Physics – Uspekhi **67** (9) 919 – 937 (2024)

METHODOLOGICAL NOTES PACS numbers: **02.50. – r**, **03.67. – a**, 42.50.Ex

Quantum random number generators, extraction of provably random bit sequences from Markov chain trajectories

I M Arbekov, S N Molotkov

DOI: https://doi.org/10.3367/UFNe.2024.02.039658

Contents

Abstract. One of the main problems in the construction of quantum random number generators — obtainment of a provably random output sequence from physical measurements, i.e., from an initial sequence generated by a physical random number generator — is investigated. The conceptual feasibility and the conditions under which randomness can be 'reached,' as well as the meaning of the 'provable randomness' term, are discussed. We consider the methods of extracting provably random bit sequences from stationary Markov chains of finite order, i.e., under the assumption of the finite depth of the dependence of the results of physical measurements on the prehistory, which is an adequate approximation of the real situation. The extraction of the output provably random bit sequence from the initial sequence of the results of physical measurements using V F Babkin's effective arithmetic coding method is demonstrated. It is shown that random bit sequences can be provably obtained even from primary sequences of the results of physical measurements, which are dependent on (correlated to) any finite depth (prehistory). We aim to reveal the connection of various approximations employed in the development and description of methods for obtaining random bit sequences with the basic physical limitations of Nature. The mathematical proofs are detailed to the level of practical algorithms applied in real random number generators. The necessary mathematical propositions are presented at an intuitive level for physicists, do not require prior knowledge in this area, and are comprehensible to undergraduate university students.

Keywords: quantum random number generators, Markov chains, random bit sequences

Truth is much too complicated
to allow anything but approximations.
John von Neumann

1. Introduction

Random numbers are widely used in various fields of science and technology: computer passwords, smart card PIN codes, and other electronic devices. The most important application of random sequences is in quantum cryptography systems — quantum key distribution, where a large number of random bits are required. In quantum cryptography systems, up to 10^8 random bits are needed to form one common key when distributing secret keys. High-speed random number generators are required to generate large-volume random sequences.

All random number generators can be divided into two types: *mathematical* and *physical*.

Mathematical generators, often called software random number generators (SRNGs), are based on a mathematical

I M Arbekov [1, a], **S N Molotkov** [1, 2, 3, 4, b]
[1] Academy of Cryptography of the Russian Federation, PO Box 100, 119331 Moscow, Russian Federation
[2] Osipyan Institute of Solid State Physics, Russian Academy of Sciences, ul. Akademika Osip'yana 2, 142432 Chernogolovka, Moscow region, Russian Federation
[3] Lomonosov Moscow State University, Faculty of Computational Mathematics and Cybernetics, Leninskie gory 1, str. 52, 119991 Moscow, Russian Federation
[4] Lomonosov Moscow State University, Quantum Technology Center, Leninskie gory 1, str. 35, 119991 Moscow, Russian Federation
E-mail: [a] arbekov53@mail.ru, [b] molotkov@issp.ac.ru

Received 25 December 2023, revised 20 February 2024
Uspekhi Fizicheskikh Nauk **194** (9) 974 – 993 (2024)
Translated by M Zh Shmatikov

transformation, usually recursive, of a certain seed. The algorithm for the mathematical transformation is publicly known; it is only the seed which is unknown. If the seed is known, the entire subsequent bit sequence is also known. For this reason, mathematical generators only produce a pseudo-random bit sequence, the 'randomness' of which is based only on an unknown seed.

Physical random number generators (PRNGs) operate based on processing the results of measurements of some physical system.

PRNGs can also be divided into two types: *classical* and *quantum*.

Classical generators are based on extracting randomness from some physical process, the evolution of which in time is described by the laws of classical physics. The evolution of any classical system, even an arbitrarily complex one, can be described by differential equations. The sequences that are obtained at the output of such a generator can hardly be called truly random, since they are completely determined by the initial conditions. In the classical domain, randomness in observing a physical system arises only as a result of the uncertainty of the initial conditions. If initial conditions are known and the evolution of a classical system is the same, the result of observations is completely deterministic.

Quantum generators are based on measuring some quantum system. Unlike classical physics, measurements in a quantum system, each time prepared in a certain and the same state, yield a random result, which is a fundamental law of Nature in the microworld. In implementing a PRNG, it is desirable to find a suitable physical system, the results of measurements in which would be of a purely quantum nature.

It seems that true randomness only exists in the quantum domain, in the sense that the result of a measurement in a quantum system, each time prepared under the same initial conditions, is fundamentally unpredictable.

In the quantum domain, probability is built into the mathematical description of a measurement in a quantum system.

Nevertheless, the following fundamental questions arise.

• What is meant by true randomness?

• Do the fundamental laws of Nature allow us to 'reach' true randomness in real physical experiments and devices? Or does Nature only allow approaching true randomness, in the spirit of von Neumann's statement quoted as an epigraph?

• In what way can it be verified that the resulting output bit sequences are truly random?

The implementation of any PRNG, either classical or quantum, includes the following stages:

— selection of a physical system (the source of 'noise'), the measurements of which yield the initial random sequence;

— estimation of the amount of randomness — the number of random 0s and 1s that can be extracted from the initial random sequence;

— extraction of random 0s and 1s from the initial random sequence — post-processing of the initial sequence, i.e., transformation of the initial random sequence into an output bit sequence;

— proof of the fact that the output bit sequence, under the approximations accepted in describing the selected physical system, is a truly random bit sequence in which the probabilities of 0 and 1 are equal to $1/2$, and all positions in the sequence are independent of each other.

It turns out that there are fundamental limitations on the generation of an initial sequence of independent results of physical measurements.

There are basic limitations on the speed of random number generation in quantum generators, which are related to the fact that the spectrum of any stable physical system must lie on the positive energy (frequency) semi-axis.[1] This fact is formalized by the well-known Wiener–Paley theorem.

• Any physical random process, either quantum or classical, is characterized by a correlation function and a spectral power density, related to each other by a pair of Fourier transforms. Physical limitations require that the spectral power density vanish at frequency values below a certain threshold. In this case, the rate of correlation function decay cannot be faster than (or equal to) exponential, which is due to the fundamental Wiener–Paley theorem. This assertion implies that the measurement results extracted from a random process at different moments in time turn out to be correlated (dependent). Measurements become formally uncorrelated only when the moments of measurement are separated in time by an infinite interval.

We now consider the spectral density

$$g(\omega) = \begin{cases} 0\,, & \omega < \omega_{\min}\,, \\ g'(\omega)\,, & \omega \geqslant \omega_{\min} > -\infty \end{cases}$$

and the correlation function

$$R(t) = \int_{-\infty}^{\infty} g(\omega) \exp\left(-\mathrm{i}\omega t\right) \mathrm{d}\omega\,.$$

According to the Wiener–Paley theorem [1, 2], for any square-integrable function, the following integral must converge:

$$\int_{-\infty}^{\infty} \frac{\left|\ln |R(t)|\right|}{1+t^2}\, \mathrm{d}t < \infty\,.$$

This condition implies that the decay rate of the function $|R(t)|$ must be slower than exponential:

$$\left|R(t)\right| \underset{t\to\infty}{\sim} \exp\left(-ct^q\right), \quad c > 0\,, \quad q < 1\,.$$

This formally shows that measurements can be independent only if the time interval between them is infinite.

Note that fundamental restrictions on the decay rate of correlations in time lead to the fact that α-decay[2] cannot be strictly exponential at large and small times (see details in [2]). The fact that the spectrum of a stable physical system lies on the positive energy (frequency) axis sets fundamental restrictions on the maximum rate of generation of random bit sequences. This issue was studied in [3, 4].

Why is the independence of the initial sequence of results of physical measurements so important? If it were possible to achieve independence between successive measurements in a finite time, the problem of high-speed PRNGs would be solved in principle, since provably efficient methods are available for extracting truly random bit sequences from an initial sequence of independent measurements (see below and [5]). However, the basic limitations of Nature do not allow

[1] Naturally, the choice of the point on the semiaxis from which the energies are measured is of no importance.

[2] It should be noted that attempts were previously made to use α-decay to generate random numbers, but this method did not find further application due to technical difficulties and low speed.

obtaining the initial sequence as a sequence of independent measurements.

The only thing that can be done is to limit ourselves to a finite depth of dependence on the history and assume that the conditional probability of any measurement result depends only on r previous outcomes, i.e., that the depth of correlations is finite.

In this case, we are talking about Markov chains of order r. This assumption that the initial sequence is a stationary Markov chain of finite order r is the broadest assumption under which a constructive construction of a provably random output sequence is generally possible (see below).

We now discuss the issue of extracting bit sequences from the initial sequence of measurement results of a physical system.

Methods for extracting random bit sequences from measurement results can be divided into two classes.

1. Probabilistic extractors.
2. Deterministic extractors.

It was previously shown in [5] that deterministic extractors work effectively for initial sequences of independent measurements.

The PRNG output sequence is considered acceptable for use in cryptography if it is obtained in accordance with the Bernoulli equiprobable trial scheme, i.e., it is a realization of a sequence of independent and equiprobable random variables. A well-known approach to testing this assumption, which involves the use of a set of statistical criteria for the agreement of the output sequence values with the hypothesis of independence and equiprobability of the generated data, is most fully presented in [6]. Theoretical issues related to finding the limit distributions of the corresponding statistics are presented in detail in book [7].

For a long time, this approach was the only instrument for checking the quality of output sequences, but its limitations are quite obvious: the criteria of agreement are successfully fulfilled by the output sequences of SRNGs, where what is random is the so-called seed — a bit segment of finite length — which is significantly smaller than the length of the sequence generated by the SRNG. The output sequence of the SRNG can hardly be considered to be obtained in accordance with the scheme of equiprobable Bernoulli trials, at least from cardinality considerations: the cardinality of the set of SRNG sequences is limited by that of the set of 'seed' binary vectors.

Informally speaking, this implies that the number of truly random bits in the output pseudorandom sequence cannot be greater than the number of seeds in the bit representation.

It may be asserted that the above-mentioned cardinality considerations, the numerical 'equivalent' of which is the concept of Shannon's limiting entropy, were used as the conceptual basis of the technique [8].

Technique [8] contains a number of tests (estimates) of the minimum entropy $H_{\min}$ per symbol. The minimum entropy is a lower bound for Shannon's entropy and, when converted to a bit, satisfies the inequality $0 \leqslant H_{\min} \leqslant 1$. The maximum value $H_{\min} = 1$ is achieved for the Bernoulli equiprobable trial scheme. The minimum entropy is a very conservative estimate of the power, since it significantly underestimates Shannon's limiting entropy.

In obtaining a numerical estimate $H_{\min} < 1$, it is proposed to perform the appropriate compression — hashing of the original sequence.

The result of applying the hash function is the output sequence of a PRNG. The issue of choosing a hash function that is efficient in some sense remains open at the moment.

For a randomly selected hash function (probabilistic extractor), the quality of the output sequence can be estimated using the well-known Leftover Hash Lemma [9].

Let there be a probability distribution $P(X)$ on the original sequence $X = (x_1, x_2, \ldots, x_N) \in \{0,1\}^N$. Based on some model assumptions regarding $P(X)$, an estimate of the minimum-entropy is constructed:

$$H_{\min} = -\frac{1}{N}\log_2 \max_{X\in\{0,1\}^N} P(X)\,,$$
$$0 < H_{\min} < 1\,.$$

To obtain the output sequence $Y = (y_1, y_2, \ldots, y_\ell) \in \{0,1\}^\ell$, $\ell < N$ (extraction of random bits), compression–hashing is used:

$$g : \{0,1\}^N \to \{0,1\}^\ell\,.$$

Application of the hash function g induces a distribution on the bit string $Y = (y_1, y_2, \ldots, y_\ell)$:

$$P_g(Y) = \sum_{X\in\{0,1\}^N : g(X)=Y} P(X)\,.$$

For a randomly selected function g from the class of universal hash functions G [10], using the Leftover Hash Lemma, one can establish the validity of an inequality that characterizes the closeness of the probability distribution $P_g(Y)$ to an equiprobable distribution:

$$\sum_{g\in G} P(g) \sum_{Y\in\{0,1\}^\ell} \left| P_g(Y) - \frac{1}{2^\ell} \right| \leqslant \sqrt{2^{-NH_{\min}+\ell}}\,.$$

For a sufficiently large length N, the right side of the inequality becomes smaller than an arbitrarily small value of ε.

The main problem here is that estimating $H_{\min}$ requires difficult-to-control model assumptions about the properties of the original sequence as a result of physical measurements and additional randomness for choosing the hash function. Multiple use of the same randomness in choosing the hash function leads (for fixed N and ℓ) to an increase in the estimation boundary ε.

The question that is discussed here is whether it is possible, based on some approximations regarding the nature of the original sequence resulting from measurements, to construct a provably random output sequence, namely, one with probability $2^{-\ell}$, ℓ being the length of the sequence. In this case, individual bits, as random variables, are independent and equally probable.

A method for constructing (extracting) a provably random output sequence with probability $2^{-\ell}$ from a sequence of independent nonequiprobable trials is known (see, for example, [5]). This technique, which uses V F Babkin's arithmetic coding method [11], is implemented in experimentally developed quantum PRNGs [12–15] based on sequential detection of attenuated laser radiation photocounts.

We now outline some issues with implementing a quantum PRNG of this type.

The underlying cause of the statistical nature of photocounts when detecting laser radiation, which is fundamentally

quantum in nature, is due to the absorption of photons by atoms. Avalanche photodetectors do not distinguish between the number of photons, so two events are random: the presence ($*$) of a photocount in a time window or its absence ($\sqcup$).

Avalanche photodetectors have a finite recovery time after the detection event. If laser radiation is strongly attenuated, the average frequency of photocounts is small, and the detector has time to recover before the next detection event. In this case, it is assumed that statistical independence of successive photocounts, which is the original PRNG sequence, is ensured.

Assuming independence of successive photocounts as random variables, a provably random output sequence with a probability of $2^{-\ell}$ can be obtained at the output of a quantum PRNG, while knowledge of the probabilities themselves, $p = P(\sqcup)$ and $1 - p = P(*)$, is not required.

Due to the low average frequency of photocounts, such quantum PRNGs are fairly slow. An attempt to increase the speed of operation leads to an increase in the average frequency of photocounts and to a corresponding negative effect — the appearance of dependences in the original sequence, where the current outcome of measurements depends on previous outcomes.

In [16, 17], methods for extracting a random sequence from Markov chain trajectories are studied. Proposed in study [17] is an original method for extracting a provably random sequence from the trajectories of a simple ($r = 1$) Markov chain with a finite number of states; many proofs of its operation are quite complex and multi-stage.

Below, following the main concept of proving randomness in [17], which consists in partitioning the probability space into subsets of equivalent (equally probable) trajectories, we generalize the results of [17] to Markov chains of an arbitrary order.

Using the design features of random bit extraction in V F Babkin's arithmetic coding method [11], we first present a detailed proof of the randomness of output sequences for an independent source, which we then extend to Markov sequences.

Following the general concept of [17], it is shown that, for the initial PRNG sequences in the form of trajectories of a Markov chain with two states, of arbitrary order r, the output sequences have the same probability $2^{-\ell}$.

