Reconciling data compression and Kolmogorov complexity

Laurent Bienvenu[1] and Wolfgang Merkle[2]

[1] Laboratoire d'Informatique Fondamentale, Université de Provence, Marseille, France, laurent.bienvenu@lif.univ-mrs.fr
[2] Institut für Informatik, Ruprecht-Karls-Universität Heidelberg, Germany, merkle@math.uni-heidelberg.de

Abstract. While data compression and Kolmogorov complexity are both about effective coding of words, the two settings differ in the following respect. A compression algorithm or compressor, for short, has to map a word to a unique code for this word in one shot, whereas with the standard notions of Kolmogorov complexity a word has many different codes and the minimum code for a given word cannot be found effectively. This gap is bridged by introducing decidable Turing machines and a corresponding notion of Kolmogorov complexity, where compressors and suitably normalized decidable machines are essentially the same concept.
Kolmogorov complexity defined via decidable machines yields characterizations in terms of the intial segment complexity of sequences of the concepts of Martin-Löf randomness, Schnorr randomness, Kurtz randomness, and computable dimension. These results can also be reformulated in terms of time-bounded Kolmogorov complexity. Other applications of decidable machines are presented, such as a simplified proof of the Miller-Yu theorem (characterizing Martin-Löf randomness by the plain complexity of the initial segments) and a new characterization of computably traceable sequences via a natural lowness notion for decidable machines.

1 Introduction

The Kolmogorov complexity of a word w with respect to a Turing machine M is defined to be the length of the shortest input on which M halts and outputs w. This shortest input is often thought as the "best compression" of w relatively to M. However, as pointed out by Shen [14]: *"[...] in the framework of Kolmogorov complexity we have no compression algorithm and deal only with the decompression algorithm."* In this paper, we address Shen's remark and have a look at effective randomness from an actual compression viewpoint.

First, in Section 2, we introduce a formal notion of compressor. Among the many possible formalizations of the notion of compressor, we study one that is natural and captures the main properties of "real world" compressors. We then argue that compression obtained by compressors in our sense is essentially the

same as dealing with Kolmogorov complexity defined via *decidable machines*, i.e., by Turing machines that have computable domains.

In the following sections, the theory of decidable machines is developed in a framework of algorithmic randomness. In Section 3, we review the celebrated characterization of Martin-Löf randomness in terms of prefix-free Kolmogorov complexity due to Schnorr and the more recent characterizations of Schnorr and Kurtz randomness in terms of bounded machines due to Downey and Griffiths [2]. We give identical or very similar charaterizations of all three notions of randomness in terms of decidable machines; to the best of our knowledge, this is the first time that all three notions are characterized using a single type of Turing machine. Similary, we argue that the characterization of computable Hausdorff dimension in terms of computable machines due to Downey et al. [1] extends to decidable machines. In Section 4, all the mentioned charaterizations are transferred to standard time-bounded Kolmogorov complexity by arguing that the latter is closely related to Kolmogorov complexity defined via decidable machines.

In Section 5, we use the characterization of Martin-Löf randomness in terms of decidable machines in order to give a simplified proof of the recent Miller-Yu theorem, which characterizes Martin-Löf randomness in terms of plain Kolmogorov complexity of initial segments.

Finally, in Section 6, we consider lowness notions. A sequence A is called low and order-low for prefix-free decidable machines in case Kolmogorov complexity with respect to such machines, when relativized to oracle A, changes by less than a constant and by less than any computable order, respectively. We obtain that any sequence is order-low for decidable machines if the sequence is computably traceable, which then implies some known facts on lowness for Schnorr and Kurtz randomness. Furthermore, exactly the computable sequences are low for prefix-free decidable machines. In what follows several proofs will be omitted due to lack of space.

We conclude the introduction by reviewing some standard concepts and notation that will be used in the sequel. A WORD is a finite binary sequence, the empty word, the unique word of length 0, is denoted by λ; the set of all words is denoted by 2^*. Unless explicitely stated differently, a SEQUENCE is an infinite binary sequence, and we write 2^ω for the set of all sequences (sometimes also referred to as Cantor space).We denote by $\sqsubseteq$ the prefix relation on $2^* \cup 2^\omega$. For every word u and for every set of words A, let

$$[u] = \{R \in 2^\omega \colon u \sqsubseteq R\} \qquad \text{and} \qquad [A] = \textstyle\bigcup_{u \in A} [u] \,.$$

Recall that Lebesgue measure on Cantor space is the unique probability measure μ on Cantor space such that for all words u holds $\mu([u]) = 2^{-|u|}$.

Following Li and Vitanyi [9], we denote by $\mathrm{C}(u)$ and $\mathrm{K}(u)$ the plain and the prefix Kolmogorov complexity of a word u. For every Turing machine M which computes a partial function from words to words and for any word u, we denote by $\mathrm{C}_M(u)$ the natural number $\inf\{|p| \colon M(p) = u\}$. If M is assumed to have prefix-free domain, we write K_M in place of C_M.

2 Compressors and decidable machines

The intuitive understanding of a compressor is a procedure that maps a word to a code for that word, where the mapping is one-to-one and hence in principle invertible. For compressors that are to be applied in practice, in addition one will surely require that coding and decoding are efficient and that redundant sources will be mapped to reasonably short codes; however, these latter requirements will not be considered here. We consider a most general notion of compressor where one simply requires that the compressor yields an effective bijection between a computable set of words and a computable set of codes.

Definition 1. *A* COMPRESSOR *is a partial computable function* $\Gamma\colon 2^* \to 2^*$ *such that* Γ *is one-to-one and the domain and range of* Γ *are computable. A compressor is* PREFIX-FREE *if its range is prefix-free.*

By Definition 1, a compressor may be undefined on some strings, but such strings can be recognized effectively. Furthermore, the definition ensures that decompression can be performed effectively, i.e., given a string $u = \Gamma(v)$ in the range of Γ, the unique preimage v of u can be found effectively. Here again, some strings can be invalid for the decompression process, i.e., they do not represent a compressed string, but these strings can also be recognized effectively. Most actual compressors (e.g., gzip) are indeed compressors in our sense.

