Theoretical Computer Science 48 (1986) 9–33
North-Holland

ON THE NOTION OF INFINITE PSEUDORANDOM SEQUENCES *

Ker-I KO**
Department of Computer Science, University of Houston, Houston, TX 77004, U.S.A.

Communicated by P. Young
Received July 1986

Abstract. Three definitions of infinite pseudorandom sequences, with respect to polynomial time and space complexity, are introduced and compared with each other. It is shown that the first two definitions, based on Martin-Löf's notion of sequential tests and Levin and Schnorr's notion of monotonic operator complexity, are equivalent with respect to polynomial space complexity, while both are strictly stronger than the third definition, which is derived from Von Mises's notion of collectives.

Introduction

In the recent literature on computational complexity, several definitions of pseudorandom sequences have been proposed. Yao [32] and Blum and Micali [3] gave a weak definition of pseudorandom sequences. It was argued in [3] that the classical strong definition of randomness, as developed by Martin-Löf [18, 20], Kolmogorov [14] and Chaitin [4, 5, 6], is nonconstructive and unnatural, and a new theory of randomness is necessary for the application to, say, cryptography. They defined a sequence generated by a random-number generator to be pseudorandom if it passes all probabilistic polynomial time statistical tests. Shamir [27], Yao [32], Blum and Micali [3], Blum, Blum and Shub [2] and Plumstead [23] contain detailed analyses of some well-known random-number generators. Wilber [31] defined a set A to be P-random if, for every set $B \in \mathrm{P}$, the set of strings for which A and B agree has density $\frac{1}{2}$. This definition is based on Von Mises's concept of 'collectives', and has been discussed by Meyer and McCreight [21]. Wilber showed the existence of an exponential time computable P-random set, as well as the existence of efficient random-number generators.

While the results obtained in these studies are interesting in the context of complexity theory and cryptography theory, it is not clear what the relation is

* Research supported in part by the NSF Grants MCS-8103479 and DCR-8501226. Part of the work was done while the author was at the Mathematical Sciences Research Institute, Berkeley, CA, U.S.A.

** Present affiliation: Department of Computer Science, SUNY at Stony Brook, Stony Brook, NY 11794, U.S.A.

0304-3975/86/$3.50 © 1986, Elsevier Science Publishers B.V. (North-Holland)

between these definitions and the classical definition of Martin-Löf, Kolmogorov and Chaitin. As it is well known, the theory of randomness based on Von Mises's concept of collectives is unsatisfactory because there are random sequences, according to the definition in this theory, which do not satisfy some important law of probability [19, 33]. Instead, Martin-Löf [18] defined random sequences to be infinite sequences that can withstand all recursively enumerable statistical tests of randomness. He proved that a random sequence must have high program size complexity and thus gave a strong justification for his definition. Kolmogorov [14] and Chaitin [5] actually defined random sequences as those with high program size complexity. Levin [16, 33] and Schnorr [24, 25, 26] defined a variation of program size complexity, called monotonic operator complexity, or process complexity, and used this complexity measure to provide a characterization of Martin-Löf's random sequences. Furthermore, this characterization holds in general cases with respect to arbitrary computable probability distributions on infinite sequences.

In this paper, we propose, based on Martin-Löf and Levin and Schnorr's approaches, two definitions of pseudorandom sequences and study their relationship. It is shown that the class of pseudorandom sequences with respect to polynomial space-bounded Martin-Löf tests is exactly the class of infinite sequences with high polynomial space-bounded monotonic operator complexity. Thus it justifies the naturalness of our approaches. However, for the class of pseudorandom sequences with respect to polynomial time-bounded Martin-Löf tests, such a characterization is not known. It is observed that if $FP = \#P$, then the two definitions, with respect to polynomial time complexity, are equivalent. Whether this condition $FP = \#P$ is necessary remains open and appears to be difficult.

Recently, interesting applications of the concept of time/space-bounded program size complexity to the study of the structure of feasible computation have been demonstrated (see [11, 28]). Sets with low time-bounded program size complexity and their relationship with sets in NP and PSPACE have been studied by Karp and Lipton [13] and many other researchers. In Section 2, we will study recursive sequences that have high polynomial time-bounded program size complexity (KT-complexity, in short). We will demonstrate the existence of a double exponential time computable sequence that has high polynomial time-bounded KT-complexity, and hence, establish the existence of a double exponential time computable pseudorandom sequence. The equivalence of the two definitions (with respect to polynomial space bounds) will be established in Section 3. In Section 4, we will compare our definitions of pseudorandom sequences with the weaker definition of Meyer and McCreight [21] and Wilber [31] and show that our definitions are strictly stronger than theirs. This comparison reveals interesting properties of the relative frequency of pseudorandom sequences. Based on this result, we propose a modification to the latter definition.

Notation. We only consider binary sequences, i.e., finite strings in $\{0, 1\}^*$ and infinite sequences in $\{0, 1\}^\infty$. Finite strings are often denoted by s and t and infinite sequences are denoted by x, y, and z. The length of a finite string s is denoted by $|s|$. For each string x of length $\geq n$ (finite or infinite), x^n denotes the initial segment

of x of length n, and $x(n)$ denotes the nth bit of x. (This notation should not be confused with extended regular expressions by which x^n denotes the repetition of x for n times. However, we *do* use 0^n and 1^n to denote a string of n 0's and a string of n 1's. 0^∞ denotes the infinite sequence with all bits equal to 0.) Natural numbers are represented by finite strings in $\{0, 1\}^*$ in its standard binary expansion. Let $d(s)$ be the string obtained from s by doubling each bit of s. We use a fixed coding scheme for the pairing function $\langle , \rangle$: $\langle s, t\rangle = d(s)01t$. Thus, $|\langle s, t\rangle| = |t| + 2|s| + 2$. We use $\#A$ to denote the cardinality of the set A. We say p is polynomial to mean that p is a polynomial function with nonnegative coefficients. We use $(\forall^\infty n)$ to denote 'for all but finitely many' and $(\exists^\infty n)$ to denote 'for infinitely many'. For each n, $\log n$ means $\log_2 n$.

We will use the standard notation for complexity classes. P denotes the class of sets computable in polynomial time. FP denotes the class of functions computable in polynomial time. NP denotes the class of sets recognizable in polynomial time by nondeterministic Turing machines (TMs). #P denotes the class of functions counting the number of accepting paths of a polynomial time nondeterministic TM. PSPACE denotes the class of sets computable in polynomial space. FPSPACE denotes the class of functions computable in polynomial space. It is obvious that $\mathrm{P} \subseteq \mathrm{NP} \subseteq$ PSPACE, and $\mathrm{FP} \subseteq \#\mathrm{P} \subseteq$ FPSPACE. Whether any of the above inclusions is proper is a major open question in complexity theory [10].

1. Polynomial time Martin-Löf tests and pseudorandom sequences

Martin-Löf [18] defined an infinite binary random sequence to be an infinite sequence which can withstand all recursively enumerable (r.e.) tests of randomness. From the statistical point of view, a test of randomness is a function f which accepts or rejects, for any finite string s and level of significance ε, the hypothesis that s is random; or, equivalently, it is a function f which, for any finite string s, outputs an integer $f(s)$ as an indicator of the 'quantity of regularity' in s such that the hypothesis that s is random is rejected on the level $\varepsilon = 2^{-m}$ iff $f(s) \geq m$. The following is a more precise definition.

Definition 1.1 (Martin-Löf [18]; Zvonkin and Levin [33]). A *Martin-Löf test* (or, simply, a *test*) is a function $f: \{0, 1\}^* \to \mathbb{N}$ such that

(i) the set $\{(s, n) \mid f(s) \geq n\}$ is r.e.,

(ii) $(\forall n)(\forall m)\ \#\{s \mid |s| = n \text{ and } f(s) \geq m\} \leq 2^{n-m}$, and

(iii) (f is sequential) $f(s) \leq f(t)$ whenever s is an initial segment of t.

Condition (ii) arises from the requirement that the probability of rejecting a string s on level 2^{-m} must be $\leq 2^{-m}$. This condition may be formulated in terms of probability measures on $\{0, 1\}^\infty$. (In this paper, we deal only with the uniform probability distribution.)

Lemma 1.2 (Zvonkin and Levin [33]). *A function* $f: \{0,1\}^* \to \mathbb{N}$ *is a test iff* (i), (iii) *and*
(ii') $\Pr\{x \in \{0,1\}^\infty \mid (\exists n) f(x^n) \geq m\} \leq 2^{-m}$ *hold.*

Now, an infinite random sequence can be defined as a sequence for which there is a significance level ε such that no Martin-Löf test can ever reject any initial segment of x on level ε.

Definition 1.3 (Martin-Löf [18]; Zvonkin and Levin [33]). An infinite sequence $x \in \{0,1\}^\infty$ is *random* if, for any Martin-Löf test f, $\lim_{n\to\infty} f(x^n) < \infty$.

It is shown in [18] that, with probability 1, a sequence x in $\{0,1\}^\infty$ is random. This means that Definition 1.3 is not too strong.

We now give a definition of infinite pseudorandom sequences based on the concept of Martin-Löf tests. While a random sequence is a sequence which cannot be rejected by any effective method of testing randomness, our definition of pseudorandom sequences requires that it cannot be *easily* rejected by any *efficient* method of testing randomness. In other words, we will only consider tests of randomness which have polynomial time complexity, i.e., tests in FP. (Sometimes we also consider tests which have polynomial space complexity.) More generally, let C be a class of functions.

Definition 1.4. A Martin-Löf test f is called a *C-test* if $f \in C$.

It then appears natural to define, for any complexity class C, a sequence $x \in \{0,1\}^\infty$ to be pseudorandom with respect to C-tests if, for all C-tests f, $\lim_{n\to\infty} f(x^n) < \infty$. However, the following lemma shows that if $C = \mathrm{FP}$, then these sequences are exactly random sequences as defined by Martin-Löf.

