THEORY PROBAB. APPL.
Vol. 32, No. 3

Translated from Russian Journal

ALGORITHMS AND RANDOMNESS*

A. N. KOLMOGOROV AND V. A. USPENSKII

(Translated by Bernard Seckler)

CONTENTS

Introduction. This paper presents the contents of the like-titled report given on September 8, 1986, to the First All-World Congress of the Bernoulli Society of Mathematical Statistics and Probability Theory held in Tashkent. Although algorithms and randomness may appear to be diametrically opposed and even incompatible subjects, this is only so at first glance. By scrutinizing our ideas on randomness and algorithms more closely, we are able to observe a connection between them.

However paradoxical this may seem, all of today's known mathematical definitions of randomness can be formulated in terms of the theory of algorithms. At the same time, when algorithms are constructed, randomization can be applied in a fruitful way. The article therefore concerns two things. The first is the use of algorithms to define randomness. The second is the use of randomness to develop algorithms.

Comment of the second author. Although the basic contents of the article (and especially the first two chapters) rest on the ideas and publications of A. N. Kolmogorov (who was the second author's immediate teacher), the first author did not have the opportunity to familiarize himself with the final version of the text. Therefore, the second author is responsible for any of its possible flaws as well as for the use of the name Kolmogorov in such expressions as "Kolmogorov suggested," "Kolmogorov's ideas," "Kolmogorov's theorem," "Kolmogorov stochastic property," and so on. The

* Received by the editors April 17, 1987.

notions and terms "typical sequence" and "chaotic sequence" were approved by Kolmogorov and the term "stochastic sequence" (as applied to sequences having the property of frequency stability in admissible subsequences) is due to him.

A great amount of help in the preparation of the Tashkent report was rendered by R. Freivald and Aleksandr Shen'. It is the authors' obligation and pleasure to express our gratitude to them.

1. Algorithmic Definition of Randomness: Infinite Case.

1.1. From finite chains to infinite sequences. If someone were to say to us that he had flipped a fair coin twenty times and, denoting heads by one and tails by zero, had obtained a result such as

(I) 10001011101111010000

or such as

(II) 01111011001101110001,

we would hardly be surprised. However if we were told that the result of the flips was

(III) 00000000000000000000

(twenty zeros), then we would be startled or we would in general not trust the properness of the experiment and we would even have doubts about it. The question arises, why? An exhaustive clarification would belong, we should think, to the domain of psychology and therefore lies beyond the scope of our presentation.

Apparently, the chains (I) and (II) are perceived to be random while the chain (III) is regarded as nonrandom. Here and elsewhere we have the Bernoulli distribution with equally likely outcomes in mind.

But what do the words "perceived to be random" mean? Classical probability theory does not give an answer to this important question. Quite often one hears the following explanation: the probability of the chain (III) is too small being equal to 2^{-20}. But the fact is that the chains (I) and (II) have the exact same probability. This is a trap into which even prominent specialists happen to fall.

A proper answer may be obtained on the basis of algorithmic notions. Let us repeat the question whose answer we are seeking. Under conditions of a uniform Bernoulli distribution, consider finite chains of outcomes that are equally likely and independent—in other words, finite chains of equally likely and independent binary digits. Can we distinguish random chains from nonrandom ones? Of course, the question itself is reasonable only for chains that are not too short.

Thus, our main objective is to define the notion of randomness for sufficiently long finite chains. It proves fairly difficult however to solve this problem directly. Therefore as so often happens, we shall exchange our stated goal with another one, but close to our original goal. Namely, we shall seek a definition of randomness for infinite sequences. Stated otherwise, we shall go from finite chains (I), (II) and (III) to infinite continuations of them (I...), (II...), and (III...):

(I...) 10001011101111010000...

(II...) 01111011001101110001...

(III...) 00000000000000000000....

Here (I...) and (II...) are arbitrary infinite binary sequences (formed by flipping a coin at random) whose beginning segments are the respective chains (I) and (II). And (III...) is an infinite sequence of zeros.

One could say that we are considering infinity to be an upper approximation to the finite. We are counting on finding a solution to our problem (to define randomness) for the infinite case in order to use it as an approximate solution for the finite case.

It is highly nontrivial that such a solution can indeed be found and that we have the right to say "random sequence" not only as a euphemism for something having probability one but also in a perfectly strict sense. In other words, it turns out to be possible to define randomness for an individually chosen infinite sequence. Thus the question whether an individually chosen sequence is random or nonrandom becomes perfectly sensible. By the same token, the set R of all random infinite sequences is not vague but is clearly delineated. In fact, as we shall see, it is possible to give a precise definition of the set R in algorithmic terms. This remarkable fact occupies a central position in algorithmic probability theory.

We introduce the following notation: Ξ for the set of all binary words (i.e., binary chains consisting of the digits 0 and 1); $|x|$ for the length of a chain $x \in \Xi$ (not to be confused with the notation $|A|$ for the cardinality of a set A); and Ω for the set of all infinite sequences of the digits 0 and 1.

When saying "sequence" we shall have in mind an infinite sequence and when saying "chain" a finite chain.

Thus we go from a consideration of the set Ξ to the set Ω. On the set Ω there is defined a uniform Bernoulli distribution or, if convenient, a uniform Bernoulli measure μ such that $\mu(\Omega)=1$. We assume that Ω contains the subset R of all sequences that are random in some informal reasonable sense. Our immediate aim is to define such an R in a perfectly precise way. But to this end, we should learn something about random sequences.

1.2. Three properties of a random sequence. Random sequences possess three fundamental properties. Each of these may be taken as the basis of a definition of randomness. We shall first analyze the three properties on an intuitive level and then give precise formulations.

The first property is that of *being typical.* This property was pointed out by Martin-Löf. Every random sequence is typical. Thus the sequences (I...) and (II...) are perceived to be typical whereas the sequence (III...) is perceived on the contrary to be very special. Practically speaking, the property of being typical is the property of belonging to any reasonable majority. In choosing some object at random, we have confidence in the fact that this object will fall precisely in such a majority. In particular, on choosing a sequence at random, we have the justifiable expectation of obtaining a typical sequence. Therefore, $R \subset T$, where T is the class of all typical sequence in Ω.