Compressors as just defined and Kolmogorov complexity both are about effective coding of words. The main difference is that a compressor has to produce a code for a word in one shot and in fact every word has at most a single code. For the standard notions of Kolmogorov complexity, on the other hand, a word has several codes. While decoding is effective and the set of pairs of codes and coded words is computably enumerable, in general there is no effective way to go from a word to its shortest code. So if we want to have a notion of Kolmogorov complexity that corresponds to the type of coding done by compressors, we have to define Kolmogorov complexity with respect to a restricted class of Turing machines.

Definition 2. *A Turing machine that computes a partial function on the set of words is called* DECIDABLE *if its domain is decidable.*

Decidable Turing machines can be normalized in the sense that superfluous codes are removed and possessing a code becomes a computable property.

Definition 3. *A decidable Turing machine* M *is called* NORMALIZED *if the range of* M *is decidable and for all words* w *there is at most one word* p *where* $M(p)=w$.

As usual, a Turing machine is called PREFIX-FREE if its domain is prefix-free.

Proposition 4. *Given a decidable Turing machine* M *one can effectively find a decidable Turing machine* M' *that is normalized such that for all words* w *holds* $\mathrm{K}_{M'}(w) \leq \mathrm{K}_M(w) + 1$. *In addition, if* M *is prefix-free, then* M' *can be chosen to be prefix-free, too.*

Proof. For a given decidable Turing machine M, define M' as follows. For any word w, in case there is some word p where $M(p) = w$ and $|p| \leq 2|w|$, let p_w be the least such word and let $M'(0p_w) = w$. If there is no such word p, let $M'(1^{|w|}0w) = w$. □

Proposition 5 shows that compressors and normalized decidable Turing machines are essentially the same. Recall that the inverse of a partial function f from words to words that is one-to-one is the partial function $f^{-1}\colon v \mapsto \min\{u\colon f(u) = v\}$.

Proposition 5. *For any compressor Γ, the inverse mapping Γ^{-1} is just the partial function computed by some normalized decidable Turing machine. For any normalized decidable Turing machine M the inverse mapping M^{-1} is a compressor. The same correspondence is given for the subclasses of prefix-free compressors and prefix-free decidable machines.*

Proof. It suffices to observe that both compressors and normalized decidable Turing machines are by definition just partial computable bijections between a decidable domain and a decidable range. □

In the sequel, we will derive several results on decidable Turing machines, however, by the close correspondence with compressors, all these results could be reformulated in terms of compressors. For further use, recall the notion of computable Turing machine, as introduced by Downey and Griffiths.

Definition 6 (Downey and Griffiths). *A Turing machine that computes a partial function on the set of words is called* COMPUTABLE *if its domain D is prefix-free and $[D]$ has computable Lebesgue measure.*

Observe that any computable Turing machine is in particular decidable; the converse is false even when attention is restricted to prefix-free Turing machines. Downey and Griffiths [2] introduced the notion of computable machine in order to give a machine characterization of Schnorr randomness and Kurtz randomness. In Section 3, we will give alternative characterizations of these randomness notions in terms of decidable machines.

3 Characterizing randomness notions by decidable machines

We first review the notions of a Martin-Löf random, a Schnorr random, and a Kurtz random sequence, and then we characterize these notions in terms of the complexity of initial segments with respect to decidable machines.

Definition 7. *A* MARTIN-LÖF TEST *is a uniformly recursively enumerable sequence $V_0, V_1, \ldots$ of sets of words where $[V_n]$ has Lebesgue measure at most 2^{-n}. A* SCHNORR TEST *is a Martin-Löf test where in addition the Lebesgue measure of $[V_n]$ is exactly 2^{-n}. A* KURTZ TEST *is a Martin-Löf test where in addition the sets $[V_n]$ are finite and a canonical index for V_n can be computed from n.*

A sequence $V_0, V_1, \ldots$ of sets of words COVERS *a sequence X if X is contained in the intersection of the open sets $[V_n]$ (i.e., if X has a prefix in every set V_i). A sequence R is* MARTIN-LÖF RANDOM *if it cannot be covered by a Martin-Löf test. A sequence R is* SCHNORR RANDOM *if it cannot be covered by a Schnorr test. A sequence R is* KURTZ RANDOM *if it cannot be covered by a Kurtz test.*

In the above definition of a Schnorr test, the measure condition on the uniformly recursively enumerable family $V_0, V_1, \ldots$ actually implies that this family is uniformly computable. This is no longer true in the case of Martin-Löf tests: some are not uniformly computable. However, it is well-known that every Martin-Löf test can be turned into an equivalent Martin-Löf test that is uniformly computable.

Lemma 8. *For every Martin-Löf test $V_0, V_1, \ldots$ there exists a Martin-Löf test $U_0, U_1, \ldots$ such that $[U_n] = [V_n]$ for all n and where the set $\{(u, n) \colon u \in V_n\}$ is decidable. (We will call a Martin-Löf test with the latter property a* DECIDABLE *Martin-Löf test.)*

One of the most celebrated results in algorithmic randomness is Schnorr's characterization of Martin-Löf randomness in terms of prefix-free Kolmogorov complexity. Propositions 10 and 11 assert that very similar characterizations are true in a setting of prefix-free decidable machines.

Theorem 9 (Schnorr). *A sequence R is Martin-Löf random if and only if $\mathrm{K}(R[0..n]) \geq n - \mathrm{O}(1)$.*

Proposition 10. *A sequence R is Martin-Löf random if and only if for all prefix-free decidable machines M, $\mathrm{K}_M(R[0..n]) \geq n - \mathrm{O}(1)$.*

Proposition 11. *There exists a prefix-free decidable machine M such that any sequence R is Martin-Löf random if and only if $\mathrm{K}_M(R[0..n]) \geq n - \mathrm{O}(1)$.*

The implications from left to right in Propositions 10 and 11 are immediate from Schnorr's theorem because for any prefix-free machine M there is a constant d such that for all words u holds $\mathrm{K}(u) \leq \mathrm{K}_M(u)+d$. The proof of the other direction is similar to the proof of the corresponding implication in Schnorr's theorem, i.e., one applies the Kraft-Chaitin Theorem in order to go from a Martin-Löf test that covers a set X to a prefix-free machine that has short codes for the prefixes of X; in order to actually obtain prefix-free decidable machines, it suffices to start with a decidable Martin-Löf test according to Lemma 8, where in the case of Proposition 11 this test is chosen to be universal.