It is necessary to clarify once again what the term 'provable randomness' means.

It is shown that, if the correlation depth is finite, our method produces a truly random sequence of 0s and 1s at the output.

True randomness is usually understood as the fact that any position in the output sequence of 0s and 1s is realized strictly with probability 1/2, and each position is independent of the others.

It turns out that it is impossible to prove this statement straightforwardly, even applying the employed assumption of a finite correlation depth.

An equivalent statement is proved instead.

Namely, it is shown that the output sequences of 0s and 1s for any length ℓ contain all possible combinations of 0s and 1s,

$$\overbrace{(000\ldots000)}^{\ell}, \overbrace{(000\ldots001)}^{\ell}, \ldots, \overbrace{(111\ldots111)}^{\ell},$$

and all such sequences have the same probability $2^{-\ell}$, regardless of the correlation depth. In this case, individual bits, as random variables, are independent and equally probable, i.e., they are truly random.

This is the actual situation. Ideal randomness is too 'strong and complex' an information resource. As shown below, even under the approximations to the real situation used, proving true randomness is a task that is far from trivial.

The consideration consists of two stages.

In the first one, an effective and provable method for obtaining random sequences of 0s and 1s from a sequence of independent trials is presented.

In the second stage, it is shown how the problem of extracting provably random bits from the Markov chain trajectory can be reduced to the problem of extracting from a sequence of independent trials.

2. Source of independent states

If people do not believe that mathematics is simple,
it is only because they do not realize how complicated life is.
John von Neumann

Prior to showing how provable extraction of randomness from a sequence of independent trials is obtained, we present an informal explanation. The general idea can be understood using the example of tossing an asymmetric coin, which has the probability of getting Heads (H) p or Tails (T) of $1 - p$; the probability p is unknown; each toss is independent of the others.

We now split the entire sequence — obtained by tossing — into blocks of the same length but with different numbers H and T. Blocks with the same number of H and T have the same probability, equal to $p^{n_H}(1-p)^{n_T}$ (n_H, n_T are the numbers of H and T in the block), and differ only in the permutation of the symbols H and T, $n_H + n_T$ being the length of the block.

Blocks with the same number of H and T are assigned to one class. Let the number of blocks in some class be N_{n_H,n_T}. We renumber the blocks (sequences) in each class, starting the numbering from 0. We assume (as a simplification) that the number of equally probable blocks in a class is a power of 2, i.e., $N_{n_H,n_T} = 2^L$, where L is the number of binary digits required to represent numbers from 0 to $N_{n_H,n_T} - 1 = 2^L - 1$.

In other words, each ith block ($0 \leqslant i \leqslant 2^L - 1$) in the class of equally probable blocks is associated with a number whose binary representation yields a binary sequence of length L.

Binary sequences reproduce all possible combinations of 0s and 1s of length L_i once, and are therefore equally probable in their class.

Thus, by processing in the sequence of their appearance, blocks of measurement results (detector reading — H, no reading — T) — blocks of independent tests — and by sequentially connecting (concatenating) binary sequences, we obtain, provided that the readings are independent, truly random sequences of 0s and 1s.

Intuitively, the situation can be understood as follows: each block is realized with some probability $p_H^{n_H} p_T^{n_T}$; the source produces one of the truly random sequences of 0s and 1s of length L_i with equal probability. This is equivalent to having N sources, the number of which is equal to that of classes of sequences (N is the length of a block of Hs and Ts). Each source 'turns on' with probability $p_H^{n_H} p_T^{n_T}$ and produces one of the equally probable binary sequences of length L_i,

which are then concatenated into a common output sequence of 0s and 1s.

The question arises: if everything is clear enough at the intuitive level, what is the problem?

The problem is in the numbering of blocks in each class 'on the fly.' The point is that the number of possible blocks in a class is exponentially large. For example, with the length of the processed block $L = 64$ (which is associated with the architecture of a 64-bit processor), this number is $2^{64} \approx 10^{22}$. Numbering via a table requires computer memory of 10^{22} bits. To clearly illustrate the scale of such memory, we give the following example. Let the memory capacity of an 'average' flash drive be 1 GB (10^9 bits). A backpack can hold 1000 flash drives, a total of $1000 \times 10^9 = 10^{12}$ bits. Thus, 10^{10} — ten billion — backpacks are needed, or more than a backpack of flash drives for each inhabitant of Earth, to obtain the memory of $10^{10} \times 10^3 \times 10^9 = 10^{22}$ bits.

Apparently, this method of numbering is not feasible.

However, there is a method of numbering 'on the fly,' presented below, which only requires 64×64 bits of memory and 64 processing steps, i.e., the resources required are 64^3 bits.

We now proceed to a detailed derivation of the method of numbering 'on the fly' and extraction of bit sequences of 0s and 1s from independent Bernoulli sequences.

Let there be a source generating Bernoulli (independent) sequences of two symbols. Random sequences of 0s and 1s are extracted from the Bernoulli sequences in two steps.

Step 1: numbering of Bernoulli sequences 'on the fly' — Babkin's method.

Step 2: a block of random 0s and 1s is formed based on the obtained number of the Bernoulli sequence and consecutive blocks are concatenated into an output random sequence.

2.1 First stage: numbering of Bernoulli sequences — Babkin's method, binary alphabet

Let there be a source that generates symbols of a binary alphabet $A = \{s_1, s_2\}$.

Consider a sequence — a block of length n — which contains k symbols s_1. There are C_n^k such blocks in total. Let k symbols s_1 occur at positions $(i_1, i_2, \ldots, i_k)$, $1 \leqslant i_1 < i_2 < \ldots < i_k \leqslant n$. We assign to the block a number,

$$\mathrm{Num}\,(i_1, i_2, \ldots, i_k) = C_{i_1-1}^1 + C_{i_2-1}^2 + \ldots + C_{i_{k-1}-1}^{k-1} + C_{i_k-1}^k\,,$$

where it is set that $C_j^i = 0$ if $j < i$. This equality provides a method for numbering blocks using Babkin's technique [11].

Proposition 1.

The following relations hold:

$$\min_{i_1, i_2, \ldots, i_k} \mathrm{Num}\,(i_1, i_2, \ldots, i_k) = 0\,,$$

$$\max_{i_1, i_2, \ldots, i_k} \mathrm{Num}\,(i_1, i_2, \ldots, i_k) = C_n^k - 1\,.$$

Proposition 2.

The relation between blocks with k events s_1 at positions $(i_1, i_2, \ldots, i_k)$ and numbers $\mathrm{Num}\,(i_1, i_2, \ldots, i_k)$ *is a one-to-one correspondence*.

Thus, any sequence containing exactly k symbols s_1 at positions $(i_1, i_2, \ldots, i_k)$ is uniquely assigned a number $\mathrm{Num}\,(i_1, i_2, \ldots, i_k)$.

2.1.1 Stream numbering according to table. Blocks are numbered sequentially as the event s_1 arrives.

Table 1.

	1	2	3	4	5	…	$n-1$	n
i_1	0	1	C_2^1	C_3^1	C_4^1	…	C_{n-2}^1	C_{n-1}^1
i_2	0	0	1	C_3^2	C_4^2	…	C_{n-2}^2	C_{n-1}^2
i_3	0	0	0	1	C_4^3	…	C_{n-2}^3	C_{n-1}^3
…	…	…	…	…	…	…	…	…
i_{n-1}	0	0	0	0	0	…	0	1

The block size n is specified; the table of binomial coefficients (Table 1) of size $(n-1) \times n$ is calculated once. The value of k is not fixed in advance.

The numbering of sequences is reduced to moving along a certain trajectory on the table with sequential summation of binomial coefficients.

If event s_1 occurs for the first time at position m_1, the value of the binomial coefficient at the intersection of the row with number i_1 (the first event s_1) with the column with number m_1 is taken.

If event s_1 occurs for the second time at position m_2 ($m_2 > m_1$), the value of the binomial coefficient at the intersection of the row with number i_2 and the column with number m_2 is taken and added to the previous value of the binomial coefficient.

If event s_1 occurs for the kth time at position m_k ($m_k > m_{k-1}$), the value of the binomial coefficient at the intersection of the row with number i_k and the column with number m_k is taken and added to the previous sum of the binomial coefficients.

The process halts when the entire block of size n has been scanned. According to the previous section, the number $\mathrm{Num}\,(m_1, m_2, \ldots, m_k)$ of the block with events s_1 and s_2 is obtained as a binary representation — these are not yet random bits.

After the number of a specific sequence of s_1 and s_2 is obtained, a block of random 0s and 1s is extracted from its binary representation.

2.2 Second stage: extraction of random binary sequences from sequence number

The cardinality of the set of blocks $\mathcal{R}_n(k)$ with k events s_1 and $n-k$ events s_2 is $|\mathcal{R}_n(k)| = C_n^k$. Blocks are numbered from 0 to $C_n^k - 1$.

Let n be even. Consider the representation of $|\mathcal{R}_n(k)|$ as a sum,

$$\left|\mathcal{R}_n(k)\right| = 2^{r_m} + \ldots + 2^{r_1} + 2^{r_0}\,, \quad r_m > r_{m-1} > \ldots > r_1 > r_0\,.$$

Let a block be realized that has a composition $(i_1, i_2, \ldots, i_k)$ of events s_1. The block number has a binary decomposition of the form

$$\mathrm{Num}\,(i_1, i_2, \ldots, i_k) = \varepsilon_{r_m+1} 2^{r_m+1} + \varepsilon_{r_m} 2^{r_m} + \varepsilon_{r_m-1} 2^{r_m-1} + \ldots + \varepsilon_1 2^1 + \varepsilon_0 2^0\,, \quad \varepsilon_r \in \{0, 1\}\,,$$

and the corresponding binary representation

$$(\varepsilon_{r_m+1}, \varepsilon_{r_m}, \varepsilon_{r_m-1}, \ldots, \varepsilon_1, \varepsilon_0)\,.$$

Extraction of a block $\{\varepsilon\}$ of random 0s and 1s is performed from the binary representation $(\varepsilon_{r_m+1}, \varepsilon_{r_m}, \varepsilon_{r_m-1}, \ldots, \varepsilon_1, \varepsilon_0)$ of the number $\mathrm{Num}\,(i_1, i_2, \ldots, i_k)$. It is performed differently, depending on the range of numbers

Table 2. Babkin's algorithm, binary output, $n = 8$, $k = 1$.

Positions of s_1 and s_2 (i_1)	Number $N(i_1)$	Binary representation	Random block $\{\varepsilon\} = \varepsilon_{r_0-1}, \ldots, \varepsilon_0$
$\boldsymbol{s_1}s_2s_2s_2s_2s_2s_2s_2$	0	000	000
...	1	001	001
	2	010	010
$j = 0$	3	011	011
	4	100	100
	5	101	101
...	6	110	110
$s_2s_2s_2s_2s_2s_2s_2\boldsymbol{s_1}$	$7 = 2^{r_0} - 1$	111	111

between 0 and $C_n^k - 1$ in which the number $\mathrm{Num}\,(i_1, i_2, \ldots, i_k)$ of the current block lies.

Namely:

Number	Block $\{\varepsilon\}$ of random 0s and 1s
$0 \leqslant \mathrm{Num}\,(i_1, i_2, \ldots, i_k) \leqslant 2^{r_0} - 1$,	$\varepsilon_{r_0-1}, \ldots, \varepsilon_0$,
$2^{r_0} \leqslant \mathrm{Num}\,(i_1, i_2, \ldots, i_k) \leqslant 2^{r_0} + 2^{r_1} - 1$,	$\varepsilon_{r_1-1}, \ldots, \varepsilon_0$,
$2^{r_0} + 2^{r_1} \leqslant \mathrm{Num}\,(i_1, i_2, \ldots, i_k) \leqslant 2^{r_0} + 2^{r_1} + 2^{r_2} - 1$,	$\varepsilon_{r_2-1}, \ldots, \varepsilon_0$,
...	...
$2^{r_0} + \ldots + 2^{r_m} \leqslant \mathrm{Num}\,(i_1, i_2, \ldots, i_k) \leqslant 2^{r_0} + \ldots + 2^{r_m} - 1$,	$\varepsilon_{r_m-1}, \ldots, \varepsilon_0$.

We number the rows (inequalities) as $0, \ldots, j, \ldots, m$. The jth row — the subclass — contains 2^{r_j} different numbers $\mathrm{Num}\,(i_1, i_2, \ldots, i_k)$, which uniquely correspond to binary vectors from the space $\{0,1\}^{r_j}$. Then, for each current number $\mathrm{Num}\,(i_1, i_2, \ldots, i_k)$, the corresponding block $\{\varepsilon\} = \varepsilon_{r_j-1}, \ldots, \varepsilon_0$, consisting of 0s and 1s, is output.

Let us consider examples illustrating the general method, for $n = 8$, $k = 1, 2$.

Example 1. $n = 8$, $k = 1$.

$$\left|\mathcal{R}_n(k)\right| = \frac{8!}{1!\,7!} = 8 = 2^3\,, \quad m = 0, \quad r_0 = 3\,.$$

The length of the binary output (Table 2, fourth column) is 3.

Example 2. $n = 8$, $k = 2$.

$$\left|\mathcal{R}_n(k)\right| = \frac{8!}{2!\,6!} = 28 = 2^4 + 2^3 + 2^2,$$

$$m = 2, \quad r_2 = 4, \quad r_1 = 3, \quad r_0 = 2\,.$$

The length of the binary output (Table 3, fourth column) is 2, 3, and 4.

Note that the binary output after Babkin numbering and extraction of the block of 0s and 1s (fourth columns in Tables 2 and 3) contains all binary vectors of fixed length exactly once.

2.3 Babkin's algorithm, *m*-ary alphabet

Consider a source that generates symbols from the m-ary alphabet $A = \{s_1, \ldots, s_m\}$.

Let $\mathcal{R}_n(k_1, \ldots, k_m)$ be a set of blocks of length n, where there are $k_1, \ldots, k_m$ symbols $s_1, \ldots, s_m$, $k_1 + \ldots + k_m = n$. The total of such blocks is

$$\left|\mathcal{R}_n(k, \ldots, k_m)\right| = \frac{n!}{k_1!\,k_2! \ldots k_m!}\,.$$

In study [11], for the m-ary alphabet, V F Babkin proposed an algorithm for assigning 'on the fly' the number $0 \leqslant \mathrm{Num}\,(\ldots) \leqslant |\mathcal{R}_n(k_1, \ldots, k_m)| - 1$ to a specific block from the set $\mathcal{R}_n(k_1, \ldots, k_m)$.

Then, the extraction of random bits from the binary representation $\mathrm{Num}\,(\ldots)$ occurs similarly to the case of the binary alphabet $A = \{s_1, s_2\}$. The corresponding table becomes richer: it can contain binary outputs of greater length.

Table 3. Babkin's algorithm, binary output, $n = 8$, $k = 2$.