The second property is that of *being chaotic.* This property was pointed out by Kolmogorov. Each random sequence is chaotic in the sense that it has no simple law governing the alternation of its terms. For instance, the sequences (I...) and (II...) are chaotic while the sequence (III...) is not chaotic. Thus, $R \subset C$, where C is the class of all chaotic sequences in Ω.

Finally, the third property is embodied in the *stability of frequencies.* This property was pointed out by von Mises. The frequency of zeros in the beginning segments of a random sequence (if the occurrences of 0 and 1 are equally likely) must converge to $\frac{1}{2}$. Moreover, this effect must be observed not only for the entire sequence as a whole but also for any of its properly chosen subsequences (we underscore the phrase properly chosen). Let us agree to refer to sequences that possess the designated property as *stochastic* sequences. The fact that a sequence is stochastic implies, for example, that if we should remove every third term of the sequence (or even any term whose position in the sequence is a prime number), then the frequency of zeros in the beginning segments of a subsequence formed in this way will tend to $\frac{1}{2}$. The property of being stochastic is satisfied by (I...) and (II...) but not by (III...). Thus, $R \subset S$, where S is the class of all stochastic sequences in Ω.

In attributing the notions of being typical, chaotic and stochastic to Martin-Löf, Kolmogorov, and von Mises, we were not trying to say that these authors used the terms "typical," "chaotic" and "stochastic," Martin-Löf and Kolmogorov used the word "random" while von Mises used the German word "Kollektiv," The terms "typical," "chaotic" and "stochastic" were actually used for the first time in the sense just described in front of the large audience at the Bernoulli Congress in Tashkent on September 8, 1986.

We proceed now to give precise mathematical definitions of the three properties of randomness that were just considered "on a philosophical level," i.e., precise definitions of the classes T, C and S. In many of the technical details, we shall follow the presentation in [30], [31]. And we shall assume as before that a uniform Bernoulli measure μ has been defined on Ω.

1.3. Typical sequences: definition. We intend to give a precise meaning to the assertion that there are many typical sequences and that each typical sequence belongs to a reasonable majority. At first glance, it would seem that the mathematical equivalent of reasonable majority should be the notion of a set of measure 1. Then the class T of all typical sequences should be taken to be the intersection of all subsets of Ω having measure 1. Unfortunately, such an intersection is empty. Our more precise definition of reasonable majority has turned out to be improper. A certain natural algorithmic analogue of the notion of a set of measure 1 will serve as a more precise proper definition. The formulation of such an analogue is due to Martin-Löf [8].

Thus following [8], we shall modify the very definition of a set of measure 1 in an algorithmic direction. Since each set of measure 1 is the complement of a suitable set of measure 0, it suffices to state the notion of a set of measure 0 in algorithmic terms. This we now proceed to do.

Each finite chain $x \in \Xi$ generates a ball $\Gamma_x \subset \Omega$ consisting of all of the possible infinite continuations of x so that $\mu(\Gamma_x) = 2^{-|x|}$. It is well known that a set $M \subset \Omega$ has measure 0 or is *negligible* if and only if to each positive rational number ε there is a sequence of chains $x_0, x_1, \cdots, x_k \in \Xi$, $k = 0, 1, \cdots$, such that the corresponding sequence of balls satisfies the conditions

(i) $\bigcup \Gamma_{x_k} \supset M$,

(ii) $\sum \mu(\Gamma_{x_k}) < \varepsilon$.

We now confine ourselves to just computable sequences of chains. A sequence $x_0, x_1, \cdots$ is said to be *computable* if there exists an algorithm for computing x_k when k is specified. For a given sequence, many such algorithms are possible and each of these algorithms, in turn, may be realized through different programs offering their own *texts*. Taking $\langle x_k \rangle$ to be an arbitrary fixed computable sequence, we shall refer to any program of any algorithm computing x_k with respect to k as ***the program*** of the sequence $\langle x_k \rangle$.

Suppose that there is an algorithm which provides a program fnr a computable sequence $\langle x_k \rangle$ satisfying conditions (i) and (ii) for any positive rational number ε. In that event, we shall say that the set M ***effectively has measure*** 0 or has ***effective measure*** 0 or is ***effectively negligible*** and we shall write $\mu(M) =_{\text{eff}} 0$. We then declare that a set A ***effectively has measure*** 1 or has ***effective measure*** 1, which will be written $\mu(A) =_{\text{eff}} 1$, whenever the complement of A has effective measure 0, i.e., $\mu(\Omega \backslash A) =_{\text{eff}} 0$.

And now in our first—and unsuccessful—attempt to define T, it is necessary to replace measure 1 by effective measure 1. That this will be successful is assured by Martin-Löf's celebrated theorem of 1966 which was formulated initially in [9, p. 610] in terms of so-called universal tests (a formulation in terms of measure may be found in [14, § 35]). Martin-Löf's theorem says: *the intersection of all sets of effective measure* 1 *is not only nonempty but it also has effective measure* 1. Thus, this intersection is the smallest set of effective measure 1. It is called the *constructive support* of the measure [9, p. 614]. We now proclaim this set to be the class T of all typical sequences.

Terminological comment. Sequences belonging to the constructive support of a measure are often referred to as *Martin-Löf random sequences*; see, for example, [30], [31].

The property of being typical is encountered in many theorems of probability theory. In fact, take any theorem stating that some property, say the law of the iterated logarithm, is satisfied by "almost all infinite sequences". The exact meaning of the phrase in quotation marks is that the set of all sequences for which the property in question holds has measure 1. An analysis of the proof of any such theorem shows that the set of all sequences possessing the property not only has measure 1 but it also has this measure effectively. Consequently, every typical sequence possesses the considered property. In particular, the law of the iterated logarithm holds for an arbitrary typical sequence. (Similar considerations caused Martin-Löf to identify randomness with being typical [8], i.e., to declare that $R = T$. This was the first mathematically precise and simultaneously adequate definition of randomness.)