Lemma 1.5. *Let* $x \in \{0,1\}^\infty$. *If for all* FP*-tests* f, $\lim_{n\to\infty} f(x^n) < \infty$, *then* x *is random.*

Proof. Assume that x is not random. Then there is a Martin-Löf test f such that $\lim_{n\to\infty} f(x^n) = \infty$, and $G_f = \{(s,k) \mid f(s) \geq k\}$ is r.e.

Let M be a Turing machine (TM) enumerator for G_f. We construct another TM M' as follows: On input s with $|s| = n$, M' simulates M on the empty string for n moves, and outputs $k_s = \max^*\{k \mid M$ enumerates some pair (t,k) in n moves with t being an initial segment of $s\}$, where $\max^*(A) = \max(A)$ if $A \neq \emptyset$, and $=0$ if $A = \emptyset$.

Clearly, M' computes a function $g: \{0,1\}^* \to \mathbb{N}$ in polynomial time. Also, for all $s, t \in \{0,1\}^*$, $g(t) \leq g(s)$ if t is an initial segment of s. We claim that g is an FP-test.

Since f is sequential, $(t,k) \in G_f$ implies $(s,k) \in G_f$ whenever t is an initial segment of s. This implies that, for all s, $g(s) \leq \max\{k \mid (s,k) \in G_f\} = f(s)$. Therefore,

$$\Pr\{x \in \{0,1\}^\infty \mid (\exists n)\, g(x^n) \geq m\} \leq \Pr\{x \in \{0,1\}^\infty \mid (\exists n) f(x^n) \geq m\} \leq 2^{-m}.$$

So, g is an FP-test.

Finally, we observe that $\lim_{n\to\infty} f(x^n) = \infty$ implies $\lim_{n\to\infty} g(x^n) = \infty$ because, for any large k, there is an n such that, for some initial segment t of x, (t, k) will be generated by M in n moves, and thus $g(x^n) \geq k$. This completes the proof. □

In the above lemma, the class FP may be replaced by any reasonable complexity class C. So it shows that simply putting time bounds on the test functions does not provide a bigger class of random sequences. Thus, for the definition of pseudorandom sequences, we must allow the sequence $\{f(x^n)\}$ to diverge to infinity. This suggests the following general definition.

Definition 1.6. Let C be a complexity class, and F a class of nondecreasing, unbounded functions from $\mathbb{N}$ to $\mathbb{N}$. A sequence $x \in \{0,1\}^\infty$ is *pseudorandom* with respect to C-tests and diverging rates F if for any C-test f there is a function $g \in F$ such that $(\forall^\infty n) f(x^n) \leq g(n)$.

In other words, a sequence x is pseudorandom with respect to C-tests and the diverging rate g if no C-test f can reject the hypothesis that x is random on the significance level 2^{-m} by examining only the first $g^{-1}(m)$ bits of x.

Obviously, different complexity bounds C on the tests and different diverging rates F give different classes of pseudorandom sequences. We will, however, consider only pseudorandom sequences with respect to a small class of tests and diverging rates. Intuitively, we say a sequence x is (polynomially) pseudorandom if, for any polynomial p, no test f can detect, in time $p(m)$, enough 'quantity of regularity' in x to reject the hypothesis that x is random on the significance level 2^{-m}. Comparing this requirement with Definition 1.6, we arrive at the following two classes of pseudorandom sequences. Let LOG be the class of all functions $\lambda n[(\log n)^k]$, $k \geq 0$.

Definition 1.7. (a) PR1 $= \{x \in \{0,1\}^\infty \mid x$ is pseudorandom with respect to FP-tests and diverging rates LOG$\}$.

(b) PSR1 $= \{x \in \{0,1\}^\infty \mid x$ is pseudorandom with respect to FPSPACE-tests and diverging rates LOG$\}$.

We note that PR1 is the largest class of infinite sequences satisfying our informal requirement for polynomial pseudorandomness. If we allow the diverging rate to grow a little faster, for instance, letting $g(n) = n^{1/k}$ for some $k > 1$, then an FP-test f may be able to reject x as nonrandom on the level 2^{-m} by examining only the first m^k bits of x (thus using only time $p(m)$ for some polynomial p). On the other hand, if we require a slower diverging rate, for instance $g(n) = (\log\log n)^k$ for some $k > 0$, then no FP-test f can reject a pseudorandom sequence x on level 2^{-m} even if it spends an exponential amount of time to get $f(x^{2^m})$. So, this definition would give a much smaller class of pseudorandom sequences.

In [20], Martin-Löf showed that for the class of Martin-Löf tests, there is a universal test f_u such that, for any test f, there is a constant c such that $f_u(s) + c \geq f(s)$

for all $s \in \{0, 1\}^*$. Is there such a universal FP-test for the class of FP-tests? Probably not. First, such a universal FP-test must be able to simulate all FP-tests and so cannot be itself computed in polynomial time. Second, it is essential that the class of FP-tests be recursively presentable in order for the universal FP-test to simulate them systematically. (A class F of functions is *recursively presentable* if there is an r.e. set of TMs $\{M_1, M_2, \ldots\}$ such that $F = \{f_i \mid f_i$ is the function computed by $M_i\}$.) In general, for a (reasonable) complexity class C, the class of C-tests does not have a universal test in C. On the other hand, if the class of C-tests is recursively presentable, then a universal test can be found with complexity slightly higher than the complexity bounds for the class C. We do not know if the class of FP-tests is recursively presentable. In the following, we will show that the class of FPSPACE-tests is recursively presentable, and hence has a universal test f_u which is computable in space g for any superpolynomial function g.

Theorem 1.8. *The class of* FPSPACE*-tests is recursively presentable.*

Proof. Let $\{M_i\}$ be an enumeration of the class of polynomial space TMs. We assume that each M_i computes a total function in space $p_i(n)$. The following algorithm describes a TM M_i' for each i. Let $<$ be the lexicographic order on $\{0, 1\}^*$.

input: s {let $n := |s|$}.
begin
  if $n = 0$ **then** output 0 **and** halt;
  for all u of length $\leq n$ **do**
    for all initial segments v of u **do**
      if $M_i(v) > M_i(u)$
      then recursively compute and output $M_i'(s^{n-1})$ **and** halt;
  for $k := 0$ **to** n **do**
    for $m := 0$ **to** $p_i(k)$ **do**
      if $\#\{t \mid |t| = k$ and $M_i(t) \geq m\} > 2^{k-m}$
      then recursively compute and output $M_i'(s^{n-1})$ **and** halt;
  output $M_i(s)$ **and** halt
end.

We claim that the function f_i computed by M_i' is an FPSPACE-test. First, we note that the amount of space required to perform the computation of $f_i(s)$ except recursive calls of $M_i'(s^{n-1})$ is $O(p_i(n))$. Since we make at most n levels of recursive calls, the total space requirement is only $O(n \cdot p_i(n))$. So, f_i is polynomial space computable. Furthermore, $f_i(s) \neq f_i(s^{n-1})$ implies $f_i(s) = M_i(s) > f_i(s^{n-1})$. Thus, by induction, $f_i(t) \leq f_i(s)$ for all initial segments t of s.

Finally, let $A_{n,m} = \{t \mid |t| = n$ and $f_i(t) \geq m\}$. We check that, for all n, m, $\#A_{n,m} \leq 2^{n-m}$. First, this statement is true for $n = 0$. Assume that $\#A_{n,m} \leq 2^{n-m}$ for all m. Consider $\#A_{n+1,m}$. Suppose, by way of contradiction, that $\#A_{n+1,m} > 2^{n+1,m}$. Then, by the inductive hypothesis, there must be a string s of length $n+1$ such that

$f_i(s) \geq m > f_i(s^n)$. That is, $f_i(s)$ must be equal to $M_i(s)$ and hence, during the computation of M_i', we must have had

(1) $M_i(v) \leq M_i(u)$ for all u of length $n+1$ and all initial segments v of u, and

(2) $\#\{t \mid |t| = n+1, M_i(t) \geq m\} \leq 2^{n+1-m}$.

Condition (1) implies that, for all t of length $n+1$, $f_i(t) \leq M_i(t)$ and so

$$\#A_{n+1,m} \leq \#\{t \mid |t| = n+1, M_i(t) \geq m\}.$$

But this violates condition (2) and leads to a contradiction. Thus we have proved that $\#A_{n+1,m} \leq 2^{n+1-m}$, and hence, $\#A_{n,m} \leq 2^{n-m}$ for all n and m.

Conversely, if f is an FPSPACE-test computed by M_i, then we must have

(3) $(\forall u)(\forall v)$ (v is an initial segment of u) implies $(M_i(v) \leq M_i(u))$, and

(4) $(\forall k)(\forall m)\ \#\{t \mid |t| = k \text{ and } M_i(t) \geq m\} \leq 2^{k-m}$.

Thus, the function f_i computed by M_i' is exactly f. □

Now a universal test f_u for the class of FPSPACE-tests may be defined as follows: for each $s \in \{0, 1\}^*$, $f_u(s) = \max\{f_i(s) - i \mid i \leq |s|\}$. It can be seen that f_u is a Martin-Löf test. Furthermore, if x is not in PSR1, then $(\forall k)(\exists^\infty n)\ f_u(x^n) > (\log n)^k$. That is, whenever x is not in PSR1, the test f_u can catch it. Assume that the function f_i is computable in space $\lambda n[n^{\log i}]$. Then the universal test f_u is computable in space $\lambda n[n^{\log n}]$. However, since f_u needs to simulate all FPSPACE-tests f_i, it does not have a polynomial space bound, and so is not an FPSPACE-test.

