.

Construction If $x \in A[s+1] - A[s]$ consider the least $\langle e, i\rangle$ such that $x \leq r(e,i)[s]$ and enumerate x into A_{1-i}.

Verification By induction we show that each requirement is satisfied and its restraint reaches a limit. Suppose that there is a stage s_0 such that for all $\langle e', i'\rangle < \langle e, i\rangle$ requirement $R_{\langle e',i'\rangle}$ is met and $r(e', i')[s]$ remains constant for all $s \geq s_0$. Without loss of generality we may assume that s_0 is large enough so that all numbers enumerated in A after s_0 are larger than the final values of $r(e', i')$, $\langle e', i'\rangle < \langle e, i\rangle$.

By the choice of s_0, after that stage all numbers enumerated into A_i will be larger than the current value of $r(e,i)$. If $R_{\langle e,i\rangle}$ was not met, the length of agreement $l(e,i)$ and the restraint $r(e,i)$ tend to infinity. Hence A_i is computable, hence K-trivial. Since $R_{\langle e,i\rangle}$ is not met, it follows that A_{1-i} is K-trivial. Since A is the disjoint union of A_0 and A_1 we have $A \equiv_T A_0 \oplus A_1$. Then A is K-trivial, given that K-triviality is closed under the join operator. This contradicts the assumption about A. Hence $R_{\langle e,i\rangle}$ is met.

To conclude the induction step it suffices to show that $r(e,i)$ reaches a limit. But this is a direct consequence of its definition and the fact that $R_{\langle e,i\rangle}$ is met.

The proof of Theorem 5.2 can be written for $\leq_C$ instead of $\leq_K$ with no essential changes. This trivial modification gives the following.

Theorem 5.3. Let A be a c.e. set such that $A >_C \emptyset$. Then A is the disjoint union of two c.e. sets A_0, A_1 such that $A_0|_C A_1$ and $A_0, A_1 <_C A$.

Finally we note that Theorems 5.2 and 5.3 can be combined with a number of other c.e. splitting theorems, like the Sacks splitting theorem for the Turing degrees. In other words, in the conclusion we can add the conditions $A_0, A_1 <_T A$ and $A_0|_T A_1$. This merely requires a direct combination of the two splitting constructions, hence we do not elaborate.

6 Ideas for Further Research

In this paper we studied the complexity of the collection of strings with trivial Kolmogorov complexity. A basic open question that remains is whether it is truth table complete independently of the underlying optimal universal machine, in the case of plain Kolmogorov complexity. Another question (see Section 3) is whether this set is n-c.e. for some $n \in \mathbb{N}$, and whether the answer depends on the choice of the underlying optimal universal machine.

Another way of studying this collection is from a resource bounded point of view (e.g. time complexity). This is the approach that was taken by Allender, Buhrman and Koucký, for example, in Ref. [1] for the study of the collection of the nonrandom strings. In this spirit we can ask

What can be efficiently reduced to the strings with trivial Kolmogorov complexity?

The issues that we discussed in this paper can also be investigated in the case of another interesting and closely related set of strings, the so-called *strongly random strings*.

Definition 6.1 (Strongly random strings). A string σ is called strongly K_c-random if $K(\sigma) > |\sigma| + K(|\sigma|) - c$.

These strings were studied by Solovay[16] and Miller[12]. They are also called 'strongly K-random with constant c'. Notice that these are highly random strings, given that there is a constant d such that $K(\tau) < |\tau| + K(|\tau|) + d$ for all strings τ. Clearly the complement of the set of strongly K_c-random strings is a variation of the set of nonrandom strings. However Miller[12] showed that, in contrast with the set of nonrandom strings, it is not computably enumerable (for sufficiently large c). The methodology presented in this paper may be applicable to the study of the complexity of this collection.

References

[1] Allender E, Buhrman H, Koucký M. What can be efficiently reduced to the Kolmogorov-random strings? Ann. Pure Appl. Logic, 2006, 138(1-3): 2–19.

[2] Barmpalias G. Elementary differences between the degrees of unsolvability and the degrees of compressibility. Ann. Pure Appl. Logic, 2010, 161(7): 923–934.

[3] Beigel R, Buhrman H, Fejer PA, Fortnow L, Grabowski P, Longpré L, Muchnik A, Stephan F, Torenvliet L. Enumerations of the kolmogorov function. Journal of Symbolic Logic, 2006, 71(2): 501–528.

[4] Barmpalias G, Lewis AEM, Soskova M. Randomness, lowness and degrees. Journal of Symbolic Logic, 2008, 73(2): 559–577.

[5] Barmpalias G, Vlek C. Kolmogorov complexity of initial segments of sequences and arithmetical definability. Theoret. Comput. Sci., 2011. In press.

[6] Calude CS, Nies A. Chaitin Ω numbers and strong reducibilities. J. UCS, 1997, 3(11): 1162–1166 (electronic).

[7] Downey RG,Hirschfeldt DR, LaForte G. Randomness and reducibility. J. Comput. System Sci., 2004, 68(1): 96–114.
[8] Figueira S, Stephan F, Wu GH. Randomness and universal machines. J. Complexity, 2006, 22(6): 738–751.
[9] Kolmogorov AN. Three approaches to the definition of the concept "quantity of information". Problemy Peredavci Informacii, 1965, 1(vyp.1): 3–11.
[10] Kummer M. On the complexity of random strings (extended abstract). STACS 96 (Grenoble, 1996), volume 1046 of Lecture Notes in Computer Science. Springer, Berlin. 1996. 25–36.
[11] Li M, Vitányi P. An introduction to Kolmogorov complexity and its applications. Graduate Texts in Computer Science. Springer-Verlag, New York, second edition, 1997.
[12] Joseph S. Miller. Contrasting plain and prefix-free Kolmogorov complexity. Preprint.
[13] Muchnik AA, Ye S. Positselsky. Kolmogorov entropy in the context of computability theory. Theoret. Comput. Sci., 2002, 271(1-2): 15–35.
[14] Nies A. Computability and Randomness. Oxford University Press, 2009.
[15] Soare RI. Recursively enumerable sets and degrees. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. A study of computable functions and computably generated sets.
[16] Solovay R. Handwritten manuscript related to Chaitin's work. IBM Thomas J. Watson Research Center, Yorktown Heights, NY, 215, 1975.
[17] Stephan F. The complexity of the set of nonrandom numbers. Randomness and complexit, World Sci. Publ., Hackensack, NJ, 2007. 217–230.
[18] Sterkenburg T. Sequences with trivial initial segment complexity [MS Dissertation]. University of Amsterdam, February 2011.