Positions of s_1 and i_2, i_1, i_2	Number $N(i_1, i_2)$	Binary representation	Random block $\{\varepsilon\} = \varepsilon_{r_{j-1}}, \ldots, \varepsilon_0$
$\boldsymbol{s_1}\boldsymbol{s_1}s_2s_2s_2s_2s_2s_2$	0	00000	00
...	1	00001	01
$j = 0$	2	00010	10
	$3 = 2^{r_0} - 1$	00011	11
	4	00100	100
	5	00101	101
	6	00110	110
$j = 1$	7	00111	111
	8	01000	000
	9	01001	001
	10	01010	010
	$11 = 2^{r_1} + 2^{r_0} - 1$	01011	011
	12	01100	1100
	13	01101	1101
	14	01110	1110
	15	01111	1111
	16	10000	0000
	17	10001	0001
$j = 2$	18	10010	0010
	19	10011	0011
	20	10100	0100
	21	10101	0101
	22	10110	0110
	23	10111	0111
	24	11000	1000
	25	11001	1001
	26	11010	1010
$s_2s_2s_2s_2s_2s_2\boldsymbol{s_1}\boldsymbol{s_1}$	$27 = 2^{r_2} + 2^{r_1} + 2^{r_0} - 1$	11011	1011

We note again that, for a fixed length, the binary output of Babkin's algorithm in the corresponding table for the m-ary alphabet will also contain in the fourth column all binary vectors of fixed length exactly once. This circumstance is one of the main points for the further proof of the equiprobability of binary sequences at the output of Babkin's algorithm.

Babkin's algorithm for the m-ary alphabet is used further to prove the randomness of bits extracted from the trajectory of a simple Markov chain of order $r = 1$ with m states $\{s_1, \ldots, s_m\}$. In study [18], Babkin's mth algorithm was used to generate random sequences in quantum random number generators based on homodyne detection — 'vacuum fluctuations.'

When extracting provably random bits from the practical case of a Markov chain of order $r \geqslant 2$ with two states $\{s_1, s_2\}$, which is of interest to us, the algorithm for the m-ary alphabet is not needed: Babkin's algorithm for the binary alphabet is operative.

2.4 Main concept of proving equiprobability: partitioning original probability space into classes of equivalent (equally probable) sequences

Chance deals with order in disorder
while chaos deals with disorder in order.
K R Rao

The epigraph is emotional and is more of a controversial pun than a mathematical definition. Randomness has a clear mathematical definition. Randomness, in our case the true

randomness of sequences of 0s and 1s understood as the equiprobability and independence of the occurrence of 0s and 1s in each position, is the highest degree of disorder — unpredictability, rather than order among disorder. Shannon entropy for independent trials is a measure of disorder; for a truly random sequence of 0s and 1s, the entropy has a maximum value equal to one per position.

Before proceeding to formal proofs, we present informal qualitative considerations regarding the extraction of randomness from the original sequences.

Consider a simple example. We toss a die with equally probable outcomes $\Omega = \{1, 2, \ldots, 6\}$, which we call elementary events, and extract random bits $\varepsilon \in \{0, 1\}$ when an even or odd outcome occurs, respectively. It is easy to see that, in this case, with the same probability of events in the even $\{2, 4, 6\}$ and odd $\{1, 3, 5\}$ subsets, we obtain the equiprobability of random bits:

$$\Pr(\varepsilon = 0) = \Pr(\varepsilon = 1) = \frac{1}{2}.$$

A more complex construction of extracting equally probable random bits from observations of elementary events of the original probability space can be easily imagined.

Suppose that the entire probability space $\Omega = \{\omega\}$ is divided into subsets (classes) of equally probable elementary events:

$$\Omega = \bigcup_i S_i, P(\omega) = P(\omega'), \text{ once } \omega, \omega' \in S_i.$$

If in each subset (class) S_i the number of elementary events generating the value $\varepsilon = 0$ is equal to that of elementary events generating the value $\varepsilon = 1$, then, apparently, $\Pr(\varepsilon = 0) = \Pr(\varepsilon = 1)$. Calculating the probabilities $\Pr(\varepsilon = 0)$ and $\Pr(\varepsilon = 1)$ is the summation of the probabilities of elementary events over the corresponding subsets. Conditional equiprobability also occurs in the sense that

$$\Pr\left(\varepsilon = 0 \,|\, \omega \in \Omega_\varepsilon\right) = \Pr\left(\varepsilon = 1 \,|\, \omega \in \Omega_\varepsilon\right) = \frac{1}{2},$$

where Ω_ε is the set of elementary events that generate random bits ε.

The principle of partitioning equally probable elementary events into classes is used to prove the equiprobability of output sequences.

A practically working Babkin algorithm processes the input sequence in blocks.

Let us assume, for example, that the input of the algorithm is a Bernoulli sequence of three blocks of the same size $n = 8$:

$$X = (X_1, X_2, X_3).$$

This is one of the possible elementary events of the original probability space $\Omega = \{X\} = \{s_1, s_2\}^{24}$.

Let us denote by

— $Y_i = \Psi(X_i)$ — the binary output of Babkin's algorithm from block X_i; $|Y_i|$ is the length of the binary output, $i = \overline{1, 3}$,

— $Y = Y_1 \,\|\, Y_2 \,\|\, Y_3 = \Psi(X_1) \,\|\, \Psi(X_2) \,\|\, \Psi(X_3)$ is the full output of Babkin's algorithm from the sequence X as a concatenation of individual outputs.

To simplify the notation, we also write $Y = \Psi(X)$; $|Y|$ is the length of the full output.

Let us consider the full binary output $Y = (01011100)$, $|Y| = 8$.

To calculate the probability of this output, we must sum the probabilities of those elementary events X that yield $Y = (01011100)$.

We now consider the subset $S \subseteq \Omega = \{X = (X_1, X_2, X_3)\}$, where blocks (X_1, X_2, X_3) contain exactly $k = 1, 2, 2$ events s_1, respectively

This forms a class of S elementary events, where $X = (X_1, X_2, X_3)$ differs from $X' = (X'_1, X'_2, X'_3)$ by a permutation within the blocks, while, as is easy to see, the probabilities of sequences X and X' are the same:

$$P_S(X) = P_S(X') = \left[P(s_1)\right]^5 \left[P(s_2)\right]^{24-5}.$$

We call the class S of equally probable sequences an equivalence class.

Is it possible to obtain the output $Y = (01011100)$, $|Y| = 8$ from sequences $X \in S$?

Examination of the fourth columns of Tables 2 and 3 shows that

— the first block X_1 yields at the output $Y_1 = \Psi(X_1)$ with a bit length of $|Y_1| = 3$,

— the second block X_2 yields at the output $Y_2 = \Psi(X_2)$ with a bit length of $|Y_2| = 2, 3, 4$,

— the third block X_3 yields at the output $Y_3 = \Psi(X_3)$ with a bit length of $|Y_3| = 2, 3, 4$,

the possible length $|Y| = |Y_1| + |Y_2| + |Y_3|$ varying from 7 to 11.

Partial binary output $Y_i = \Psi(X_i)$ is the binary output from the number X_i in Babkin's algorithm, i.e., $Y_i = \Psi(\mathrm{Num}(X_i))$.

Figure 1 below shows the partition of sequence numbers X_i into blocks. The size of each block is a power of two. The length of the binary output from each block (dashed and dotted lines) is the same and does not depend on the bit composition, and the output bit strings of length $|Y_i| = |Y'_i|$ range over all bit combinations of 0s and 1s (see Tables 2, 3).

The length of the full binary output $|Y| = 8$ can be obtained as $|Y| = 3 + 2 + 3$ or as $|Y| = 3 + 3 + 2$, the specific binary output $Y = (01011100)$ being obtained in the class S as a concatenation of partial binary outputs (see Tables 2 and 3):

$$Y_1 = 010 \,\|\, Y_2 = 11 \,\|\, Y_3 = 100$$

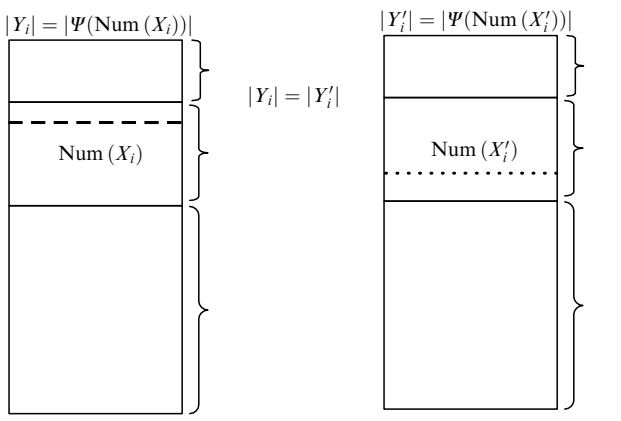

Figure 1. Partitioning numbers of sequences from one equivalence class into blocks.

or

$$\widetilde{Y}_1 = 010 \,\|\, \widetilde{Y}_2 = 111 \,\|\, \widetilde{Y}_3 = 00,$$

i.e.,

$$Y = (01011100) = 010 \,\|\, 11 \,\|\, 100 = 010 \,\|\, 111 \,\|\, 00\,.$$

Since, for a fixed length, the binary output in the right column in Tables 2 and 3 contains all binary vectors exactly once, it may be asserted that, for $Y = (01011100)$ in the class S, there are exactly two preimages — two sequences X and $\widetilde{X}$, such that $\Psi(X) = \Psi(\widetilde{X}) = Y$.

We also assert that, for $Y = (01011100)$ in the class S, there are exactly two admissible partitions of a specific output Y, namely (Y_1, Y_2, Y_3) and $(\widetilde{Y}_1, \widetilde{Y}_2, \widetilde{Y}_3)$, for which the corresponding preimages X and $\widetilde{X}$ are found.

The set of admissible partitions Y for the class S is denoted as $\mathbb{Y}_S(Y)$, and its cardinality as $|\mathbb{Y}_S(Y)|$, in this case $|\mathbb{Y}_S(Y)| = 2$.

It is extremely important to note that, for any other binary output of the same length, for example $Y' = (11010011)$, also by the construction of Babkin's algorithm, we obtain exactly two concatenations of partial outputs:

$$Y'_1 = 110 \,\|\, Y'_2 = 10 \,\|\, Y'_3 = 011\,,$$

$$\widetilde{Y}'_1 = 110 \,\|\, \widetilde{Y}'_2 = 100 \,\|\, \widetilde{Y}'_3 = 11\,.$$

These concatenations correspond to their two preimages from the set S.

From this, we conclude that the cardinalities of the sets of admissible partitions of the outputs $Y = (01011100)$ and $Y' = (11010011)$ are the same:

$$\left|\mathbb{Y}_S(Y)\right| = \left|\mathbb{Y}_S(Y')\right| = 2\,.$$

The corresponding generalizing conclusion can be made for the remaining possible lengths $|Y| = 7 \div 11$ of the binary output of Babkin's algorithm.

Namely, for any Y, Y' with the same length $|Y| = |Y'|$, the cardinalities of the sets of admissible partitions for the class S are the same:

$$\left|\mathbb{Y}_S(Y)\right| = \left|\mathbb{Y}_S(Y')\right|\,.$$

It is easy to see that such equivalence classes S consisting of equally probable sequences can be constructed for any composition $k = k_1, k_2, k_3$, $0 \leqslant k_i \leqslant 8$ of events s_1 in blocks X_1, X_2, X_3. Thus, we obtain a partition of the probability space $\Omega = \{X = (X_1, X_2, X_3)\}$ into equivalence classes S consisting of equally probable sequences.

Note that a specific binary output $Y = (01011100)$ can be obtained for some input sequence $X \in S'$, $S' \neq S$, i.e., in another equivalence class. Again, we can assert that, for any Y and Y' with the same length $|Y| = |Y'|$, the cardinalities of the sets of admissible partitions for the class S' are the same:

$$\left|\mathbb{Y}_{S'}(Y)\right| = \left|\mathbb{Y}_{S'}(Y')\right|\,.$$

Thus, for complete output sequences Y, Y' with the same length $|Y| = |Y'|$ in each class of equally probable sequences, there is the same number of preimages — sequences X. From this, we can informally conclude that, for Y and Y', the probabilities must be equal: $P(Y) = P(Y')$.

The above reasoning illustrates the proof of the equiprobability of the output binary sequence for the general case of an m-ary alphabet $A = \{s_1, \ldots, s_m\}$.

2.5 Proof of equiprobability of binary sequence at output of Babkin's algorithm

Let us consider the general case of a source generating symbols from the m-ary alphabet $A = \{s_1, \ldots, s_m\}$.

Let the input of Babkin's algorithm be a sequence of X independent trials from a finite probability scheme with outcomes $\{s_1, \ldots, s_m\}$ — this is the original probability space. The probabilities $\{P(a_1), \ldots, P(a_m)\}$ are unknown.

Below, we will present a proof of equiprobability of the binary sequence at the output of Babkin's algorithm. We use the above-discussed idea of partitioning the set of input sequences into equivalence classes as a basis. This technique will be used further, when considering Markov chain trajectories at the input.

Thus, a practically working Babkin algorithm processes the input sequence in blocks. Without loss of generality, we assume that the input of the algorithm is a sequence

$$X = (X_1, X_2, \ldots, X_M),\ X_1 \in \{s_1, \ldots, s_m\}^{n_1}, \ldots, X_M \in \{s_1, \ldots, s_m\}^{n_M}\,, \tag{1}$$

where M is the number of blocks being processed, and $n_1, \ldots, n_M$ are the sizes of the blocks being processed.

We divide all possible sequences $X \in \{s_1, \ldots, s_m\}^{(n_1 + \ldots + n_M)}$ into equivalence classes and use G to denote the set of classes.

Definition.

Two sequences

$$X = (X_1, X_2, \ldots, X_M), \quad X' = (X'_1, X'_2, \ldots, X'_M)$$

belong to the same equivalence class S if and only if the block X'_j is a permutation of the block X_i, for any $i = \overline{1, M}$.

We denote the permutation as

$$X'_i \equiv X_i\,.$$

Note that the equivalence class S corresponds to a certain specific composition of symbols $\{s_1, \ldots, s_m\}$ in the blocks $X_1, X_2, \ldots, X_M$.

Namely, for each block X_i the numbers $k_1^{(i)}, \ldots, k_m^{(i)}$, $k_1^{(i)} + \ldots + k_m^{(i)} = n_i$ are fixed, where $k_j^{(i)}$ is the number of occurrences of the symbols s_j in the blocks X_i.

Proposition 3.

Any two sequences $X = (X_1, X_2, \ldots, X_M)$ and $X' = (X'_1, X'_2, \ldots, X'_M)$ belonging to the same equivalence class S have the same probability:

$$P_S(X) = P_S(X')\,.$$

The proof easily follows from the initial assumption that the input sequence X is a sequence of independent trials over a finite probabilistic scheme.

Thus, the partitioning into equivalence classes is a partition of the entire set of input sequences into classes of equally probable sequences.

Let the input sequence $X \in S$ belong to some equivalence class.

Let $Y = \Psi(X) \in \{0, 1\}^*$ be the full binary output of Babkin's algorithm as a concatenation of partial outputs $Y_i = \Psi(X_i)$.

Here, we introduced the notation $\{0,1\}^*$, emphasizing the uncertainty of the length $|Y|$ of the output binary sequence, which depends on the specific composition of the outcomes $\{s_1,\dots,s_m\}$ in the blocks.