2. Time-bounded program size complexity

Kolmogorov [14] and Chaitin [4, 5, 6] introduced the concept of program size complexity and, based on this concept, proposed a definition of random sequences. To be more precise, the program size complexity (or, the Kolmogorov complexity) of a finite string is the length of the shortest TM program that prints it; and a finite string is random if its Kolmogorov complexity is 'almost' equal to its length. Intuitively, the complexity of a finite string is a measure of the amount of information contained in the string, i.e., the minimum information that is sufficient to compute the string. However, the amount of resources, such as time and space, required to compute the string is not measured by its Kolmogorov complexity. Recently, time- and space-bounded Kolmogorov complexity has been introduced and has been demonstrated useful in complexity theory [8, 11, 15, 28]. This generalized Kolmogorov complexity is also closely related to the notion of circuitry complexity as studied by Pippenger [22] and Karp and Lipton [13]. In this section, we will study the basic properties of time- and space-bounded program size complexity and, in particular, the existence of recursive sequences that have high time-bounded program size complexity.

Recall that the Kolmogorov complexity of a string $s \in \{0, 1\}^*$, with respect to a Turing machine (TM) M, is $K_M(s) = \min^*\{|t| \mid M(t) = s\}$, where $\min^*(A) = \min(A)$

if $A \neq \emptyset$, and $=\infty$ if $A = \emptyset$. By adding time or space bounds on TMs, we have the following definition (cf. [8, 11, 28]).

Definition 2.1. The *time-* and *space-bounded program size complexities* (KT- and KS-*complexity*, in short) of a string $s \in \{0,1\}^*$, with respect to a TM M, are

$$\mathrm{KT}^k_M(s) = \min^*\{|t| \,|\, M(t) \text{ halts and prints } s \text{ in } k \text{ moves}\}$$

and

$$\mathrm{KS}^k_M(s) = \min^*\{|t| \,|\, M(t) \text{ halts and prints } s \text{ using } k \text{ cells}\},$$

respectively.

The following observation on the universal TM U is well known.

Observation 2.2. *There is a universal TM U and a polynomial p_0 such that, for all inputs $\langle \tilde{M}, s\rangle$, where $\tilde{M}$ encodes a TM M, if $M(s)$ halts in k moves (or, using k cells), then $U(\langle \tilde{M}, s\rangle)$ halts and prints $M(s)$ in $p_0(k)$ moves (using $p_0(k)$ cells, respectively).*

Using this universal TM U, the KT_U-complexity is optimal in the sense that, for all TM M, there is a constant c such that

$$(\forall s \in \{0,1\}^*)(\forall k)\ \mathrm{KT}^{p_0(k)}_U(s) \leq \mathrm{KT}^k_M(s) + c$$

because $M(t) = s$ implies $U(\langle \tilde{M}, t\rangle) = s$ and $|\langle \tilde{M}, t\rangle| = |t| + 2|\tilde{M}| + 2$. So, in the rest of this paper, unless otherwise stated, we will use this fixed TM U for the KT- or KS-complexity measure, and omit the subscript U. We sometimes say a string t is the *shortest TM program* for a string s to mean that t is the shortest string such that $U(t) = s$.

In addition to the absolute program size complexity, Kolmogorov [14] also introduced the *conditional* program size complexity in which the information about the length of the string to be computed is given without charge. Loveland [17] modified it and introduced the *uniform* program size complexity which requires a program for a string s, when given a length $i \leq |s|$, to output the initial segment s^i of s. This prevents the use of the length $|s|$ from being used to provide information about the bits of the string s. In the following we define the conditional and uniform KT-complexity measures.

Definition 2.3. Let k be an integer.

(a) The *time-bounded conditional program size complexity* of $s \in \{0,1\}^*$ (with respect to a two-input universal TM U) is

$$\mathrm{KT}^k(s|n) = \min^*\{|t| \,|\, U(t, n) \text{ halts and prints } s \text{ in } k \text{ moves}\},$$

where $n = |s|$.

(b) The *time-bounded uniform program size complexity* of $s \in \{0,1\}^*$ is

$$\mathrm{KT}^k(s;n) = \min^*\{|t| \,|\, (\forall i \leq n)\ U(t,i) \text{ halts and prints } s^i \text{ in } k \text{ moves}\},$$

where $n = |s|$.

The *space-bounded conditional and uniform complexity* $\mathrm{KS}^k(s|n)$ and $\mathrm{KS}^k(s;n)$ are similarly defined.

Notation. Let f be a function and s a string of length n. We will write $\mathrm{KT}^f(s)$ and $\mathrm{KT}^f(s|n)$ to denote $\mathrm{KT}^{f(n)}(s)$ and $\mathrm{KT}^{f(n)}(s|n)$, respectively.

Remark 2.4. The generalized circuitry complexity introduced by Karp and Lipton [13] (called by them 'nonuniform complexity') may be considered as another form of the time-bounded Kolmogorov complexity defined on *sets* of strings. For example, for a set $A \subseteq \{0,1\}^*$, let A^n denote the set $\{s \in A \,|\, |s| \leq n\}$. We may define the time-bounded program size complexity of A^n with respect to a time function f as $\mathrm{CT}^f(A^n) = \min\{|t| \,|\, (\forall s, |s| \leq n)\ U(\langle t, s\rangle)$ halts and outputs $\chi_A(s)$ in $f(n)$ moves$\}$. Using this notation, the class P/poly of sets with small circuits defined in [13] is just the class of sets A with the following property:

$$(\exists \text{polynomials } p, q)(\forall n)\ \mathrm{CT}^p(A^n) \leq q(n).$$

Now we consider recursive sequences with high KT-complexity. In the following, we work only with the conditional KT-complexity. The analogous results about the uniform KT-complexity will be discussed at the end of the section.

First, we consider sequences whose initial segments have high KT-complexity infinitely often. Martin-Löf [20] proved that, with probability 1, an infinite sequence x has the maximal conditional complexity: $(\exists c)(\exists^\infty n)\ \mathrm{K}(x^n|n) \geq n - c$. The next theorem shows that recursive sequences cannot have such maximal KT-complexity, even with a polynomial time bound.

Theorem 2.5. *Let x be an infinite recursive sequence. Then, there exists an unbounded function f, and a polynomial p such that* $(\forall^\infty n)\ \mathrm{KT}^p(x^n|n) \leq n - f(n)$.

Proof. Assume that x is recursive, and let M be a TM which computes x; i.e. for all m, $M(m) = x^m$. Consider the following algorithm M' for computing x^n.

Algorithm M'. On input (t, n), simulate $M(0)$, $M(1), \ldots$ for n moves. Let $x^m = M(m)$ be the longest output that we get in n moves. Then output the first n bits of the sequence $x^m t 0^\infty$.

Let $g(n) = \max\{m \,|\, \text{the computation of } M(0), M(1), \ldots, M(m) \text{ halts in } \leq n \text{ moves}\}$. Then, $(\forall n)(\exists t, |t| = n - g(n))\ M'(t, n) = x^n$. Since M' is a polynomial time-bounded TM, we can encode this TM together with its input (t, n) and simulate it

with the universal TM U. This gives us the following bound on the KT-complexity of x:

$$(\exists \text{polynomial } p)(\exists c)\ \mathrm{KT}^p(x^n|n) \leq n - g(n) + c.$$

Obviously, $g(n) \to \infty$ as $n \to \infty$. So the theorem is proved. □

Corollary 2.6. *If $x \in \{0,1\}^\infty$ is computable in time $O(2^{kn})$ for some $k \geq 1$, then*

$$(\exists \text{polynomial } p)(\forall^\infty n)\ \mathrm{KT}^p(x^n|n) \leq n - \log n.$$

The next theorem shows that the upper bound given by Theorem 2.5 is the best we can have.

Theorem 2.7. *For any unbounded recursive function f, there is a recursive sequence x such that $(\forall \text{polynomial } p)(\exists^\infty n)\ \mathrm{KT}^p(x^n|n) \geq n - f(n)$.*

Proof. Assume, without loss of generality, that $f(n) < n$ for all n. Define a function $g(n)$ inductively:

$$g(0) = 0, \qquad g(n+1) = \min\{m \mid m \geq n, f(m) \geq g(n)\}.$$

Since f is unbounded, g is well-defined and recursive. Furthermore, $g(n+1) > g(n)$ for all n.

We define, for each $n \geq 1$, the subsequence $x(g(n-1)+1) \ldots x(g(n))$ (i.e., from the $(g(n-1)+1)$st bit to the $g(n)$th bit of x) to be distinct from those bits of the outputs of $U(t, g(n))$ for all t of length $< g(n) - g(n-1)$ such that $U(t, g(n))$ halts in $2^{g(n)}$ moves. Since there are less than $2^{g(n)-g(n-1)}$ such strings t and there are $g(n) - g(n-1)$ many bits of x to be defined, this diagonalization always works. (Indeed, it works in time $2^{O(g(n))}$.)

Now, for any polynomial p, we have

$$(\forall^\infty n)\ \mathrm{KT}^p(x^{g(n)}|g(n)) \geq g(n) - g(n-1) \geq g(n) - f(g(n)).$$

This completes the proof. □

Next we study recursive sequences whose initial segments have high KT-complexity almost everywhere. First we recall that Martin-Löf [20] has proved that if f is a recursive function satisfying the condition $\sum_{n=0}^{\infty} 2^{-f(n)} = \infty$, then, for every infinite sequence x,

$$(\exists^\infty n)\ \mathrm{K}(x^n|n) \leq n - f(n).$$

We can modify his proof to show a similar upper bound for KT-complexity.

