Random languages for non-uniform complexity classes*

Martin Mundhenk Rainer Schuler

Universität Ulm
Abt. Theoretische Informatik
D-7900 Ulm
West Germany

Abstract

A language A is considered to be random for a class $\mathcal{C}$ if for every language B in $\mathcal{C}$ the fraction of the strings where A and B coincide is approximately 1/2.

We show that there exist languages in $\mathbf{DSPACE}(f(n))$ which are random for the non-uniform class $\mathbf{DSPACE}(g(n))/h(n)$, where n, $g(n)$ and $h(n)$ are in $o(f(n))$. Non-uniform complexity classes were introduced by Karp and Lipton [Karp and Lipton 1980] and allow an advice string that depends only on the length of the input as additional information. This paper extends a result by Wilber [Wilber 1983] who proved bounds for the existence of random languages for (uniform) time and space classes. Huynh [Huynh 1987] provides a result for the special case of **P/poly**-random languages in **EXPSPACE**.

Here we explore a different method using strings with high generalized Kolmogorov complexity [Hartmanis 1983]. A characterization of the non-uniform space classes in terms of Kolmogorov complexity is given. This generalizes a result of [Balcázar, Díaz, and Gabarró 1987b] where characterizations of the class **PSPACE/poly** are given.

*This work was supported by the Deutsche Forschungsgemeinschaft, grant No. Scho 3/2.

1 Introduction

In 1965 Hartmanis, Lewis, and Stearns [Hartmanis, Lewis, Stearns 1965] proved a strong hierarchy of space bounded complexity classes. They showed that for any two space constructible functions f and g which are $\Omega(\log(n))$, if $g(n) = o(f(n))$ then $\mathbf{DSPACE}(g(n))$ is a proper subset of $\mathbf{DSPACE}(f(n))$. That is, there exist languages in $\mathbf{DSPACE}(f(n))$ such that every Turing-machine which accepts the language needs more than $g(n)$ space for infinitely many inputs, or equivalently, if we allow the machine to give wrong answers on some inputs we get infinitely many wrong answers. Thus we get that for all languages $B \in \mathbf{DSPACE}(g(n))$: $|\{x \mid x \in A \Leftrightarrow x \notin B\}| = \infty$.

Wilber considerably strengthened this result by showing that this hierarchy of space bounded classes also holds under the stronger requirement that there exist languages in $\mathbf{DSPACE}(f(n))$ such that every $O(g(n))$ space bounded Turing machine is wrong on approximately half of all strings. This idea has been discussed by Meyer and McCreight [Meyer and McCreight 1971] who showed that for every recursive function f there exists a language which can only be approximated by an f-space bounded computation with error probability 1/2. They call such a language random for the class of f-space bounded computations.

In this paper we are concerned with the randomness of languages with respect to non-uniform complexity classes. Non-uniform complexity classes were defined by Karp and Lipton [Karp and Lipton 1980]. They allow (as opposed to uniform complexity classes) an advice string for each input length as additional input. As a main result we prove that under essentially the same hypothesis as the space hierarchy theorem the class $\mathbf{DSPACE}(f(n))$ contains random languages even if the $O(g(n))$ space bounded Turing machine is allowed $o(f(n))$ additional bits of nonuniform advice. We define certain $p(n)$-periodic languages in $\mathbf{DSPACE}(f(n))$ and show that they are random for the class $\mathbf{DSPACE}(g(n))/h(n)$ when $h(n)$ is $o(p(n))$ and $o(f(n))$. A language A is $p(n)$-periodic if for each n the characteristic string of A for strings of length n is a prefix of y^k for some y, $|y| = p(n)$, and some $k \geq 1$. The strings y used here to define the language A are strings with an amount of generalized Kolmogorov complexity which cannot be computed in $\mathbf{DSPACE}(g(n))/h(n)$. Generalized Kolmogorov complexity was introduced by Hartmanis [Hartmanis 1983], and it measures how much a string can be compressed such that it can be restored in a given time or a given space. It has already been applied in various areas [Li and Vitanyi 1988, Kolmogorov and Uspenskii 1987].

In [Huynh 1987] Huynh already showed the existence of $\mathbf{P/poly}$-random languages in $\mathbf{EXPSPACE}$. As an interesting corollary from our proof technique it follows that in any (reasonable) non-uniform complexity class there exist languages that are random for all uniform complexity classes. Therefore, we complement the theorem of Huynh by showing the existence of an $\mathbf{EXPSPACE}$-random language in **P/poly**.

The above definition of random languages (and of random strings, if defined analogously) can be seen as an extension of a randomness definition due to Church [Church 1940], based on von Mises concept of collectives. This definition, like Church's, is relatively weak because it admits random strings which do not satisfy the law of iterated logarithm. A stronger, succesful definition of randomness was introduced by Martin-Löf [Martin-Löf 1966]. This and closely related work on Kolmogorov complexity [Kolmogorov 1965, Solomonoff 1964, Chaitin 1966] have been extended by Ko [Ko 1986], Huynh [Huynh 1986a, Huynh 1986b], and Lutz [Lutz 1988] to define randomness relative to complexity classes, but these definitions seem to be too strong to admit a hierarchy theorem as sharp as the one reported here.