For $Y\in\{0,1\}^*$, we denote by B_Y the set of sequences $X\in\{s_1,\dots,s_m\}^{(n_1+\dots+n_M)}$ such that $\Psi(X)=Y$. The set B_Y is the union of all preimages over all classes. It is clear that there may exist classes S with an empty set of preimages.

Proposition 4.

For any class $S\in G$, the cardinality of the set $|S\cap B_{Y'}|=|S\cap B_Y|$ whenever the lengths coincide: $|Y'|=|Y|$.

Proof.

Let a specific output $Y\in\{0,1\}^*$ with length $|Y|$ be given. Consider the class S. It consists of the original sequence of blocks $X=(X_1,X_2,\dots,X_M)$ and all permutations inside the blocks.

We associate with the full binary output $Y\in\{0,1\}^*$ a partition into partial binary outputs

$$Y_1,\dots,Y_M,\quad |Y|=|Y_1|+|Y_2|+\dots+|Y_M|,$$

such that the concatenation

$$Y_1\|\dots\|Y_M=Y.$$

We define S_i as the set consisting of all permutations of the block X_i, $i=\overline{1,M}$ and introduce the notation

$$S_i(Y_i)=\{X_i\in S_i:\Psi(X_i)=Y_i\}.$$

By the construction of Babkin's algorithm (the fourth columns of Tables 2 and 3), for a fixed length $|Y_i|$, the cardinality $|S_i(Y_i)|=1$ or $|S_i(Y_i)|=0$ if the length $|Y_i|$ is not suitable for the set S_i.

Consider $Y'\in\{0,1\}^*$, which does not coincide with Y but has the same length $|Y'|=|Y|$ and the corresponding partition:

$$Y'_1,\dots,Y'_M,\ |Y'|=|Y'_1|+|Y'_2|+\dots+|Y'_M|,$$

$$|Y'_i|=|Y_i|.$$

Again, by the construction of Babkin's algorithm, the cardinality $|S_i(Y'_i)|=1$ or $|S_i(Y_i)|=0$.

Therefore, exactly one sequence $X_i,X'_i\in S_i$ (i.e., one of all permutations S_i) in Babkin's algorithm yields partial outputs Y_i and Y'_i or, generally speaking, does not yield them if the length $|Y_i|$ is not suitable for the set S_i.

Therefore, it is not possible for every partition Y_1, $Y_2,\dots,Y_n$, such that the concatenation $Y_1\|\dots\|Y_M=Y$, to find a sequence

$$X=(X_1,X_2,\dots,X_M)\in S$$

such that $\Psi(X)=Y$.

For example, the first block X_1 may be scanty — containing, for example, only the pair of outcomes $\{s_1,s_2\}$, $k_1^{(1)}+k_2^{(1)}=n_1$. Then, the length $|Y_1|$ of the binary output of the first block cannot be large, which, generally speaking, is admissible for another partition.

We call the partition $Y_1,Y_2,\dots,Y_M$, such that the concatenation $Y_1\|\dots\|Y_M=Y$ is admissible for a given class S if $|S_i(Y_i)|=1$ for all $i=\overline{1,M}$, and denote as $\mathbb{Y}_S(Y)$ the set of admissible partitions.

Thus, we are interested in partitions $(Y_1,Y_2,\dots,Y_M)\in\mathbb{Y}_S(Y)$ that belong to the set of admissible partitions.

The above reasoning regarding the constructive construction of Babkin's algorithm allows us to conclude that the cardinality of $|\mathbb{Y}_S(Y)|$ depends on the class S and on the length $|Y|$, rather than on the bit composition of Y, i.e.,

$$\bigl|\mathbb{Y}_S(Y')\bigr|=\bigl|\mathbb{Y}_S(Y)\bigr|$$

once $|Y'|=|Y|$.

It follows that

$$\bigl|S\cap B_Y\bigr|=\sum_{\substack{Y_1,Y_2,\dots,Y_M:\\ |Y_1|+|Y_2|+\dots+|Y_M|=|Y|}}\prod_{i=1}^{n}\bigl|S_i(Y_i)\bigr|$$
$$=\sum_{(Y,Y_2,\dots,Y_M)\in\mathbb{Y}_S(Y)}1=\bigl|\mathbb{Y}_S(Y)\bigr|$$

are the cardinalities of the set of admissible partitions.

As was established above, the cardinality of $|\mathbb{Y}_S(Y)|$ depends only on the class S and on the length $|Y|$.

Then, for Y', $|Y'|=|Y|$, we have

$$\bigl|S\cap B_{Y'}\bigr|=\sum_{\substack{Y'_1,Y'_2,\dots,Y'_M:\\ |Y'_1|+|Y'_2|+\dots+|Y'_M|=|Y'|}}\prod_{i=1}^{n}\bigl|S_i(Y'_i)\bigr|$$
$$=\sum_{(Y'_1,Y'_2,\dots,Y'_M)\in|\mathbb{Y}_S(Y')|}1=\bigl|\mathbb{Y}_S(Y')\bigr|=\bigl|\mathbb{Y}_S(Y)\bigr|.$$

Therefore, $|S\cap B_{Y'}|=|S\cap B_Y|$ whenever $|Y'|=|Y|$.

Proposition 4 is proved.

If, for a given class S, there are no preimages for any partition $Y_1,Y_2,\dots,Y_M$, i.e., $S_i(Y_i)=\{X_i\in S_i:\Psi(X_i)=Y_i\}=\emptyset$ for all $i=\overline{1,M}$, then $|\mathbb{Y}_S(Y)|=0$ and, therefore, $|S\cap B_Y|=|S\cap B_{Y'}|=0$.

We now formulate a theorem on the equiprobability of the binary output of Babkin's algorithm.

Theorem 1.

Let a sequence $X=(X_1,X_2,\dots,X_M)$ of independent trials from a finite probability scheme with outcomes $\{s_1,\dots,s_m\}$ be used as input to Babkin's algorithm.

Then, the binary output $Y\in\{0,1\}^\ell$ obtained for any possible ℓ has the probability

$$P(Y)=2^{-\ell}.$$

Proof.

Consider the class S corresponding to a certain composition of symbols $\{s_1,\dots,s_m\}$ in blocks $X_1,X_2,\dots,X_M$.

Above, in Proposition 3, it was established that, for any $X,X'\in S$, the probability $P_S(X)=P_S(X')$.

Then, for $Y\in\{0,1\}^\ell$, we have

$$P(Y)=P(X\in B_Y)$$
$$=\sum_{S\in G}P(X\in S)\,P(X\in B_Y\mid X\in S)$$
$$=\sum_{S\in G}P(X\in S)\,\frac{P(X\in S\cap B_Y)}{P(X\in S)}$$
$$=\sum_{S\in G}P(X\in S)\,\frac{P_S(X)\,|S\cap B_Y|}{P_S(X)\,|S|}$$
$$=\sum_{S\in G}P(X\in S)\,\frac{|S\cap B_Y|}{|S|}.$$

Now, consider any other output $Y' \in \{0,1\}^{\ell}$. For coinciding lengths $|Y'| = |Y|$, it follows from Proposition 4 that the cardinalities of the preimages are also the same:

$$\left|S \cap B_{Y'}\right| = \left|S \cap B_{Y}\right| .$$

From this, we immediately conclude that

$$P(Y) = P(Y') .$$

Since this equality holds for any $Y, Y' \in \{0,1\}^{\ell}$, it follows that $P(Y) = 2^{-\ell}$.

Theorem 1 is proved.

A comment is relevant here.

1. As we noted above, it may turn out that in some classes S there is no sequence $X = (X_1, X_2, \ldots, X_M)$ that generates Y of a given length $|Y| = \ell$, which, generally speaking, narrows the original probability space. Thus, we conclude that the reasoning above, in fact, justifies the equiprobable choice of Y under the condition that the length $|Y| = \ell$ is maintained (fixed).

2. The proof of the equiprobability of the binary output is based on the construction of equivalence classes, i.e., classes of equally probable sequences. It is assumed that when implementing Babkin's algorithm for another sequence $X' = (X'_1, X'_2, \ldots, X'_M)$ in the equivalence class, the order of processing blocks does not change:

$$(Y'_1, Y'_2, \ldots, Y'_M) = \left(\Psi(X'_1), \Psi(X'_2), \ldots, \Psi(X'_M)\right) .$$

For an independent sequence, this assumption is natural, since the blocks are processed in a single stream of incoming symbols.

For the Markov chains considered below, preserving the order of processing blocks in an equivalence class is key to proving the equiprobability of the binary output of Babkin's algorithm. Preservation of the order of processing blocks in an equivalence class gives rise to a feature of the implementation of the algorithm for reading blocks for processing, in which they are queued.

3. Markov chain

Before providing a detailed explanation of how provably true randomness is extracted from correlated sequences — Markov chain trajectories with finite dependence depth in the measurement history — we give a qualitative explanation of how the problem can be reduced to the previous case of independent measurements, Bernoulli sequences.

The general idea also consists in partitioning different Markov chain trajectories into equivalence classes of trajectories that, as a whole, are equally probable. If this can be done, we can also enumerate the trajectories in each class of equally probable trajectories, similar to how this was done above by the enumeration of Bernoulli sequences in one class of equally probable sequences. Next, each number is associated with a block of truly random 0s and 1s, after which the blocks are concatenated.

The main problem is how to split the various trajectories of the Markov chain, corresponding to correlated sequences of measurement results, into equivalence classes.

It turns out that each trajectory can be associated with a closed graph, the Euler graph. The vertices correspond to states, or, informally speaking, to the measurement result combined with the prehistory. The edges correspond to transitions between the states of the Markov chain.

It is shown below that the trajectories in one class of equally probable trajectories include such trajectories of the Markov chain which correspond to Euler graphs that differ from each other in the sequence of traversing the edges in the graph.

The central point is that only such an order of the traversal of edges of a closed graph is allowed that does not affect the edge connecting the end and the beginning of the Markov chain trajectory.

After dividing equally probable Markov chain trajectories into classes, the problem of extracting truly random sequences is reduced to that of extracting random sequences from independent trials, the solution to which was given above.

We now proceed to a detailed derivation. Next, we consider Markov chain trajectories — elementary events

$$X = x_1 x_2 \ldots x_N ,$$

where $x_i \in A = \{s_1, \ldots, s_m\}$ belongs to the set of chain states. Specified are the initial distribution,

$$P(s_1), \ldots, P(s_m), \quad \sum_{i=1}^{m} P(s_i) = 1 ,$$

and the matrix of transition probabilities of size $m \times m$,

$$\left\|P(s_j \,|\, s_i)\right\|, \quad i, j = \overline{1, m}, \quad \sum_{j=1}^{m} P(s_j \,|\, s_i) = 1 .$$

The probability of an elementary event (trajectory) X is defined as

$$P(X) = P(x_1 x_2 \ldots x_N) = P(x_1) \prod_{i=1}^{N-1} P(x_{i+1} \,|\, x_i) .$$

The matrix of transition (conditional) probabilities contains in the condition the dependence on only one previous state. This is the case of a stationary Markov chain of order $r = 1$. Strictly speaking, for a chain to be stationary, a certain condition for the initial distribution must also be satisfied, but it is not essential for us.

Next, for a Markov chain with m states of order $r = 1$, we describe an algorithm for obtaining a provably random binary sequence and extend it to the practical case of interest to us, i.e., a Markov chain with two states $\{s_1, s_2\}$ of order $r \geqslant 2$.

3.1 Construction of equivalence classes, connection with Eulerian graphs

To prove the equiprobability of output sequences, it is necessary to partition the trajectories of Markov chains into classes with the same probability. An intuitively understandable method of partitioning is based on some facts from graph theory.

Surprisingly, the problem of partitioning Markov chains (measurement results) into equivalence classes is connected with Eulerian graphs, which arose in the well-known problem of a single walk around the Königsberg bridges, the problem that was solved by Leonhard Euler (see, for example, [19]).[3]

[3] Recall that a single bypass of the Königsberg bridges turned out to be impossible, which is due to the specific configuration of the connections between the bridges.

Thus, the immediate goal is to construct equivalence classes (equally probable) of Markov chain trajectories.

For a Markov chain trajectory $X = x_1x_2\ldots x_N$, a set of π-sequences is generated,

$$\pi(X) = \left[\pi_1(X), \pi_2(X), \ldots, \pi_m(X)\right],$$

where $\pi_i(X)$ is a subsequence of states from $X = x_1x_2\ldots x_N$ following the state s_i:

$$\pi_i(X) = \{x_{j+1} : x_j = s_i,\ 1 \leqslant j \leqslant N\}.$$

For example, for $X = s_1\boldsymbol{s_4}s_2s_1\boldsymbol{s_3}s_2s_3s_1\boldsymbol{s_1s_2}s_3s_4s_1$, we obtain π-sequences

$$\pi(X) = \left[\pi_1(X) = (\boldsymbol{s_4s_3s_1s_2}), \pi_2(X) = (s_1s_3s_3),\right.$$
$$\left.\pi_3(X) = (s_2s_1s_4), \pi_4(X) = (s_2s_1)\right].$$

Proposition 5.

The sequence X is uniquely determined by x_1 and $\pi(X)$.

Proposition 5 implies that, if X is already a trajectory of a Markov chain, and we do not know it, the trajectory of X is uniquely reconstructed based on the initial state x_1 and $\pi(X) = [\pi_1(X), \pi_2(X), \ldots, \pi_m(X)]$ [17].

We denote $Y \equiv X$ if Y is any permutation of X, and denote $Y \overset{\bullet}{\equiv} X$ if Y is a permutation of X with a fixed tail (last element).

For example:

$$s_1s_2s_2s_3 \equiv s_3s_2s_2s_1,$$
$$s_2s_3s_2\boldsymbol{s_1} \overset{\bullet}{\equiv} s_3s_2s_2\boldsymbol{s_1}.$$

Proposition 6.

Two trajectories of a Markov chain $X = x_1x_2\ldots x_N$ and $X' = x'_1x'_2\ldots x'_N$ with the same initial $x_1 = x'_1$ have the same probability if $\pi_i(X') \equiv \pi_i(X)$ for all $1 \leqslant i \leqslant m$.

Proof.

Note that the probability

$$P(X) = P(x_1)P(x_2\,|\,x_1)\ldots P(x_N\,|\,x_{N-1})$$
$$= P(x_1)\prod_{i=1}^{m}\prod_{s_j\in\pi_i(X)} P(s_j\,|\,s_i),$$

and the probability

$$P(X') = P(x'_1)\prod_{i=1}^{m}\prod_{s_j\in\pi_i(X')} P(s_j\,|\,s_i).$$

If $P(x_1) = P(x'_1)$ and $\pi_i(X') \equiv \pi_i(X)$ for all $1 \leqslant i \leqslant m$, then, rearranging the terms in $\pi_i(X)$, it is easy to find that $P(X') = P(X)$.

Proposition 6 is proved.

Thus, for $x_1 = x'_1$, a permutation of symbols inside π-sequences — $\pi_i(X) \equiv \pi_i(X')$ for all $1 \leqslant i \leqslant m$ — does not change the probability of the Markov chain trajectory, but after a permutation of symbols, the π-sequences must correspond to the Markov chain trajectory. In some cases, after a permutation of symbols inside π-sequences, it is not possible to construct a Markov chain trajectory.