Downey and Griffiths [2] characterized Schnorr randomness by the complexity of initial segements with respect to computable machines; Proposition 13 states a related characterization in terms of decidable Turing machines. In connection with the latter, recall that a function $h \colon \mathbb{N} \to \mathbb{N}$ is called an ORDER if h is nondecreasing and unbounded.

Proposition 12 (Downey and Griffiths). *A sequence R is Schnorr random iff for every computable Turing machine M holds $\mathrm{K}_M(R[0..n]) \geq n + \mathrm{O}(1)$.*

Proposition 13. *A sequence R is Schnorr random iff for every computable order h and for every prefix-free decidable Turing machine M,*

$$\mathrm{K}_M(R[0..n]) \geq n - h(n) + \mathrm{O}(1) \; .$$

Downey and Griffiths [2] also used computable machines to give a machine characterization of Kurtz randomness. Proposition 15 asserts a completely similar characterization in terms of decidable machines, where interestingly the prefix-free property makes no difference. Furthermore, the last condition in Proposition 15 gives a characterization of Kurtz randomness similar to the characterization of Schnorr randomness in Proposition 13.

Proposition 14 (Downey and Griffiths).
The following assertions are equivalent.

(i) *R is not Kurtz random.*
(ii) *There exists a computable machine M and a computable function $f\colon \mathbb{N} \to \mathbb{N}$ such that $\mathrm{K}_M(R[0..f(d)]) \leq f(d) - d$ holds for all d.*

Proposition 15. *The following assertions are equivalent.*

(i) *R is not Kurtz random.*
(ii) *There exists a prefix-free decidable machine M and a computable function $f\colon \mathbb{N} \to \mathbb{N}$ such that $\mathrm{K}_M(R[0..f(d)]) \leq f(d) - d$ holds for all d.*
(iii) *There exists a decidable machine M and a computable order h such that $\mathrm{C}_M(R[0..n]) \leq n - h(n)$ holds for all n.*
(iv) *There exists a decidable machine M and a computable function $f\colon \mathbb{N} \to \mathbb{N}$ such that $\mathrm{C}_M(R[0..f(d)]) \leq f(d) - d$ holds for all d.*

A characterization of computable dimension in terms of computable machines has been obtained by Downey et al. [1], and we conclude this section by a similar characterization in terms of decidable machines. In the context of time-bounded complexity, an equivalent formulation of the latter characterization has been previously demonstrated by Hitchcock [5], see Proposition 20.

Proposition 16. *For every sequence R holds*

$$\dim_{\mathrm{comp}}(R) = \inf_M \liminf_{n\to+\infty} \frac{\mathrm{C}_M(R[0..n])}{n},$$

where the infimum is over all decidable Turing machines M.

4 Time-bounded Kolmogorov complexity

Since a decidable Turing machine is required to have a computable domain, it is not hard to show that a decidable Turing machine is the same as a Turing machine that obeys a time bound t but is not required to be defined on all inputs, i.e., on any input p the machine runs for at most $t(|p|)$ steps and then either

produces an output or alternatively may decide not to terminate. In contrast to this, time-bounded Kolmogorov complexity is usually defined in terms of time-bounded machines where the time-bound is required with respect to the length of the output, that is the machine may again be undefined on certain inputs, but whenever the machine outputs a word w then the corresponding computation runs for at most $t(|w|)$ many steps.

Definition 17. *Fix an additively optimal machine U. For any computable function $t : \mathbb{N} \to \mathbb{N}$ and any word w, let*

$$\mathrm{C}^t(w) = \min\{|p| \colon U(p) \text{ outputs } w \text{ after at most } t(|w|) \text{ steps of computation}\}.$$

$K^t(w)$ is defined similarly, taking U optimal among the prefix-free machines.

Lemma 18. *For every decidable machine M, there exists a computable time bound t such that $\mathrm{C}^t \leq \mathrm{C}_M + \mathrm{O}(1)$. For every prefix-free decidable machine M, there exists a computable time bound t such that $\mathrm{K}^t \leq \mathrm{K}_M + \mathrm{O}(1)$.*

The converse of Lemma 18 is not true, but the following weaker statement will be sufficient for our purposes.

Lemma 19. *For every computable time bound t and every computable order h, there exists a decidable machine M such that for every word w and all $k \in \mathbb{N}$,*

$$C^t(w) \leq |w| - k \quad \Longrightarrow \quad C_M(w) \leq |w| - k + h(k) \; .$$

A similar statement holds for K^t and prefix-free decidable machines.

By Lemma 18 and Lemma 19, all our previous results can be interpreted in terms of time-bounded Kolmogorov complexity.

Proposition 20. *(a) A sequence R is Martin-Löf random iff for every computable time bound t, $K^t(R[0..n]) \geq n + \mathrm{O}(1)$.*
(b) There exists a computable time bound t_0 such that every sequence R is Martin-Löf random iff $K^{t_0}(R[0..n]) \geq n + \mathrm{O}(1)$.
(c) A sequence R is Schnorr random iff for every computable time bound t and every computable order g, $K^t(R[0..n]) \geq n - g(n) + \mathrm{O}(1)$.
(d) A sequence R is Kurtz random iff for every computable time bound t and every computable order g, $K^t(R[0..n]) \geq n - g(n)$ for infinitely many n (and this equivalence remains true with C^t in place of K^t).
(e) For every sequence R, $dim_S(R) = \inf \liminf \frac{K^t(R[0..n])}{n}$, the infimum being taken over all computable time bounds t (and this equation remains true with C^t in place of K^t).

Assertion (e) was proved earlier by Hitchcock [5]. Assertion (c) is an improvement of a result of Lathrop and Lutz [7], who demonstrated that the right-hand side condition is necessary for R to be computably random.

5 The Miller-Yu Theorem

After Schnorr [13] characterized Martin-Löf randomness in terms of the prefix Kolmogorov complexity of initial segments, the question whether there is a similar characterization in terms of plain complexity remained open for more than three decades until recently Miller and Yu [11] gave a positive answer. A simplified proof of their result is obtained by using the characterization of Martin-Löf randomness via prefix-free decidable machines from Proposition 11.