1.1 Preliminaries

First we give some notations and the necessary definitions. For a more detailed introduction see for example [Schöning 1986, Balcázar, Díaz, and Gabarró 1987a, Balcázar, Díaz, and Gabarró 1987b]. In the following we will consider only languages and strings over the alphabet $\{0,1\}$. Let $x^{(i)}$ denote the i-th string under lexicographic order and A^n denote the language A restricted to strings of length n. $\langle\ \rangle$ denotes an easy to compute and easy to invert pairing function, $x \circ y$ denotes the concatenation of the strings x and y, and y^k the k-fold concatenation of the string y.

Let a self-delimiting description of a string x be the concatenation of a length description n of x followed by x. n is the length of x where bits equal to 1 are replaced by 11, bits equal to 0 are replaced by 00 and the end of n is marked by a 01. This description has two important properties. First, it is possible (for a Turing machine) to decode the string x from a (possibly infinite) input and second, the length of the self-delimiting description of x is $|x| + 2\log|x|$. We will in the following denote the self-delimiting description of a string x by $\overline{x}$.

For a function f let $O(f)$, $o(f)$, $\omega(f)$, and $\Omega(f)$ have their usual meanings, namely for functions g and f, $g = O(f)$ if $\limsup_{n\to\infty} \frac{g(n)}{f(n)} < \infty$, $g = o(f)$ if $\lim_{n\to\infty} \frac{g(n)}{f(n)} = 0$, $g = \omega(f)$ if $\liminf_{n\to\infty} \frac{g(n)}{f(n)} = \infty$, and $g = \Omega(f)$ if $\liminf_{n\to\infty} \frac{g(n)}{f(n)} > 0$.

Time and space bounded generalized Kolmogorov complexity were introduced by Hartmanis [Hartmanis 1983]. We give a slightly modified definition for notational convenience. For a Turing machine M and integers l and s let

$$\mathbf{KS}_M[l,s] = \{x \quad | \quad (\exists y) : |y| \leq l \text{ and } M(y) = x \text{ using no more than } s \text{ space}\}$$

be the space bounded Kolmogorov complexity (relative to M). For fixed integers this is a finite set, but we can still let l and s be unbounded functions. The following fact can be easily verified [Hartmanis 1983]:

There exist a universal Turing machine U and a constant a such that, for every Turing machine M, there is a constant c such that, for every l and s,

$$\mathbf{KS}_M[l,s] \quad \subseteq \quad \mathbf{KS}_U[l+c, a\cdot s+c].$$

In the following we fix such a universal machine U and omit the subscript U when no confusion arises. The next two observations can be shown with the same simulation argument.

There exist constants l and s such that for all integers l_1, l_2, s_1, and s_2 and all strings x and y:

If $x \in \mathbf{KS}[l_1, s_1]$ and $y \in \mathbf{KS}[l_2, s_2]$ and x XOR y is the bit-wise XORing of x and y then

$$x \text{ XOR } y \in \mathbf{KS}[2\log\log l_1 + \log l_1 + l_1 + l_2 + l, (s_1+s_2)\cdot s],$$

and if $x \in \mathbf{KS}[l_1, s_2]$ then for all space constructible functions $p(n) < n$ the i-th substring of length $p(|x|)$ of x (i.e. from bit $i\cdot p(|x|)$ to bit $(i+1)\cdot p(|x|)$) is in

$$\mathbf{KS}[2\log\log|x| + \log|x| + l_1 + l, (s_1 + p(|x|))\cdot s].$$

The non-uniform complexity classes as introduced by Karp and Lipton are defined as follows [Karp and Lipton 1980].

For a complexity class $\mathcal{C}$ and a class of functions $\mathcal{F}$ from $\mathbb{N}$ to Σ^* let $\mathcal{C}/\mathcal{F}$ denote the non-uniform complexity class of all languages A such that there exists a language $B \in \mathcal{C}$ and a function $h \in \mathcal{F}$ such that for all $x \in \Sigma^*$:

$$x \in A \Longleftrightarrow \langle x, h(|x|)\rangle \in B.$$

We will use also the notation $\mathcal{C}/f$ for a function $f : \mathbb{N} \to \mathbb{N}$ to denote the class $\mathcal{C}/\mathcal{F}$, where $\mathcal{F} = \{g : \mathbb{N} \to \Sigma^* \mid |g(n)| = O(f(n))\}$.

1.2 Non-uniform classes and Kolmogorov complexity

There exists a strong connection between non-uniform complexity classes and resource bounded Kolmogorov complexity. [Balcázar, Díaz, and Gabarró 1987b] gives a characterization of the class **PSPACE/poly** in terms of space bounded Kolmogorov complexity. Here a general characterization is given.