Theorem 2.8. *Let f be a function such that (a) $f(n) < n$ for all $n \geq 1$, (b) $\sum_{n=0}^{\infty} 2^{-f(n)} = \infty$, and (c) the function $h(n) = 2^{n-f(n)} - 1$ is computable in time $r(n)$ for some polynomial r. Also let $x \in \{0,1\}^\infty$. Then, there exist a polynomial p and a constant c such that*

$$(\exists^\infty n)\ \mathrm{KT}^p(x^n|n) \leq n - f(n) + c.$$

Remark 2.9. Actually, Theorem 2.8 holds for all functions f' such that $f'(n) \leq f(n)$ for some f satisfying conditions (a), (b), and (c). An example is $f'(n) \leq \lceil \log n \rceil$.

Proof of Theorem 2.8. Define a function succ on all finite strings: if $s = 1^n$ for some n, then $\text{succ}(s) = 0^n$; otherwise $\text{succ}(s) = s + 1$. Then, inductively define sets A_n as follows: $A_0 := \{\varepsilon\}$. Let s be the last string in A_{n-1} with respect to the function succ, i.e., s is the (unique) string in A_{n-1} whose successor $\text{succ}(s)$ is not in A_{n-1}. Define A_n to contain $2^{n-f(n)} - 1$ many strings of length n, starting with the string $t = \text{succ}(s1)$ and containing the next $2^{n-f(n)} - 2$ successors. The sets A_n are well-defined because each set A_n has exactly one string without a successor in A_n.

Define, for each n, $B_n = \{x \in \{0,1\}^\infty \mid x^n \in A_n\}$. Also define a circular order on $\{0,1\}^\infty$ as the natural lexicographic order on $\{0,1\}^\infty$, with the extra rule that 1^∞ is immediately followed by 0^∞. Then, B_n contains an 'interval' of sequences (with respect to the circular order) which is immediately followed by the interval B_{n+1}. Furthermore, assuming the uniform probability measure on $\{0,1\}^\infty$, we have $\Pr(B_n) = 2^{-f(n)} - 2^{-n}$. Since $\sum_{n=1}^\infty 2^{-f(n)} = \infty$, the sets $\{B_n\}_{n=1}^\infty$ circularly cover $\{0,1\}^\infty$ infinitely many times. Or, equivalently, for any $x \in \{0,1\}^\infty$, $x^n \in A_n$ infinitely often.

Now, consider the following function g: $g(t, n) =$ the tth string in A_n with respect to succ. We claim that $g(t, n)$ can be computed in time $p(n)$ for some polynomial p. Define $u_n =$ the last string in A_n, and $h(n) = 2^{n-f(n)} - 1$. We observe that, for each $n > 0$, $u_n = \text{succ}^{(h(n))}(u_{n-1}1)$ where $\text{succ}^{(k)} =$ the composition of k succ's. It is easy to see that the function $\lambda k, s[\text{succ}^{(k)}(s)]$ is just $(s + k) \bmod 10^{|s|}$ and is polynomial time computable. So, $g(t, n)$ can be computed by successively calculating u_0, $u_1, \ldots, u_{n-1}$, and $g(t, n) = \text{succ}^{(t)}(u_{n-1}1)$ and is computable in time $p(n)$ for some polynomial p.

Since, for any $x \in \{0,1\}^\infty$, x^n occurs in A_n infinitely often, we have, for all $x \in \{0,1\}^\infty$, $(\exists^\infty n)(\exists t, |t| \leq n - f(n))\ g(t, n) = x^n$. By encoding the TM program for g into a string of fixed length, we have

$$(\exists \text{polynomial } q)(\exists c)(\exists^\infty n)\ \text{KT}^q(x^n \mid n) \leq n - f(n) + c. \qquad \square$$

The next theorem shows that if $\sum_{n=0}^\infty 2^{-f(n)} < \infty$, then we can find recursive sequences with KT-complexity almost as high as $n - f(n)$ almost everywhere. The proof technique is a refinement of Meyer and McCreight's weighted priority diagonalization [21].

Theorem 2.10. *Let f be a nondecreasing, unbounded recursive function such that $\sum_{n=0}^\infty 2^{-f(n)}$ converges to a real number $\alpha < \infty$. Then, for any recursive function ϕ, there is a recursive sequence x such that*

$$(\forall^\infty n)\ \text{KT}^\phi(x^n \mid n) \geq n - f(n) - \lfloor \log n \rfloor.$$

Proof. The idea of the proof is to construct a recursive sequence x bit by bit such that for each bit $x(n)$ we try to diagonalize against exponentially many TM programs.

Since a TM program t may be used to compute any initial segment of x, our diagonalization process works on pairs (t, n) for all strings t and all numbers n.

We first define a function $h:\{0,1\}^*\to\mathbb{N}$ by $h(t)=|t|+f(|t|)$. Then, let S be the set of all pairs (t, n) to be diagonalized; i.e., $S=\{(t,n)\mid t\in\{0,1\}^*,\ n\in\mathbb{N}$ and $n-\lfloor\log n\rfloor\geq h(t)\}$. Also, for each pair $(t,n)\in S$, define a weight $w(t,n)=2^{-(n-\lfloor\log n\rfloor)}$. Note that the function $g(n)=n-\lfloor\log n\rfloor$ is nondecreasing and has the property that, for all n, $g(n)<g(n+2)$. Thus, $\sum_{g(n)\geq k}2^{-g(n)}\leq 2\cdot\sum_{m=k}^{\infty}2^{-m}=2^{-(k-2)}$. It follows that the total weight of pairs in S is

$$\sum_{(t,n)\in S}w(t,n)=\sum_{m=0}^{\infty}\sum_{|t|=m}\sum_{g(n)\geq h(t)}2^{-g(n)}$$

$$\leq\sum_{m=0}^{\infty}\sum_{|t|=m}2^{-(h(t)-2)}=\sum_{m=0}^{\infty}2^{-(f(m)-2)}=4\alpha.$$

Now we describe an algorithm for x. The algorithm proceeds in stages. At stage k, it determines the kth bit of x. Prior to stage 1, we assign to each pair (t,n) in S the initial weight $w(t,n)=2^{-g(n)}$, and all pairs in S are *uncancelled*. In each stage, some pairs may be cancelled and some may double their weights.

Stage k: For $j:=0$ and 1, let $Q_j:=\{(t,n)\in S\mid k\leq n\leq 2^{2k+2}$, (t,n) is uncancelled, and $U(t,n)$ prints $x^{k-1}js$ in $\phi(n)$ moves for some s of length $n-k\}$, and $v_j:=\sum_{(t,n)\in Q_j}w(t,n)$.

If $v_0\geq v_1$, then we cancel all pairs (t,n) in Q_0, set $x(k):=1$, double weight $w(t,n)$ for all pairs (t,n) in Q_1, and go to the next stage; otherwise, we do the opposite: cancel all pairs (t,n) in Q_1, set $x(k):=0$, double weight $w(t,n)$ for all pairs (t,n) in Q_0, and go to the next stage.

End of stage k.

First, note that in each stage k, the total weight of uncancelled pairs in S is increased by $\min\{v_0,v_1\}$ and decreased by $\max\{v_0,v_1\}$ and therefore it can never exceed the initial value 4α.

Next, if the weight of a pair (t,n) is doubled m times before it is cancelled, its final weight becomes $2^{m-n+\lfloor\log n\rfloor}$. Since this value is less than or equal to the total weight 4α, we have $m-n+\lfloor\log n\rfloor\leq\log\alpha+2$. That is, each pair $(t,n)\in S$ can be doubled at most $n-\lfloor\log n\rfloor+\log\alpha+2$ times.

Now, assume that t_n is the shortest string such that $U(t_n,n)$ prints x^n in $\phi(n)$ moves. Then the pair (t_n,n) is never cancelled. However, if $(t_n,n)\in S$, then (t_n,n) must have been included in $Q_0\cup Q_1$ from stage $\frac{1}{2}\lfloor\log n\rfloor$ to stage n, and hence its weight must have been doubled $n-\frac{1}{2}\lfloor\log n\rfloor-1$ times. Since a pair (t,n) in S can be doubled at most $n-\lfloor\log n\rfloor+\log\alpha+2$ times, we have $n-\frac{1}{2}\lfloor\log n\rfloor-1\leq n-\lfloor\log n\rfloor+\log\alpha+2$; or, $n\leq 2^7\alpha^2$. Thus, for almost all n, $(t_n,n)\notin S$. That is,

$$(\forall^{\infty}n)\ n-\lfloor\log n\rfloor<|t_n|+f(|t_n|);$$

or,

$$(\forall^{\infty} n)\ \mathrm{KT}^{\phi}(x^n|n) > n - \lfloor \log n \rfloor - f(n)$$

because $n \geq |t_n|$ implies $f(n) \geq f(|t_n|)$. □

The following corollary will be used in the next section to show the existence of a double exponential time computable sequence in PR1.

Corollary 2.11. *Let $\phi(n) = 2^{cn}$ for some constant c. Then, there is an infinite sequence x, computable in time $2^{2^{O(n)}}$, such that $(\forall^{\infty} n)\ \mathrm{KT}^{\phi}(x^n|n) \geq n - 3 \log n$.*

Proof. In the proof of Theorem 2.10, let $f(n) = \lfloor 2 \log n \rfloor$. Then, at stage k, we need to simulate $U(t, n)$ for $2^{2^{2k+2}}$ many pairs (t, n), each for $\leq \phi(n) \leq 2^{2^{2k+c+2}}$ moves. So, x^n can be computed in time $2^{2^{O(n)}}$. □

Remark 2.12. It is easy to check that all of the above results also hold for the conditional KS-complexity. They also hold for the uniform KT- and KS-complexity. We first make two general observations about the relationship between the conditional and uniform KT-complexity.

Observation 2.13. $(\forall s)(\forall k)\ \mathrm{KT}^k(s|n) \leq \mathrm{KT}^k(s; n)$, *where* $n = |s|$.