Proposition 21 (Miller and Yu). *There is a computable function $G\colon \mathbb{N} \to \mathbb{N}$ such that the sum $\sum_{n\in\mathbb{N}} 2^{-G(n)}$ converges and such that for any sequence R the following assertions are equivalent.*

(i) R is Martin-Löf random.
(ii) For every computable function $g\colon \mathbb{N} \to \mathbb{N}$ such that $\sum_{n\in\mathbb{N}} 2^{-g(n)}$ converges it holds that $\mathrm{C}(R[0..n-1]) \geq n - g(n) - \mathrm{O}(1)$.
(iii) It holds that $\mathrm{C}(R[0..n-1]) \geq n - G(n) - \mathrm{O}(1)$.

Proof. For completeness, we review the standard proof of the implication (i)→(ii). If (ii) is false, then there is a computable function g where $\sum_{n\in\mathbb{N}} 2^{-g(n)}$ converges and such that for arbitrarily large d there is some n where

$$\mathrm{C}(R[0..n-1]) \leq n - g(n) - d\,. \tag{1}$$

But for any such d and n, inequality (1) remains true with $g(n)$ replaced by $\mathrm{K}(n)$ because $\mathrm{K}(n) \leq g(n) + \mathrm{O}(1)$ holds by the Kraft-Chaitin Theorem [3] and assumption on g. Hence for any such d and n, the prefix w of R of length n has a prefix-free code of length at most $n - d/2 + \mathrm{O}(1)$, which consists of a prefix-free code for n of length $\mathrm{K}(n)$, followed by a prefix-free code for $n - \mathrm{K}(n) - \mathrm{C}(w)$ plus a plain code for w of length $\mathrm{C}(w)$. Consequently, R is not Martin-Löf random.

We now construct a computable function G with the required convergence property, where the implication (ii)→(iii) is then immediate, and we conclude by giving an alternative proof of (iii)→(i). Let M be the prefix-free decidable machine of Proposition 11. For all $n, c \in \mathbb{N}$, let $A^c_n = \{u\colon |u| = n \text{ and } K_M(u) \leq |u| - c\}$ and $a^c_n = \mathrm{Card}(A^c_n)$. Furthermore, let $b^c_n = 2^c a^c_n$ and $b_n = \sum_{c\in\mathbb{N}} b^c_n$; observe that the sums of the latter type are actually finite because A^c_n is empty for $c > n$. This way we have

$$\sum_{n\in\mathbb{N}} b_n \frac{1}{2^n} = \sum_{n,c\in\mathbb{N}} b^c_n \frac{1}{2^n} = \sum_{n,c\in\mathbb{N}} a^c_n \frac{1}{2^{n-c}} \leq 1\,, \tag{2}$$

where the equalities hold by definition and the inequality holds because M is prefix-free. Now, if we let $G(n) = n - \log(b_1 + ... + b_n)$, then G is computable and by definition of G and elementary rearrangements of terms one obtains

$$\sum_{n\in\mathbb{N}} 2^{-G(n)} \leq \sum_{n\in\mathbb{N}} \frac{b_1 + \cdots + b_n}{2^n} \leq 2 \sum_{n\in\mathbb{N}} \frac{b_n}{2^n} \leq 2\,.$$

Next consider any word w in A^c_n. Since the A^c_n's are uniformly computable, the word w can be obtained effectively from a description that contains c together with the index of w in the enumeration of the union of the sets $A^c_0, A^c_1, \ldots$ where the elements of A^c_n are enumerated before those of A^c_{n+1}. Hence it holds that

$$\begin{aligned} C(w) &\leq 2\log c + \log(a^c_1 + a^c_2 ... + a^c_n) + \mathrm{O}(1) \\ &\leq 2\log c + \log(2^{-c}b_1 + 2^{-c}b_2 + ... + 2^{-c}b_n) + \mathrm{O}(1) \\ &\leq n - G(n) - c + 2\log c + \mathrm{O}(1) \leq n - G(n) - c/2 + \mathrm{O}(1)\ , \end{aligned} \tag{3}$$

where the second and third inequality hold by definition of the b_n's and of G, respectively. Now if the sequence R is not Martin-Löf random, then by Proposition 11 and definition of the A^c_n's, there are arbitrarily large c such that for some n the prefix w of R of length n is in A^c_n and thus w and c satisfy the chain of inequalities (3), hence (iii) is false. □

6 Lowness and order-lowness

In the area of algorithmic randomness, various lowness notions have been studied and have been shown to interact interestingly with each other and with other notions [3, 12]. In general, a sequence A is called low for a certain concept, if the concept does not change (or at least does not change significantly) when the concept is relativized to the sequence A. For example, a sequence A is low for K if the the standard prefix-free Kolmogorov complexity K and it relativized version with oracle A differ at most by some additive constant. In connection with complexity notions that are not defined via a universal machine, such as Kolmogorov complexity defined in terms of prefix-free decidable machines, lowness usually means that for any machine of the given type with oracle there is another machine without oracle such that complexity with respect to the latter is not significantly larger than complexity with respect to the former. Note in this connection that a prefix-free decidable machine with oracle A is an oracle Turing machine with oracle A that on oracle A has prefix-free domain that can be computed with oracle A; we write K^A_M for the corresponding relativized notion of Kolmogorov complexity.

Definition 22. *A sequence A is* LOW FOR PREFIX-FREE DECIDABLE MACHINES *if for every prefix-free decidable machine M with oracle A, there exists a prefix-free decidable machine M' such that for all words w,*

$$\mathrm{K}_{M'}(w) \leq \mathrm{K}^A_M(w) + \mathrm{O}(1).$$

A sequence A is ORDER-LOW FOR PREFIX-FREE DECIDABLE MACHINES *if for every prefix-free decidable machine M with oracle A and any computable order h, there exists a prefix-free decidable machine M' such that for all words w,*

$$\mathrm{K}_{M'}(w) \leq \mathrm{K}^A_M(w) + h(\mathrm{K}^A_M(w)) + \mathrm{O}(1). \tag{4}$$

The notion of order-low is similar to a lowness notion for standard prefix-free Kolmogorov complexity that has been introduced and has been shown to be equivalent to strong jump-traceability by Figueira, Nies, and Stephan [4]. Somewhat similar to their equivalence result, we obtain a characterization of computable traceability in terms of order-lowness for prefix-free decidable machines. Recall the concept of computable traceability.

