Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Statistical Planning and Inference 133 (2005) 95–110
journal of statistical planning and inference
www.elsevier.com/locate/jspi

Using information theory approach to randomness testing☆

B.Ya. Ryabko*, V.A. Monarev

Department of Applied Mathematics & Cybernatics, Siberian State University of Telecommunication & Computer Science, Kirov str. 86, Novosibirsk 630102, Russian Federation

Received 1 July 2003; accepted 20 February 2004
Available online 1 July 2004

Abstract

We address the problem of detecting deviations of binary sequence from randomness,which is very important for random number (RNG) and pseudorandom number generators (PRNG). Namely, we consider a null hypothesis H_0 that a given bit sequence is generated by Bernoulli source with equal probabilities of 0 and 1 and the alternative hypothesis H_1 that the sequence is generated by a stationary and ergodic source which differs from the source under H_0. We show that data compression methods can be used as a basis for such testing and describe two new tests for randomness, which are based on ideas of universal coding. Known statistical tests and suggested ones are applied for testing PRNGs. Those experiments show that the power of the new tests is greater than of many known algorithms.

MSC: 62B10; 62G10; 62M07; 62M10; 94A29

Keywords: Hypothesis testing; Randomness testing; Random number testing; Universal code; Information Theory; Random number generator; Shannon entropy

1. Introduction

The randomness testing of random number and pseudorandom number generators is used for many purposes including cryptographic, modeling and simulation applications

☆ Supported by INTAS Grant No. 00-738 and Russian Foundation for Basic Research under Grant No. 03-01-00495.

* Corresponding author. Tel.: +7-3832-284938; fax: +1-3832-668030.
E-mail address: ryabko@adm.ict.nsc.ru (B.Ya. Ryabko)

0378-3758/$ - see front matter
doi:10.1016/j.jspi.2004.02.010

(see, for example, Knuth, 1981; L'Ecuyer, 1994; Maurer, 1992; Mezenes et al., 1996). For such applications a required bit sequence should be true random, i.e., by definition, such a sequence could be interpreted as the result of the flips of a "fair" coin with sides that are labeled "0" and "1" (for short, it is called a random sequence; see Rukhin et al., 2001). More formally, we will consider the main hypothesis H_0 that a bit sequence is generated by the Bernoulli source with equal probabilities of 0's and 1's. Associated with this null hypothesis is the alternative hypothesis H_1 that the sequence is generated by a stationary and ergodic source which generates letters from $\{0, 1\}$ and differs from the source under H_0.

In this paper we will consider some tests which are based on results and ideas of Information Theory and, in particular, the source coding theory. First, we show that a universal code can be used for randomness testing. (Let us recall that, by definition, the universal code can compress a sequence asymptotically till the Shannon entropy per letter when the sequence is generated by a stationary and ergodic source.) If we take into account that the Shannon per-bit entropy is maximal (1 bit) if H_0 is true and is less than 1 if H_1 is true (Billingsley, 1965; Gallager, 1968), we see that it is natural to use this property and universal codes for randomness testing because, in principle, such a test can distinguish each deviation from randomness, which can be described in a framework of the stationary and ergodic source model. Loosely speaking, the test rejects H_0 if a binary sequence can be compressed by a considered universal code (or a data compression method).

It should be noted that the idea to use the compressibility as a measure of randomness has a long history in mathematics. The point is that, on the one hand, the problem of randomness testing is quite important for practice, but, on the other hand, this problem is closely connected with such deep theoretical issues as the definition of randomness, the logical basis of probability theory, randomness and complexity, etc. (see Kolmogorov, 1965; Li and Vitanyi, 1997; Knuth, 1981; Maurer, 1992). Thus, Kolmogorov suggested to define the randomness of a sequence, informally, as the length of the shortest program, which can create the sequence (if one of the universal Turing machines is used as a computer). So, loosely speaking, the randomness (or Kolmogorov complexity) of the finite sequence is equal to its shortest description. It is known that the Kolmogorov complexity is not computable and, therefore, cannot be used for randomness testing. On the other hand, each lossless data compression code can be considered as a method for upper bounding the Kolmogorov complexity. Indeed, if x is a binary word, ϕ is a data compression code and $\phi(x)$ is the codeword of x, then the length of the codeword $|\phi(x)|$ is the upper bound for the Kolmogorov complexity of the word x. So, again we see that the codeword length of the lossless data compression method can be used for randomness testing.

In this paper we suggest tests for randomness, which are based on results and ideas of the source coding theory.

Firstly, we show how to build a test basing on any data compression method and give some examples of application of such test to PRNG's testing. It should be noted that data compression methods were considered as a basis for randomness testing in literature. For example, Maurer's Universal Statistical Test, Lempel–Ziv Compression Test and Approximate Entropy Test are connected with universal codes and are quite popular in practice (see, for example, Rukhin et al., 2001). In contrast to known methods, the suggested approach gives a possibility to make a test for randomness, basing on any lossless data compression method even if a distribution law of the codeword lengths is not known.

Secondly, we describe two new tests, conceptually connected with universal codes. When both tests are applied, a tested sequence $x_1x_2\dots x_n$ is divided into subwords $x_1x_2\dots x_s$, $x_{s+1}x_{s+2}\dots x_{2s},\dots,\ s\geqslant 1$, and the hypothesis H_0^* that the subwords obey the uniform distribution (i.e. each subword is generated with the probability 2^{-s}) is tested against $\mathrm{H}_1^*=\neg\mathrm{H}_0^*$. The key idea of the new tests is as follows. All subwords from the set $\{0,1\}^s$ are ordered and this order changes after processing each subword $x_{js+1}x_{js+2}\dots x_{(j+1)s}$, $j=0,1,\dots$ in such a way that, loosely speaking, the more frequent subwords have small ordinals. When the new tests are applied, the frequency of different ordinals are estimated (instead of frequencies of the subwords as for, say, chi-square test).