Observation 2.14. *There exist a polynomial p and a constant c such that, for all s of length n and for all integers k and m,*

$$\mathrm{KT}^k(s|n) \leq n - m \Rightarrow \mathrm{KT}^{p(k)}(s; n) \leq n - m + 2 \log m + c.$$

Sketch of proof for Observation 2.14. Assume that $U(t, n)$ prints s in k moves with $|t| = n - m$. Define a new TM M as follows: on input $(\langle j, u\rangle, i)$, M simulates $U(u, |u| + j)$ and prints the first i bits of its output. Then, for all $i \leq n$, $M(\langle m, t\rangle, i)$ prints s^i in $p(k)$ moves for some polynomial p. Note that $|\langle m, t\rangle| = 2|m| + |t| + 2$. This proves Observation 2.14. □

Now, from Observation 2.13, Theorems 2.7 and 2.10 and Corollary 2.11 hold for the uniform KT-complexity, too. From Observation 2.14, Theorem 2.5 and Corollary 2.6 hold for the uniform KT-complexity, because the function $f(n) - 2 \log(f(n))$ is unbounded whenever the function $f(n)$ is unbounded. Finally, from Observation 2.14, we have the following weaker form of Theorem 2.8 for the uniform KT-complexity.

Theorem 2.8'. *Let f and x be given as in Theorem* 2.8. *Then, there exist a polynomial p and a constant c such that*

$$(\exists^{\infty} n)\ \mathrm{KT}^{p}(x^{n};n) \leq n - f(n) + 2\log(f(n)) + c.$$

3. The monotonic KT-complexity and pseudorandom sequences

The relationship between Martin-Löf's random sequences and program size complexity has been observed in many forms. Martin-Löf [20] has observed the following relations:

$$\begin{aligned}&(\exists c)(\exists^{\infty} n)\ \mathrm{K}(x^{n}|n) \geqslant n - c\\ &\quad\Rightarrow\ x \text{ is random}\\ &\quad\Rightarrow\ (\forall f, f \text{ recursive and } \textstyle\sum 2^{-f(n)} < \infty)(\forall^{\infty} n)\ \mathrm{K}(x^{n}|n) \geqslant n - f(n).\end{aligned}$$

Kolmogorov [14] and Chaitin [4, 5] actually defined a random sequence x to be the one with high K-complexity almost everywhere. Levin [16, 33] and Schnorr [24, 25, 26] used a variation of K-complexity, called monotonic operator complexity, or process complexity, to give an exact characterization of random sequences. We give, in the following, a brief review of this work.

Definition 3.1 (Schnorr [24, 25]). A TM M is called a *monotonic operator* if, for any s and t in the domain of M, $M(s)$ is an initial segment of $M(t)$ whenever s is an initial segment of t.

The class of monotonic operators are recursively presentable, and hence there is a universal monotonic operator UM.

Observation 3.2. *There exists a universal monotonic operator* UM *such that, for any monotonic operator M, there exist a polynomial p and a constant c such that*

$$(\forall s)(\forall k)\ \mathrm{K}_{\mathrm{UM}}^{p(k)}(s) \leq \mathrm{K}_{M}^{k}(s) + c.$$

Definition 3.3. The *monotonic operator complexity* of a string s is $\mathrm{KM}(s) = \min\{|t| \mid \mathrm{UM}(t) = s\}$.

Theorem 3.4 (Levin [16], Schnorr [25]). *Let $x \in \{0,1\}^{\infty}$. Then x is random in the sense of Martin-Löf iff* $(\exists c)(\forall n)\ \mathrm{KM}(x^{n}) \geqslant n - c$.

In addition, this characterization can be generalized to the definition of finite random strings and that of infinite random sequences with respect to arbitrary computable probability measures.

In this section, we follow this approach to define an infinite pseudorandom sequence to be the one with high monotonic operator complexity with respect to polynomial time (or, space) bound.

Definition 3.5. Let k be an integer. The *time- and space-bounded monotonic operator complexities* of a string s are

$$\mathrm{KMT}^k(s) = \min\{|t| \,|\, \mathrm{UM}(t) \text{ prints } s \text{ in } k \text{ moves}\}$$

and

$$\mathrm{KMS}^k(s) = \min\{|t| \,|\, \mathrm{UM}(t) \text{ prints } s \text{ using } k \text{ cells}\},$$

respectively.

Following the convention of Section 2, for any function f, we write $\mathrm{KMT}^f(s)$ and $\mathrm{KMS}^f(s)$ to denote $\mathrm{KMT}^{f(|s|)}(s)$ and $\mathrm{KMS}^{f(|s|)}(s)$, respectively.

Definition 3.6

(a) $\mathrm{PR2} = \{x \in \{0,1\}^\infty \,|\, (\forall \text{polynomial } p)(\exists k)(\forall^\infty n)\ \mathrm{KMT}^p(x^n) \geq n - (\log n)^k\}$.

(b) $\mathrm{PSR2} = \{x \in \{0,1\}^\infty \,|\, (\forall \text{polynomial } p)(\exists k)(\forall^\infty n)\ \mathrm{KMS}^p(x^n) \geq n - (\log n)^k\}$.

The following theorem is a polynomial space analogue of Theorem 3.4.

Theorem 3.7. $\mathrm{PSR1} = \mathrm{PSR2}$.

Proof. ($\mathrm{PSR1} \subseteq \mathrm{PSR2}$): For a fixed polynomial p, define a function $f: \{0,1\}^* \to \mathbb{N}$ as follows:

$$f(s) = \max\{m - \mathrm{KMS}^p(s^m) \,|\, m \leq |s|\}.$$

We claim that f is an FPSPACE-test.

First, we note that, in order to compute $f(s)$, we need only to simulate $\mathrm{UM}(t)$, for all t of length $|t| \leq |s|$, each using $p(|s|)$ cells. So, it is clear that $f \in \mathrm{FPSPACE}$.

Next, if s is an initial segment of t, and if $f(s) = m - \mathrm{KMS}^p(s^m)$ for some $m \leq |s|$, then $f(t) \geq m - \mathrm{KMS}^p(t^m) = f(s)$. So, f satisfies the sequential property.

Finally, for each k, we define $A_k = \{x \in \{0,1\}^\infty \,|\, (\exists n) f(x^n) \geq k\}$, and check that $\Pr(A_k) \leq 2^{-k}$. Assume, by way of contradiction, $\Pr(A_k) > 2^{-k}$. Then we can find a finite number of strings $s_1, \ldots, s_h$ such that

(1) for any $i, j \leq h$, s_i is not a prefix of s_j if $i \neq j$;

(2) $\sum_{i=1}^h 2^{-|s_i|} > 2^{-k}$, and

(3) $\mathrm{KMS}^p(s_i) \leq |s_i| - k$, for all $i = 1, \ldots, h$.

Property (3) implies that there are $t_1, \ldots, t_h$ such that $\mathrm{UM}^p(t_i) = s_i$ and $|t_i| \leq |s_i| - k$, for $i = 1, \ldots, h$. Since UM is a monotonic operator, property (1) implies that t_i is not a prefix of t_j if $i \neq j$. However, $\sum_{i=1}^h 2^{-|t_i|} \geq \sum_{i=1}^h 2^{-|s_i|+k} > 1$, and it implies that some t_i is a prefix of some t_j, $i \neq j$, and hence gives a contradiction. So, we have proved that $\Pr(A_k) \leq 2^{-k}$, and also the claim.

Now, assume that $x \in \mathrm{PSR1}$. Then there is an integer k such that $(\forall^\infty n)\ f(x^n) \leq (\log n)^k$. But, from the definition of f, this exactly means that $(\forall^\infty n)\ \mathrm{KMS}^p(x^n) \geq n - (\log n)^k$. Since this holds for arbitrary polynomials p, we have $x \in \mathrm{PSR2}$.

(PSR2 $\subseteq$ PSR1): Assume that $x \notin$ PSR1. Then, there exists an FPSPACE-test f such that

$$(\forall k)(\exists^{\infty} n)\, f(x^n) > (\log n)^k.$$

We need to find a polynomial space-bounded monotonic operator M such that

$$(\forall k)(\exists^{\infty} n)(\exists t_n)\, M(t_n) = x^n \text{ and } |t_n| \leq n - (\log n)^k.$$

For fixed n and k, the set $B_{n,k} = \{w \mid |w| = n, f(w) > (\log n)^{k+1}\}$ has size $\# B_{n,k} \leq 2^{n-(\log n)^{k+1}}$. Recall that our pairing function $\langle\, ,\rangle$ is defined as follows: $\langle s, t\rangle = d(s)01t$, where $d(s)$ is the string obtained by doubling each bit of s. So, $|\langle s, t\rangle| = 2|s| + |t| + 2$. We now define a TM M as follows:

(1) $\text{domain}(M) = \{(n, t) \mid |t| = \lceil n - (\log n)^{k+1} \rceil\}$;

(2) on input $\langle n, t\rangle$, ignoring the leading 0's of t and using t as an integer, M outputs the tth string in $B_{n,k}$ (if $t > \# B_{n,k}$, then M outputs the first string in $B_{n,k}$).

It is easy to see that M operates in polynomial space because $f \in$ FPSPACE. We note that if $\langle n, t\rangle, \langle m, u\rangle \in \text{domain}(M)$ and $\langle n, t\rangle$ is an initial segment of $\langle m, u\rangle$, then we must have $n = m$ because the first occurrence of "01" in $\langle n, t\rangle = d(n)01t$ and $\langle m, u\rangle = d(m)01u$ determines the length of n and m. But, then, $|t| = |u|$ and hence, $t = u$. This implies that M is a monotonic operator. Furthermore, for each n with $f(x^n) > (\log n)^{k+1}$, $M(\langle n, t\rangle) = x^n$ for some t with $|\langle n, t\rangle| \leq n - (\log n)^{k+1} + 2|n|$. So, by the universality of UM, we have

$$(\exists^{\infty} n)\, \text{KMS}^p(x^n) \leq n - (\log n)^{k+1} + 2 \log n + c$$

for some polynomial p and some constant c (since $|n| = \lfloor \log n \rfloor + 1$). This shows that $x \notin$ PSR2 and completes the proof. □

For the classes PR1 and PR2, we are not able to show their equivalence. We observe that if P = PSPACE, then PR1 = PSR1 and PR2 = PSR2 and therefore they are equivalent. In fact, they are equivalent under the weaker assumption FP = #P.