Definition 23. *A sequence A is computably traceable if there is a computable order h such that for any function f that is computable with oracle A there is a computable sequence of canonical indices for finite sets $F_0, F_1, \ldots$ where for all i,*

$$(i) \quad f(i) \in F_i, \qquad \text{and} \qquad (ii) \quad |F_i| \leq h(i).$$

Proposition 24. *A sequence A is computably traceable if and only if A is order-low for prefix-free decidable machines.*

Terwijn and Zambella [16] observed that the notion of computable traceability is robust in the following sense. The notion remains the same if in its definition it is required that for any computable order h and any function f computable in A there is a trace $F_0, F_1, \ldots$ with $f(i) \in F_i$ and $|F_i| \leq h(i)$; that is, if according to Definiton 23 a sequence A is computably traceable with respect to some computable order h, then the sequence is computably traceable with respect to any computable order h. Indeed, the definition holds then even with respect to any order that is computable in A, because by Remark 25, for any computably traceable sequence A and any order g computable in A there is a computable order h where $h(n) \leq g(n)$ for all n.

Recall that by definition sequence A is hyperimmune-free, if for any function f that is computable with oracle A there is a computable function g such that $f(n) \leq g(n)$ for all n. Furthermore, observe that every computably traceable sequence A is hyperimmune-free, where it suffices to let $g(n)$ be the maximum value in F_i where $F_0, F_1, \ldots$ is a computable trace for a given function f that is computable in A.

Remark 25. *For any hyperimmume-free sequence A, thus in particular for any computably traceable sequence A, and for any order h that is computable in A there is a computable order g such that for all n holds $g(n) \leq h(n)$ (in fact, this property characterizes the hyperimmune-free sequences).*

For a proof, h being given, consider the A-computable function f defined by: $f(k) = \max\{n : h(n) \leq k\}$. If A is hyperimmune-free, there is a computable function f' such that for all n, $f(n) \leq f'(n)$, and we can assume that f' is increasing. Set $g(k) = \max\{n : f'(n) \leq k\}$ for all k. Since f' is computable and increasing, g is a computable order. Moreover, for all n, $g(n) \leq h(n)$. Indeed, suppose that for some n, one has $g(n) > h(n)$. Then:

$$n \geq f'(g(n)) > f'(h(n)) \geq f(h(n)) \geq n$$

(the inequalities following from the definition of g, the fact that f' is increasing, the fact that $f' \geq f$ and the defintion of f respectively), a contradiction.

Remark 26. *The concept of order-low for prefix-free decidable machines is not changed if one requires in its definition just that* (4) *is satisfied for some fixed computable order in place of all such orders. For a proof, observe that the equivalence in Proposition 24 extends to the altered concepts by literally the same proof, and conclude using Remark 25.*

Remark 27. *The concept of order-low for prefix-free decidable machines is not changed if one requires in the definition of a sequence A being order-low that* (4) *is not just satisfied for all computable orders but for all orders that are computable with oracle A. For a proof it suffices to observe that if a sequence A is order-low, then A is computably traceable by Proposition 24, hence by Remark 25 for any A-computable order there is a computable order that grows at least as slowly.*

Proposition 24 and the assertions of Remarks 26 and 27 remain true when considering dedicable machines (and corresponding notions of order lowness) in place of prefix-free decidable machines.

Terwijn and Zambella [16] and Kjos-Hanssen, Nies, and Stephan [8] demonstrated that the class of computably traceable sequences coincides with the class of sequences that are low for Schnorr null tests and with the class of sequences that are low for Schnorr randomness, respectively. Furthermore, it is known that the computably traceable sequences form a strict subclass of the class of sequences that are low for Kurtz randomness [15]. We obtain part of these results as a corollary to Propositions 13, 15, and 24.

Corollary 28. *Any computably traceable sequence A is low for Schnorr randomness and low for Kurtz randomness.*

Proof. Let A be a computably traceable sequence, and R a Schnorr random sequence. We shall prove that R is A-Schnorr random. Let M be a prefix-free decidable machine with oracle A, and h be an A-computable order. Up to normalizing M as we did in the proof of Proposition 4, let us suppose that $K^A_M(w) \leq 2|w|$ for all words w. By Remark 25, there exists a computable order g such that $g \leq h/2$. Set $g'(n) = g(n/2)$ for all n, and notice that g' is a computable order. By Proposition 24, let M' be a prefix-free decidable machine such that $K_{M'} \leq K^A_M + g'(K^A_M) + O(1)$. Since R is Schnorr random, by Proposition 13, $K_{M'}(R[0..n]) \geq n - g(n) - O(1)$ for all n. Hence

$$\begin{aligned} K^A_M(R[0..n]) &\geq n - g(n) - g'(K^A_M(R[0..n])) - O(1) \\ &\geq n - g(n) - g'(2n) - O(1) \\ &\geq n - 2g(n) - O(1) \\ &\geq n - h(n) - O(1) \end{aligned}$$

By Proposition 13 (relativized to A), R is A-Schnorr random. The proof for Kurtz randomness is similar. □

Finally, the following proposition shows that, in contrast to order-lowness, lowness for decidable machines is a trivial lowness notion.

Proposition 29. *A sequence is low for prefix-free decidable machines if and only if the sequence is computable.*

Proof. Let A be a sequence that is low for prefix-free decidable machines. By Proposition 10, A is low for Martin-Löf randomness, hence Δ^0_2 (see [12]). Moreover, since A is low for prefix-free decidable machines, it is in particular order-low, which by Proposition 24 implies that A is computably traceable. As a Δ^0_2 computably traceable sequence is necessarily computable, we are done. □