The natural question is how to choose the block length s in such schemes. We show that, informally speaking, the block length s should be taken quite large due to the existence of so called *two-faced processes*. More precisely, it is shown that for each integer s^* there exists such a process ξ that for each binary word u the process ξ creates u with the probability $2^{-|u|}$ if the length of the u ($|u|$) is less than or equal to s^*, but, on the other hand, the probability distribution $\xi(v)$ is very far from uniform if the length of the words v is greater than s^*. (So, if we use a test with the block length $s\leqslant s^*$, the sequences generated by ξ will look like random, in spite of ξ is far from being random.)

The outline of the paper is as follows. In Section 2 the general method for construction randomness testing algorithms basing on lossless data compressors is described. Two new tests for randomness, which are based on constructions of universal coding, as well as the two-faced processes, are described in the Section 3. In Section 4 the new tests are experimentally compared with methods from "A statistical test suite for random and pseudo-random number generators for cryptographic applications", which was recently suggested by Rukhin et al. (2001). It turns out that the new tests are more powerful than known ones.

2. Data compression methods as a basis for randomness testing

2.1. Randomness testing based on data compression

Let A be a finite alphabet and A^n be the set of all words of the length n over A, where n is an integer. By definition, $A^*=\bigcup_{n=1}^{\infty}A^n$ and A^∞ is the set of all infinite words $x_1x_2\dots$ over the alphabet A. A data compression method (or code) φ is defined as a set of mappings φ_n such that $\varphi_n: A^n\to\{0,1\}^*$, $n=1,2,\dots,$ and for each pair of different words $x,y\in A^n$ $\varphi_n(x)\neq\varphi_n(y)$. Informally, it means that the code φ can be applied for compression of each message of any length n, $n>0$ over alphabet A and the message can be decoded if its code is known.

Now, we can describe a statistical test which can be constructed basing on any code φ. Let n be an integer and $\hat{\mathrm{H}}_0$ be a hypothesis that the words from the set A^n obey the uniform distribution, i.e., $p(u)=|A|^{-n}$ for each $u\in\{0,1\}^n$. (Here and below $|x|$ is the length if x is a word, and the number of elements if x is a set.) Let a required level of significance (or a Type I error) be α, $\alpha\in(0,1)$. The following main idea of a suggested test is quite natural: The well compressed words should be considered as non-random and $\hat{\mathrm{H}}_0$ should be rejected. More exactly, we define a critical value of the suggested test by

$$t_\alpha = n \log |A| - \log(1/\alpha) - 1. \tag{1}$$

(Here and below $\log x = \log_2 x$.)

Let u be a word from A^n. By definition, the hypothesis $\hat{\mathrm{H}}_0$ is accepted if $|\varphi_n(u)| > t_\alpha$ and rejected, if $|\varphi_n(u)| \leqslant t_\alpha$. We denote this test by $\Gamma^{(n)}_{\alpha,\varphi}$.

Theorem 1. *For each integer n and a code φ, the Type I error of the described test $\Gamma^{(n)}_{\alpha,\varphi}$ is not larger than α.*

Proof is given in the appendix.

Comment 1. The described test can be modified in such a way that the Type I error will be equal to α. For this purpose we define the set A_γ by

$$A_\gamma = \{x : x \in A^n \ \& \ |\varphi_n(x)| = \gamma\}$$

and an integer g for which the two following inequalities are valid:

$$\sum_{j=0}^{g} |A_j| \leqslant \alpha |A|^n < \sum_{j=0}^{g+1} |A_j|. \tag{2}$$

Now the modified test can be described as follows:

If for $x \in A^n$ $|\varphi_n(x)| \leqslant g$ then $\hat{\mathrm{H}}_0$ is rejected, if $|\varphi_n(x)| > (g+1)$ then $\hat{\mathrm{H}}_0$ is accepted and if $|\varphi_n(x)| = (g+1)$ the hypothesis $\hat{\mathrm{H}}_0$ is accepted with the probability

$$\left(\sum_{j=1}^{g+1} |A_j| - \alpha |A|^n \right) \Big/ \ |A_{g+1}|$$

and rejected with the probability

$$1 - \left(\sum_{j=1}^{g+1} |A_j| - \alpha |A|^n \right) \Big/ \ |A_{g+1}|.$$

(Here we used a randomized criterion, see for definition, for example, Kendall and Stuart, 1961, part 22.11.) We denote this test by $\Upsilon^{(n)}_{\alpha,\varphi}$.

Claim 1. *For each integer n and a code φ, the Type I error of the described test $\Upsilon^{(n)}_{\alpha,\varphi}$ is equal to α.*

Proof is given in the appendix.

We can see that this criterion has the level of significance (or Type I error) exactly α, whereas the first criterion, which is based on critical value (1), has the level of significance that could be less than α. In spite of this drawback, the first criterion may be more useful due to its simplicity. Moreover, such an approach gives a possibility to use a data compression method ψ for testing even in case where the distribution of the length $|\psi_n(x)|, \ x \in A^n$ is not known.

Comment 2. We have considered codes, for which different words of the same length have different codewords (in Information Theory sometimes such codes are called non-singular.) Quite often a stronger restriction is required in Information Theory. Namely, it is required that each sequence $\varphi_n(x_1)\varphi_n(x_2)\ldots\varphi(x_r), \ r \geqslant 1$, of encoded words from the set $A^n, \ n \geqslant 1$, can be uniquely decoded into $x_1x_2\ldots x_r$. Such codes are called uniquely decodable. For example, let $A = \{a, b\}$, the code $\psi_1(a) = 0, \psi_1(b) = 00$, obviously, is non-singular, but is not uniquely decodable. (Indeed, the word 000 can be decoded in both *ab* and *ba*.) It is well known in Information Theory that a code φ can be uniquely decoded if the following Kraft inequality is valid:

$$\Sigma_{u \in A^n} 2^{-|\varphi_n(u)|} \leqslant 1, \tag{3}$$

see, for e.g., Gallager (1968).