1.4. Chaotic sequences: equivalent definitions. Being chaotic signifies a complexity of structure. Let K be a measure of the complexity of finite binary chains so that for any $y \in \Xi$ the value of $K(y)$ is a natural number. Thus, $K : \Xi \to \mathrm{N}$. What K represents exactly will be determined subsequently. At this point, the general notion that the natural number $K(y)$ reflects some informal idea of the complexity of a binary chain is sufficient for us. Then the fact that a sequence is chaotic $a_0, a_1, \cdots$ signifies that the complexity of its beginning segments $K(a_0)$, $K(a_0, a_1), \cdots$ grows sufficiently fast.

It is reasonable to interpret the phrase "sufficiently fast" in the sense of "the fastest possible rate." However, it is possible to specify each binary chain of length n by using up at most n bits of information. Therefore, in any natural measure of complexity, the complexity of an n-term binary chain cannot be of order greater than n. These remarks justify the following formula as a mathematical expression of the fact that the complexity of the beginning segments of a sequence $a_0, a_1, \cdots$ grow at the fastest possible rate:

$$K(a_0, a_1, \cdots, a_{n-1}) \geqq n - c$$

for all n and some constant c not depending on n but depending on the entire sequence $\langle a_s \rangle$ (cf. [34, § 6]). It remains to clarify what such a K is.

We start out with the following simple considerations. There are objects y and descriptions of them encoded in the form of binary chains belonging to Ξ. Each object has many descriptions in general. If among these descriptions there is at least one short one, then the object under consideration is simple. But if all descriptions of this object are long, then this object is complex. Thus, complexity of an object is nothing more than the length of its shortest description.

However a straightforward development of these simple and natural ideas meets up with serious difficulties associated with the familiar Richard-Berry paradox. The gist of this paradox is manifested in the expression "the smallest natural number that cannot be described using fewer than fifty words"—but this expression contains less than fifty words and as such it does describe some natural number. Therefore, one has to be careful.

Thus, let Y be a fixed set of objects. Their descriptions belong to Ξ. Each object may have many descriptions and each description can be the description of many objects. Thus, in the general case we have a set E consisting of all possible pairs $\langle x, y\rangle$ in which x is a description of y. This E is a subset of the cartesian product $\Xi \times Y$ and is said to be a process describing elements of Y by means of chains in Ξ. It is meaningful to interpret an *arbitrary* set $E \subset \Xi \times Y$ as a process describing elements in Y by means of chains in Ξ. Whenever E is a descriptive process and $\langle x, y\rangle \in E$, we shall call x a description of the object y by or relative to the process E.

The complexity $K_E(y)$ of an object y relative to a descriptive process E is by definition the smallest length of the description of this y by E, i.e.,

$$K_E(y) = \min (|x| \,|\, \langle x, y\rangle \in E).$$

As is usual in such cases, if y has no description at all in E, then we put $K_E(y) = \infty$.

Let Y be fixed and let $\mathfrak{A}$ be a family of processes describing the objects in Y. For certain important families $\mathfrak{A}$ associated with algorithms, we have Kolmogorov's theorem of 1965: $\mathfrak{A}$ contains an optimum descriptive process which in a certain sense furnishes the shortest possible descriptions. More precisely [7]: there exists an $A \in \mathfrak{A}$ such that, for every $E \in \mathfrak{A}$,

$$K_A(y) \leqq K_E(y) + c_E$$

with c_E a constant not depending on y.

The complexity of an object y relative to an optimum descriptive process is defined to be the *entropy* of y. (One should not forget that we have a fixed specific family $\mathfrak{A}$.) Any two entropies (corresponding to two optimum descriptive processes) can differ at most by an additive constant. More precisely, if A' and A'' are two optimum descriptive processes, then $|K_{A'}(y) - K_{A''}(y)| \leqq c$, where c is independent of y. Therefore, it is possible to say simply "the entropy of object y" without indicating a specific optimum descriptive process. We then understand that the entropy is determined to within at least an additive constant.

The entropy of an object y is usually denoted by $K(y)$.

Starting from here, we shall confine ourselves to the case where the set of objects being considered—our previous Y—is Ξ. For this case, we now declare a binary sequence $a_0, a_1, \cdots$ to be by definition *chaotic* if and only if

$$K(a_0, a_1, \ldots, a_{n-1}) \geqq n - c,$$

for some constant c and all n, where K is the entropy. This definition depends of course on $\mathfrak{A}$.

It remains to choose a suitable family $\mathfrak{A}$. In order to accomplish this, we shall describe a certain $\mathfrak{A}$ which leads to the concept of entropy introduced by Levin in 1973 [16]. This entropy was called monotone complexity by the author. We shall call it *monotone entropy.*

We shall say that a descriptive process E *preserves comparability* whenever it possesses the following property:

$$\begin{gathered} \langle x_1, y_1\rangle \in E \;\&\; \langle x_2, y_2\rangle \in E \\ \&\; (x_2 \text{ is a continuation of } x_1) \\ \Rightarrow (y_2 \text{ is a continuation of } y_1) \\ \text{or } (y_1 \text{ is a continuation of } y_2). \end{gathered}$$

We shall confine ourselves to *countable* descriptive processes. It is precisely here that algorithms will make their appearance because a nonempty set is by definition countable if it is in the range of values of a computable sequence. We take $\mathfrak{A}$ to be the family of all descriptive processes that simultaneously preserve comparability and are countable. The entropy L corresponding to such a family A is called *monotone entropy.* It has to be used when defining the property of being chaotic. We now proceed to give the definitive definition of the class C of all chaotic sequences: by definition,

$$\langle a_0, a_1 \cdots \rangle \in C$$

if and only if, for all n,

$$L(a_0, a_1, \cdots, a_{n-1}) \geqq n - c,$$

where c is independent of n.