References

1. R. Downey, W. Merkle and J. Reimann. *Schnorr dimension.* Mathematical Structures in Computer Science 16:789–811, 2006.
2. R. Downey and E. Griffiths. *On Schnorr randomness.* Journal of Symbolic Logic 69(2):533–554, 2004.
3. R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity.* Manuscript, 2007.
4. S. Figueira, A. Nies, and F. Stephan. *Lowness properties and approximations of the jump.* In WOLLIC 2005, Electronic Notes in Theoretical Computer Science 143:45–57, Elsevier 2006.
5. J. M. Hitchcock. PhD dissertation, Iowa State University, 2003.
6. J. M. Hitchcock and J. H. Lutz. *Why computational complexity requires stricter martingales.* Theory of computing systems 39(2):277–296, 2006.
7. J. Lathrop and J. Lutz. *Recursive computional depth.* Information and Computation 153:137-172, 1999.
8. B. Kjos-Hanssen, A. Nies, and F. Stephan. *Lowness for the class of Schnorr random reals.* SIAM Journal on Computing 35(3): 647–657, 2005.
9. M. Li and P. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications*, second edition. Springer, 1997.
10. P. Martin-Löf. *The definition of random sequences.* Information and Control 9(6):602–619, 1966.
11. J. S. Miller and L. Yu. *On initial segment complexity and degrees of randomness.* Transactions of the American Mathematical Society. To appear.
12. A. Nies. *Lowness properties and randomness.* Advances in Mathematics 197:274–305, 2005.
13. C.-P. Schnorr. *A unified approach to the definition of random sequences.* Mathematical Systems Theory 5:246–258, 1971.
14. A. Shen. *Algorithmic Information Theory and Kolmogorov Complexity.* Lecture notes of an introductory course. Uppsala University Technical Report 2000-034. Available online at `http://www.it.uu.se/publications/reports/2000-034`.
15. F. Stephan and Liang Yu. *Lowness for weakly 1-generic and Kurtz-random.* TAMCS 2006, LNCS 3959:756–764, Springer, 2006.
16. S. Terwijn and D. Zambella. *Computable randomness and lowness.* Journal of Symbolic Logic 66(3):1199–1205, 2001.

Appendix

Proof of Lemma 8

Proof. Let n be fixed. We construct V_n as follows. We enumerate U_n, and, when a word u is enumerated in U_n after s steps of enumeration, we add in V_n all the

extensions of u of length $max(|u|, s)$. Of course this can be done uniformly in n, and hence $\{V_n\}$ is a Martin-Löf test. Moreover, to decide whether or not a word w belongs to V_n, it is sufficient to compute the $|v|$ first elements $u_1, ..., u_{|v|}$ in the enumeration of U_n, and to notice that v is in V_n if and only if v an extension of u_s of length $max(|u_s|, s)$ for some $s \leq |v|$. □

Proof of Proposition 10

Proof. Since for all prefix-free machines M we have $K \leq K_M + O(1)$, we only need to prove that if R if not Martin-Löf random, then there exists a decidable prefix-free machine M such that $\inf K_M(R[0..n]) - n = -\infty$. Let R be non Martin-Löf random, and let $\{V_n\}$ be a decidable Martin-Löf test covering it (there exists such a test by the previous lemma). From this test, we contruct the machine M. By a classical Kraft-Chaitin argument, there exists a computable function $\pi : 2^* \times \mathbb{N} \to 2^*$ which range is prefix free and such that for all (u, n) with $u \in V_n$, $|\pi(u, n)| = |u| - \lfloor n/2 \rfloor + c$ for some constant $c \in \mathbb{N}$. We then define M by $M(\pi(u, n)) = u$. We show that M is decidable. Suppose $w \in dom(M)$, then $w = \pi(u, n)$ for some (u, n) s.t. $u \in V_n$. In this case $|w| = |u| - \lfloor n/2 \rfloor + c$, and $|u| \geq n$. Hence, $n \leq 2|w|$. Thus, to check if $w \in dom(M)$ it is sufficient to compute $\pi(u, n)$ for all (u, n) such that $u \in V_n$, $n \leq 2|w|$ and $|w| = |u| - \lfloor n/2 \rfloor + c$. Since $\{V_n\}$ is decidable and since there are finitely many such (u, n), this shows that M is decidable. Now, for any n and any u such that $u \in V_n$, we have by definition of M: $K_M(u) = |u| - \lfloor n/2 \rfloor + c$, and hence $\inf K_M(R[0..n]) - n = -\infty$. □

Proof of Proposition 13

Proof. First assume that R is not Schnorr random, that is, there is a normed computable martingale d and a computable order g such that $d(R[0..n]) \geq g(n)$ for infinitely many n. Let g' be a computable order such that $g'(n) = \mathrm{o}(g(n))$, for example, $g' = \log g$ (log being the floor of the logarithm of base 2). For all $k \in \mathbb{N}$, let A_k be the set of all words that are minimal among all words u such that $d(u) \geq 2^{k+1} g'(|u|)$. By construction, $R \in \bigcap_k [A_k]$. Moreover, by Theorem **??**, we have

$$2^{k+1} \sum_{u \in A_k} 2^{-|u| + \log(g'(|u|))} = \sum_{u \in A_k} 2^{-|u| + k + 1 + \log(g'(|u|))} \leq \sum_{u \in A_k} 2^{-|u|} d(u) \leq 1$$

and hence the sum of $2^{-|u| + \log(g'(|u|))}$ over all k and all u in A_k is at most 1. By a standard Kraft-Chaitin argument, there exists some partial computable function $\pi \colon \mathbb{N} \times 2^* \to 2^*$ whose domain is $\{(k, u) \in \mathbb{N} \times 2^* \colon u \in A_k\}$, whose range is prefix free and such that for all $(k, u) \in dom(\pi)$, $|\pi(k, u)| = |u| - \log(g'(|u|))$. Let us show that the range of π is computable. By definition of A_k, for all $u \in A_k$, $|u| - \log(g'(|u|)) \geq k$. Thus, in order to know if a word w lies in range(π), it suffices to compute $\pi(k, u)$ for all $k \leq |w|$ and all u such that $|u| - log(g'(|u|)) = |w|$ (there are only finitely many such couples (k, u) since $\log(g'(|u|)) = o(|u|)$, and

they can be effectively found since g' is computable) and check whether or not w belongs to the computed values.

Let now M be the machine of domain range(π) such that for all $(k,u) \in dom(\pi)$, $M(\pi(k,u)) = u$ (M is a computable function for the same reason as above). Thus, for all $u \in A_k$, $\mathrm{C}_M(u) \leq |u| - \log(g'(|u|))$, and since arbitrarily long prefixes of R belong to $\bigcup_k A_k$, it does not hold that $\mathrm{K}_M(R[0..n]) \geq n - \log(g'(n)) + \mathrm{O}(1)$ ($\log(g'(n))$ being an order).