For a language A, let χ_A denote its (infinite) characteristic string, χ_A^n denote its characteristic string up to the n-th word (in lexicographic order) and $\chi_A(n)$ denote its characteristic string for words of length n. All space bounds $g(n)$ for the non-uniform complexity classes $\mathbf{DSPACE}(g(n))/h(n)$ considered here are $\Omega(\log n)$.

Lemma 1.1 *For each A in* $\mathbf{DSPACE}(g(n))/h(n)$ *there is a constant c such that for all n,*

$$\chi_A(n) \in \mathbf{KS}[\log n + c \cdot h(n), c \cdot (n + g(n))].$$

Proof: Let $A \in \mathbf{DSPACE}(g(n))/h(n)$, then there exists a function f, $|f(n)| = O(h(n))$, and a Turing machine M operating in space $O(g(n))$ such that M on input $\langle x, f(|x|)\rangle$ accepts if and only if $x \in A$.

Using a self-delimiting description $\overline{f(n)}$ of $f(n)$ we get $|\overline{f(n)} \circ n| = O(h(n)) + \log n$. Consider the machine M' which on input $\overline{f(n)} \circ n$ cycles through all words w of length n and outputs (for each word w) 1 if M accepts $\langle w, f(n)\rangle$ and 0 otherwise.

Clearly M' computes $\chi_A(n)$ within space $O(n + g(n))$ from an input of length $O(h(n)) + \log n$. Hence (using the property of the universal Turing machine U) we get for some constant c and all n

$$\chi_A(n) \in \mathbf{KS}[\log n + c \cdot h(n), c \cdot (n + g(n))].$$

$\Box$

If the advice function $h(n)$ is the constant zero-function we obtain the uniform space classes:

Corollary 1.2 *For each A in* $\mathbf{DSPACE}(f(n))$ *there is a constant c such that for all n,*

$$\chi_A(n) \in \mathbf{KS}[\log n + c, c \cdot (n + f(n))].$$

The converse of the above does not hold, since $\mathbf{KS}[\log n, f(n)]$ also defines non-recursive languages. But if $g(n) > n, h(n) > \log n$ we get the following characterization of the non-uniform complexity classes.

Theorem 1.3 *Let g, h be functions such that $g(n) = \Omega(n)$ and $h(n) = \Omega(\log n)$. A language A is in* $\mathbf{DSPACE}(g(n))/h(n)$ *if and only if there is a constant c such that for all n,*

$$\chi_A(n) \in \mathbf{KS}[c \cdot h(n), c \cdot g(n)].$$

1.3 Random languages

Several slightly different definitions for random languages have been used in the literature already [Meyer and McCreight 1971, Wilber 1983, Huynh 1987] and it is a priori not at all clear whether they are equivalent. We give three definitions below and show that they are equivalent for all complexity classes of interest in this paper.

We distinguish the following two census functions. $cens_A(n)$ is the cardinality of a language A restricted to strings of length n, i.e. the cardinality of A^n.

$Cens_A(x)$ denotes the cardinality of a language A restricted to strings that are (lexicographically) smaller than x, i.e. the cardinality of $A \cap \{y|y \leq x\}$.

For any two languages A, B let $A \triangle B$ be the language of all strings on which A and B disagree: $A \triangle B = (A \cap \overline{B}) \cup (\overline{A} \cap B)$.

Definition 1.4 *[Meyer and McCreight 1971] Let $x^{(n)}$ denote the n-th string under lexicographically ordering. A language A is* Church-random *for a class $\mathcal{C}$ if for all $B \in \mathcal{C}$*

$$\lim_{n \to \infty} \frac{\mathrm{Cens}_{\mathrm{A \triangle B}}(x^{(n)})}{n} = \frac{1}{2}.$$

Definition 1.5 *[Huynh 1987, Wilber 1983] A language A is* length-random *for a class $\mathcal{C}$ if for all $B \in \mathcal{C}$*

$$\lim_{n \to \infty} \frac{\mathrm{cens}_{\mathrm{A \triangle B}}(n)}{2^n} = \frac{1}{2}.$$

Definition 1.6 *A language A is* sum-length-random *for a class $\mathcal{C}$ if for all $B \in \mathcal{C}$*

$$\lim_{n \to \infty} \frac{\sum_{i=0}^{n} \mathrm{cens}_{\mathrm{A \triangle B}}(i)}{2^{n+1}} = \frac{1}{2}.$$

We say a language $B \in \mathcal{C}$ *approximates* A or it is a witness that A is not random for $\mathcal{C}$ with respect to one of the random definitions above if the respective equation does not hold.
We can now prove the following two facts:

Fact 1: Let $\mathcal{C} = \mathbf{DSPACE}(g(n))/h(n)$ be a non-uniform complexity class where $h(n) = \Omega(n)$. Then A is *Church-random* for $\mathcal{C}$ if and only if A is *sum-length-random* for $\mathcal{C}$.