Remark 1. An equivalent definition of monotone entropy was suggested by Schnorr in 1977 [22, § 4]. (A transparent proof of the fact that Levin's formulation and Schnorr's formulation lead to the exact same entropy may be found in [29, pp. 34-35]. We point out that applied to any two entropies K_1 and K_2, the phrase "the same" has the following meaning: $|K_1(y) - K_2(y)| \leqq c$ for some c not depending on y.) Earlier, in 1973, Schnorr [17] had proposed another notion of entropy (which he subsequently discarded), which he called *process complexity.* Process complexity leads to the same class C (as does monotone entropy); this is a trivial consequence of the theorems of Levin and Schnorr, which we shall discuss in § 1.5. However, process complexity does not coincide with monotone entropy even up to an additive constant; the difference of these two entropies is unbounded (as V'yugin showed [29, p. 35]).

Remark 2. The concept of entropy as a measure of the complexity of a finite object can be formulated in another way without using the "object-description" relationship. We are thinking here first of all of the approach relying on various versions of the so-called a priori probability of a binary chain. One such version is given in [15, § 3.3]. In [16, Theorem 3], it was established that the entropy that comes up on the basis of this version leads to the same class C (as does monotone entropy). Another version of a priori probability will be discussed below in § 4.1. This version also leads to the very same class C.

Remark 3. It would be false to assume however that any conceivable (or even any reasonable) measure of complexity of a binary chain is fit to define the class C with the help of an inequality such as $K(a_0, \cdots, a_{n-1}) \geqq n - c$. For instance, as a quite natural measure of complexity of a chain y one could take the value of $H(y|\varnothing)$ or of $H(y||y|)$, where $G(y|x)$ is the conditional entropy described in § 2.2 below, $\varnothing$ is the empty chain and $|y|$ is the length of a chain y. But then there will be the following effect: whatever the sequence $a_0, a_1, \cdots$ in Ω may be, neither the difference $n - H(a_0, \cdots, a_{n-1}|\varnothing)$ nor the difference $n - H(a_0, \cdots, a_{n-1}|n)$ is bounded. The unboundedness of the second difference is an easy consequence of the unboundedness of the first. And the unboundedness of the first difference was revealed by Martin-Löf [8] (a proof may be found, for example, in [13, Thm. 2.2]).

A useful comparative presentation of the various versions of entropy has been published by V'yugin [29].

Terminological comment. The sequences belonging to the class C defined in this section are referred to as *Kolmogorov random sequences* in the review articles [30], [31].

1.5. Random sequences: definition. Thus, we have two classes of binary sequences (both of them now being defined precisely): the class T of typical sequences and the class C of chaotic sequences. In [16], Levin established that C coincides with T. Schnorr also proved this coincidence in [17]. Each of them started from his own definition of C, which results, respectively, from Levin's monotone entropy or Schnorr's process complexity. Thus, there are actually two different theorems. Nevertheless, we find it natural to refer to the assertion that the classes T and C coincide as the Levin-Schnorr theorem.

It was already mentioned in § 1.3 that each typical sequence obeys the law of the iterated logarithm, and to prove this, it suffices to repeat the standard proof of the theorem on the law of the iterated logarithm. Thus, every typical sequence obeys this law. It is noteworthy that this last fact can be proved-directly. Such a proof was devised by Vovk [41]. In other words, he created a proof of the theorem on the law of the iterated logarithm by means of algorithmic entropy. One gets the impression that this proof penetrates the essence of randomness more deeply than the standard textbook proofs. Thus the approach being developed permits one not only to translate traditional probability-theoretic facts into the language of algorithmic concepts, on the contrary, these concepts can be of use in perfectly classical parts of probability theory.

The Levin-Schnorr theorem, which says that the classes T and C coincide, furnishes valid grounds for proclaiming C or T to be the class of all genuinely random sequences. Thus, we can identify our earlier slightly vague R with this class $T = C$. Henceforth the class R will be treated as being defined rigorously.

1.6. Stochastic sequences: attempts at a definition. One says that a binary sequence $a_0, a_1, a_2, \cdots$ possesses the *property of frequency stability* with limit p if $\lim (\nu_n/n) = p$, $n \to \infty$, where ν_n is the number of zeros in a beginning segment $a_0, a_1, \cdots, a_{n-1}$ of length n. To simplify the presentation in what follows, we shall treat just the case $p = \frac{1}{2}$ and hence the phrase "with limit p" will be omitted.

By definition, a binary sequence is *stochastic* whenever any suitably chosen subsequence of it has frequency stability. The main thing here is to define the phrase "suitably chosen".

We assume that each subsequence is formed from the original sequence by making a selection of its terms. It is therefore assumed that a certain rule exists which accomplishes this selection and extracts certain terms of the original sequence $a_a, a_1, \cdots$ so as to form from them a subsequence $a_{\gamma_0}, a_{\gamma_1}, \cdots$. If the selection rule is admissible, then a subsequence $\langle a_{\gamma_i} \rangle$ formed in this way will have the property of frequency stability.

The notion of admissible selection rule should not depend of course on the particular sequence to which the selection rule is applied; it should be the same for all conceivable sequences. It is thus assumed that there is a family of admissible selection rules. A sequence is said to be stochastic (relative to the given family!) if each infinite subsequence of it constructed with the help of one of the admissible selection rules possesses frequency stability.

But what selection rules are admissible? Here is a trivial example of an inadmissible selection rule: extract exactly those terms of an original sequence that are zero. This rule is inadmissible because the decision as to whether or not to pick out the term a_s of the sequence $a_0, a_1, \cdots$ should depend of course just on the earlier observed values and not on the value of a_s itself.

The above well-known approach to randomness goes back to the classical papers of von Mises of the first third of the century [1], [2]. But von Mises did not give and could not give a precise definition of an admissible selection rule. Indeed, such a definition requires algorithmic ideas.