Suppose conversely that there exists a computable order g and a prefix-free, decidable machine M such that for all k, $\mathrm{K}_M(R[0..n]) \leq n - g(n) - k$ for infinitely many n. We can assume that $g(n) = \mathrm{o}(n)$. For all $k \in \mathbb{N}$, let A_k be the set of all words that are minimal among all words u such that $\mathrm{K}_M(u) \leq |u| - g(|u|) - k$. Obviously, we have $K_M \geq K$ (up to an additive constant). Thus, $\{A_k\}$ is a uniformly computable family of prefix-free subsets of 2^* such that for all $k \in \mathbb{N}$: $\sum_{u\in A_k} 2^{-|u|+g(|u|)} \leq \sum_{u\in A_k} 2^{-K(u)-k} \leq 2^{-k}$. Let d be the martingale defined as follows:

$$d(w) = \sum_{k\in\mathbb{N}} \sum_{u\in A_k} 2^{-|u|+g(|u|)} d_u^{g(|u|)/2}(w)$$

- d is a martingale as the weighted sum of martingales, where the sum of the weights is bounded.
- d is computable: We first show that $d(\lambda)$ is computable. Indeed, $d(\lambda) = \sum_{k\in\mathbb{N}} \sum_{u\in A_k} 2^{-|u|+g(|u|)}$, and for all k, $\sum_{u\in A_k} 2^{-|u|+g(|u|)} \leq 2^{-k}$. This allows us, to get an approximation of $d(\lambda)$ by 2^{-n}, to compute only the finite sum $\sum_{k\leq n} \sum_{u\in A_k} 2^{-|u|+g(|u|)} d_u^{g(|u|)/2}(w)$. By induction, for any word w, if $d(w)$ is computable one can compute $d(w0)$ (and symmetrically $d(w1)$) since

$$d(w0) - d(w) = \sum_{k\in\mathbb{N}} \sum_{u\in A_k} 2^{-|u|+g(|u|)} \left(d_u^{g(|u|)/2}(w0) - d_u^{g(|u|)/2}(w) \right)$$

(which is in fact a finite sum because $d_u^{g(|u|)/2}(w) = 1$ for every u such that $g(|u|) \geq 2|w|$)

- Finally, if $w \in \bigcup_k A_k$, $d(w) \geq 2^{-|w|+g(|w|)} d_w^{g(|w|)/2}(w)$ i.e. $d(w) \geq 2^{g(|w|)/2}$. Since $2^{g(n)/2}$ is an order and there are infinitely many prefixes of R in $\bigcup_k A_k$, this shows that R is not Schnorr random. □

Proof of Proposition 15

Proof. $(i) \rightarrow (ii)$: Let d be a computable martingale and g a computable order such that for all n, $d(R[0..n]) \geq g(n)$. For $k \in \mathbb{N}$, set $g^{-1}(k) = \min\{n \in \mathbb{N} \colon g(n) = k\}$ and $A_k = \{u \colon |u| = g^{-1}(2^{2k})$ *and* $d(u) \geq 2^{2k}\}$. In the same way as in the proof of Proposition 13 (using a Kraft-Chaitin argument), there exists a decidable machine M with prefix-free domain such that for all k and all $u \in A_k$, $K_M(u) = |u| - k$. Thus, for all k: $K_M(R[0..g^{-1}(2^{2k})]) \leq g^{-1}(2^{2k}) - k$. Hence, taking $f(n) = g^{-1}(2^{2n})$, we get the desired result.
$(ii) \rightarrow (iii)$ is immediate.
$(iii) \rightarrow (iv)$: Just take $f = h^{-1}$.

$(iv) \to (i)$: For all n, set $B_n = \{u\colon |u| = f(2n)\ and\ C_M(u) \leq |u| - 2n\}$. Let the d be the martingale defined by

$$d(w) = \sum_{n\in\mathbb{N}} \sum_{u\in B_n} 2^{-f(2n)+n} d_u(w)$$

For all n, $\sum_{u\in B_n} d_u(w) \leq \mathrm{Card}(B_n)2^{|w|} \leq 2^{f(2n)-2n+|w|}$. Hence, for al n the quantity $\sum_{u\in B_n} 2^{-f(2n)+n} d_u(w)$ in the above sum is bounded by $2^{-n+|w|}$, which ensures that this sum is computable up to any required precision, i.e. d is computable. Moreover, for all n, $R([0..f(2n)-1]) \in B_n$ and hence for all $m \geq f(2n)$:

$$\sum_{u\in B_n} 2^{-f(2n)+n} d_u(R[0..m]) \geq 2^n$$

Hence, setting $g(n) = 2^{f^{-1}(2n)}$, we get the desired result. □

Proof of Proposition 24

Proof. Let A be any sequence that is order-low for prefix-free decidable machines and let f be a function that is computable in A. Then there is a prefix-free decidable machine M relative to oracle A which on input $u_x = 1^{|x|}0x$ ouputs $f(x)$. Accordingly, there is a prefix-free decidable machine M' such that for some computable order g and for all x there is some word p_x of length at most $n_x = |u_x| + g(|u_x|)$ where $M'(p_x) = f(x)$. Since M' is decidable, this yields a computable trace for f of size at most 2^{n_x+1}, where the latter bound does not depend on f, hence A is computably traceable.

Next, let A be any sequence that is computably traceable, let M be any prefix-free decidable machine with oracle A, and let h be any order that is computable in A. Let f be the A-computable function such that $f(n)$ contains for all words p such that $h(|p|) \leq 2n + 1$ in appropriately encoded form the information on whether M terminates on input x and if so, the value $M(x)$. Let g be a computable order that grows so slowly that $g(0) + \ldots + g(i) \leq 2i - 1$ holds for all $i > 0$. Furthermore, by assumption on A, let $F_0, F_1, \ldots$ be a computable trace of f where the sets F_i have size of at most $g(i)$. Let $z_1, z_2, \ldots$ be the sequence of values that are contained in the sets F_i where $z_1, \ldots, z_{|F_0|}$ are the values in $F(0)$ in ascending order, $z_{|F_0|+1}, \ldots, z_{|F_0|+|F_1|}$ are the values in $F(1)$ in ascending order, and so on. Consider the machine M' which on input 1^t0x first checks that z_t is an appropriate encoding of the the behavior of a prefix-free machine on a set of words that includes x. If the latter is wrong, then M' does not terminate whereas, otherwise, M' simply copies the behavior at place x of the machine encoded in z_t, i.e., either does not terminate or outputs the same value.