Fact 2: Let $\mathcal{C}$ be any complexity class. Then A is *length-random* for $\mathcal{C}$ if and only if A is *sum-length-random* for $\mathcal{C}$.

Proof of Fact 1: By the definitions it follows immediately that every Church-random language is sum-length-random. For the converse assume A is a language not Church-random for $\mathcal{C}$. Then there exist a language $B \in \mathcal{C}$ and an $\epsilon > 0$ such that the set

$$W = \left\{ x^{(m)} \mid \mathrm{Cens}_{\mathrm{A \triangle B}}(x^{(m)}) > (\frac{1}{2} + \epsilon) \cdot m \right\}$$

is infinite. Now we can define a new language $Z = \bigcup_{n=0}^{\infty} Z_n$ which witnesses that A is not sum-length-random for $\mathcal{C}$. Let the $Z_n \subseteq \Sigma^n$ be defined as follows.

1. If $W \cap \Sigma^n = \emptyset$, then $Z_n = B \cap \Sigma^n$.

2. If $W \cap \Sigma^n \neq \emptyset$, let $y = \max(W \cap \Sigma^n)$. Define $L = \{x | 0^n \leq x \leq y\}$, and $R = \{x | y < x \leq 1^n\}$.

   If $|(A \triangle B) \cap R| \geq \frac{1}{2}|R|$ set $Z_n = B \cap \Sigma^n$, otherwise set $Z_n = (B \cap L) \cup (\overline{B} \cap R)$.

Note that $\mathrm{Cens}_{\mathrm{A \triangle Z}}(x) \geq \mathrm{Cens}_{\mathrm{A \triangle B}}(x)$ for all x. Since $B \in \mathcal{C}$, $h(n) = \Omega(n)$, and the considered non-uniform complexity classes are closed under union, intersection, and complementation, we also have $Z \in \mathcal{C}$.

There are infinitely many $n > 0$ such that $W \cap \Sigma^n \neq 0$. Fix such an n, and let $y = x^{(m)} = \max(W \cap \Sigma^n)$. Then

$$
\begin{aligned}
\sum_{i=0}^{n} \mathrm{cens}_{\mathrm{A \triangle Z}}(i) &= \mathrm{Cens}_{\mathrm{A \triangle Z}}(1^{n-1}) + |(A \triangle B) \cap L| + |(A \triangle Z) \cap R| \\
&\geq \mathrm{Cens}_{\mathrm{A \triangle B}}(1^{n-1}) + |(A \triangle B) \cap L| + \frac{1}{2}|R| \\
&= \mathrm{Cens}_{\mathrm{A \triangle B}}(x^{(m)}) + \frac{1}{2}(2^{n+1} - m) \\
&> \left(\frac{1}{2} + \epsilon\right) m + \frac{1}{2}(2^{n+1} - m) \\
&= 2^n + \epsilon m \\
&\geq 2^n(1 + \epsilon)
\end{aligned}
$$

so

$$
\frac{\sum_{i=0}^{n} \mathrm{cens}_{\mathrm{A \triangle B}}(i)}{2^{n+1}} > \frac{1}{2} + \frac{\epsilon}{2}.
$$

Since this holds for infinitely many n Z is a witness that A is not sum-length-random for $\mathcal{C}$.

□

The proof of Fact 2 is straightforward. From now on "$\mathcal{C}$-random" means "length-random for $\mathcal{C}$".

2 Main results

What is the necessary complexity for the characteristic strings of a language which is random for a non-uniform complexity class? Clearly, if infinitely many characteristic strings of a language A could be computed in $\mathcal{C}$, for a non-uniform complexity class $\mathcal{C}$, then A is not $\mathcal{C}$-random.

Proposition 2.1 *Let $\mathcal{C} = \mathbf{DSPACE}(g(n))/h(n)$ be a non-uniform complexity class, where $g = \Omega(n)$, and L be any $\mathcal{C}$-random language. Then for every constant c and for all but finitely many n:*

$$\chi_L(n) \notin \mathbf{KS}[c \cdot h(n), c \cdot g(n)].$$

Proof: Assume that for some constant c and infinitely many n $\chi_L(n) \in \mathbf{KS}[c \cdot h(n), c \cdot g(n)]$. Hence the universal Turing machine U computes $\chi_L(n)$ for infinitely many n within space $c \cdot g(n)$ from an input of length $c \cdot h(n)$. Then there exists a Turing machine M, which on input $\langle x, y\rangle$, $|y| = h(|x|)$, simulates U on input y and accepts if and only if the x-th bit of the output of U is 1.

Since the simulation of U on input y and the counting up to the x-th bit can be done in space $O(n + g(n))$ and M accepts L^n correctly for infinitely many n, the language accepted by M is a witness that L is not random for $\mathcal{C}$.

□

We will now construct random languages for non-uniform complexity classes. The languages defined have a special structure.