If it is known that the code is uniquely decodable, the suggested critical value (1) can be changed. Let us define

$$\hat{t}_\alpha = n \log |A| - \log(1/\alpha). \tag{4}$$

Let, as before, u be a word from A^n. By definition, the hypothesis $\hat{\mathrm{H}}_0$ is accepted if $|\varphi_n(u)| > \hat{t}_\alpha$ and rejected, if $|\varphi_n(u)| \leqslant \hat{t}_\alpha$. We denote this test by $\hat{\Gamma}^{(n)}_{\alpha,\varphi}$.

Claim 2. *For each integer n and a uniquely decodable code φ, the Type I error of the described test $\hat{\Gamma}^{(n)}_{\alpha,\varphi}$ is not larger than α.*

P. roof is given in the appendix.

So, we can see from (1) and (4) that the critical value is larger, if the code is uniquely decodable. On the other hand, the difference is quite small and (1) can be used without a large loose of the test power even in a case of the uniquely decodable codes.

It should not be a surprise that the level of significance (or a Type I error) does not depend on the alternative hypothesis H_1, but, of course, the power of a test (and the Type II error) will be determined by H_1.

The examples of testing by real data compression methods will be given in Section 4.

2.2. Randomness testing based on universal codes

We will consider the main hypothesis H_0 that the letters of a given sequence $x_1x_2\ldots x_t$, $x_i \in A$, are independent and identically distributed (i.i.d.) with equal probabilities of all $a \in A$ and the alternative hypothesis H_1 that the sequence is generated by a stationary and ergodic source, which generates letters from A and differs from the source under H_0. (If $A = \{0, 1\}$, i.i.d. coincides with Bernoulli source.) The definition of the stationary and ergodic source and the Shannon entropy of such sources can be found in Billingsley (1965) and Gallager (1968).

We will consider statistical tests, which are based on universal coding and universal prediction. First we define a universal code.

By definition, φ is a universal code if for each stationary and ergodic source (or a process) π the following equality is valid with probability 1 (according to the measure π):

$$\lim_{n\to\infty} (|\varphi_n(x_1 \dots x_n)|)/n = h(\pi), \tag{5}$$

where $h(\pi)$ is the Shannon entropy. (Such codes exist, see Ryabko, 1984.) It is well known in Information Theory that $h(\pi) = \log|A|$ if H_0 is true, and $h(\pi) < \log|A|$ if H_1 is true (see, for e.g., Billingsley, 1965; Gallager, 1968). From this property and (5) we can easily yield the following theorem.

Theorem 2. *Let φ be a universal code, $\alpha \in (0,1)$ be a level of significance and a sequence $x_1x_2\dots x_n,\ n\geqslant 1$, be generated by a stationary ergodic source π. If the described above test $\Gamma^{(n)}_{\alpha,\varphi}$ is applied for testing H_0 (against H_1), then, with probability 1, the Type I error is not larger than α, and the Type II error goes to 0, when $n\to\infty$.*

So, we can see that each good universal code can be used as a basis for randomness testing. But converse proposition is not true. Let, for example, there be a code, whose codeword length is asymptotically equal to $(0.5+h(\pi)/2)$ for each source π (with probability 1, where, as before, $h(\pi)$ is the Shannon entropy). This code is not good, because its codeword length does not tend to the entropy, but, obviously, such code could be used as a basis for a test of randomness. So, informally speaking, the set of tests is larger than the set of universal codes.

Note that the close problems were considered by Bailey (1976), who obtained many important results in this field.

3. Two new tests for randomness and two-faced processes

Firstly, we suggest two tests which are based on ideas of universal coding, but they are described in such a way that can be understood without any knowledge of Information Theory.

3.1. The "book stack" test

Let, as before, there be given an alphabet $A = \{a_1, \dots, a_S\}$, a source, which generates letters from A, and two following hypotheses: the source is i.i.d. and $p(a_1) = \cdots = p(a_S) = 1/S$ (H_0) and $H_1 = \neg H_0$. We should test the hypotheses basing on a sample $x_1x_2\dots x_n,\ n\geqslant 1$, generated by the source. When the "book stack" test is applied, all letters from A are ordered from 1 to S and this order is changed after observing each letter x_t according to the formula

$$v^{t+1}(a) = \begin{cases} 1 & \text{if } x_t = a, \\ v^t(a) + 1 & \text{if } v^t(a) < v^t(x_t), \\ v^t(a) & \text{if } v^t(a) > v^t(x_t), \end{cases} \tag{6}$$

where v^t is the order after observing $x_1 x_2 \ldots x_t$, $t = 1, \ldots, n$, v^1 is defined arbitrarily. (For e.g., we can define $v^1 = \{a_1, \ldots, a_S\}$.) Let us explain (6) informally. Suppose that the letters of A make a stack, like a stack of books and $v^1(a)$ is a position of a in the stack. Let the first letter x_1 of the word $x_1 x_2 \ldots x_n$ be a. If it takes i_1th position in the stack ($v^1(a) = i_1$), then take a out of the stack and put it on the top. (It means that the order is changed according to (6).) Repeat the procedure with the second letter x_2 and the stack obtained, etc.

It can help to understand the main idea of the suggested method if we take into account that, if H_1 is true, then frequent letters from A (as frequently used books) will have relatively small numbers (will spend more time next to the top of the stack). On the other hand, if H_0 is true, the probability to find each letter x_i at each position j is equal to $1/S$.