By construction, the machine M' is decidable and prefix-free. Now assume $M(p) = w$ where p is a code of minimum length for w with respect to M and let $i = h(|p|)$. By choice of f and g, the value $f(\lfloor i/2 \rfloor)$ contains the information on the behavior of M at place p, where this value is contained in $F_{\lfloor i/2 \rfloor}$, hence is equal to one of the values z_0 through z_{i-1}. So M' will output w on an input of the form 1^t0p where $t + 1 \leq i = h(|p|) = h(\mathrm{K}^A_M(w))$. □

Proof of Corollary 28

Proof. We give the proof for the case of Schnorr randomness and omit the very similar argument for the case of Kurtz randomness. The characterization of Schnorr randomness given in Proposition 13 relativizes, i.e., for any sequences A and X, the sequence X is not Schnorr random relative to A if there is an A-computable order g and a prefix-free decidable Turing machine relative to A such that for infinitely many n holds

$$\mathrm{K}^A_M(X[0...n]) < n - g(n) + \mathrm{O}(1). \tag{5}$$

By Remark 25 there is a computable order h where $h(n) \leq g(n)$; in particular, inequality (5) remains true with g replaced by h. Let $h'(n) = \lfloor g(n)/3 \rfloor$. By Proposition refprop:orderlow, there is prefix-free decidable Turing machine M' such that for all words w holds $\mathrm{K}_{M'}(w) \leq \mathrm{K}^A_M(w) + h'(n) + \mathrm{O}(1)$, hence for infinitely many n we have

$$\mathrm{K}_{M'}(X[0...n]) < n - h'(n) + \mathrm{O}(1),$$

and R is not Schnorr random according to Proposition 13.

Proof of Proposition 29

Proof. In order to proof the non-trivial direction let A be any sequence that is low for decidable prefix-free machines. Then by Corollary 28, the sequence A is low for Kurtz randomness and low for Schnorr randomness, where all sequences of the former type are hyperimmune-free [**?**] and all sequences of the latter type are Δ^0_2 [12], while a sequence that has both properties must be computable. □

Proof of Proposition ??

Proof. (a) Suppose that R is not mp-Schnorr random, that is there exits some computable martingale process d and some computable order g such that $d(R[0..n]) \geq g(n)$ for infinitely many n, where we can assume $d(\lambda) < 1$.

Fix a computable order g' such that $2^{g'(n)} = o(g(n))$, e.g., $g'(n) = \log\log g(n)$. For all $k > 0$, let A_k be the set of words that are minimal among all words u that satisfy

$$d(u) \geq g(|u|) \geq 2^{g'(n)+k} ,$$

and observe that R has a prefix in every set A_k. Then A_k is prefix-free and closed under $\approx_d$, hence, by the generalized fairness condition for martingales,

$$2^k \sum_{u \in A_k} 2^{-|u|+g'(|u|)} = \sum_{u \in A_k} 2^{-|u|} 2^{g'(|u|)+k} \leq \sum_{u \in A_k} 2^{-|u|} d(u) \leq 1 .$$

Thus the sum of $2^{-|u|+g'(|u|)}$ over all k and all u in A_k is at most 1. Furthermore, the sets A_k are uniformly computable, hence by yet another standard Kraft-Chaitin argument, there is a Turing machine M such that for all k and all words u in A_k,

$$\mathrm{C}_M(u) = |u| - g'(|u|) . \tag{6}$$

Moreover, the domain of M can be chosen to be computable because by construction the set A_k does not contain words of length less than $2k$ and any alleged code of M of length k can only code words in A_1 through A_{2k} of length at most $2k$. In summary, we obtain a computable order g' and a decidable Turing machine M such that R has infinitely many prefixes u where (6) holds, hence R is not Schnorr random according to Proposition 12.
(b) Suppose that R is not mp-Schnorr random, that is there exits some computable martingale process d and some computable order g such that $d(R[0..n]) \geq g(n)$ for all n. By the exact same argument as for the part $(i) \to (ii)$ of Proposition 15, there exist a computable function $f : \mathbb{N} \to \mathbb{N}$ such that for all n and a decidable prefix-free machine M such that $K_M(R[0..f(n)]) \leq f(n) - n$ for all n. Hence by Proposition 15, R is not Kurtz random. □

Proof of Lemma 18

Proof. Let M be a decidable machine. By Proposition 4, there exists a normalized machine M' such that $C_{M'} \leq C_M + O(1)$. Now, define the following time bound:

$$t(n) = \max\{\text{time}(M', p_w) \colon w \in \text{range}(M') \text{ and } |w| = n\}$$

where p_w is the program such that $M'(p_w) = w$ (it is unique by the definition of normalized machines), and $\text{time}(M', p_w)$ is the number of computation steps in the computation of $M'(p_w)$. Again by definition of normalized machines, t is computable as $\text{range}(M')$ is. The fact that $C_{M'} = C^t_{M'}$ is clear, and by the invariance theorem, for some $c > 0$, $C^{c\,t\,\log t} \leq C^t_{M'} + O(1)$. □

Proof of Lemma 19

Proof. Here again we only prove the plain complexity case. Let t be a computable time bound, let h a computable order and by abuse of notation let $h^{-1}(k)$ be equal to the maximum number m where $h(m) \leq k$. Let U be the machine such that $C^t = C^t_U$. Define the machine M as follows: on an input 0^r1p, M simulates $U(p)$ during $t(h^{-1}(r+1) + |p| + 1)$ steps. If the computation halts within these number of steps, set $M(0^r1q) = U(p)$, otherwise $M(p)$ is undefined. Clearly, M is a decidable machine. Let now w be such that $C^t(w) = |w| - k$. Let q be a witness of this, i.e. $|q| = |w| - k$ and U on input q outputs w in no more than $t(|w|)$ steps. By definition of M, we have $M(0^{h(k)-1}1q) = U(q) = w$ (since $h^{-1}(h(k)) + |q| + 1 \geq |q| + k \geq |w|$). Thus, $C_M(w) \leq h(k) + |q| \leq |w| - k + h(k)$. □