The first attempt to present a precise definition of admissible selection rule dates back to 1940 and is due to Church, one of the founders of the modern theory of computability [3]. He required the existence of an algorithm that could determine whether or not to select the term a_s depending on the values of the preceding terms $a_0, a_1, \cdots, a_{s-1}$ of a considered sequence. Thus the domain of this algorithm is the set of all finite binary chains and the range is the two-element set {Yes, No}. The algorithm operates as follows. Let the sequence being considered be $a_0, a_1, \cdots$ and suppose that its beginning segment $a_0, a_1, \cdots, a_{s-1}$ has been admitted as the input to the algorithm. Then if "Yes" results at the output, the term a_s must be chosen for inclusion in the subsequence being generated; but if "No" is the output, then a_s is passed over. The computable function $\varphi: \Xi \to \{\text{Yes, No}\}$, defined by this algorithm, is uniquely determined by the set $\{x|\varphi(x)=\text{"Yes"}\}$. Therefore, in order to specify a Church admissible selection rule, it suffices to designate some decidable set $D \subset \Xi$ and then to put $\varphi(x)=$ "Yes" if $x \in D$ and $\varphi(x)=$ "No" if $x \in \Xi \backslash D$. (It will be recalled that in algorithmic theory a subset $D \subset X$ is said to be *decidable* whenever there exists an algorithm which answers the question: "$x \in D$?" for each $x \in X$.)

It is essential that any new sequence $a_{\gamma_0}, a_{\gamma_1}, \cdots$ formed from $a_0, a_1, \cdots$ by means of some Church admissible rule be a *strict* subsequence of $a_0, a_1, \cdots$. The word "strict" means that the terms of the subsequence should proceed one after the other in the same order as in the "larger sequence," i.e.,

(1) $$\gamma_0 < \gamma_1 < \gamma_2 < \cdots.$$

The notion of a Church admissible selection rule gives rise in a natural way to the notion of a Church stochastic sequence. By definition, a sequence is *Church stochastic* if and only if to each Church admissible selection rule, the subsequence formed from the original one by applying this rule possesses the property of frequency stability providing it is infinite.

Unfortunately, the class CS of Church stochastic sequences turns out to be too wide. In particular, it is possible for a sequence to be Church stochastic yet not satisfy the law of the iterated logarithm (see, for example, [13]). Thus $R \neq CS$ (although of course $R \subset CS$).

The preceding discussion shows the necessity of creating a new and more general definition of an admissible selection rule. Such a definition was proposed by Kolmogorov in 1963 in Remark 2 of the article [5]. We shall present this definition in the words of the report [35] (keep in mind that the original sequence to which the rule is applied is $x_1, x_2, \cdots$):

"According to [5], a selection rule is specified by means of an algorithm (or if convenient by means of a Turing machine). The choice of the next term in the subsequence is accomplished in this way. The input information consists of a finite set of numbers $n_1, n_2, \cdots, n_k$ and terms $x_{n_1}, x_{n_2}, \cdots, x_{n_k}$ of the original sequence. The output of the algorithm is composed of two parts: first, the number n_{k+1} of the next term subject to investigation (this number must not be the same as any of the $n_1, \cdots, n_k$; as to the order in which the $n_1, \cdots, n_k$ proceed, no restrictions are imposed here); second, designating whether $x_{n_{k+1}}$ is selected just for investigation or else the algorithm is to include this term in the subsequence.

"At the next step of operation of the algorithm, its input now consists of a larger collection of numbers $n_1, \cdots, n_{k+1}$ and the values $x_{n_1}, x_{n_2}, \cdots, x_{n_{k+1}}$. The algorithm begins its operation with the empty set.

"As compared to [3], our extension consists in the fact that the order of succession of the terms in the subsequence chosen is not necessarily the same as their order in the original sequence."

Thus the main feature of Kolmogorov's definition is that the requirement (1) has been discarded and the terms of the sequence are allowed to proceed in a new order.

A more explicit formulation of Kolmogorov's definition will be given below. This definition of a Kolmogorov admissible selection rule leads to a narrower class KS of *Kolmogorov stochastic* sequences such that

$$R \subset KS \subset CS, \qquad KS \neq CS.$$

It is easy to see that any sequence obtained from a Kolmogorov stochastic sequence by a computable permutation of its terms is itself a Kolmogorov stochastic sequence. Church stochastic sequences do not have this important property of randomness: Loveland constructed an example of a Church stochastic sequence in [10] which ceases to be Church stochastic after a certain computable permutation of its terms, which shows in particular that $KS \neq CS$.

We do not have an example available violating any of the laws of probability theory by any Kolmogorov stochastic sequence. Nevertheless, it is still unknown whether KS coincides with R. It is not even known whether any subsequence formed by applying a Kolmogorov admissible rule to a Kolmogorov stochastic sequence is itself a Kolmogorov stochastic. (Each notion of admissible selection rule leads to a corresponding notion of being stochastic. It is reasonable to require that each sequence which is obtained from a stochastic sequence by means of an admissible selection rule be itself stochastic. Church's definition satisfies this requirement. As was just mentioned, the question as to whether Kolmogorov's definition satisfies this requirement remains an open one.)

Moreover, it has been hypothesized that $KS \neq R$. This hypothesis seems all the more plausible in that the inequality $KS \neq R$ follows from an assertion made by Kolmogorov in 1969; however the proof of that assertion has been lost. Thus the question "Is $KS = R$ true or false?" must be regarded as an open one. (The assertion that we are discussing is contained in the concluding paragraphs of Section 2 of [12]. It says that there exists a Kolmogorov stochastic sequence x satisfying $H(x^l) = O(\log l)$. Here x^l is the beginning segment of length l of the sequence x and $H(w) = H(w|\varnothing)$, where $H(w|u)$ will be defined below in § 2.2 and $\varnothing$ is the empty chain.)

We now formulate Kolmogorov's notion of selection rule (and thereby the notion of being Kolmogorov stochastic) in a clearer form.