Let us proceed with the description of the test. The set of all indexes $\{1, \ldots, S\}$ is divided into r, $r \geqslant 2$, subsets $A_1 = \{1, 2, \ldots, k_1\}$, $A_2 = \{k_1 + 1, \ldots, k_2\}, \ldots, A_r = \{k_{r-1} + 1, \ldots, k_r\}$. Then, using $x_1 x_2 \ldots x_n$, we calculate how many $v^t(x_t)$, $t = 1, \ldots, n$, belong to a subset A_k, $k = 1, \ldots, r$. We define this number as n_k (or, more formally, $n_k = |\{t : v^t(x_t) \in A_k, t = 1, \ldots, n\}|$, $k = 1, \ldots, r$). Obviously, if H_0 is true, the probability of the event $v^t(x_t) \in A_k$ is equal to $|A_j|/S$. Then, using a "common" chi-square test we test the hypothesis $\hat{\mathrm{H}}_0 = P\{v^t(x_t) \in A_k\} = |A_k|/S$ basing on the empirical frequencies $n_1, \ldots, n_r$, against $\hat{\mathrm{H}}_1 = \neg \hat{\mathrm{H}}_0$.Let us recall that the value

$$x^2 = \sum_{i=1}^{r} \frac{(n_i - n(|A_i|/S))^2}{n(|A_i|/S)} \tag{7}$$

is calculated, when chi-square test is applied, see, for e.g., Kendall and Stuart (1961). It is known that x^2 asymptotically follows the χ-square distribution with $(k-1)$ degrees of freedom (χ^2_{k-1}) if $\hat{\mathrm{H}}_0$ is true. If the level of significance (or a Type I error) of the χ^2 test is α, $\alpha \in (0, 1)$, the hypothesis $\hat{\mathrm{H}}_0$ is accepted when x^2 from (7) is less than the $(1-\alpha)$-value of the χ^2_{k-1} distribution; see, for e.g., Kendall and Stuart (1961).

We do not describe the exact rule how to construct the subsets $\{A_1, A_2, \ldots, A_r\}$, but we recommend to perform some experiments for finding the parameters, which make the sample size minimal (or, at least, acceptable). The point is that there are many cryptographic and other applications where it is possible to implement some experiments for optimizing the parameter values and, then, to test hypothesis basing on independent data. For example, in case of testing a PRNG it is possible to seek suitable parameters using a part of generated sequence and then to test the PRNG using a new part of the sequence.

Let us consider a simple example. Let $A = \{a_1, \ldots, a_6\}$, $r = 2$, $A_1 = \{a_1, a_2, a_3\}$, $A_2 = \{a_4, a_5, a_6\}$, $x_1 \ldots x_8 = a_3 a_6 a_3 a_3 a_6 a_1 a_6 a_1$. If $v_1 = 1, 2, 3, 4, 5, 6$, then $v_2 = 3, 1, 2, 4, 5, 6$, $v_3 = 6, 3, 1, 2, 4, 5$, etc., and $n_1 = 7$, $n_2 = 1$. We can see that the letters a_3 and a_6 are quite frequent and the "book stack" indicates this nonuniformity quite well. (Indeed, the average values of n_1 and n_2 equal 4, whereas the real values are 7 and 1, correspondingly.)

Examples of practical applications of this test will be given in Section 4, but here we make two notes. Firstly, we pay attention to the complexity of this algorithm. The “naive” method of transformation according to (6) could take the number of operations proportional to S, but there exist algorithms, which can perform all operations in (6) using $O(\log S)$ operations. Such algorithms can be based on AVL-trees, see, for e.g., Aho et al. (1976).

The last comment concerns with the name of the method. The “book stack” structure is quite popular in Information Theory and Computer Science. In Information Theory this structure was firstly suggested as a basis of an universal code by Ryabko (1980) and was rediscovered by Bently et al. (1986) and Elias (1987) (see also a comment of Ryabko, 1987 about a history of this code). In English language literature this code is frequently called as “Move-to-Front” (MTF) scheme as it was suggested by Bently et al. (1986). Now this data structure is used in a caching and many other algorithms in Computer Science under the name “Move-to-Front”. It is also worth noting that the book stack was firstly considered by a soviet mathematician M.L. Cetlin as an example of a self-adaptive system in 1960s, see Rozanov (1971).

3.2. The order test

This test is also based on changing the order $v^t(a)$ of alphabet letters but the rule of the order change differs from (6). To describe the rule we first define $\lambda^{t+1}(a)$ as a count of occurrences of a in the word $x_1 \dots x_{t-1}x_t$. At each moment t the alphabet letters are ordered according to v^t in such a way that, by definition, for each pair of letters a and b $v^t(a) \prec v^t(b)$ if $\lambda^t(a) \leqslant \lambda^t(b)$. For example, if $A = \{a_1, a_2, a_3\}$ and $x_1x_2x_3 = a_3a_2a_3$, the possible orders can be as follows: $v^1 = (1, 2, 3)$, $v^2 = (3, 1, 2)$, $v^3 = (3, 2, 1)$, $v^4 = (3, 2, 1)$. In all other respects this method coincides with the book stack. (The set of all indexes $\{1, \dots, S\}$ is divided into r subsets, etc.)

Obviously, after observing each letter x_t the value $\lambda^t(x_t)$ should be increased and the order v^t should be changed. It is worth noting that there exist a data structure and algorithm, which allow maintaining the alphabet letters ordered in such a way that the number of operations spent is constant, independently of the size of the alphabet. This data structure was described by Moffat (1999) and Ryabko and Rissanen (2003).

3.3. Two-faced processes and the choice of the block length for a process testing

There are quite many methods for testing H_0 and H_1, where the bit stream is divided into words (blocks) of the length $s, s \geqslant 1$, and the sequence of the blocks $x_1x_2 \dots x_s$, $x_{s+1} \dots x_{2s}, \dots$, is considered as letters, where each letter belongs to the alphabet $B_s = \{0, 1\}^s$ and has the probability 2^{-s}, if H_0 is true. For instance, both above described tests, methods from Ryabko et al. (2004) and quite many other algorithms belong to this kind. That is why the questions of choosing the block length s will be considered here.