The issue of which permutations are admissible in constructing a class of equally probable trajectories is one of the key ones in [17]. It is solved using the following Proposition.

Proposition 7 (Admissible permutations).

Let the following be given:

1) *a Markov chain trajectory $X = x_1x_2\ldots x_N$ with the last state $x_N = s_\chi$,*

2) *π-sequences $\pi(X) = [\pi_1(X), \ldots, \pi_\chi(X), \ldots, \pi_m(X)]$,*

3) *a set of sequences $[\Lambda_1, \ldots, \Lambda_\chi, \ldots, \Lambda_m]$ such that*

$\Lambda_\chi \equiv \pi_\chi(X)$ is any permutation, s_χ is the last state in X,

$\Lambda_i \overset{\bullet}{\equiv} \pi_i(X)$, $i \neq \chi$, is a permutation with a fixed tail.

Then, there exists a trajectory of a Markov chain $X' = x'_1x'_2\ldots x'_N$ with initial state $x'_1 = x_1$, final state $x'_N = x_N$, and π-sequences $\pi(X') = [\Lambda_1, \Lambda_2, \ldots, \Lambda_m]$.

The proof of this statement in [17] is quite cumbersome and involves a number of intermediate results.

We present an intuitive proof of this statement with some comments.

We associate the Markov chain trajectories with a directed graph. The vertices of the graph are the states of the chain labeled with edges — transitions between states in accordance with the Markov chain trajectory.

Let a trajectory of a Markov chain with the number of states $m = 4$ and length $N = 9$ be given:

$$X = s_1s_3s_1s_2s_4s_1s_2s_1s_3s_2,$$

where the last state $s_\chi = s_2$, and the π-sequences are

$$\pi(X) = \left[\pi_1 = (s_3s_\chi s_\chi s_3), \pi_\chi = (s_4s_1), \pi_3 = (s_1s_\chi), \pi_4 = (s_1)\right].$$

We introduce a fictitious state s_0 preceding the first element $x_1 = s_1$. We close the graph with an edge originating from the last element s_χ and entering s_0. We represent the closure as an edge denoted by a dotted line.

We represent the new trajectory as

$$X^* = s_0 \overset{1}{\to} s_1 \overset{2}{\to} s_3 \overset{3}{\to} s_1 \overset{4}{\to} s_\chi \overset{5}{\to} s_4 \overset{6}{\to} s_1 \overset{7}{\to} s_\chi \overset{8}{\to} s_1 \overset{9}{\to} s_3 \overset{10}{\to} s_\chi \overset{11}{\to} s_0.$$

Apparently, there is a one-to-one correspondence between the original trajectory of the Markov chain X and the trajectory X^*.

The corresponding graph is displayed in Fig. 2.

Each vertex of the graph has an even power: the number of the edges incoming to and outgoing from each vertex is the same. We then have a Euler graph. Starting from s_0, we can

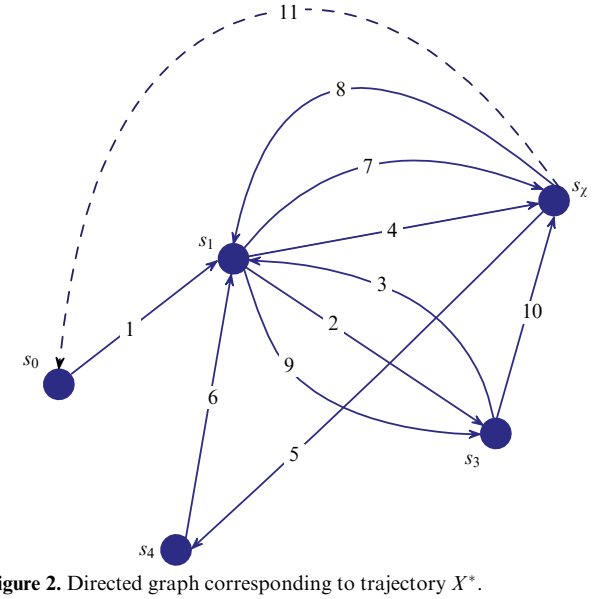

Figure 2. Directed graph corresponding to trajectory X^*.

completely traverse the graph and return to the state s_0, passing each edge only once.

The trajectories X^*, similar to the trajectories of the Markov chain X, correspond to a complete traversal of the corresponding graph.

The trajectory X^* has the π-sequences 'extended' compared to the trajectory of the Markov chain X:

$$\pi(X) = \left[\pi_1 = (s_3 s_\chi s_\chi s_3), \pi_\chi = (s_4 s_1), \pi_3 = (s_1 s_\chi), \pi_4 = (s_1)\right],$$

$$\pi(X^*) = \left[\pi_0^* = (s_1), \pi_1^* = (s_3 s_\chi s_\chi s_3), \pi_\chi^* = (s_4 s_1 s_0), \pi_3^* = (s_1 s_\chi), \pi_4^* = (s_1)\right].$$

Disregarding the unimportant block $\pi_0^* = (s_1)$, one can clearly see that the main difference is in the block $\pi_\chi^* = (s_4 s_1 s_0)$ corresponding to the last element of the chain χ. In this block, the closing state s_0 is added.

From here, it is easy to conclude that, for Proposition 7 to be fulfilled, it is sufficient to show the existence of a new trajectory — a complete traversal of the graph — for some transposition in the new blocks π_i^* that does not affect the last states in the blocks π_i^*. In the original block $\pi_\chi = (s_4 s_1)$, this transposition can already affect the last state of the block s_1. Moving sequentially along the transpositions, we can proceed to any permutations in the π-sequences that satisfy the conditions of Proposition 7.

It is clear that, if after an admissible transposition (in some block π_i^*) a complete traversal of the graph is possible, it is implied that the existence of the corresponding trajectory of the Markov chain is shown.

In constructing the graph edges, movement is performed between blocks π_i^*. The direction to the next block is indicated by the corresponding current state in the block. As soon as it is passed, it is excluded from the states of the block.

If, during transposition, the last states in blocks π_i^* are left unaffected, in constructing a new trajectory (movement between blocks), the last element in the block will only be excluded when there is no new entry into this block, and further movement will occur along the remaining blocks to continue in this way until the very end when all states are exhausted. In this way, a complete traversal of the corresponding graph can be constructed.

This also applies to the block $\pi_\chi^* = (s_4 s_1 s_0)$, where the last element (in the block) is s_0. Therefore, a transposition affecting the penultimate element in the block $\pi_\chi^* = (s_4 s_1 s_0)$ is possible. This implies that any transposition or any permutation in block $\pi_\chi(X)$ of the original trajectory X is possible.

We now illustrate the construction of trajectories using the example of

$$X^* = s_0 \xrightarrow{1} s_1 \xrightarrow{2} s_3 \xrightarrow{3} s_1 \xrightarrow{4} s_\chi \xrightarrow{5} s_4 \xrightarrow{6} s_1 \xrightarrow{7} s_\chi \xrightarrow{8} s_1 \xrightarrow{9} s_3 \xrightarrow{10} s_\chi \xrightarrow{11} s_0 .$$

We take an *admissible* transposition in block $\pi_1^* = (s_3 s_\chi s_\chi s_3) \to \pi_1^{**} = (s_\chi s_3 s_\chi s_3)$. Then,

$$\pi(X^{**}) = \left[\pi_0^{**} = (s_1), \pi_1^{**} = (s_\chi s_3 s_\chi s_3), \pi_\chi^{**} = (s_4 s_1 s_0), \pi_3^{**} = (s_1 s_\chi), \pi_4^{**} = (s_1)\right],$$

$$X^{**} = s_0 \xrightarrow{1} s_1 \xrightarrow{2} s_\chi \xrightarrow{3} s_4 \xrightarrow{4} s_1 \xrightarrow{5} s_3 \xrightarrow{6} s_1 \xrightarrow{7} s_\chi \xrightarrow{8} s_1 \xrightarrow{9} s_3 \xrightarrow{10} s_\chi \xrightarrow{11} s_0.$$

Note that trajectory X^{**} is a new complete traversal of the Euler graph.

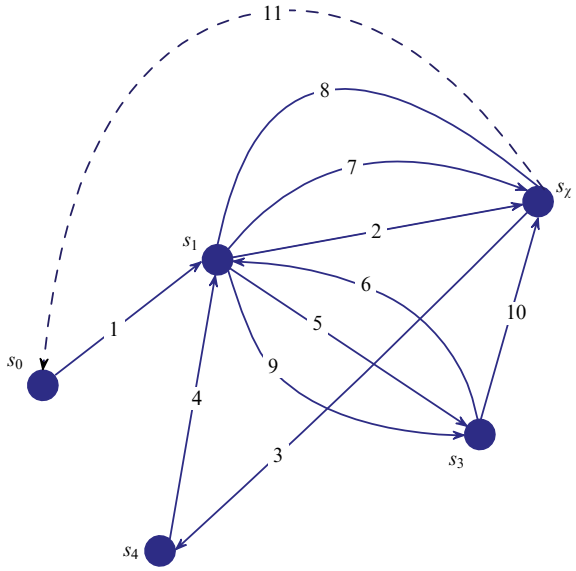

Figure 3. Directed graph corresponding to trajectory X^{**}.

We now take an *inadmissible* transposition in block $\pi_3^* = (s_1 s_\chi) \to \pi_3^{***} = (s_\chi s_1)$. Then,

$$\pi(X^{***}) = \left[\pi_0^{***} = (s_1), \pi_1^{***} = (s_3 s_\chi s_\chi s_3), \pi_\chi^{***} = (s_1 s_4 s_0), \pi_3^{***} = (s_\chi s_1), \pi_4^{***} = (s_1)\right],$$

$$X^{***} = s_0 \xrightarrow{1} s_1 \xrightarrow{2} s_3 \xrightarrow{3} s_\chi \xrightarrow{4} s_1 \xrightarrow{5} s_\chi \xrightarrow{6} s_4 \xrightarrow{7} s_1 \xrightarrow{8} s_\chi \xrightarrow{9} s_0 \xrightarrow{10} ? .$$

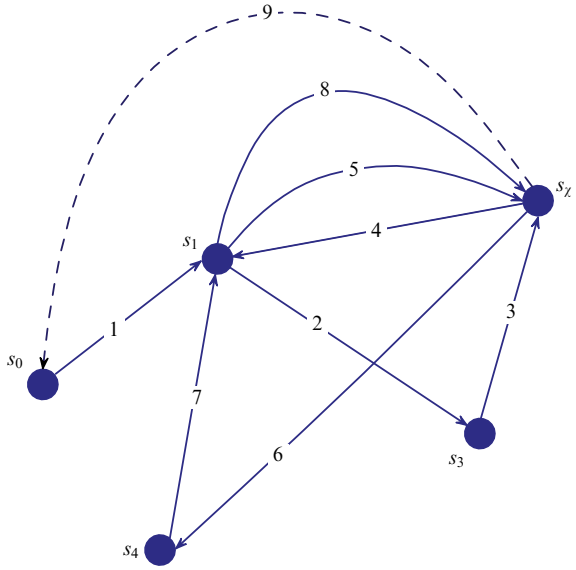

Figure 4. Directed graph corresponding to trajectory X^{***}.

The trajectory is interrupted at step 9, where, before the states s_3 in $\pi_1^{***} = (***s_3)$ and s_1 in block $\pi_3^{***} = (*s_1)$ are exhausted, we return to block $\pi_0^{***} = (s_1)$, where moving to state s_1 is already forbidden.

The resulting Proposition 7 enables all possible trajectories of the Markov chain $X \in \{s_1, s_2, \ldots, s_m\}^N$ to be split into equivalence classes (equally probable trajectories).

Namely, consider some Markov chain trajectory $X = x_1 x_2 \ldots x_N$ as defining a class S and include $X' = x_1' x_2' \ldots x_N'$ in the class S if:

1) $x'_1 = x_1$ and $x'_N = x_N = s_\chi$ for some χ — the beginning and end of the trajectories X and X' are the same,

2) $\pi_\chi(X') \equiv \pi_\chi(X)$ is any permutation,

3) $\pi_i(X') \overset{\bullet}{\equiv} \pi_i(X)$, $i \neq \chi$, is a permutation with a fixed tail.

According to Proposition 7, Markov chain trajectories exist for such π-sequences. It follows from Proposition 6 that they are equally probable:

$$P_S(X') = P_S(X)\,.$$

3.2 Algorithm A: memory-intensive random bit extraction

Below, we consider Algorithm A for extracting random bits, which requires storing all π-sequences of a Markov chain trajectory in memory.

3.2.1 Description of Algorithm A

Input: Markov chain trajectory $X = x_1 x_2 \ldots x_N$, $x_i \in \{s_1, s_2, \ldots, s_m\}$.

Output: bit sequence $Y = \Psi(X) \in \{0,1\}^*$.

Algorithm A:

• generates π-sequences $\pi(X) = \{\pi_1(X), \pi_2(X), \ldots, \pi_\chi(X), \ldots, \pi_m(X)\}$, $s_\chi = x_N$ is the last element of the chain,

• for all $i \neq \chi$, the block $\pi_i(X)$ without the last element of the block $\pi_i(X)^{|\pi_i(X)|-1}$ is taken,

• for all $i \neq \chi$ blocks $\pi_i(X)^{|\pi_i(X)|-1}$, the m-ary Babkin algorithm is applied to the blocks: $Y_i = \Psi(\pi_i(X)^{|\pi_i(X)|-1})$,

• the block $\pi_\chi(X)$ is used entirely: $Y_\chi = \Psi(\pi_\chi(X))$,

• partial binary outputs are concatenated:

$$Y = \Psi(X) = \Psi\big(\pi_1(X)^{|\pi_1(X)|-1}\big) \,\|\, \Psi\big(\pi_2(X)^{|\pi_2(X)|-1}\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_\chi(X)\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_m(X)^{|\pi_m(X)|-1}\big)\,.$$

3.2.2 Equiprobability of binary output

Theorem 2 (Algorithm A).

Let the trajectory of the Markov chain $X = x_1 x_2 \ldots x_N$, $x_i \in \{s_1, s_2, \ldots, s_m\}$ be used as input for Algorithm A, $\Psi(\ldots)$, which is the binary output of the m-ary Babkin algorithm,

$$Y = \Psi(X) = \Psi\big(\pi_1(X)^{|\pi_1(X)|-1}\big) \,\|\, \Psi\big(\pi_2(X)^{|\pi_2(X)|-1}\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_\chi(X)\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_m(X)^{|\pi_m(X)|-1}\big)\,.$$

Then, the binary output $Y \in \{0,1\}^\ell$ obtained for some ℓ has probability

$$P(Y) = 2^{-\ell}\,.$$

Proof.

Below, we give the proof of Theorem 2 reported in [17], adding some useful clarifications and modifications.

For any Markov chain trajectory, we have a one-to-one correspondence:

$$X = x_1 x_2 \ldots s_\chi \Longleftrightarrow \pi(X) = \big\{\pi_1(X), \ldots, \pi_\chi(X), \ldots, \pi_m(X)\big\}\,. \tag{2}$$

All trajectories of the Markov chain are divided into equivalence classes (equally probable trajectories) $S \in G$. Trajectories X' and X belong to the same class S if

1) $x'_1 = x_1$, $x'_N = s_\chi$,

2) $\pi(X') = \big\{\pi_1(X') \overset{\bullet}{\equiv} \pi_1(X), \ldots, \pi_\chi(X') \equiv \pi_\chi(X), \ldots, \pi_m(X') \overset{\bullet}{\equiv} \pi_m(X)\big\}$.