Definition 2.2 *A language A is* p(n)-periodic *if for every n there exist a string $y \in \Sigma^{p(n)}$ and a positive integer k such that $\chi_A(n)$ is a prefix of y^k. y is called a* witness *that A is $p(n)$-periodic.*

As can be easily seen every $p(n)$-periodic language is in $\mathbf{DSPACE}(n+p(n))/p(n)$ and every language is 2^n-periodic.

Lemma 2.3 *Let f be a space-constructible function with $f(n) = \omega(n)$. Let $p(n) = \min(f(n), 2^n)$. Then for every constant b there exist a constant c and a $p(n)$-periodic language $A \in \mathbf{DSPACE}(f(n))$ such that for all n:*

$$\chi_A(n) \notin \mathbf{KS}[p(n) - c, b \cdot f(n)].$$

Proof: Let b_1 and c_1 be constants such that if $\chi_A(n)$ is in $\mathbf{KS}[p(n) - d, b \cdot f(n)]$ for some constant d and space constructible $p(n)$, then the first $p(n)$ bits of $\chi_A(n)$ are in $\mathbf{KS}[p(n) - d + c_1, b_1 \cdot b \cdot f(n)]$.

Consider for each n the lexicographically first string z of length $p(n)$ such that

$$z \notin \mathbf{KS}[p(n) - 1, (b \cdot b_1) \cdot f(n)]$$

and call it z_n. Such strings exist by an obvious counting argument. A $p(n)$-periodic language A is defined in the following way:

Let $bin(x)$ be the string x treated as binary number, where leading zeros are ignored. A string x of length n is in A if and only if the ($bin(x)$ mod $|z_n|$)-th bit of z_n is 1, i.e.

$$A = \{x \mid |x| = n \text{ and } bit_x(z_n) = 1\},$$

where $z_n = min_z(|z| = p(n)$ and $z \notin \mathbf{KS}[p(n)-1, (b \cdot b_1) \cdot f(n)])$ and $bit_x(z_n)$ is the value of the ($bin(x)$ mod $|z_n|$)-th bit of string z_n.

Clearly the characteristic strings of A, $\chi_A(n)$ are not in $\mathbf{KS}[p(n)-c_1-1, b \cdot f(n)]$, otherwise we get that $z_n \in \mathbf{KS}[p(n)-1, (b \cdot b_1) \cdot f(n)]$.

To see that A is in $\mathbf{DSPACE}(f(n))$ we define the following deterministic machine M:

input x, $|x| = n$
for all z, $|z| = p(n)$ (in lexicographical order) **do**
    if (**for** all y, $|y| \leq p(n) - 1$,
            $U(y) \neq z$ or U on input y uses more space than $(b \cdot b_1) \cdot f(n)$)
      then if $bit_x(z) = 1$
            then *accept*
            else *reject*

The machine M accepts the language A. The two nested for-loops ensure that the lexicographically first string z, of length $p(n)$, not in $\mathbf{KS}[p(n)-1, (b \cdot b_1) \cdot f(n)]$ is found.

The space used by M is essentially the space needed to simulate U plus the space to keep track of the values of z and y. The simulation of U is in space $O(f(n))$ and z, y are of length $p(n)$. Thus, M runs in space $O(f(n))$.

□

Lemma 2.4 *There exists a constant s such that for all $\epsilon > 0$ there exist a constant $l < 1$ and an n_0, such that for all strings z of length $n > n_0$ containing less than $(\frac{1}{2} - \epsilon) \cdot n$ bits equal to 1:*

$$z \in \mathbf{KS}[l \cdot n, s \cdot n].$$

Proof: First we define an order on the set of all strings in a way such that strings with only a few bits equal to 1 get a low index. For $x \in \{0,1\}^n$, let $r(x) = \#\{k \mid k \leq n, bit_k(x) = 1\}$ be the number of bits equal to 1 in a string x. Then we define the order $\prec$ on $\Sigma^n \times \Sigma^n$ as follows: $s \prec t$ if $(r(x) < r(y))$ or ($r(x) = r(y)$ and x precedes y in the lexicographical order).

Lemma 2.5 *[Ko 1986] The function* dec*, defined by* $\mathrm{dec}(n, m) =$ *the m-th string in $\{0,1\}^n$ in the order $\prec$, is computable in time $O(n^2)$ and in space $O(n)$.*

Lemma 2.6 *[Chaitin 1966] Let ϵ be a real number between 0 and $\frac{1}{2}$. Then, for any $n \geq 1$,*

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

for some constant c, where $H(\epsilon) = -(\frac{1}{2}+\epsilon) \cdot \log(\frac{1}{2}+\epsilon) - (\frac{1}{2}-\epsilon) \cdot \log(\frac{1}{2}-\epsilon)$ is the entropy function.