As it was mentioned in the introduction there exist two-faced processes, which, on the one hand, are far from being truly random, but, on the other hand, they can be distinguished from truly random only in the case when the block length s is large. From the information theoretical point of view the two-faced processes can be simply described as follows. For a two-faced process, which generates letters from $\{0, 1\}$, the limit Shannon entropy is (much) less than 1 and, on the other hand, the s-order entropy (h_s) is maximal ($h_s = 1$ bit per letter) for relatively large s.

We describe two families of two-faced processes $T(k, \pi)$ and $\bar{T}(k, \pi)$, where $k=1, 2, \ldots,$ and $\pi \in (0, 1)$ are parameters. The processes $T(k, \pi)$ and $\bar{T}(k, \pi)$ are Markov chains of the connectivity (memory) k, which generate letters from $\{0, 1\}$. It is convenient to define them inductively. The process $T(1, \pi)$ is defined by conditional probabilities $P_{T(1,\pi)}(0/0) = \pi$, $P_{T(1,\pi)}(0/1) = 1 - \pi$ (obviously, $P_{T(1,\pi)}(1/0) = 1 - \pi$, $P_{T(1,\pi)}(1/1) = \pi$). The process $\bar{T}(1, \pi)$ is defined by $P_{\bar{T}(1,\pi)}(0/0) = 1 - \pi$, $P_{\bar{T}(1,\pi)}(0/1) = \pi$. Assume that $T(k, \pi)$ and $\bar{T}(k, \pi)$ are defined and describe $T(k + 1, \pi)$ and $\bar{T}(k + 1, \pi)$ as follows

$$P_{T(k+1,\pi)}(0/0u) = P_{T(k,\pi)}(0/u), \quad P_{T(k+1,\pi)}(1/0u) = P_{T(k,\pi)}(1/u),$$

$$P_{T(k+1,\pi)}(0/1u) = P_{\bar{T}(k,\pi)}(0/u), \quad P_{T(k+1,\pi)}(1/1u) = P_{\bar{T}(k,\pi)}(1/u)$$

and vice versa,

$$P_{\bar{T}(k+1,\pi)}(0/0u) = P_{\bar{T}(k,\pi)}(0/u), \quad P_{\bar{T}(k+1,\pi)}(1/0u) = P_{\bar{T}(k,\pi)}(1/u),$$

$$P_{\bar{T}(k+1,\pi)}(0/1u) = P_{T(k,\pi)}(0/u), \quad P_{\bar{T}(k+1,\pi)}(1/1u) = P_{T(k,\pi)}(1/u)$$

for each $u \in B_k$ (here vu is a concatenation of the words v and u). For example,

$$P_{T(2,\pi)}(0/00) = \pi, \quad P_{T(2,\pi)}(0/01) = 1 - \pi,$$

$$P_{T(2,\pi)}(0/10) = 1 - \pi, \quad P_{T(2,\pi)}(0/11) = \pi.$$

The following theorem shows that the two-faced processes exist.

Theorem 3. *For each $\pi \in (0, 1)$ the s-order Shannon entropy (h_s) of the processes $T(k, \pi)$ and $\bar{T}(k, \pi)$ equals 1 bit per letter for $s = 0, 1, \ldots, k$ whereas the limit Shannon entropy (h_∞) equals $-(\pi \log_2 \pi + (1 - \pi) \log_2(1 - \pi))$.*

The proofs of the theorem is given in the appendix, but here we consider examples of "typical" sequences of the processes $T(1, \pi)$ and $\bar{T}(1, \pi)$ for π, say, $\frac{1}{5}$. Examples are: 010101101010100101 ... and 000011111000111111000 ... . We can see that each sequence contains approximately one half of 1's and one half of 0's. (That is why the first order Shannon entropy is 1 per a letter.) On the other hand, both sequences do not look like truly random, because they, obviously, have too long subwords like either 101010.. or 000..11111... (In other words, the second order Shannon entropy is much less than 1 per letter.) Hence, if a randomness test is based on estimation of frequencies of 0's and 1's only, then such a test will not be able to find deviations from randomness.

So, if we revert to the question about the block length of tests and take into account the existence of two-faced processes, it seems that the block length could be taken as large as possible. But it is not so. The following informal consideration could be useful for choosing the block length. The point is that statistical tests can be applied if words from the sequence

$$x_1 x_2 \ldots x_s, x_{s+1} \ldots x_{2s}, \ldots, x_{(m-1)s+1} x_{(m-1)s+2} \ldots x_{ms} \tag{8}$$

are repeated (at least a few times) with high probability (here ms is the sample length). Otherwise, if all words in (8) are unique (with high probability) when H_0 is true, a sensible test cannot be constructed basing on a division into s-letter words. So, the word length s should be chosen in such a way that some words from sequence (8) are repeated with high probability, when H_0 is true. So, now our problem can be formulated as follows. There is a binary sequence $x_1x_2 \dots x_n$ generated by the Bernoulli source with $P(x_i=0)=P(x_i=1)=\frac{1}{2}$ and we want to find such a block length s that sequence (8) with $m = \lfloor n/s \rfloor$, contains some repetitions (with high probability). This problem is well known in the probability theory and sometimes called as the birthday problem. Namely, the standard statement of the problem is as follows. There are $S = 2^s$ cells and m $(=n/s)$ pellets. Each pellet is put in one of the cells with the probability $1/S$. It is known in Probability Theory that, if $m=c\sqrt{S}$, $c > 0$ then the average number of cells with at least two pellets equals $c^2(1/2 + \circ(1))$, where S goes to ∞; see Kolchin et al. (1976). In our case, the number of cells with at least two pellets is equal to the number of the words from sequence (8) which are met two (or more) times. Having into account that $S=2^s$, $m=n/s$, we obtain from $m=c\sqrt{S}$, $c > 0$ an informal rule for choosing the length of words in (8):

$$n \asymp s2^{s/2}, \tag{9}$$

where n is the length of a sample $x_1x_2 \dots x_n$, s is the block length. If s is much larger, sequence (8) does not have repeated words (in case H_0) and it is difficult to build a sensible test. On the other hand, if s is much smaller, large classes of the alternative hypotheses cannot be tested (due to existence of the two-faced processes). It is worth noting that it is impossible to have a universal choice of s, because it is impossible to avoid the two-faced phenomenon. In other words, this fact can be explained basing on the following known result of Information Theory: it is impossible to have guaranteed rate of code convergence universally for all ergodic sources; see Bailey (1976) and Ryabko (1984). That is why, it is impossible to choose a universal length s. On the other hand, there are many applications where the word length s can be chosen experimentally. (But, of course, such experiments should be performed on the independent data.)