Observation 3.8. (a) PR2 $\subseteq$ PR1 *if* FP = #P.

(b) PR1 $\subseteq$ PR2 *if* P = NP.

Proof. (a): Consider the proof of PSR2 $\subseteq$ PSR1 of Theorem 3.7. The function $g(n, k, s) = \#\{t \in B_{n,k} \mid t \leq s\}$ is in #P if $f \in$ FP. So, if FP = #P, then $g \in$ FP and hence, $M(\langle n, t\rangle)$ can be computed in time $q(n)$ for some polynomial q by binary searching for s such that $g(n, k, s) = t$.

(b): Consider the proof of PSR1 $\subseteq$ PSR2 of Theorem 3.7 and define, for each polynomial p, a function

$$f(s) = \max\{m - \text{KMT}^p(s^m) \mid m \leq |s|\}.$$

Now, let $Q = \{\langle s, i\rangle \mid (\exists u, |u| \leq i)$ UM(u) prints s in $p(|s|)$ moves$\}$. Then, it is clear

that $Q \in \mathrm{NP}$. If $\mathrm{P} = \mathrm{NP}$, then $Q \in \mathrm{P}$, and so $f \in \mathrm{FP}$ because $f(s) = \max\{j - i \mid \langle s^j, i\rangle \in Q\}$. □

We do not know whether the conditions $\mathrm{FP} = \#\mathrm{P}$ and $\mathrm{P} = \mathrm{NP}$ are necessary. It appears to be an interesting open question. In particular, whether $\mathrm{P} = \mathrm{NP}$ is necessary for $\mathrm{PR1} \subseteq \mathrm{PR2}$ is closely related to the following question: is the set

$$A = \{\langle s, 0^i, 0^j\rangle \mid (\exists t, |t| \leq i)\ U(t) \text{ prints } s \text{ in } j \text{ moves}\}$$

NP-complete? Hartmanis [12] has pointed out the importance of this question in connection with studies of generalized Kolmogorov complexity.

Another interesting question related to the equivalence of the two definitions is that in case they are not equivalent, which one is a better definition of pseudorandomness. It appears to need further studies before we can give a satisfactory answer to this question.

We use the above proofs to show the existence of a double exponential time computable pseudorandom sequence x in PR1.

Corollary 3.9. (a) *There is a sequence* $x \in \mathrm{PSR1}$ *that is computable in space* $2^{2^{O(n)}}$.

(b) *There is a sequence* $x \in \mathrm{PR1}$ *that is computable in time* $2^{2^{O(n)}}$

Proof. (a): By Theorem 3.7, if x satisfies

$$(\forall^\infty n)\ \mathrm{KMS}^{2^n}(x^n) > n - \lfloor 4 \log n \rfloor,$$

then $x \in \mathrm{PSR1}$. The KS-complexity version of Corollary 2.11 showed the existence of an x computable in space $2^{2^{O(n)}}$ which satisfies

$$(\forall^\infty n)\ \mathrm{KS}^{2^n}(x^n \mid n) > n - \lfloor 3 \log n \rfloor.$$

Since $\mathrm{KMS}^{2^n}(s) \geq \mathrm{KS}^{2^n}(s \mid n) - \log n$ if $|s| = n$, part (a) follows.

(b): In the proofs of Theorem 3.7 and Observation 3.2(a), we defined, for each test f and each integer k, a set $B_{n,k}$ and a monotonic operator M such that $M(\langle n, t\rangle)$ prints the tth string in $B_{n,k}$. Note that the tth string in $B_{n,k}$ can be found by simulating $f(w)$ for all w of length n, and so $M(\langle n, t\rangle)$ runs in time $2^n \cdot p(n)$ if f runs in time $p(n)$. Furthermore, if $x \notin \mathrm{PR1}$, then x^n occurs among the first $2^{n - \lfloor \log n \rfloor^{k+1}}$ strings of $B_{n,k}$ for infinitely many n. In other words, if x satisfies

$$(\forall^\infty n)\ \mathrm{KMT}^{2^{2^n}}(x^n) > n - \lfloor 4 \log n \rfloor,$$

then $x \in \mathrm{PR1}$. Now, part (b) follows from Corollary 2.11 and the fact that

$$\mathrm{KMT}^{2^{2^n}}(x^n) \geq \mathrm{KT}^{2^{2^n}}(x^n \mid n) - \log n. \qquad \square$$

The questions of the existence of exponential space computable sequences in PSR1 and the existence of exponential time computable sequences in PR1 remain

open. It seems that Wilber's technique [31] of constructing exponential time computable P-random sets cannot apply to our setting, because our definition of pseudorandomness is strictly stronger than his definition (as will be shown in the next section).

4. Relative frequency and pseudorandom sequences

Based on Von Mises's notion of 'collectives', we may give a third definition of infinite pseudorandom sequences $x \in \{0,1\}^\infty$ as follows (cf. [7, 9, 21, 29, 30, 31]).

Definition 4.1. A sequence $x \in \{0,1\}^\infty$ is in PR3 (PSR3) if, for any polynomial time (space, respectively) computable function $f: \{0,1\}^* \to \{0,1\}$,

$$\lim_{n\to\infty} \#\{k \mid k \leq n, f(x^{k-1}) = x^k\}/n = \tfrac{1}{2}.$$

In other words, x is in PR3 if, for any polynomial time algorithm which predicts the nth bit $x(n)$ from the previous $n-1$ bits x^{n-1}, the probability of success is no better than tossing an unbiased coin.

In this section we will show that this definition of pseudorandomness is strictly weaker than the definitions PR1 and PR2. Thus the definition given above is probably not adequate in the sense that there exists a sequence x that is pseudorandom by this definition but for which we may find a polynomial time testing function f such that $f(x^n) \geq n^{1/3}$ infinitely often. Our results here agree with the analogous results on random sequences and hence serve as another justification for our definitions of PR1 and PR2.

We first state several lemmas. The first lemma gives an information-theoretic bound and is due to Chaitin [4].

Lemma 4.2. *Let ε be a real number between 0 and $\frac{1}{2}$. Then, for any $n \geq 1$,*

$$\log\binom{n}{\lfloor \frac{1}{2}n - \varepsilon n \rfloor} \leq n \cdot H(\varepsilon) + c$$

for some constant c, where $H(\varepsilon) = -(\frac{1}{2}+\varepsilon)\cdot\log(\frac{1}{2}+\varepsilon) - (\frac{1}{2}-\varepsilon)\cdot\log(\frac{1}{2}-\varepsilon)$.

Next we define an order $\prec$ on $\{0,1\}^n$. For $s \in \{0,1\}^n$, let $u(s)$ = the number of 1's in $s = \#\{k \mid k \leq n, s(k) = 1\}$. Then, $\prec$ is defined as follows: $s \prec t$ if $(u(s) < u(t))$ or $(u(s) = u(t)$ and s precedes t under the lexicographic order).

Lemma 4.3. *The function g, defined by $g(n, m)$ = the m-th string in $\{0,1\}^n$ under order $<$, is computable in time $p(n)$ for some polynomial p.*

Proof. First, we can determine the number of 1's in $g(n, m)$ as $k = \min\{j \mid \sum_{i=0}^{j} \binom{n}{i} \geq m\}$. Thus, we need only to find the m'th string in $\{0,1\}^n$ which has k 1's and $n-k$ 0's, where $m' = m - \sum_{i=0}^{k-1} \binom{n}{i}$. Let us call it $h(n, k, m')$.

We note that the leftmost 1 of $h(n, k, m')$ can be determined as the $(n-i+1)$st bit, where $i = \min\{j \mid \binom{j}{k} \geq m'\}$. (We call the leftmost bit the first bit.) Now, after determining the leftmost 1 of $h(n, k, m')$, the rest of the string $h(n, k, m')$, between the $(n-i+2)$nd and nth bits, is just the m''th string of length $i-1$ and having $k-1$ 1's, where $m'' = m' - \binom{i-1}{k}$. That is, $h(n, k, m') = 0^{n-i}1h(i-1, k-1, m'')$. By repeating the computation of the leftmost 1 in $h(i-1, k-1, m'')$, we can get all 1's in $g(n, m) = h(n, k, m')$ in k iterations.

It is clear that the above computation runs in time $O(n^2)$. □

Lemma 4.4. *The function g', defined by $g'(n, t)$ = the unique number m such that $g(n, m) = t$, is computable in time $p(n)$ for some polynomial p.*

Proof. The function g' can be computed by reversing the algorithm in the proof of Lemma 4.3. We omit it here. □

Theorem 4.5. (a) PR1 ⊆ PR3.
(b) PR2 ⊆ PR3.

Proof. The proof is based on the following simple observation: there are relatively few strings of length n with more than $\frac{1}{2}n + \varepsilon n$ 0's, even for a small ε. Therefore, these strings (i.e., strings not in PR3) can be rejected by FP-tests, and can be computed easily from short TM programs.