A *Kolmogorov admissible selection rule* is specified by two computable functions f and g from Ξ to N. It is not assumed that these functions are necessarily defined on all of Ξ. To apply the rule to a sequence $a_0, a_1, \cdots = a(0), a(1), \cdots$ we first construct a sequence of numbers k using the recursion formula

$$k(n) = f(a(k(0)), a(k(1)), \cdots, a(k(n-1))),$$

which we apply until all the numbers $k(0), k(1), \cdots, k(n)$ are defined and distinct from one another. As soon as a first n arises such that $k(n)$ is either undefined or coincides with $k(s)$ for some $s < n$ (providing such an n exists at all), the formation of the sequence k is terminated; in that event, k is a finite chain of length n. We are interested however only in the case where k is an infinite sequence. At the next step, certain terms of the sequence k must be removed while the order of succession of remaining terms remains unchanged. Namely, we leave the term $k(m)$ in the sequence if and only if $g(a(k(0)), \cdots, a(k(l-1)))$ is defined for all $l \leqq m$ and $g(a(k(0)), \cdots, a(k(m-1))) = 0$. The sequence k' obtained in this way is by definition the result of applying the Kolmogorov admissible selection rule $K_{f,g}$ to the sequence $a_0, a_1, \cdots = a(0), a(1), \cdots$.

A sequence $a = a_0, a_1, \cdots$ is called *Kolmogorov stochastic* whenever for any computable functions f and g for which the sequence b formed from a by applying the rule $K_{f,g}$ is infinite, this b possesses frequency stability.

As already stated, this definition appeared for the first time in Kolmogorov's article of 1963 [5, Remark 2]. In 1966, it was rediscovered by Loveland [11]. Some useful observations were made by Shen' in [32].

Terminological comment. The review articles [30] and [31] refer to Church stochastic sequences as "Mises-Church random sequences" and to Kolmogorov stochastic sequences as "Mises-Kolmogorov-Loveland random sequences."

If the hypothesis $KS \neq R$ is valid, then the following problem arises. Does there exist a "good" definition of admissible selection rule for which the class of stochastic sequences (corresponding to this definition) coincides with R? The word "good" means that the notion to be defined should be sufficiently general while remaining reasonable. (A step in the direction of discovering such a notion was made by Shen' in [32]. Some natural requirements that such a notion must satisfy were discussed above in this section.) From a philosophical standpoint, the question amounts to whether von Mises' ideas can be reduced to a proper conception of randomness.

1.7. Computable distributions and other generalizations. Until now, we have spoken only about uniform Bernoulli distributions. Mutatis mutandis, our considerations can be generalized: for stochastic sequences—to an arbitrary Bernoulli distribution on Ω (with probability of occurrence of zero not necessarily equal to $\frac{1}{2}$); for typical and chaotic sequences—to arbitrary probability distributions on Ω.

A measure μ on Ω (in particular, a probability distribution) is said to be *computable* if there exists a computable function h such that, for each $x \in \Xi$ and each positive rational number ε, its value $h(x, \varepsilon)$ is a rational ε-approximation to the real number $\mu(\Gamma_x)$:

$$h(x, \varepsilon) \in \mathbf{Q} \;\&\; |h(x, \varepsilon) - \mu(\Gamma_x)| < \varepsilon.$$

In particular, any Bernoulli distribution with a computable real p as the probability of zero is a special case of a computable distribution. (In fact, a number p is said to be computable whenever there exists a computable function furnishing a rational ε-approximation to p for each positive rational ε.)

If μ is any computable distribution on Ω, then it is possible to construct the classes T and C and to prove that $T = C$. The class T is defined as before to be the constructive support of the measure, i.e., the smallest subset of Ω having effective measure 1. In order to define the class C, we have to replace the inequality $K(a_0, a_1, \cdots, a_{n-1}) \geqq n - c$ of § 1.4 by the inequality

$$K(a_0, a_1, \cdots, a_{n-1}) \geqq \log_2 \mu(\Gamma_{a_0, a_1, \cdots, a_{n-1}}) - c.$$

We now have at our disposal the definition of randomness for the case of an arbitrary computable distribution μ on Ω: a binary sequence is said to be *random relative to* μ whenever it belongs to the class $T = C$, where T and C have been determined for this μ.

It is possible to consider the *conditional probability* $\pi(x; P)$, P any distribution on Ω, that an arbitrary unspecified sequence $\omega_0, \omega_1, \cdots$ will have a one occurring after a specified beginning segment $x \in \Xi$:

$$\pi(x; P) = \mathbf{P}(\omega_{n+1} = 1 | \omega_0, \omega_1, \cdots, \omega_n = x) = \frac{P(\Gamma_{x_1})}{P(\Gamma_x)}.$$

Of course, the value of $\pi(x; P)$ will depend essentially on P so that the difference of $\pi(x; P')$ and $\pi(x; P'')$ for two distributions P' and P'' may be appreciable. This difference proves to be small if P' and P'' are both computable and x is the length of the beginning segment of a pre-assigned sequence which is random relative to P' and

P'' simultaneously. This fact was revealed by Vovk [37]. Here is a precise statement of Vovk's theorem: Let P' and P'' be arbitrary computable distributions on Ω and let $a_0, a_1, \cdots$ be a random sequence relative to P' and P'' simultaneously. Then

$$\pi(a_0, a_1, \cdots, a_n; P') - \pi(a_0, a_1, \cdots, a_n; P'') \to 0$$

as $n \to \infty$. Thus $\pi(x; P')$ and $\pi(x; P'')$ are close to one another whenever the chain $x = a_0, a_1, \cdots, a_n$ is long and we are entitled to speak about the conditional probability of occurrence of a one after a specified beginning segment of zeros and ones—the entire sequence viewed as a whole is random relative to some distribution (unknown to us!).

Until now, we have spoken only about the randomness of binary sequences. However the above algorithmic approach can be applied also to more general situations. For example, Asarin [38], [39] developed a definition of typical sequences in the style of Martin-Löf as it applies to Wiener random processes and gave a definition of a typical trajectory for a Brownian motion. He also suggested a definition of a chaotic trajectory in terms of entropy in the style of Kolmogorov and he proved that both definitions actually determine the same class of trajectories.

2. Algorithmic Definition of Randomness: Finite Case.

2.1. Introduction. It seems natural to call a chain random if it cannot be written down in a more condensed form, i.e., if the shortest program for generating it is as long as the chain itself. But in order to embody this in a strict mathematical definition, we have to expend a little effort.