4. The experiments

In this part, we describe some experiments carried out to compare new tests with known ones. We will compare order test, book stack test, tests which are based on standard data compression methods, and tests from Rukhin et al. (2001). The point is that the tests from Rukhin and others are selected basing on comprehensive theoretical and experimental analysis and can be considered as the state-of-the-art in randomness testing. Besides, we will also test the method recently published by Ryabko et al. (2004), because it was published later than the book of Rukhin et al.

We used data generated by the PRNG “RANDU” (described in Dudewicz and Ralley, 1981) and random bits from “The Marsaglia Random Number CDROM”, see: http://stat.fsu.edu/diehard/cdrom/. RANDU is a linear congruent generators (LCG), which is defined by the following equality:

$$X_{n+1} = (AX_n + C) \bmod M,$$

where X_n is nth generated number. RANDU is defined by parameters $A = 2^{16} + 3$, $C = 0$, $M = 2^{31}$, $X_0 = 1$. Those kinds of sources of random data were chosen because random bits from "The Marsaglia Random Number CDROM" are considered as good random numbers, whereas it is known that RANDU is not a good PRNG. It is known that the lowest digits of X_n are "less random" than the leading digits (Knuth, 1981), that is why in our experiments with RANDU we extract an eight-bit word from each generated X_i by formula $\hat{X}_i = \lfloor X_i/2^{23} \rfloor$.

The behavior of the tests was investigated for files of different lengths (see the tables below). We generated 100 different files of each length and applied each mentioned above test to each file with level of significance 0.01 (or less, see below). So, if a test is applied to a truly random bit sequence, on average 1 file from 100 should be rejected. All results are given in the tables, where integers in boxes are the number of rejected files (from 100). If a number of the rejections is not given for a certain length and test, it means that the test cannot be applied for files of such a length.

Table 1 contains information about testing of sequences of different lengths generated by RANDU, whereas Table 2 contains results of application of all tests to 5 000 000-bit sequences either generated by RANDU or taken from "The Marsaglia Random Number CDROM". For example, the first number of the second row of the Table 1 is 56. It means that there were 100 files of the length 5×10^4 bits generated by PRNG RANDU. When the Order test was applied, the hypothesis H_0 was rejected 56 times from 100 (and, correspondingly, H_0 was accepted 44 times.) The first number of the third line shows that H_0 was rejected 42 times, when the Book stack test was applied to the same 100 files. The third number of the second line shows that the hypothesis H_0 was rejected 100 times, when the Order test was applied for testing of 100 100000-bit files generated by RANDU, etc.

Let us first give some comments about the tests, which are based on popular data compression methods RAR and ARJ. In those cases we applied each method to a file and first estimated the length of compressed data. Then we use the test $\Gamma^{(n)}_{\alpha,\varphi}$ with the critical value (1) as follows. The alphabet size $|A| = 2^8 = 256$, $n \log |A|$ is simply the length of file (in bits) before compression, (whereas n is the length in bytes). So, taking $\alpha = 0.01$, from (1), we see that the hypothesis about randomness (H_0) should be rejected, if the length of compressed file is less than or equal to $n \log |A| - 8$ bits. (Strictly speaking, in this case $\alpha \leqslant 2^{-7} = \frac{1}{128}$.) So, taking into account that the length of computer files is measured in bytes, this rule is very simple: if the n-byte file is really compressed (i.e. the length of the encoded file is $n - 1$ bytes or less), this file is not random (and H_0 is rejected). So, the following tables contain numbers of cases, where files were really compressed.

Let us now give some comments about parameters of the considered methods. As it was mentioned, we investigated all methods from the book of Rukhin et al. (2001), the test of Ryabko et al. (2004) (RSS test for short), the described above two tests based on data compression algorithms, the order tests and the book stack test. For some tests there are parameters, which should be specified. In such cases the values of parameters are given in

Table 1
Number of files generated by PRNG RANDU and recognized as non-random for different tests and different file lengths (in bits)

Name of test/Length of file	5×10^4	10^5	5×10^5	10^6
Order test	56	100	100	100
Book stack	42	100	100	100
Parameters for both tests		$s = 20,\ \lvert A_1 \rvert = 5\sqrt{2^s}$		
RSS	4	75	100	100
Parameters	$s = 16$	$s = 17$	$s = 20$	
RAR	0	0	100	100
ARJ	0	0	99	100
Frequency	2	1	1	2
Block frequency	1	2	1	1
Parameters	$M = 1000$	$M = 2000$	$M = 10^5$	$M = 20\,000$
Cumulative sums	2	1	2	1
Runs	0	2	1	1
Longest run of ones	0	1	0	0
Rank	0	1	1	0
Discrete Fourier transform	0	0	0	1
Nonoverlapping templates	—	—	—	2
Parameters				$m = 10$
Overlapping templates	—	—	—	2
Parameters				$m = 10$
Universal statistical	—	—	1	1
Parameters			$L = 6$	$L = 7$
			$Q = 640$	$Q = 1280$
Approximate entropy	1	2	2	7
Parameters	$m = 5$	$m = 11$	$m = 13$	$m = 14$
Random excursions	—	—	—	2
Random excursions variant	—	—	—	2
Serial	0	1	2	2
Parameters	$m = 6$	$m = 14$	$m = 16$	$m = 8$
Lempel–Ziv complexity	—	—	—	1
Linear complexity	—	—	—	3
Parameters				$M = 2500$

the table in the row, which follows the test results. There are some tests from the book of Rukhin et al., where parameters can be chosen from a certain interval. In such cases we repeated all calculations three times, taking the minimal possible value of the parameter, the maximal one and the average one. Then the data for the case when the number of rejections of the hypothesis H_0 is maximal, is given in the table.