Now the logarithm of the number of possible strings of length n with up to $(\frac{1}{2}-\epsilon)\cdot n$ many bits equal to 1 is estimated by

$$\begin{aligned} \log \sum_{k=0}^{\lfloor (\frac{1}{2}-\epsilon)n \rfloor} \binom{n}{k} &\leq \log \left(n \cdot \binom{n}{\lfloor (\frac{1}{2} - \epsilon)n \rfloor} \right) \\ &\leq n \cdot H(\epsilon) + log(n) + c \qquad \text{for } \epsilon > 0. \end{aligned}$$

Thus we get:

The length of the index v of z is less than $n \cdot H(\epsilon) + \log n + c$, where $H(\epsilon) < 1$. Hence, for every $\epsilon > 0$, there exist constants $l < 1$ and n_0, such that for all strings z of length $n > n_0$ we have $|v| \leq l \cdot n$.

There exist constants s and n_0 such that for all $n > n_0$ $z = dec(n, v)$ can be computed in space $s \cdot n$. Note that by lemma 2.5 the constant s is independent of ϵ.

□

Lemma 2.7 *Let p, g and h be functions such that p and g are space-constructible, $h(n) = o(2^n)$, and $p(n) \leq 2^n$. Then there exists a constant d_2 such that for every $p(n)$-periodic language A which is not* **DSPACE**$(g(n))/h(n)$*-random, there exist constants $\alpha < 1$ and d_1 such that for infinitely many n,*

$$\chi_A(n) \in \mathbf{KS}[\alpha \cdot p(n) + n + d_1 \cdot (h(n) + \log n), d_2 \cdot p(n) + d_1 \cdot (n + g(n))].$$

Proof: Let A be a $p(n)$-periodic language which is not **DSPACE**$(g(n))/h(n)$-random. Then there exists a language B in **DSPACE**$(g(n))/h(n)$ such that

$$\lim_{n \to \infty} \frac{\mathrm{cens}_{\mathrm{A \triangle B}}(n)}{2^n} = \frac{1}{2}$$

does not hold. This means, that for some $\epsilon > 0$ and infinitely many n,

$$\left| \frac{\mathrm{cens}_{\mathrm{A \triangle B}}(n)}{2^n} - \frac{1}{2} \right| \geq \epsilon.$$

W.l.o.g. we assume that there are infinitely many n such that

$$\mathrm{cens}_{\mathrm{A}\triangle\mathrm{B}}(n) \leq (\frac{1}{2} - \epsilon) \cdot 2^n.$$

Let S be a (infinite) sequence of such numbers. For the following we fix such an $n \in S$ large enough to satisfy the condition of lemma 2.4. We will show how to compute the string z witnessing that A^n is $p(n)$-periodic.

Consider $\chi_{A\triangle B}(n)$ (for such an $n \in S$) to be partitioned in $\frac{2^n}{p(n)}$ substrings each of length $p(n)$. Then there exists at least one substring where the number of bits equal to 1 is less than $(\frac{1}{2} - \epsilon) \cdot p(n)$. Otherwise the total number of bits equal to 1 would exceed $(\frac{1}{2} - \epsilon) \cdot 2^n$. Call the first such substring $u_{A\triangle B}$ and let i be its number.

Now we can compute z by bitwise XORing $u_{A\triangle B}$ and and the i-th substring of $\chi_B(n)$ of length $p(n)$. (Since $(A \triangle B) \triangle B = A$, and A is $p(n)$-periodic.) The complexity of z can be derived in the following way.

$$u_{A\triangle B} \in \mathbf{KS}[l \cdot p(n), s \cdot p(n)] \text{ by lemma 2.4,}$$

for a constant s independent of ϵ (and hence of A and B) and a constant $l < 1$ depending on ϵ, and

$$\chi_B(n) \in \mathbf{KS}[c_1 \cdot (h(n) + \log n), c_1 \cdot (n + g(n))] \text{ by lemma 1.1,}$$

for a constant c, and therefore the i-th substring of $\chi_B(n)$

$$u_B \in \mathbf{KS}[2 \log\log n + \log n + n + c_2 \cdot (h(n) + \log n), c_2 \cdot (n + g(n))]$$

for a constant c_2. Choosing $c_3 = c_2 + 2$

$$u_B \in \mathbf{KS}[n + c_3 \cdot (h(n) + \log n), c_3 \cdot (n + g(n))].$$

Hence we get for $z = u_{A\triangle B}$ XOR u_B, the bitwise XOR of $u_{A\triangle B}$ and u_B:

$$z \in \mathbf{KS}[l \cdot p(n) + n + 2c_3 \cdot (h(n) + \log n), c_4(s \cdot p(n) + c_3 \cdot (n + g(n)))],$$

Let d_1 be the maximum of $2c_3$ and $c_3 \cdot c_4$ and $d_2 = c_4 \cdot s$, and since $\chi_A(n) = z^k$ for some k the lemma follows. Note that d_2 is independent of A and B since s is independent and c_4 is a constant factor needed to compute the XOR of two strings. $\square$

Theorem 2.8 *Let $f, g,$ and h be space constructible functions, where $n, g(n)$, $h(n) = o(f(n))$ and $h(n) = o(2^n)$. Then there exists a language in* $\mathbf{DSPACE}(f(n))$ *which is* $\mathbf{DSPACE}(g(n))/h(n)$*-random.*