The question "what chains of zeros and ones of length n are random?" is clearly irrelevant for $n = 2$ and is meaningful only for n sufficiently large. Moreover, even for very large n, there is no clearcut boundary between random and nonrandom chains of length n. In fact, if we take a "random" chain of very large length n and replace successively the ones by zeros digit after digit, then we arrive in the final analysis at a "nonrandom" chain of the same length of only zeros. But at no stage of the process does the incipient formation of a "nonrandom" chain out of a "random" one manifest itself. (This is one of the manifestations of the pile paradox.)

Thus in contrast to the infinite case, we cannot split the set of all chains of length n into a subset of all "random" chains and a subset of all "nonrandom" chains. The correct question is not "Is a given chain random?" but rather "By how much is a given chain random?". It is expedient to define also the degree of randomness of a finite chain not in an absolute sense but relative to some finite set M containing the chain. (For comparison sake, it may be observed that randomness in an infinite chain was defined in the previous chapter relative to all of Ω with a Bernoulli distribution defined on Ω.)

These considerations led the first author to introduce the notion of "defect of randomness of an element y relative to a finite set M assuming that $y \in M$". The larger is this defect, the less random is y as an element of M. As a function of y and M, defect of randomness is determined only up to at most an additive constant. The role of M here is similar to the role of the probability distribution on Ω in the infinite case. It is right to say that the defect of y relative to M is the defect of y relative to the uniform distribution on M, where all the elements of M are regarded as equally likely with probability $1/|M|$ ($|M|$ is the cardinality of M).

2.2. Conditional complexity and conditional entropy. In this section, some subsidiary notions will be explained. We shall need the notions of conditional complexity and conditional entropy of an object given that an object x has already been specified or is known [12]. If we resort to informal language, then conditional complexity is the length of the shortest description of an algorithm that transforms x into y (under the assumption that some method of describing algorithms has been given). But if we talk rigorously, then *conditional* complexity of y given x is the length of the shortest program of a computable function whose value at x equals y:

$$H_F(y|x) = \min(|p|\,|f_p(x) = y).$$

Here x, y, $p \in \Xi$, f_p is a computable function with program p, $|p|$ is the length of p and F is subject to clarification.

The notation $f_p(x)$ indicates that we actually have a function of two variables p and x:

$$F(p, x) \simeq f_p(x)$$

(the symbol $\simeq$ between two expressions means that the values of both expressions are simultaneously defined or undefined and if defined they coincide). The function F is computable and can be treated as a *programming method.* Thus, the definition given above rests on the method F and this fact is expressed by means of the subscript F: the expression $H_F(y|x)$ is the conditional complexity of an object y relative to the method F given that x is known.

Among the programming methods there are the so-called *optimum* methods. A programming method Φ is said to be *optimum* if, for any other method F,

$$H_\Phi(y|x) \leqq H_F(y|x) + c_F,$$

where c_F is a constant not depending on either x or y.

If Φ is an optimum method, then $H_\Phi(y|x)$ is called the *conditional entropy* of the object y given x. When speaking about entropy, we may omit the subscript Φ since the entropies corresponding to any two optimum Φ differ by at most an additive constant. Thus we can simply write $H(y|x)$ for the conditional entropy of y given x. This conditional entropy estimates the length of the shortest description of object y that can be achieved with the help of x.

The preceding developments are sufficiently clear if x and y are binary chains. We shall need to handle situations however in which x and y are both finite sets of such chains. For that purpose, it is necessary to secure some natural way of coding finite sets of chains by means of binary chains and then to replace the considered set by its code. (Various reasonable ways of doing the coding lead to definitions that differ by at most an additive constant.)

2.3. The defect of randomness. We are finally ready to define defect of randomness (or perhaps it is better to say: defect of being chaotic). Let M be an arbitrary set of finite binary chains and let y be an element of M. Then the *defect of randomness* of an element y relative to M is by definition

$$d(y|M) = \log_2|M| - H(y|M),$$

where $|M|$ is the cardinality of M.

We now make some comments apropos of this formula. Suppose that M has been specified. Then each $y \in M$ can be identified uniquely by its serial number in the lexicographic ordering of M. In order to label this serial number and thereby the element y itself, it suffices to expand at most $\log_2|M|$ digits. The logarithmic term in the defect formula is just the standard description of y by means of M. As for the computable $H(y|M)$, it is by definition (see § 2.2) the length of the shortest program of the algorithm transforming M into y (and moreover for the best programming method!). Any such program may be interpreted as some description of y that makes use of M. We see that the defect of randomness $d(y|M)$ estimates the difference in length between two descriptions of y by means of the set M—between the standard and shortest descriptions. This explanation shows in particular that

$$H(y|M) \leqq \log_2 |M| + c$$

and so the defect of randomness is "almost positive":

$$d(y|M) \geqq -c$$

for some positive constant c not depending on M and y.

When $d(y|M)$ is large, this means that there is a description of the element y with the help of M which is considerably shorter than the description mentioned above by means of a serial number. In that event, it is reasonable to treat y as an element of a very special form and so it is not random.

We point out that there are comparatively few elements with a large defect of randomness:

$$|\{y|d(y|M) \geqq k\}| \leqq |M|/2^{k-1},$$

since the number of all descriptions of length at most l does not exceed 2^{l+1}.

Those elements y in M that have a sufficiently small defect of randomness $d(y|M)$ are declared to be *sufficiently random* elements of M.

2.4. Δ-randomness. Let Δ be some number. It is expedient to introduce the notion of Δ-randomness of an element y relative to M. We shall say that $y \in M$ is *Δ-random* relative to M if $d(y|M) \leqq \Delta$. Then sufficiently random elements of a set M can be defined to be those that are Δ-random for Δ sufficiently small. Precisely such a definition was given in [35].

If M is taken to be the set of all binary chains of fixed length n, then Δ-randomness of a chain y will mean that $n - H(y|M) \leqq \Delta$. Thus, in this important case, those chains whose conditional entropy (i.e., "shortest descriptive length") is close to their length can be treated as random chains.