The choice of parameters for RSS, the book stack test and the order test was made on the basis of special experiments, which were carried out for independent data. (Those algorithms are implemented as a Java program and can be found on the internet, see http://web.ict.nsc.ru/~rng/.) In all cases such experiments have shown that for all three algorithms the optimal blocklength is close to the one defined by informal equality (9).

We can see from the tables that the new tests can detect non-randomness more efficiently

Table 2
Number of 5 000 000-bit files generated by PRNG RANDU and random, which are recognized as non-random

Name of test/ Kind of file	*RANDU*	*Random*
Order test	100	3
Book stack	100	0
Parameters for both tests	$s = 24$, $\lvert A_1\rvert = 5\sqrt{2^s}$	
RSS	100	1
Parameters	$s = 24$	$s = 24$
RAR	100	0
ARJ	100	0
Frequency	2	1
Block frequency	2	1
Parameters	$M = 10^6$	$M = 10^5$
Cumulative sums	3	2
Runs	2	2
Longest run of ones	2	0
Rank	1	1
Discrete Fourier transform	89	9
Nonoverlapping templates	5	5
Parameters	$m = 10$	$m = 10$
Overlapping templates	4	1
Parameters	$m = 10$	$m = 10$
Universal statistical	1	2
Parameters	$L = 9$	$L = 9$
	$Q = 5120$	$Q = 5120$
Approximate entropy	100	89
Parameters	$m = 17$	$m = 17$
Random excursions	4	3
Random excursions variant	3	3
Serial	100	2
Parameters	$m = 19$	$m = 19$
Lempel–Ziv complexity	0	0
Linear complexity	4	3
Parameters	$M = 5000$	$M = 2500$

than the known ones. Seemingly, the main reason is that RSS, book stack tests and order test deal with such large blocklength as it is possible, whereas many other tests are focused on other goals. The second reason could be an ability for adaptation. The point is that the new tests can find subwords, which are more frequent than others, and use them for testing, whereas many other tests are looking for particular deviations from randomness.

In conclusion, we can say that the obtained results show that the new tests, as well as the ideas of Information Theory in general, can be useful tools for randomness testing.

Acknowledgements

The authors wish to thank one of anonymous reviewers for information about a unpublished thesis of David Harold Bailey.

Appendix A

Proof of Theorem 1. First, we estimate the number of words $\varphi_n(u)$ whose length is less than or equal to an integer τ. Obviously, at most one word can be encoded by the empty codeword, at most two words by the words of the length $1, \ldots,$ at most 2^i can be encoded by the words of length i, etc. Having taken into account that the codewords $\varphi_n(u) \neq \varphi_n(v)$ for different u and v, we obtain the inequality

$$|\{u : |\varphi_n(u)| \leqslant \tau\}| \leqslant \sum_{i=0}^{\tau} 2^i = 2^{\tau+1} - 1.$$

From this inequality and (1) we can see that the number of words from the set $\{A^n\}$, whose codelength is less than or equal to $t_\alpha = n \log |A| - \log(1/\alpha) - 1$, is not greater than $2^{n \log |A| - \log(1/\alpha)}$. So, we obtained that

$$|\{u : |\varphi_n(u)| \leqslant t_\alpha\}| \leqslant \alpha |A|^n.$$

Taking into account that all words from A^n have equal probabilities if H_0 is true, we obtain from the last inequality, (1) and the description of the test $\Gamma_{\alpha,\varphi}^{(n)}$ that

$$Pr\{|\varphi_n(u)| \leqslant t_\alpha|\} \leqslant (\alpha |A|^n / |A|^n) = \alpha$$

if H_0 is true. The theorem is proved. □

Proof of Claim 1. The proof is based on a direct calculation of the probability of rejection for a case where H_0 is true. From the description of the test $\Upsilon_{\alpha,\varphi}^{(n)}$ and definition of g (see (8)) we obtain the following chain of equalities:

$$\begin{aligned}
&Pr\{H_0 \; is \; rejected\} \\
&\quad = Pr\{|\varphi_n(u)| \leqslant g\} + Pr\{|\varphi_n(u)| = g+1\} \\
&\qquad \times \left(1 - \left(\sum_{j=1}^{g+1} |A_j| - \alpha |A|^n\right) \Big/ |A_{g+1}|\right) \\
&\quad = \frac{1}{A^n} \left(\sum_{j=0}^{g} |A_j| + |A_{g+1}|\right. \\
&\qquad \left. \times \left(1 - \left(\sum_{j=1}^{g+1} |A_j| - \alpha |A|^n\right) \Big/ |A_{g+1}|\right)\right) = \alpha.
\end{aligned}$$

The claim is proved. □

Proof of Claim 2. We can think that $\hat{t}_\alpha$ in (4) is an integer. (Otherwise, we obtain the same test taking $\lfloor \hat{t}_\alpha \rfloor$ as a new critical value of the test.) From the Kraft inequality (3) we obtain that

$$1 \geqslant \sum_{u \in A^n} 2^{-|\varphi_n(u)|} \geqslant |\{u : |\varphi_n(u)| \leqslant \hat{t}_\alpha\}| 2^{-\hat{t}_\alpha}.$$

This inequality and (4) yield:

$$|\{u : |\varphi_n(u)| \leqslant \hat{t}_\alpha\}| \leqslant \alpha |A|^n.$$

If H_0 is true then the probability of each $u \in A^n$ equals $|A|^{-n}$ and from the last inequality we obtain that

$$Pr\{|\varphi(u)| \leqslant \hat{t}_\alpha\} = |A|^{-n} |\{u : |\varphi_n(u)| \leqslant \hat{t}_\alpha\}| \leqslant \alpha,$$

if H_0 is true. The claim is proved.