Proof: Assume the hypothesis. Let $p(n) = \min(f(n), 2^n)$ and choose d_2 as in lemma 2.7. Let $b = 2 \cdot d_2$ and choose c and A as in lemma 2.3. Remark that $A \in \mathbf{DSPACE}(f(n))$. Let $\alpha < 1$ and d_1 be arbitrary constants. Let $l(n) = \alpha \cdot p(n) + n + d_1 \cdot (h(n) + \log n)$ and $s(n) = d_2 \cdot f(n) + d_1 \cdot (n + g(n))$. Since $n + g(n) = o(f(n))$ and $n + h(n) + \log n = o(p(n))$, there exists an n_0 such that

$$\mathbf{KS}[l(n), s(n)] \subset \mathbf{KS}[p(n) - c, b \cdot f(n)]$$

for all $n \geq n_0$. It follows by the choice of A that $\chi_A(n) \notin \mathbf{KS}[l(n), s(n)]$ for all $n \geq n_0$. Since such n_0 exist for each choice of α and d_1, it follows by lemma 2.7 that A is $\mathbf{DSPACE}(g(n))/h(n)$-random.

□

With the same proof technique it is possible to show the following claim concerning random languages for non-uniform time classes.

Claim 2.9 *There exist languages in* $\mathbf{DTIME}(2^{n^{1+\epsilon}})$ *that are* $\mathbf{P}/\log$*-random.*

As an immediate consequence of our main theorem we get the result of Huynh and the result of Wilber as special cases. The result of Huynh is strengthend to **PSPACE/poly**-random languages in **EXPSPACE**, which also follows by the proof of Huynh.

Corollary 2.10 *[Huynh 1987] There exist languages in* **EXPSPACE** *that are random for* **PSPACE/poly**.

Corollary 2.11 *[Wilber 1983] Let f, g be functions, where $f(n) = \omega(n)$ and $g(n) = o(f(n))$. Then there exist languages in* $\mathbf{DSPACE}(f(n))$ *that are random for* $\mathbf{DSPACE}(g(n))$.

Wilber [Wilber 1983] proved this result already for functions $f(n) = \omega(log(n))$ using a different method. As an immediate consequence of the construction of our proof we get:

Corollary 2.12 *Let f be a function where $f(n) = \omega(n)$. Then there exists a constant c such that for infinitely many n there exist strings of length n in* $\mathbf{KS}[log(n)+c, c \cdot f(n)]$ *which are not in* $\mathbf{KS}[n-1, f(n)]$.

Proof: Consider the following machine M. M simulates the universal machine U for every string $z \leq n$ until the first is found such that U on input y uses more space than $f(n)$ or $U(y) \neq z$, for all $|y| \leq n - 1$. This search can be done in space $O(f(n))$. Hence with the property of the universal Turing machine we get $z \in \mathbf{KS}[log(n) + c, c \cdot f(n)]$.

□

Up to now we showed how to construct languages in a class $\mathcal{C}_1$ which are random for a class $\mathcal{C}_2$ using the greater computational power of $\mathcal{C}_1$. This yielded computable random languages for classes containing non-computable languages, i.e. non-uniform classes.

Now we will show how one can use the power of uncomputability for the construction of random languages.

Theorem 2.13 *Let $\mathcal{C}$ be a uniform complexity class. Let f and g be functions such that $f(n) = \omega(n)$, $g(n) = o(f(n))$, and $g(n) = o(2^n)$. Then there exist languages L in $\mathbf{P}/f(n)$ which are random for $\mathcal{C}/g(n)$.*

Proof: We use a similar proof technique as in our main theorem. Let $y_1, y_2, \ldots$ be a sequence of strings such that y_i is a string of length $f(i)$ which is not in $\mathbf{K}[f(i)-1]$. $\mathbf{K}[f(n)]$ denotes the Kolmogorov complexity if no bound on the computation is given. Define L to be an $f(n)$-periodic language, where for all i and $k = \lceil \frac{2^i}{f(i)} \rceil$ $\chi_L(i)$ is a prefix of $(y_i)^k$. Clearly L is in $\mathbf{P}/f(n)$.

Assume that for a function $g = o(f)$ there exists a language $A \in \mathcal{C}/(g(n))$ that approximates L. Then w.l.o.g. there are infinitely many n such that the characteristic string of $(A \triangle L)^n$ contains few bits equal to 1. Each such characteristic string has a substring of length $f(n)$ with less than $(\frac{1}{2} - \epsilon) \cdot f(n)$ many bits equal to 1. Let i_n ($i_n \leq \frac{2^n}{f(n)}$) be the indices of these substrings.

These substrings can therefore (cf. Lemma 2.4) be described by strings of length $d \cdot f(n)$ for some $d < 1$. Thus from some n on we can compute infinitely many strings y_n from an input less than $d \cdot f(n) + |i_n| + |M_A| + g(n) + 2 \cdot \log(n) \leq f(n) - 1$. This contradicts the high Kolmogorov complexity of all strings y_n.