For each function $f: \{0,1\}^* \to \{0,1\}$ and string s of length n, define

$$w(f, s) = \#\{k \mid k \leq n, f(s^{k-1}) = s(k)\}.$$

Assume that $x \notin$ PR3. Then there is a function $f \in$ FP such that $\liminf_{n\to\infty} w(f, x^n)/n \neq \frac{1}{2}$. Without loss of generality, we assume that $\liminf w(f, x^n)/n < \frac{1}{2}$ (otherwise, just replace f by $f'(s) = 1 - f(s)$). Thus, we have $(\exists \varepsilon > 0)(\exists^\infty n)$ $w(f, x^n) \leq \frac{1}{2}n - \varepsilon n$. We claim that, for each n and k, there are exactly $\binom{n}{k}$ many strings s of length n such that $w(f, s) = k$.

Proof of claim: If s has length $|s| = n$ and satisfies the condition $w(f, s) = k$, then there are k positions $i_1, \ldots, i_k$ between 1 and n such that, for each $i \leq n$, $f(s^{i-1}) = s(i)$ iff $i \in \{i_1, \ldots, i_k\}$. Now, if s and t, both of length n, determine the same set of positions $\{i_1, \ldots, i_k\}$, then, by a simple inductive proof, we have $s = t$. So, each string s such that $w(f, s) = k$ uniquely determines a set $\{i_1, \ldots, i_k\}$, and there are exactly $\binom{n}{k}$ many such strings.

Therefore, there are $\leq \sum_{k=0}^{\lfloor n/2-\varepsilon n\rfloor} \binom{n}{k}$ many strings s such that $w(f, s) \leq \frac{1}{2}n - \varepsilon n$. By Lemma 4.2,

$$\log \sum_{k=0}^{\lfloor n/2-\varepsilon n\rfloor} \binom{n}{k} \leq \log\left(n \cdot \binom{n}{\lfloor \frac{1}{2}n - \varepsilon n\rfloor}\right) \leq n \cdot H(\varepsilon) + \log n + c$$

for some constant c. Since $H(\varepsilon) < 1$ for all $\varepsilon > 0$, there is a $\delta > 0$ such that

$$\log \sum_{k=0}^{\lfloor n/2-\varepsilon n\rfloor} \binom{n}{k} \leq (1-\delta)n.$$

Now, for part (a), we consider the following algorithm M. Let K be the least integer such that $K \geq \sum_{i=1}^{\infty} 2^{-(\delta i/2)}$.

Algorithm *M*.
input: s {Let $n := |s|$}.
begin
  for $i := 1$ **to** n **do**
    if $f(s^{i-1}) = s(i)$ **then** $t(i) := 1$ **else** $t(i) := 0$;
  {denote this t as $t(f, s)$}
  for $i := 1$ **to** n **do**
    $k(i) := \lfloor (1 - \frac{1}{2}\delta)i \rfloor - \lceil \log(g'(i, t^i)) \rceil - K$;
    {g' is defined in Lemma 4.4}
  output $(\max\{k(i) \mid i \leq n\})$
end.

The algorithm M computes a function $h : \{0, 1\}^* \to \mathbb{N}$. We claim that h is an FP-test. First, by Lemma 4.4, $g' \in \mathrm{FP}$ and hence $h \in \mathrm{FP}$. To see that h is a test, we note that, for each function f, the function $\lambda s[t(f, s)]$ is a one-one mapping from $\{0, 1\}^n$ to $\{0, 1\}^n$. Thus,

$$\begin{aligned}
\#\{s \mid |s| = n, h(s) \geq m\} &= \#\{t \mid |t| = n, (\exists i \leq n)\, k(i) \geq m\} \\
&\leq \sum_{i=1}^{n} \#\{t \mid |t| = n, g'(i, t^i) \leq 2^{(1-\delta/2)i-m-K}\} \\
&\leq \sum_{i=1}^{n} 2^{n-\delta i/2-m-K} \leq 2^{n-m-K} \cdot \sum_{i=1}^{n} 2^{-(\delta i/2)} \\
&\leq 2^{n-m-K} \cdot K \leq 2^{n-m}
\end{aligned}$$

because for each j there are only 2^j many strings s of length i such that $g'(i, s) \leq 2^j$. So, h is a test. Finally, it is obvious from the definition of h that h is a sequential test.

Now, for each x^n such that $w(f, x^n) \leq \frac{1}{2}n - \varepsilon n$, we have $\log(g'(n, x^n)) \leq (1-\delta)n$, and so $h(x^n) \geq \frac{1}{2}\delta n - K$. Therefore, we have

$$(\exists \delta > 0)(\exists c)(\exists^{\infty} n)\; h(x^n) \geq \tfrac{1}{2}\delta n - c.$$

It follows that $x \notin \mathrm{PR1}$, and part (a) is proven.

For part (b), consider the following algorithm M'.

Algorithm M'.
input: $u = \langle n, t\rangle$.
begin
  if $|t| \neq \lceil (1-\delta)n \rceil$ **then** undefined
  else begin
    $m :=$ the integer represented by t, ignoring leading zeros;
    $s := g(n, m)$; {g is defined in Lemma 4.3}
    for $i := 1$ **to** n **do**
      if $s(i) = 1$ **then** $v(i) := f(v^{i-1})$ **else** $v(i) := 1 - f(v^{i-1})$;
    output (v^n)
  end
end.

We note that, by our coding scheme for the pairing function $\langle , \rangle$, for any u_1, $u_2 \in \text{domain}(M')$, u_1 is not a prefix of u_2 unless $u_1 = u_2$. So, M' is a monotonic operator. Furthermore, $M'(\langle n, t\rangle)$ outputs the tth string in $\{0,1\}^n$ under an order $<'$ that satisfies the property that $w(f, s_1) < w(f, s_2)$ implies $s_1 <' s_2$. Thus, for infinitely many n, x^n is computed by $M'(\langle n, t\rangle)$ with $|t| \leq (1-\delta)n$. Or, there is a polynomial p such that

$$(\exists^\infty n)\ \mathrm{KMT}^p(x^n) \leq (1-\delta)n + \log n.$$

It follows that $x \notin \mathrm{PR2}$. □

Theorem 4.6. (a) $\mathrm{PR1} \neq \mathrm{PR3}$.
(b) $\mathrm{PR2} \neq \mathrm{PR3}$.

Proof. Let $u(s)$ be the number of 0's in s, i.e., $u(s) = \#\{k \mid k \leq |s|, s(k) = 0\}$. We will construct a sequence x with the following property:

$$(\forall^\infty n) u(x^n) \approx n(2^{-1} + (2\log n)^{-1}).$$

Since $\lim_{n\to\infty} (2\log n)^{-1} = 0$, the relative frequency of 0's in x is $\frac{1}{2}$ and hence $x \in \mathrm{PR3}$. However, $n/(2\log n)$ many extra 0's in x^n allow us to find a test to reject x^n, and to find a short monotonic TM program to compute x^n.

Let $y \in \{0,1\}^\infty$ be an arbitrary random sequence (in the sense of Definition 1.3). Then $y \in \mathrm{PR1}$ (and $y \in \mathrm{PR2}$), and so $y \in \mathrm{PR3}$. That is, $\lim_{n\to\infty} u(y^n)/n = \frac{1}{2}$. Without loss of generality, assume that there is an infinite sequence $\{n_0, n_1, \ldots\}$ such that $u(y^{n_i}) \geq \frac{1}{2} n_i$. Now, we insert some 0's into y to form a new sequence $x \in \{0,1\}^\infty$: insert two 0's between $y(4)$ and $y(5)$; and insert, for each $n > 2$, $\lceil 2^n/n \rceil - \lceil 2^{n-1}/(n-1) \rceil$ many 0's between $y(2^n)$ and $y(2^n+1)$. We claim two properties of x:

(1) $x \in \mathrm{PR3}$, and
(2) $(\exists^\infty n)\ u(x^n) \geq n(2^{-1} + (2\log n)^{-1})$.

Proof of claim (1): Assume that $x \notin \mathrm{PR3}$, and so there is a function $f \in \mathrm{FP}$ and an $\varepsilon > 0$ such that $(\exists^\infty n)\ w(f, x^n) \geq \frac{1}{2}n + \varepsilon n$. (Recall that $w(f, x^n) = \#\{k \mid k \leq n, f(x^{k-1}) = x(k)\}$.) Define a function f' as follows: On input s with $|s| = n-1$, first insert two 0's between $s(4)$ and $s(5)$ and then insert, for each $k > 2$ such that $2^k \leq n-1$, $\lceil 2^k/k \rceil - \lceil 2^{k-1}/(k-1) \rceil$ 0's between $s(2^k)$ and $s(2^k+1)$; call the new string t; then, output $f(t)$.

Now consider $w(f', y_n)$. Let $k = \lfloor \log n \rfloor$. Then,

$$w(f', y^n) \geq w(f, x^{n+\lceil 2^k/k \rceil}) - \lceil 2^k/k \rceil.$$

So, we have $(\exists^\infty n)\ w(f', y^n) \geq \frac{1}{2}n + \varepsilon n - n/\log n$; or,

$$\limsup_{n\to\infty} \frac{w(f', y^n)}{n} \geq \lim_{n\to\infty} \left(\frac{1}{2} + \varepsilon - \frac{1}{\log n}\right) = \frac{1}{2} + \varepsilon.$$

This contradicts the assumption that $y \in \mathrm{PR3}$. So, x must be in PR3.

Proof of claim (2): For each $n \in \{n_0, n_1, \ldots\}$, we have $u(y^n) \geq \frac{1}{2}n$. Let $k_n = 2^{\lfloor \log n \rfloor}/\lfloor \log n \rfloor$. Then, for each $n \in \{n_0, n_1, \ldots\}$,

$$u(x^{n+k_n}) \geq \tfrac{1}{2}n + k_n \geq \tfrac{1}{2}n + n/(2 \log n).$$

So, $(\exists^\infty n)\ u(x^n) \geq n(2^{-1} + (2 \log n)^{-1})$.