Proof of Theorem 3. We prove the theorem for the process $T(k, \pi)$, but this proof is valid for $\bar{T}(k, \pi)$, too. First we show that

$$p^*(x_1 \ldots x_d) = 2^{-d}, \tag{10}$$

$(x_1 \ldots x_d) \in \{0, 1\}^d$, $d = 1, \ldots, k$, is a stationary distribution for the processes $T(k, \pi)$ (and $\bar{T}(k, \pi)$) for all $k = 1, 2, \ldots,$ and $\pi \in (0, 1)$. For any values of k, $k \geqslant 1$, (10) will be proved if we show that the system of

$$\begin{aligned} P_{T(k,\pi)}(x_1 \ldots x_d) = {} & P_{T(k,\pi)}(0x_1 \ldots x_{d-1}) P_{T(k,\pi)}(x_d/0x_1 \ldots x_{d-1}) \\ & + P_{T(k,\pi)}(1x_1 \ldots x_{d-1}) P_{T(k,\pi)}(x_d/1x_1 \ldots x_{d-1}) \end{aligned}$$

has the solution $p(x_1 \ldots x_d) = 2^{-d}$, $(x_1 \ldots x_d) \in \{0, 1\}^d$, $d = 1, 2, \ldots, k$. It can be easily seen for $d = k$, if we take into account that, by definition of $T(k, \pi)$ and $\bar{T}(k, \pi)$, the equality $P_{T(k,\pi)}(x_k/0x_1 \ldots x_{k-1}) + P_{T(k,\pi)}(x_k/1x_1 \ldots x_{k-1}) = 1$ is valid for all $(x_1 \ldots x_k) \in \{0, 1\}^k$. From this equality and the law of total probability we immediately obtain (10) for $d < k$.

Let us prove the second claim of the theorem. From the definition $T(k, \pi)$ and $\bar{T}(k, \pi)$ we can see that either $P_{T(k,\pi)}(0/x_1 \ldots x_k) = \pi$, $P_{T(k,\pi)}(1/x_1 \ldots x_k) = 1 - \pi$ or $P_{T(k,\pi)}(0/x_1 \ldots x_k) = 1 - \pi$, $P_{T(k,\pi)}(1/x_1 \ldots x_k) = \pi$. That is why $h(x_{k+1}/x_1 \ldots x_k) = -(\pi \log_2 \pi + (1 - \pi) \log_2(1 - \pi))$ and, hence, $h_\infty = -(\pi \log_2 \pi + (1 - \pi) \log_2(1 - \pi))$. The theorem is proved. □

References

Aho, A.V., Hopcroft, J.E., Ulman, J.D., 1976. The Design and Analysis of Computer Algorithms. Addison-Wesley, Reading, MA.

Bailey, D.H., 1976. Sequential schemes for classifying and predicting ergodic processes. Ph.D. Dissertation, Stanford University.

Bently, J.L., Sleator, D.D., Tarjan, R.E., Wei, V.K., 1986. A locally adaptive data compression scheme. Comm. ACM 29, 320–330.

Billingsley, P., 1965. Ergodic Theory and Information. Wiley, New York.

Dudewicz, E.J., Ralley, T.G., 1981. The Handbook of Random Number Generation and Testing With TESTRAND Computer Code. American Series in Mathematical and Management Sciences, vol. 4. American Sciences Press, Columbus, OH.
Elias, P., 1987. Interval and recency rank source coding: two on-line adaptive variable-length schemes. IEEE Trans. Inform. Theory 33 (1), 3–10.
Gallager, R.G., 1968. Information Theory and Reliable Communication. Wiley, New York.
Kendall, M.G., Stuart, A., 1961. The Advanced Theory of Statistics, vol. 2. Inference and Relationship, London.
Knuth, E.E., 1981. The art of computer programming, vol. 2. Addison-Wesley, Reading, MA.
Kolchin, V.F., Sevast'yanov, B.A., Chistyakov, V.P., 1976. The Random Allocation. Nauka, Moscow.
Kolmogorov, A.N., 1965. Three approaches to the quantitative definition of information. Problems Inform. Transmission 1, 3–11.
L'Ecuyer, P., 1994. Uniform random numbers generation. Ann. Oper. Res.
Li, M., Vitanyi, P., 1997. An Introduction to Kolmogorov Complexity and its Applications. 2nd Edition. Springer, New York.
Maurer, U., 1992. A universal statistical test for random bit generators. J. Cryptol. 5 (2), 89–105.
Mezenes, A., van Oorschot, P., Vanstone, S., 1996. Handbook of Applied Cryptography. CRC Press, Boca Raton, FL.
Moffat, A., 1999. An improved data structure for cumulative probability tables. Software Pract. Exper. 29 (7), 647–659.
Rozanov, Yu.A., 1971. The Random Processes. Nauka, Moscow.
Rukhin, A., et al., 2001. A statistical test suite for random and pseudorandom number generators for cryptographic applications. NIST Special Publication 800-22 (with revision dated May 15, 2001). http://csrc.nist.gov/rng/SP800-22b.pdf.
Ryabko, B.Ya., 1980. Information compression by a book stack. Problems Inform. Transmission 16 (4), 16–21.
Ryabko, B.Ya., 1984. Twice-universal coding. Problems Inform. Transmission 3, 173–177.
Ryabko, B.Ya., 1987. A locally adaptive data compression scheme (Letter). Comm. ACM 30 (9), 792.
Ryabko, B., Rissanen, J., 2003. Fast adaptive arithmetic code for large alphabet sources with asymmetrical distributions. IEEE Comm. Lett. 7 (1), 33–35.
Ryabko, B.Ya., Stognienko, V.S., Shokin, Yu.I., 2004. A new test for randomness and its application to some cryptographic problems. J. Statist. Plann. Inference 123, 365–376.