□

Note that the only property of $\mathcal{C}$ which is needed is that $\mathcal{C}$ contains only computable languages.

Corollary 2.14 *There exist languages in* $\mathbf{P}/\mathbf{poly}$ *which are random for every class of computable languages.*

As a special case we get that there exists an **EXPSPACE**-random language in $\mathbf{P}/\mathbf{poly}$. This is interesting because we already showed that there is a $\mathbf{P}/\mathbf{poly}$-random language in **EXPSPACE**. This situation that two classes contain random languages for each other class can be used to create new languages which are random for both classes.

Let $\mathcal{C}_1, \mathcal{C}_2$ be complexity classes which are closed under symmetric difference $\triangle$ such that

1. there exists a language $L_1 \in \mathcal{C}_1$ which is random for $\mathcal{C}_2$

2. there exists a language $L_2 \in \mathcal{C}_2$ which is random for $\mathcal{C}_1$.

Then $L_3 = L_1 \triangle L_2$ is random for C_1 and for C_2 and L_3 is in the boolean closure of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Acknowledgements: The authors would like to thank an unknown referee for helpful suggestions and corrections. The authors would like to thank Uwe Schöning for consistent help and encouragement. Also many thanks to José Balcázar for the discussions during his visit in Koblenz.

References

[Balcázar, Díaz, and Gabarró 1987a] J. Balcázar, J. Díaz, and J. Gabarró. *Structural Complexity 1.* Springer 1987.

[Balcázar, Díaz, and Gabarró 1987b] J. Balcázar, J. Díaz, and J. Gabarró. On characterizations of the class PSPACE/poly. *Theoretical Computer Science*, 52(1987):251–267.

[Blum and Micali 1982] 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.

[Chaitin 1966] G.J. Chaitin. On the length of programs for computing finite binary sequences. *Journal of the Association for Computing Machinery*, 13(1966):547–569.

[Church 1940] A. Church. On the concept of a random sequence. *Bull. Amer. Soc.* 46(1940):130–135.

[Hartmanis 1983] J. Hartmanis Generalized Kolmogorov complexity and the structure of feasible computations. *Proc. 24th IEEE Symp. on Foundations of Computer Science* 1983:439–445.

[Hartmanis, Lewis, Stearns 1965] J. Hartmanis, P.M. Lewis II, and R.E. Stearns. Hierarchies of memory limited computations *Proc. 6th Annual IEEE Symp. on Switching Circuits Theory and Logicals Design.* 1965:179–190.

[Huynh 1986a] D.T. Huynh. Some observations about the randomness of hard problems. *SIAM Journal on Computing*, 15(1986):1101–1105.

[Huynh 1986b] D.T. Huynh. Resource-bounded complexity of hard languages. *Proc. 1st Conference on Stucture in Complexity Theory*, 1988:184–195.

[Huynh 1987] D.T. Huynh. On solving hard problems by polynomial-size circuits. *Information Processing Letters*, 24(1987):171–176.

[Karp and Lipton 1980] R.M. Karp and R.J. Lipton. Some connections between nonuniform and uniform complexity classes. *Proc. 12th Annual ACM Symposium on Theory of Computing* 1980:302–309.

[Kolmogorov 1965] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1(1965):1–7.

[Kolmogorov and Uspenskii 1987] A.N. Kolmogorov and V.A. Uspenskii Algorithms and Randomness, translated in *Probability and Its Applications*, 32(1987):389–412.

[Ko 1986] K.I. Ko. On the notion of infinite pseudorandom sequences. *Theoretical Computer Sciences*, 48(1986):9–13.

[Li and Vitanyi 1988] M. Li and P.M.B. Vitanyi. Two decades of applied Kolmogorov complexity. *Proc. 3rd Conference on Structure in Complexity Theory* IEEE 1988:80–101.

[Lutz 1988] J.H. Lutz. Pseudorandom sources for BPP. *Proc. 3rd Conference on Structure in Complexity Theory* IEEE 1988:175–180.

[Martin-Löf 1966] P. Martin-Löf. On the definition of random sequences. *Information and Control*, 9 (1966):602–619.

[Meyer and McCreight 1971] A.R. Meyer and E.M. McCreight. Computationally complex and pseudo-random zero-one valued functions, in: Z. Kohavi and A. Paz (eds.). *Theory of Machines and Computations*:19–42, Academic Press 1971.

[Schöning 1986] U. Schöning. *Complexity and Structure.* Lecture Notes in Computer Science 211, Springer 1986.

[Solomonoff 1964] R.J. Solomonoff. A formal theory of inductive inference, Part 1 and Part 2. *Information and Control* 7(1964):1–22, 224–254.

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

[A.C. Yao 1982] A.C. Yao. Theory and application of trapdoor functions. *Proc. 23rd IEEE Symp. on Foundations of Computer Science* 1982:80–91.