It follows from (2) and the equality of the probabilities of the trajectories,

$$P_S(X') = P_S(X)\,,$$

that the probabilities of the π-sequences are equal:

$$P_S\big(\pi_1(X'), \ldots, \pi_\chi(X'), \ldots, \pi_m(X')\big) = P_S\big(\pi_1(X), \ldots, \pi_\chi(X), \ldots, \pi_m(X)\big)\,. \tag{3}$$

The binary output of Babkin's algorithm for Algorithm A is

$$Y = \Psi(X) = \Psi\big(\pi_1(X)^{|\pi_1(X)|-1}\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_\chi(X)\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_m(X)^{|\pi_m(X)|-1}\big) = Y_1 \,\|\, \ldots \,\|\, Y_m\,.$$

The blocks

$$\pi_1(X)^{|\pi_1(X)|-1}, \ldots, \pi_\chi(X), \ldots, \pi_m(X)^{|\pi_m(X)|-1}$$

feature lengths $n_1, \ldots, n_\chi, \ldots, n_m$ and some specific composition of symbols $\{s_1, \ldots, s_m\}$ in the blocks.

In this representation, the implementation of Babkin's algorithm is similar to that for an independent source (1) with the number of blocks $M = m$ and, this time, sequential processing of the blocks

$$\pi_1(X)^{|\pi_1(X)|-1}, \ldots, \pi_\chi(X), \ldots, \pi_m(X)^{|\pi_m(X)|-1}\,.$$

The difference is that one must first wait for the complete realization of the Markov chain $X = x_1 x_2 \ldots x_N$, form $\pi_1(X), \ldots, \pi_\chi(X), \ldots, \pi_m(X)$, and then, prior to applying Babkin's algorithm, exclude the extreme element in all blocks $\pi_i(X)$ except $\pi_\chi(X)$, where χ corresponds to the last element of the trajectory: $x_N = s_\chi$.

Similar to the case of an independent source, we denote by B_Y the set of Markov chain trajectories $X = x_1 x_2 \ldots x_N$ such that $\Psi(X) = Y$.

For a given equivalence class S, we have an admissible partition $Y_1, Y_2, \ldots, Y_m$ such that

$$Y = Y_1 \,\|\, \ldots \,\|\, Y_m = \Psi\big(\pi_1(X)^{|\pi_1(X)|-1}\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_\chi(X)\big) \,\|\, \ldots \,\|\, \Psi\big(\pi_m(X)^{|\pi_m(X)|-1}\big)\,.$$

It can be easily seen that an incomplete set being included in the equivalence class permutations does not cancel the constructive property of Babkin's algorithm: exactly one sequence $\pi_i(X)^{|\pi_i(X)|-1}$ or $\pi_i(X')^{|\pi_i(X')|-1}$ from the equivalence class yields Y_i or Y'_i whenever $|Y_i| = |Y'_i|$. From this, the main conclusion regarding the equality of the cardinalities of the preimages $|S \cap B_Y| = |S \cap B_{Y'}|$ easily follows.

Further, using the same probability of π-sequences belonging to the same equivalence class S (3), the proof of Theorem 2 is easily completed by analogy with the proof of Theorem 1 for an independent source.

Theorem 2 is proved.

3.3 Algorithm B: extracting random bits 'on the fly'

Using Algorithm A, equiprobable binary sequences can be generated from the input trajectory of a Markov chain $X = x_1 x_2 \ldots x_N$.

However, Algorithm A has certain drawbacks:

— the entire input sequence must be stored,

— the input cannot be a stream or an infinitely long sequence, because no output random bits 0 and 1 can be generated until the entire input trajectory of the Markov chain has been received,

— Algorithm A can not be computed in the expected linear time.

In practice, Algorithm B for generating random bits 0 and 1 'on the fly' turns out to be effective. This algorithm is not memory intensive and does not require storing the entire trajectory, but produces random 0s and 1s as the states of the Markov chain arrive.

3.3.1 Description of Algorithm B

Input: Markov chain trajectory $X = x_1 x_2 \dots x_N$, $x_i \in \{s_1, s_2, \dots, s_m\}$.

Algorithm B parameter: *window size* ϖ. *A window is needed to cut consecutive blocks from* $\pi_i(X)$.

Output: bit sequence $Y = \Psi(X) \in \{0,1\}^*$.

Algorithm B:

• parallelizes the current states of the Markov chain trajectory into current π-sequences $\pi_1(X), \dots, \pi_m(X)$, adding the next state s_j to the sequence $\pi_i(X)$ only if state s_i precedes s_j,

• if the sequence $\pi_i(X)$ has a current block F_{ik}, $k = 1, 2, \dots$, of size ϖ added, then:

— it is immediately sent for transformation $\Psi(F_{ik})$ if the last element in F_{ik} is s_i,

— if the last element in F_{ik} is not s_i, the block F_{ik} waits to be sent for processing until s_i appears in the Markov chain,

— the block F_{ik} is not sent for processing at all if it fails to wait for s_i to appear.

• partial binary outputs by blocks are concatenated:

$$Y = \Psi(F_{i_1 j_1}) \parallel \Psi(F_{i_2 j_2}) \parallel \dots \parallel \Psi(F_{i_L j_L}) .$$

Here, $i_1 j_1, i_2 j_2, \dots, i_L j_L$ are the numbers of blocks from $\pi_{i_1}(X), \pi_{i_2}(X), \dots, \pi_{i_L}(X)$ sequentially arriving for processing. It can be asserted that filled blocks in parallel π-sequences $\pi_1(X), \dots, \pi_m(X)$ are queued for processing. The sequence in which they are read by Algorithm B does not coincide with the natural-temporal appearance of blocks. As shown below, the order of reading blocks by Algorithm B for a given Markov chain trajectory is also preserved when reading blocks of any Markov chain trajectory in the equivalence class S (defined below), which opens a direct way of proving the equiprobability of the binary sequence at the output of Babkin's algorithm by analogy with an independent source.

Let the trajectory of the Markov chain $X = x_1 x_2 \dots x_N$ with the last element $x_N = s_\chi$ be read using Algorithm B.

For all $1 \leqslant i \leqslant m$, we can write

$$\pi_i(X) = F_{i1} F_{i2} \dots F_{i\alpha_i} E_i ,$$

where F_{ij}, $1 \leqslant j \leqslant \alpha_i$, are the blocks used to generate the output binary data.

Note that E_i is the remaining 'piece' of the output sequence $\pi_i(X)$ from which binary data is not produced.

For all bit-producing segments, we have

$$|F_{ij}| = \varpi ,$$

and

$$0 \leqslant |E_\chi| < \varpi , \quad 0 < |E_i| \leqslant \varpi , \quad i \neq \chi . \tag{4}$$

Comment.

Condition (4) appears due to the way Algorithm B is constructed.

1. Let the last element in the last block $F_{i\alpha_i}$ be i. Then, it may be the very last one, i.e., $i = \chi$. In this case, exactly $|E_\chi| = 0$.

2. Let $i \neq \chi$ be the last element of the entire trajectory. Then, to send the last block $F_{i\alpha_i}$ for processing, one must wait for s_i, but it is not the last one in the entire trajectory. Therefore, if we managed to wait for s_i, then something follows it and is included in E_i, i.e., a non-empty addition with length $|E_i| > 0$ appears. If we fail to wait for such s_i, the length of the last block $|E_i| = \varpi$, and, according to Algorithm B, it is not sent for processing.

Example 3. $N = 29$, $m = 2$, $\varpi = 3$.

$$X = s_1 s_2 s_2 s_1 s_1 s_2 s_2 s_1 s_1 s_2 s_2 s_1 s_1 \mathbf{s}_2 s_2 s_1 s_1 \mathbf{s}_2 s_2 s_1 s_1 s_2 s_2 s_1 s_1 s_2 s_2 s_1 s_1 ,$$

$$\pi_1(X) = \overbrace{s_2 - - s_1 s_2}^{F_{11}} - - \overbrace{s_1 s_2 - - s_1}^{F_{12}} \overbrace{s_2 - - s_1 s_2}^{F_{13}} - - \overbrace{s_1 s_2 - - s_1}^{F_{14}} \overbrace{s_2 - - s_1}^{E_1} ,$$

$$\pi_2(X) = - \overbrace{s_2 s_1 - - s_2}^{F_{21}} \overbrace{s_1 - - s_2 s_1}^{F_{22}} - - \overbrace{s_2 s_1 - - s_2}^{F_{23}} \overbrace{s_1 - - s_2 s_1}^{F_{24}} - - \overbrace{s_2 s_1}^{E_2} - .$$

The order is which blocks are read by Algorithm B is

$$F_{21} \dots F_{11} \dots F_{12} \dots F_{22} \dots F_{23} \dots F_{13} \dots F_{14} \dots F_{24} .$$

The natural-temporal order of reading blocks is

$$F_{11} \dots F_{21} \dots F_{22} \dots F_{12} \dots F_{13} \dots F_{23} \dots F_{24} \dots F_{14} .$$

As we can see, these orders of reading blocks are not the same.

Theorem 3 (Algorithm B).

Let the trajectory of the Markov chain $X = x_1 x_2 \dots x_N$, $x_i \in \{s_1, s_2, \dots, s_m\}$, *be used as input for Algorithm B,* $\Psi(\dots)$, *the binary output of the m-ary Babkin algorithm,*

$$Y = \Psi(X) = \Psi(F_{i_1 j_1}) \parallel \Psi(F_{i_2 j_2}) \parallel \dots \parallel \Psi(F_{i_L j_L}) .$$

Then, the binary output $Y \in \{0,1\}^\ell$ *obtained for some* ℓ *has probability*

$$P(Y) = 2^{-\ell} .$$

Proof.

Below, we present the proof of Theorem 3 reported in [17], adding some useful clarifications and changes.

We divide all possible trajectories of the Markov chain $X \in \{s_1, s_2, \dots, s_n\}^N$ into equivalence classes (of equal probability).

Consider a trajectory X as defining a class S and include X' in class S if:

1) $x_1 = x'_1$ and $x_N = x'_N$ — the beginning and end coincide;

2) for all $1 \leqslant i \leqslant m$

$$\begin{aligned} \pi_i(X) &= F_{i1} F_{i2} \dots F_{i\alpha_i} E_i , \\ \pi_i(X') &= F'_{i1} F'_{i2} \dots F'_{i\alpha_i} E_i , \end{aligned} \tag{5}$$

where F_{ij} and F'_{ij} are blocks used to generate output data;

3) for all i, j we have $F_{ij} \equiv F'_{ij}$, i.e., a complete permutation.

Note that the unprocessed parts E_i for trajectories X and X' are the same.

According to Proposition 7 (Acceptable permutations), such a trajectory X' exists. It is sufficient to note here that, according to the comment given above, a complete permutation of the entire sequence $\pi_i(X)$ is only possible when $i = \chi$, i.e., the last element of the entire trajectory X, and it may happen that the length $|E_\chi| = 0$. If $i \neq \chi$, a non-empty addition $|E_i| > 0$ necessarily appears. Since X' is skipped when proceeding to the construction of E_i, these conditions correspond exactly to the conditions of Proposition 7. In addition, according to Proposition 6, trajectories in the class S are equally probable.

In sequences $\pi_i(X) = F_{i1}F_{i2}\dots F_{i\alpha_i}E_i$, $1 \leqslant i \leqslant m$, the sequence of blocks $F_{i1}, F_{i2}, \dots, F_{i\alpha_i}$ determines the order in which these blocks were read for processing precisely for the states following the s_ith state.

At the same time, the order of reading blocks is

$$F_{11}, \dots, F_{1\alpha_1}, F_{21}, \dots, F_{2\alpha_2}, \dots, F_{m1}, \dots, F_{m\alpha_m}. \tag{6}$$

The real-time appearance of states of the Markov chain by Algorithm B can be quite arbitrary, but strictly determined by the original Markov chain $X = x_1x_2\dots x_N$ (see Example 3 above). To simplify the reasoning, we assume that it is exactly like this (6).

Now, assume that the order of reading blocks by Algorithm B for all trajectories $X' = x'_1x'_2\dots x'_N$ from the class S is preserved and equal to

$$F'_{11}, \dots, F'_{1\alpha_1}, F'_{21}, \dots, F'_{2\alpha_2}, \dots, F'_{m1}, \dots, F'_{m\alpha_m}.$$

Similar to the case of an independent source, B_Y denotes the set of trajectories of the Markov chain $X = x_1x_2\dots x_N$ such that $\Psi(X) = Y$, and, for a given equivalence class S, we have an admissible partition

$$Y_{11}, \dots, Y_{1\alpha_1}, Y_{21}, \dots, Y_{2\alpha_2}, \dots, Y_{m1}, \dots, Y_{m\alpha_m},$$

such that

$$Y = \Psi(X) = \Psi(Y_{11}) \parallel \dots \parallel \Psi(Y_{1\alpha_1}) \parallel \Psi(Y_{21}) \parallel \dots$$
$$\dots \parallel \Psi(Y_{2\alpha_2}) \parallel \dots \parallel \Psi(Y_{m1}) \parallel \dots \parallel \Psi(Y_{m\alpha_m}).$$

Then, if the order of reading blocks by Algorithm B is preserved, the constructive property of Babkin's algorithm holds: exactly one sequence F_{ij}, F'_{ij} yields Y_{ij}, Y'_{ij} whenever $|Y_i| = |Y'_i|$. From this, the main relation on the equality of the cardinalities of the preimages easily follows: $|S \cap B_Y| = |S \cap B_{Y'}|$ whenever $|Y| = |Y'|$. Next, the trajectories of the Markov chain inside the equivalence classes S being equally probable, the proof of Theorem 3 is easily completed by analogy with the proof of Theorem 1 for an independent source.

It remains to prove that the order of reading blocks by Algorithm B is preserved for all trajectories of the Markov chain $X = x_1x_2\dots x_N$ from the equivalence class S.

1. It is clear that it is sufficient to prove the equality of the order of reading blocks for only one trajectory $X' = x'_1x'_2\dots x'_N$ in the class S, which differs from $X = x_1x_2\dots x_N$ by one transposition for an arbitrary i in an arbitrary block F_{ik}:

$$X' = x'_1x'_2\dots x'_N \Longleftrightarrow \left\{ \begin{array}{l} \pi_i(X') = F_{i1}F_{i2}\dots F'_{ik}\dots F_{i\alpha_i}E_i\,, \\ \pi_j(X') = F_{j1}F_{j2}\dots F_{j\alpha_j}E_j\,,\ j \neq i \end{array} \right\}.$$

Then, moving sequentially along the transpositions, we can go to any permutation with preservation of the order of reading blocks.