Next we recall that in the proof of Theorem 4.5, we have shown that there are $\leq 2^{nH(\varepsilon)+\log n+c}$ many strings s of length n such that $u(s) \geq \frac{1}{2}n + \varepsilon n$. Now replace ε by $(2 \log n)^{-1}$. We observe that $\lim_{r\to 0} 2^{1/(4r)} \cdot (1 - H(r)) = \infty$. (The second derivative of $2^{-1/(4r)}$ has a limit 0 as r tends to 0, and the second derivative of $1 - H(r)$ has a limit 4 as r tends to 0.) So, $(\forall^\infty n)\ H((2 \log n)^{-1}) \leq 1 - n^{-1/2}$. This implies that there are $\leq 2^{n-n^{1/2}+\log n+c}$ many strings in the set $S = \{s \mid |s| = n, u(s) \geq n \cdot (2^{-1} + (2 \log n)^{-1})\}$.

For part (a), we consider the function h which, on input s with $|s| = n$, outputs

$$\max\{i - \lceil i^{1/3} \rceil - \lceil \log(g'(i, s^i)) \rceil - K \mid i \leq n\},$$

where $g'(i, s^i)$ is the function defined in Lemma 4.4 and $K \geq \sum_{i=1}^{\infty} 2^{-i^{1/3}}$. Then, similar to the proof of Theorem 4.5(a), we can prove that h is an FP-test. (The only thing that needs to be checked is that

$$\#\{s \mid |s| = n, h(s) \geq m\} \leq \sum_{i=1}^{n} \#\{s \mid |s| = n, i - i^{1/3} - \log(g'(i, s^i)) - K \geq m\}$$

$$\leq \sum_{i=1}^{n} 2^{n-i} \cdot 2^{i-i^{1/3}-m-K} \leq 2^{n-m}.)$$

Now, if $u(x^n) \geq n \cdot (2^{-1} + (2 \log n)^{-1})$, then $g'(n, t) \leq 2^{n-n^{1/2}+\log n+c}$ for some constant c. This implies that

$$(\exists c')(\exists^\infty n)\ h(x^n) \geq n^{1/2} - n^{1/3} - \log n - c',$$

and hence $x \notin \mathrm{PR1}$, and part (a) is proven.

For part (b), consider algorithm M'' that operates as follows: on input $u = \langle n, t \rangle$, if $|t| \neq \lceil n - n^{1/2} + \log n + c \rceil$, then $M''(u)$ is undefined; otherwise, let m be the integer represented by t, ignoring leading zeros, and output $g(n, m)$.

Similar to Algorithm M' in the proof of Theorem 4.5(b), M'' is a monotonic TM. Also, by the estimation of the size of the set S, there is a constant c such that

$$(\exists^\infty n)(\exists t, |t| \leq n - n^{1/2} + \log n + c)\ M''(\langle n, t\rangle) = x^n.$$

As a consequence, there is a polynomial p such that

$$(\exists^\infty n)\ \mathrm{KMT}^p(x^n) \leq n - n^{1/3}.$$

and so $x \notin \mathrm{PR2}$. □

Corollary 4.7. $\mathrm{PSR1} \subsetneq \mathrm{PSR3}$.

Proof. The proofs of Theorems 4.5(a) and 4.6(a) can be carried over for the classes PSR1 and PSR3. □

We remark that the above proof shows more than just $\mathrm{PR1} \neq \mathrm{PR3}$. It actually shows that the relative frequency of 0's in a pseudorandom sequence $x \in \mathrm{PR1}$ converges to $\frac{1}{2}$ faster than the function $\lambda n[1/\log n]$ to 0. This suggests a stronger relative frequency requirement for pseudorandom sequences. Namely, a sequence $x \in \{0, 1\}^\infty$ cannot be considered as pseudorandom unless, for all $f \in \mathrm{FP}$, the relative frequency $\lambda n[w(f, x^n)/n]$ converges to $\frac{1}{2}$ at least as fast as the function $\lambda n[1/(k \cdot \log n)]$ for any constant k.

5. Concluding remarks

A pseudorandom sequence may be defined in many different forms, depending upon its application. In this paper we proposed two strong definitions of pseudorandom sequences and compared them with a third, weaker definition. The strong definitions are not intended to be the criteria for judging random-number generators. Instead, our purpose is to get better understanding of the structural relations between the notion of pseudorandomness and the notion of complexity.

One of the main questions left open here is whether there exists an exponential time computable sequence x that has high polynomial time-bounded KT-complexity almost everywhere (cf. Corollary 2.11). The priority diagonalization technique of Meyer and McCreight [21] does not seem applicable to this question. Another interesting question is to find a necessary and sufficient condition for the relation $\mathrm{PR1} = \mathrm{PR2}$ (cf. Observation 3.8). These questions ask, in general, what the relation is between computational complexity and program size complexity, and deserve further investigation.

Acknowledgment

The author thanks the referee for the helpful reports. In particular, the current formulation of the definitions PR2 and PSR2 are suggested by the referee. The

author is also grateful to Professor Juris Hartmanis and Dr. Osamu Watanabe for their encouragement.

References

[1] M. Blum, On the size of machines, *Inform. and Control* **11** (1967) 257-265.
[2] L. Blum, M. Blum and M. Shub, A simple secure pseudo-random number generator, *SIAM J. Comput.* **15** (1986) 364-383.
[3] M. Blum and S. Micali, How to generate cryptographically strong sequences of pseudo random bits, *Proc. 23rd IEEE Symp. on Foundations of Computer Science* (1982) 112-117.
[4] G.J. Chaitin, On the length of programs for computing finite binary sequences, *J. Assoc. Comput. Mach.* **13** (1966) 547-569.
[5] G.J. Chaitin, On the length of programs for computing finite binary sequences: statistical considerations, *J. Assoc. Comput. Mach.* **16** (1969) 145-159.
[6] G.J. Chaitin, A theory of program size formally identical to information theory, *J. Assoc. Comput. Mach.* **22** (1975) 329-340.
[7] A. Church, On the concept of random sequence, *Bull. Amer. Math. Soc.* **46** (1940) 130-135.
[8] R.P. Daley, Noncomplex sequences: characterizations and examples, *J. Symbolic Logic* **41** (1976) 626-638.
[9] R.A. Di Paola, Random sets in subrecursive hierarchies, *J. Assoc. Comput. Mach.* **16** (1969) 621-630.
[10] M. Garey and D. Johnson, *Computers and Intractability* (Freeman, San Francisco, CA, 1979).
[11] J. Hartmanis, Generalized Kolmogorov complexity and the structure of feasible computations, *Proc. 24th IEEE Symp. on Foundations of Computer Science* (1983) 439-445.
[12] J. Hartmanis, Personal communication, 1985.
[13] R.M. Karp and R.J. Lipton, Some connections between nonuniform and uniform complexity classes, *Proc. 12th ACM Symp. on Theory of Computing* (1980) 302-309.
[14] A.N. Kolmogorov, Three approaches to the quantitative definition of information, *Problems Inform. Transmission* **1** (1965) 1-7.
[15] L. Levin, Universal sorting problems, *Problems Inform. Transmission* **9** (1973) 265-266.
[16] L. Levin, On the notion of a random sequence, *Soviet Math. Dokl.* **14** (1973) 1413-1416.
[17] D.W. Loveland, A variant of the Kolmogorov concept of complexity, *Inform. and Control* **15** (1969) 510-526.
[18] P. Martin-Löf, On the definition of random sequences, *Inform. and Control* **9** (1966) 602-619.
[19] P. Martin-Löf, The literature on Von Mises' Kollektivs revisited, *Theoria* **35** (1969) 12-37.
[20] P. Martin-Löf, Complexity oscillations in infinite binary sequences, *Z. Wahrsch. Verw. Gebiete* **19** (1971) 225-230.
[21] A.R. Meyer and E.M. McCreight, Computationally complex and pseudorandom zero- one valued functions, in: Z. Kohavi and A. Paz, eds., *Theory of Machines and Computations* (Academic Press, New York/London, 1971) 19-42.
[22] N. Pippenger, On simultaneous resource bounds, *Proc. 20th IEEE Symp. on Foundations of Computer Science* (1979) 307-311.
[23] J. Plumstead, Inferring a sequence generated by a linear congruence, *Proc. 23rd IEEE Symp. on Foundations of Computer Science* (1982) 153-159.
[24] C.P. Schnorr, *Zufälligkeit und Wahrscheinlichkeit*, Lecture Notes in Mathematics **218** (Springer, New York, 1971).
[25] C.P. Schnorr, Process complexity and effective random tests, *J. Comput. System Sci.* **7** (1973) 376-388.
[26] C.P. Schnorr and P. Fuchs, General random sequences and learnable sequences, *J. Symbolic Logic* **42** (1977) 329-340.
[27] A. Shamir, On the generation of cryptographically strong pseudorandom sequences, *Proc. 8th Internat. Coll. on Automata, Languages, and Programming* (1981).
[28] M. Sipser, A complexity theoretic approach to randomness, *Proc. 15th ACM Symp. on Theory of Computing* (1983) 330-335.
[29] R. Von Mises, Grundlagen der Wahrscheinlichkeitsrechnung, *Math. Z.* **5** (1919) 52-99.

[30] A. Wald, Die Widerspruchsfreiheit des Kollektivbegriffes der Wahrscheinlichkeitsrechnung, *Ergebnisse eines Math. Kolloquiums* **8** (1937) 38–72.

[31] R. Wilber, Randomness and the density of hard problems, *Proc. 24th IEEE Symp. on Foundations of Computer Science* (1983) 335–342.

[32] A.C. Yao, Theory and applications of trapdoor functions, *Proc. 23rd IEEE Symp. on Foundations of Computer Science* (1982) 80–91.

[33] A.K. Zvonkin and L.A. Levin, The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms, *Russian Math. Survey* **25** (1970) 83–124.