2. Consider the part of the original trajectory of the Markov chain X^a that ends in the state x_a, which is followed the reading of block F_{ik} for conversion into bits. According to the reading of blocks by Algorithm B, $x_a = s_i$, and, in addition,

$$\pi_i(X^a) = F_{i1}F_{i2}\dots F_{ik}\,,$$
$$\pi_j(X^a) = F_{j1}F_{j2}\dots F_{jk_j}, \widehat{E}_j,\ \ j \neq i\,,$$
$$0 < |\widehat{E}_j| \leqslant \varpi\,.$$

We use the notation $\widehat{E}_j$ because X^a is part of the trajectory X.

Clearly, $|\widehat{E}_i| = 0$ because F_{ik} is the last block to be read, in which case either the last element in F_{ik} is s_i or s_i is part of some $\widehat{E}_j$, after which F_{ik} is read.

Can a part of the Markov chain trajectory be constructed by transposing the block F_{ik}?

Since the last element $x_a = s_i$, we can perform any transposition in the block F_{ik} — it is as if we made any permutation on the whole $\pi_i(X^a) = F_{i1}F_{i2}\dots F_{ik}$, including the last element.

We get

$$\pi_i(X'^a) = F_{i1}F_{i2}\dots F'_{ik}\,,$$
$$\pi_j(X'^a) = F_{j1}F_{j2}\dots F_{jk_j}\widehat{E}_j\,,\ \ j \neq i\,, \tag{7}$$
$$|\widehat{E}_j| > 0\,,$$

which is acceptable for constructing the initial segment X'^a of the Markov chain trajectory according to Proposition 7.

In addition, according to Proposition 7, the length $|X'^a| = |X^a|$ and the last elements coincide: $x'_a = x_a = s_i$. This implies that, in our configuration, for the initial segment of the Markov chain trajectory, block F'_{ik} will definitely be sent for processing. Will it be the last one?

If $x'_a = s_i$ and is the last one in F'_{ik}, F'_{ik} is sent for processing (zeroed) last.

If this is not the case, there must be another s_i that zeroes F'_{ik} before some F_{js}, and then ours, finally the last $x'_a = s_i$, must appear.

We must then get

$$\pi_i(X'^a) = F_{i1}F_{i2}\dots F'_{ik}s_i\,.$$

We have a contradiction. Therefore, block F'_{ik} is read last.

3. It is easy to see that, after the element x_a, the trajectories of X and X' coincide, and the order of reading blocks does not change.

4. Consider the initial part of the trajectory of the Markov chain X'^b, which ends at the state x_b ($b = i$), which is followed by the first element of block F'_{ik}. It is easy to see that, before the element x_b, the trajectories of X and X' coincide, and the order of reading blocks does not change.

From this, we can conclude that the order of reading blocks by Algorithm B for trajectories X and X' is the same.

Theorem 3 is proved.

We now consider an example illustrating the preservation of the order of reading blocks by Algorithm B.

Example 4. $N = 29$, $m = 3$, $\varpi = 3$.

1. *Initial trajectory*

$$X = \overbrace{s_1s_2s_2s_3s_2s_3s_1s_1s_2s_2s_3s_3s_1}^{X^b}\, s_1s_2s_1s_3s_2s_1\, \overbrace{s_3s_2s_1s_2s_3s_1s_1s_2s_3s_2}^{X_a}\,.$$

Movement along blocks

$$\pi_1(X) = \overbrace{s_2 - - - - - s_1s_2}^{F_{11}} - - - - \overbrace{s_1s_2 - s_3}^{F_{12}} - - \overbrace{s_3 - - s_2 - - s_1}^{F_{13}}\,\overbrace{s_2}^{E_1} - - ,$$

$$\pi_2(X) = -\overbrace{s_2s_3 - s_3}^{F_{21}} - - - \overbrace{s_2s_3 - - - - - s_1}^{F_{22}} - - \overbrace{s_1 - - s_1 - s_3}^{F_{23}} - - - \overbrace{s_3}^{E_2} - ,$$

$$\pi_3(X) = - - - \overbrace{s_2 - s_1 - - - - - s_3}^{F_{31}}\overbrace{s_1 - - - - - s_2 - - s_2}^{F_{32}} - - - \overbrace{s_1 - - - s_2}^{E_3}\,.$$

The order of reading blocks by Algorithm B is

$$F_{21}\ldots F_{31}\ldots F_{11}\ldots F_{22}\ldots F_{12}\ldots F_{32}\ldots F_{13}\ldots F_{23}\,.$$

2. *Trajectory after transposition in the block*

We now perform transposition in $F_{12} = s_1s_2s_3 \to F'_{12} = s_1s_3s_2$. After transposition,

$$\pi_1(X') = \{F_{11} = s_2s_1s_2, F'_{12} = s_1s_3s_2, F_{13} = s_3s_2s_1, E_1 = s_2\}\,,$$
$$\pi_2(X') = \{F_{21} = s_2s_3s_3, F_{22} = s_2s_3s_1, F_{23} = s_1s_1s_3, E_2 = s_3\}\,,$$
$$\pi_3(X') = \{F_{31} = s_2s_1s_3, F_{32} = s_1s_2s_2, E_3 = s_1s_2\}\,.$$

Trajectory

$$X' = \overbrace{s_1s_2s_2s_3s_2s_3s_1s_1s_2s_2s_3s_3s_1}^{X'^b = X^b}\, s_1s_3s_2s_1s_2s_1\, \overbrace{s_3s_2s_1s_2s_3s_1s_1s_2s_3s_2}^{X'_a = X_a}\,.$$

Movement along blocks

$$\pi_1(X') = \overbrace{s_2 - - - - - s_1s_2}^{F_{11}} - - - - \overbrace{s_1s_3 - - s_2}^{F'_{12}} - \overbrace{s_3 - - s_2 - - s_1}^{F_{13}}\,\overbrace{s_2}^{E_1} - - ,$$

$$\pi_2(X') = -\overbrace{s_2s_3 - s_3}^{F_{21}} - - - \overbrace{s_2s_3 - - - - - s_1}^{F_{22}} - \overbrace{s_1 - - s_1 - s_3}^{F_{23}} - - - \overbrace{s_3}^{E_2} - ,$$

$$\pi_3(X') = - - - \overbrace{s_2 - s_1 - - - - - s_3}^{F_{31}}\overbrace{s_1 - - s_2 - - - - s_2}^{F_{32}} - - - \overbrace{s_1 - - - s_2}^{E_3}\,.$$

The order of block reading by Algorithm B is

$$F_{21}\ldots F_{31}\ldots F_{11}\ldots F_{22}\ldots F'_{12}\ldots F_{32}\ldots F_{13}\ldots F_{23}\,.$$

The order of block reading by Algorithm B for trajectories X and X' is the same.

3.4 Example of non-uniqueness of binary output of Babkin's algorithm with natural-temporal reading of blocks for processing

One of the main points in the proof of Theorem 3 is based on the fact that, in the equivalence class S, the number of preimages of the full binary output $Y \in \{0,1\}^*$ with a fixed length $|Y| = \ell$ is the same for any binary composition of Y.

This is the case if the order of reading blocks in the equivalence class of Markov chain trajectories (π-sequences) corresponds to Algorithm B.

The question arises as to whether this property is preserved with natural-temporal reading of blocks, which is simpler in practical implementation.

We provide below an example for which this property is violated with natural-temporal reading of blocks: the number of preimages is different for some Y and Y' of the same length.

Consider the original trajectory of the Markov chain X as defining a class S, $N = 10$, $m = 2$, $\varpi = 4$:

$$X = s_1s_1s_1s_2s_2s_2s_1s_2s_2\boldsymbol{s_1}\,,$$
$$\pi_1(X) = \overbrace{s_1s_1s_2 - - - s_2}^{F_{11}} - - ,$$
$$\pi_2(X) = - - - \overbrace{s_2s_2s_1 - s_2}^{F_{21}}\overbrace{s_1}^{E_2}\,.$$

According to Algorithm B, block $F_{21} = (s_2s_2s_1s_2)$ will be processed first, since s_2 is the last element in the block, and then $F_{11} = (s_1s_1s_2s_2)$.

The order of reading by Algorithm B is preserved for all trajectories X of the equivalence class determined by permutations in blocks $F_{11} = (s_1s_1s_2s_2)$ and $F_{21} = (s_2s_2s_1s_2)$. The natural-temporal order of reading blocks is $F_{11} = (s_1s_1s_2s_2)$, $F_{21} = (s_2s_2s_1s_2)$.

Using the original Markov chain trajectory, we construct an equivalence class S.

There are six permutations inside the block $F_{11} = (\boldsymbol{s_1}\boldsymbol{s_1}s_2s_2)$ and four permutations inside the block $F_{21} = (s_2s_2\boldsymbol{s_1}s_2)$.

In total, the equivalence class S will include 24 permutations, defining 24 equally probable Markov chain trajectories.

In Tables 4 and 5 (corresponding to Tables 2 and 3 of Section 2.2), for the first block F_{11}, we denote (i_1, i_2) — the positions of occurrence of state s_1 — while, for the second block F_{21}, we denote (i_1) — the position of occurrence of state s_1. Babkin's numbering method:

$$\mathrm{Num}\,(i_1, i_2) = C^1_{i_1-1} + C^2_{i_2-1}\,, \quad \mathrm{Num}\,(i_1) = C^1_{i_1-1}\,,$$

where $C^i_j = 0$ if $j < i$. The fourth column specifies the binary output of Babkin's algorithm.

Table 4.

No.	F_{11}, permutations	Num (i_1, i_2)	Binary output
1	$(\boldsymbol{s_1}\boldsymbol{s_1}s_2s_2)$	Num $(1,2) = 0$	0
2	$(\boldsymbol{s_1}s_2\boldsymbol{s_1}s_2)$	Num $(1,3) = 1$	1
3	$(s_2\boldsymbol{s_1}\boldsymbol{s_1}s_2)$	Num $(2,3) = 2$	00
4	$(\boldsymbol{s_1}s_2s_2\boldsymbol{s_1})$	Num $(1,4) = 3$	01
5	$(s_2\boldsymbol{s_1}s_2\boldsymbol{s_1})$	Num $(2,4) = 4$	10
6	$(s_2s_2\boldsymbol{s_1}\boldsymbol{s_1})$	Num $(3,4) = 5$	11

Table 5.

No.	F_{11}, permutations	Num (i_1)	Binary output
1	$(\boldsymbol{s_1}s_2s_2s_2)$	Num (1) = 0	00
2	$(s_2\boldsymbol{s_1}s_2s_2)$	Num (2) = 1	01
3	$(s_2s_2\boldsymbol{s_1}s_2)$	Num (3) = 2	10
4	$(s_2s_2s_2\boldsymbol{s_1})$	Num (4) = 3	11

For each pair of blocks (F_{11}, F_{21}) and an additional block $E_2 = s_1$, the trajectories of the Markov chain X are constructed, and the sequence of reading blocks according to Algorithm B and the natural-temporal (NT) order are determined. For each order of reading, the binary output of Babkin's algorithm $Y = \Psi(X)$ is constructed.

For example, for blocks $F_{11} = (\boldsymbol{s_1}\boldsymbol{s_1}s_2s_2)$, $F_{21} = (s_2\boldsymbol{s_1}s_2s_2)$, which corresponds to the choice of numbers (1, 2) in the first columns of Tables 4 and 5, we have

$$X = s_1\boldsymbol{s_1}\boldsymbol{s_1}s_2s_2\boldsymbol{s_1}s_2s_2s_2s_1\,,$$
$$\pi_1(X) = \overbrace{\boldsymbol{s_1}\boldsymbol{s_1}s_2--s_2}^{F_{11}}---\,,$$
$$\pi_2(X) = ---\overbrace{s_2\boldsymbol{s_1}-s_2s_2}^{F_{21}}\overbrace{s_1}^{E_2}\,.$$

We obtain the following reading order.

1. **Algorithm B**: F_{21}, F_{11}, binary output $Y = \Psi(X) = 01\,\|\,0 = 010$.

2. **NT**: F_{11}, F_{21} binary output $Y = \Psi(X) = 0\,\|\,01 = 001$.

In Table 6, the binary outputs are ordered according to the choice of pairs of numbers (permutations) in the first columns of Tables 4 and 5:

$$\text{No. } 1 \text{ — } (1,1),\ \text{No. } 2 \text{ — } (1,2), \ldots, \text{No. } 24 \text{ — } (6,4)\,.$$

The results obtained show that, with natural-time NT block reading,

— with output length $|Y| = 3$, the number of preimages of combination $Y = \mathbf{110}$ is 2; combination $Y = \mathbf{001}$ has no preimages, and the remaining combinations $Y \in \{0,1\}^3$ have one preimage each,

— with output length $|Y| = 4$, the number of preimages of combination $Y = \mathbf{0001}$ is 2; combination $Y = \mathbf{0100}$ has no preimages, and the remaining combinations $Y \in \{0,1\}^4$ have one preimage each.

3.5 Two-state Markov chain of finite order r

The above algorithms for extracting random bits are constructed for simple ($r = 1$) stationary Markov chains with an initial distribution $P(s_i)$ and a transition probability matrix $P(s_j\,|\,s_i)$, $s_i, s_j \in \{s_1, \ldots, s_m\}$.

The probability of an arbitrary trajectory of the Markov chain $X_N = x_1x_2\ldots x_N$ is defined as

$$P(X_N) = P(x_1)\prod_{i=1}^{N-1} P(x_{i+1}\,|\,x_i)\,.$$

In practice, in constructing a quantum PRNG associated with the detection of photocounts, we are dealing with symbols of a binary alphabet $A = \{*, \sqcup\}$ and some arbitrary order $r \geqslant 2$.

In practical applications, it is convenient to represent the alphabet in binary form: $A = \{0,1\}$.

Table 6.

No.	Algorithm B reading	Binary output Y	NT reading	Binary output Y
1 (1,1)	F_{21}, F_{11}	000	F_{21}, F_{11}	000
2 (1,2)	F_{21}, F_{11}	010	F_{11}, F_{21}	001
3 (1,3)	F_{21}, F_{11}	100	F_{11}, F_{21}	010
4 (1,4)	F_{21}, F_{11}	110	F_{21}, F_{11}	**110**
5 (2,1)	F_{21}, F_{11}	001	F_{11}, F_{21}	100
6 (2,2)	F_{21}, F_{11}	011	F_{11}, F_{21}	101
7 (2,3)	F_{21}, F_{11}	101	F_{11}, F_{21}	**110**
8 (2,4)	F_{21}, F_{11}	111	F_{21}, F_{11}	111
9 (3,1)	F_{21}, F_{11}	0000	F_{11}, F_{21}	0000
10 (3,2)	F_{21}, F_{11}	0100	F_{11}, F_{21}	**0001**
11 (3,3)	F_{21}, F_{11}	1000	F_{11}, F_{21}	0010
12 (3,4)	F_{21}, F_{11}	1100	F_{21}, F_{11}	1100
13 (4,1)	F_{21}, F_{11}	0001	F_{21}, F_{11}	**0001**
14 (4,2)	F_{21}, F_{11}	0101	F_{21}, F_{11}	0101
15 (4,3)	F_{21}, F_{11}	1001	F_{21}, F_{11}	1001
16 (4,4)	F_{21}, F_{11}	1011	F_{21}, F_{11}	1101
17 (5,1)	F_{21}, F_{11}	0010	F_{21}, F_{11}	0010
18 (5,2)	F_{21}, F_{11}	0110	F_{21}, F_{11}	0110
19 (5,3)	F_{21}, F_{11}	1010	F_{21}, F_{11}	1010
20 (5,4)	F_{21}, F_{11}	1110	F_{21}, F_{11}	1110
21 (6,1)	F_{21}, F_{11}	0011	F_{21}, F_{11}	0011
22 (6,2)	F_{21}, F_{11}	0111	F_{21}, F_{11}	0111
23 (6,3)	F_{21}, F_{11}	1011	F_{21}, F_{11}	1011
24 (6,4)	F_{21}, F_{11}	1111	F_{21}, F_{11}	1111

A Markov chain with two states $A = \{0,1\}$, of order $r \geqslant 2$, is defined by the initial distribution

$$P(\varepsilon_1, \ldots, \varepsilon_r),\quad \sum_{(\varepsilon_1,\ldots,\varepsilon_r)\in\{0,1\}^r} P(\varepsilon_1, \ldots, \varepsilon_r) = 1\,,$$

and a transition probability matrix of size $2^r \times 2$

$$\|P(\varepsilon_{r+1}|\varepsilon_1, \ldots, \varepsilon_r)\|,\ \varepsilon_{r+1} \in \{0,1\},\ (\varepsilon_1, \ldots, \varepsilon_r) \in \{0,1\}^r\,.$$

The probability of the trajectory

$$E = \varepsilon_1\varepsilon_2\ldots\varepsilon_L,\quad \varepsilon_i \in \{0,1\}$$

is defined as

$$P(E) = P(\varepsilon_1, \ldots, \varepsilon_r)\prod_{i=1}^{L-r} P(\varepsilon_{i+r}\,|\,\varepsilon_i, \ldots, \varepsilon_{i+r-1})\,.$$

A transition to a simple Markov chain of order $r = 1$ can be made by enlarging the alphabet, where the number of states is $m = 2^r$.

We introduce a new alphabet,

$$A' = \{s_1, \ldots, s_m\} = \{s_1 = (0\ldots0), \ldots, s_m = (1\ldots1)\}\,,$$

by combining adjacent r bits in the trajectory $E_L = \varepsilon_1\varepsilon_2\ldots\varepsilon_L$ with a 1-bit engagement into one symbol, and obtain a trajectory of length $N = L - r$:

$$X = (\varepsilon_1\varepsilon_2\ldots\varepsilon_r)(\varepsilon_2\varepsilon_3\ldots\varepsilon_{r+1})(\varepsilon_3\varepsilon_4\ldots\varepsilon_{r+2})\ldots$$
$$\ldots(\varepsilon_{L-r+1}\varepsilon_{L-r+2}\ldots\varepsilon_L) = x_1x_2\ldots x_N\,,\quad x_i \in \{0,1\}^r\,.$$

We now consider the transition probabilities

$$P(x_{i+1}\,|\,x_i) = P(\boldsymbol{\varepsilon_{i+1}}, \ldots, \boldsymbol{\varepsilon_{i+r-1}}, \varepsilon_{i+r}\,|\,\varepsilon_i, \boldsymbol{\varepsilon_{i+1}}, \ldots, \boldsymbol{\varepsilon_{i+r-1}})\,.$$

The part of the condition $\varepsilon_{i+1}, \ldots, \varepsilon_{i+r-1}$ (highlighted in bold) is fixed, so the transition probability depends only on the value of ε_{i+r} and, as can be easily seen, is

$$\begin{aligned} P(x_{i+1}\,|\,x_i) &= P(\boldsymbol{\varepsilon_{i+1}}, \ldots, \boldsymbol{\varepsilon_{i+r-1}}, \varepsilon_{i+r}\,|\,\varepsilon_i, \boldsymbol{\varepsilon_{i+1}}, \ldots, \boldsymbol{\varepsilon_{i+r-1}}) \\ &= P(\varepsilon_{i+r}\,|\,\varepsilon_i, \ldots, \varepsilon_{i+r-1})\,. \end{aligned}$$

Then, the probability of the trajectory

$$\begin{aligned} P(X) = P(E) &= P(\varepsilon_1, \ldots, \varepsilon_r) \prod_{i=1}^{L-r} P(\varepsilon_{i+r}\,|\,\varepsilon_i, \ldots, \varepsilon_{i+r-1}) \\ &= P(\varepsilon_1, \ldots, \varepsilon_r) \prod_{i=1}^{L-r} P(\varepsilon_{i+1}, \ldots, \varepsilon_{i+r-1}, \varepsilon_{i+r}\,|\,\varepsilon_i, \ldots, \varepsilon_{i+r-1}) \\ &= P(x_1) \prod_{i=1}^{N} P(x_{i+1}\,|\,x_i)\,. \end{aligned}$$

Thus, we arrive at a stationary Markov chain of order 1.

Here, we have an essentially sparse matrix of transition probabilities: each row contains only two nonzero transition probabilities.

This is due to the fact that, for any state $s_i = (\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_r)$, a transition to only two states is possible: $(\varepsilon_2, \ldots, \varepsilon_r, 0)$ and $(\varepsilon_2, \ldots, \varepsilon_r, 1)$.

In the example of $r = 2$, the matrix of transition probabilities (Table 7) looks like what follows.

Table 7.

	00 (s_1)	01 (s_2)	10 (s_3)	11 (s_4)
00 (s_1)	$P(s_1\|s_1)$	$P(s_2\|s_1)$	0	0
01 (s_2)	0	0	$P(s_3\|s_2)$	$P(s_4\|s_2)$
10 (s_3)	$P(s_1\|s_3)$	$P(s_2\|s_1)$	0	0
11 (s_4)	0	0	$P(s_3\|s_4)$	$P(s_4\|s_4)$

The generated π-sequences

$$\{\pi_i(X) = F_{i1}F_{i2}\ldots F_{i\alpha_i}E_i, 1 \leqslant i \leqslant m\}$$

will be binary, and it is not necessary to use Babkin's m-ary algorithm.

Example of Algorithm B operation for $r = 2$, $N = 36$, $m = 4$, $\varpi = 3$.

At the top of the trajectory E, superscripts indicate the states of the chain of Markov trajectory X with an enlarged alphabet, and subscripts, the number of the step,

$$E = 01_1^2 0_2^3 0_3^1 1_4^2 0_5^3 1_6^2 1_7^4 0_8^3 1_9^2 0_{10}^3 0_{11}^1 1_{12}^2 1_{13}^4 1_{14}^4 1_{15}^4 0_{16}^3 0_{17}^1 1_{18}^2 0_{19}^3 1_{20}^2$$
$$0_{21}^3 0_{22}^1 0_{23}^1 1_{24}^2 1_{25}^4 1_{26}^4 0_{27}^3 0_{28}^1 0_{29}^1 0_{30}^1 1_{31}^2 1_{32}^4 0_{33}^3 0_{34}^1 1_{35}^2 0_{36}^3\,,$$

π-sequence:

$$\pi_1(X) = {}_{1\cdot2\cdot3}\overbrace{2_{4\cdot5\cdot6\cdot7\cdot8\cdot9\cdot10\cdot11}2_{12\cdot13\cdot14\cdot15\cdot16\cdot17}2_{18}}^{F_{11}}{}_{\cdot19\cdot20\cdot21\cdot22}$$
$$\overbrace{1_{23}2_{24\cdot\underset{29}{25}\cdot26\cdot27\cdot28}\mathbf{1}}^{F_{12}}\overbrace{1_{30}2_{31\cdot32\cdot33\cdot34}2_{35}}^{E_1}{}_{\cdot36}\,,$$

$$\pi_2(X) = {}_1\overbrace{3_{2\cdot3\cdot4}3_{5\cdot6}4_7}^{F_{21}}{}_{\cdot8\cdot9}\overbrace{\mathbf{3}_{10\cdot11\cdot12}4_{13\cdot14\cdot15\cdot16\cdot17\cdot18}3_{19}}^{F_{22}}{}_{\cdot20}$$
$$\overbrace{3_{21\cdot22\cdot23\cdot24}4_{25\cdot26\cdot27\cdot28\cdot29\cdot30\cdot31}4_{32}}^{F_{23}}{}_{\cdot33\cdot34\cdot35}\overbrace{3_{36}}^{E_2}\,,$$

$$\pi_{3(\chi)}(X) = {}_{1\cdot2}\overbrace{1_{3\cdot4\cdot5}2_{6\cdot7\cdot8}\mathbf{2}_9}^{F_{31}}{}_{\cdot10}\overbrace{1_{11\cdot12\cdot13\cdot14\cdot15\cdot16}1_{17\cdot18\cdot19}2_{20}}^{F_{32}}{}_{\cdot21}$$
$$\overbrace{1_{22\cdot23\cdot24\cdot25\cdot26\cdot27}1_{28\cdot29\cdot30\cdot31\cdot32\cdot33}1_{34}}^{F_{33}}{}_{\cdot35\cdot36}\,,$$

$$\pi_4(X) = {}_{1\cdot2\cdot3\cdot4\cdot5\cdot6\cdot7}\overbrace{3_{8\cdot9\cdot10\cdot11\cdot12\cdot13}4_{14}\mathbf{4}_{15}}^{F_{41}}$$
$$\overbrace{3_{16\cdot17\cdot18\cdot19\cdot20\cdot21\cdot22\cdot23\cdot24\cdot25}4_{26}3_{27}}^{F_{42}}{}_{\cdot28\cdot29\cdot30\cdot31\cdot32}$$
$$\overbrace{3_{33}}^{E_4}{}_{\cdot34\cdot35\cdot36}\,.$$

The order of reading blocks by Algorithm B is

$$F_{21}\ldots F_{31}\ldots F_{41}\ldots F_{22}\ldots F_{32}\ldots F_{11}\ldots$$
$$\ldots F_{12}\ldots F_{42}\ldots F_{23}\ldots F_{33}\,.$$

4. Conclusion

The epigraph given at the beginning of this review reflects in the best possible way the situation with regard to obtaining truly random sequences of 0s and 1s.

As can be seen from the above discussion, Nature sets some fundamental physical limitations on the rate of decrease of correlations in time, which, in turn, follow from the fact that the spectrum of a stable physical system must lie on the positive semi-axis of energies (frequencies). For this reason, it is possible to 'reach' true randomness only in an unlimited time, i.e., for the successive measurement results to be independent, it is necessary to separate the measurements by an unlimited time. Correlations (dependence) between measurements extend to an unlimited depth in time.

If successive measurements were independent, it would be possible to present effective methods for extracting truly random sequences of 0s and 1s from the original sequence.

Real experiments are always carried out in a finite time interval, so the results of successive measurements cannot be made independent.

In fact, Nature only allows counting and taking into account correlations between measurements to a finite depth in time. Moreover, the explicit (functional) form of correlations in a real experiment is unknown. In such a situation, approximations are a must. An adequate approximation is to take into account correlations to a finite depth, which is achieved using stationary Markov chains of finite order. In this approximation, correlations are described by transition (conditional) probabilities between measurement results. The explicit form of the transition probabilities themselves is unknown and is not required when constructing algorithms for extracting random bits.

As demonstrated above, with a finite depth of correlations, it is possible to obtain provably random output bit sequences, even if events in the original sequence are not independent. In this approach, unlike other techniques, for example, using probabilistic extractors, in fact, only one assumption — the finite depth of correlations — is employed. Thus, any approaches to obtaining true randomness, due to the fundamental limitations of Nature, are only an approximation. The issue is only how adequately a specific approx-

imation describes the real situation and how many assumptions it involves.

Proving the randomness of the output bit sequence is a nontrivial task, even in the chosen approximation, stationary Markov chains. Not all approaches allow obtaining provable randomness in the sense discussed above.

Acknowledgments
We are grateful for the active cooperation, useful discussions, and support to V O Mironkin, our colleague from the Academy of Cryptography of the Russian Federation, to K A Balygin, A N Klimov, and S P Kulik, colleagues from the Center for Quantum Technologies of Moscow State University, and to V A Kiryukhin, a specialist at the SFB Laboratory, Moscow.

References

1. Paley R, Wiener N *Fourier Transforms in the Complex Domain* (American Mathematical Society. Colloquium Publ., Vol. 19) (New York: American Mathematical Society, 1934); Translated into Russian: Wiener N, Paley R *Preobrazovanie Fur'e v Kompleksnoi Oblasti* (Moscow: Nauka, 1964)
2. Fonda L, Ghirardi G C, Rimini A *Rep. Prog. Phys.* **41** 587 (1978)
3. Molotkov S N *J. Exp. Theor. Phys.* **130** 370 (2020); *Zh. Eksp. Teor. Fiz.* **157** 442 (2020)
4. Molotkov S N *Laser Phys. Lett.* **20** 035202 (2023)
5. Arbekov I M, Molotkov S N *Phys. Usp.* **64** 617 (2021); *Usp. Fiz. Nauk* **191** 651 (2021)
6. Rukhin A et al. "A statistical test suite for random and pseudorandom number generators for cryptographic applications," NIST Special Publication 800-22, Revision 1a (Gaithersburg, MD: National Institute of Standards and Technology, 2010) https://doi.org/10.6028/NIST.SP.800-22r1a
7. Ivchenko G I, Medvedev Yu I *Vvedenie v Matematicheskuyu Statistiku* (Introduction to Mathematical Statistics) (Moscow: Izd. LKI, 2010)
8. Turan M S et al. "Recommendation for the entropy sources used for random bit generation," NIST Special Publication 800-90B (Gaithersburg, MD: National Institute of Standards and Technology, 2018) https://doi.org/10.6028/NIST.SP.800-90B
9. Impagliazzo R, Levin L A, Luby M "Pseudo-random generation from one-way functions," in *STOC89: 21st Annual ACM Symp. on the Theory of Computing, Seattle, Washington, USA May 14–17, 1989* (New York: Association for Computing Machinery, 1989) p. 12, https://doi.org/10.1145/73007.73009
10. Bennett C H et al. *IEEE Trans. Inform. Theory* **41** 1915 (1995)
11. Babkin V F *Probl. Peredachi Inform.* **7** (4) 13 (1971)
12. Molotkov S N *JETP Lett.* **105** 395 (2017); *Pis'ma Zh. Eksp. Teor. Fiz.* **105** 374 (2017)
13. Balygin K A et al. *J. Exp. Theor. Phys.* **126** 728 (2018); *Zh. Eksp. Teor. Fiz.* **153** 879 (2018)
14. Balygin K A et al. *Laser Phys. Lett.* **14** 125207 (2017)
15. Balygin K A et al. *JETP Lett.* **106** 470 (2017); *Pis'ma Zh. Eksp. Teor. Fiz.* **106** 451 (2017)
16. Blum M *Combinatorica* **6** 97 (1986) https://doi.org/10.1007/BF02579167
17. Zhou H "Randomness and noise in information systems," Ph.D. Thesis (Pasadena, CA: California Institute of Technology, 2013) Defended June 1, 2012, https://doi.org/10.7907/82KV-2H11
18. Molotkov S N *Laser Phys.* **32** 055202 (2022)
19. Ore Ø *Theory of Graphs* (American Mathematical Society. Colloquium Publ., Vol. 38) (Providence, RI: American Mathematical Society, 1962); Translated into Russian: *Teoriya Grafov* (Moscow: Nauka, 1980)