The definition of randomness just formulated can be labeled a definition in terms of being chaotic. What we have actually just defined are *chaotic chains* (more precisely, *Δ-chaotic* elements of a specified finite set of chains).

As in the infinite case, it turns out to be possible to prove that chaotic objects have the properties of being typical and stochastic.

In fact, chaotic elements of a finite set of chains M are *typical* in the following sense: no chaotic element in M can belong to any subset S of M which is simultaneously pure (which means $H(S|M)$ is small) and not large (which means that $|M|/|S|$ is large). A precise statement is that if $y \in S \subset M$ and $|v| \leqq n$ for all $v \in M$, then

$$\log_2 \frac{|M|}{|S|} - H(S|M) \leqq d(y|M) + R_n, \qquad R_n = O(\log_2 n),$$

(in this $|v|$ designates length and $|M|$ and $|S|$ cardinalities). The relation $R_n = O(\log_2 n)$ shows that the theorem is of an asymptotic nature. This feature is intrinsic to all theorems concerning randomness of finite objects. These theorems hold on the assumption that the chains are sufficiently long, the sets are sufficiently pure (i.e., they possess sufficiently small conditional entropies) and so forth.

We now proceed to *stochastic* chains. First of all, we agree to use the word "subchain" to mean any chain obtained from an original chain by the removal of some of its terms and—for the case of a *nonrigorous* subchain (decidable by the Kolmogorov selection rule)—a subsequent permutation of the remaining terms.

Bearing in mind the essentially asymptotic nature of all of our considerations, we are justified in speaking in an approximate sense about frequency stability in a finite chain—on the assumption that the chain is sufficiently long. (Similar considerations are valid for all of the current discussions.) Thus it turns out to be possible to speak about a finite chain being stochastic. Admissible selection rules were indicated for this case in [5]. They are, mutatis mutandi, rules of the same Kolmogorov type as in the infinite case (see § 1.6) and are chronologically prototypes of the latter ones. But in the finite case it is further important that a selection rule should itself be sufficiently simple. The last requirement is satisfied in particular in situations where there is a sequence of chains of unboundedly increasing lengths and the selection rule is common to all of these chains. Simplicity of a selection rule E means that when E is treated as a finite object its conditional entropy $H(E|\varnothing)$ is small. Here $\varnothing$ denotes a finite set or, according to [12], any "automatically assigned object".

As we have seen in the infinite case, the phenomenon of chaotic objects being stochastic is observed only for special distributions, namely, for Bernoulli distributions. A similar situation also prevails in the finite case. A chaotic chain and its admissible subchains actually possess the property of frequency stability but only if a considered chain is chaotic relative to some special set (containing this chain). An investigation of this topic was begun in [5]. (In that article, the author again expressed "the point of view that the basis for the applicability of the mathematical theory of probability to random phenomena of the real world is the *frequency approach to probability* in some form or another, an approach that von Mises championed vigorously as being inevitable".)

If a chain is chaotic relative to a suitable set, then it is *stochastic* in the following sense: not only does the chain possess the property of frequency stability but so does any of its admissible subchains. This feature of a chain being stochastic can be formulated as follows: the frequencies of zeros in all sufficiently long subchains obtained by means of admissible selection rules are close to one another.

Take for example the set of all binary sequences of length n. If a chain is Δ-random and Δ is sufficiently small, then as mentioned in [35], "when a subsequence is selected, the property of frequency stability will be satisfied". Consider the further example of the set of all binary sequences containing m zeros and n ones. Apply the Kolmogorov selection rule to a sufficiently chaotic element of this set. If a resulting subchain is not too short, then the frequency of zeros in this subchain is close to the frequency of zeros in the entire chain. Our last remark pertains to finite chains of rational numbers with an arithmetic mean close to zero and a standard (i.e., mean square) deviation close to one. We specify three parameters: (1) the length of the chain; (2) the denominator of its terms; and (3) a majorant of the absolute value of its arithmetic mean and the absolute value of the difference between 1 and the standard deviation. Consider the set of all finite chains determined by these three parameters and along with them a chaotic chain in this set (i.e., Δ-random for Δ small). If the parameters are connected by some inequality, then the values of the terms of the chain have a frequency distribution close to the normal distribution. Precise formulations and detailed proofs of the facts presented in this section relating to the property of chaotic finite (binary or rational) chains being stochastic have been developed by Asarin [40].

2.5. Absolutely nonrandom objects. Of course we do not exclude the possibility of a chain y having a small defect of randomness and hence being random relative to one set of chains and at the same time having a large defect of randomness and hence being nonrandom relative to another set.[1] The following natural question arises: "Do there exist absolutely nonrandom objects," i.e., objects having a large defect of randomness with respect to any simple set? The answer to this question posed by Kolmogorov

[1] The dependency of the defect of randomness of a finite chain on a probability measure is investigated in [42].

was obtained by Shen'. The answer turned out to be positive: such objects exist. Here is the precise statement: let a and b be arbitrary constants; to every n sufficiently large, there is a binary chain y_n such that $d(y_n|A) \geqq b \log_2 n$ for any set A whatsoever containing y_n for which $H(A|\varnothing) \leqq a \log_2 n$. This formulation is a simple consequence of Theorem 2 in [33].

We now make some comments about what has been discussed. A statistician may have the following sort of problem: to demonstrate that an experimental result is typical. This means that he has to propose a statistical hypothesis, or in other words, include the experimental result in a set of possible outcomes in which the actually obtained result will appear to be typical. Speaking in mathematical terms, the statistician having obtained a result y must find a simple set A that contains y as a typical element. Thus, the theorem of Shen' shows that there are outcomes for which no simple statistical model of the kind described is possible. The question of course remains whether such objects exist in the real world.

3. Randomized Algorithms: General Survey.

3.1. Introductory remarks and examples. Section 3 opens the second part of our article devoted to the utilization of randomness in algorithms. Any algorithm that involves a random selection at certain of its steps is customarily called a probabilistic or randomized algorithm.

The use of such algorithms actually conforms fully to statistical tradition. Suppose that it is necessary to determine the arithmetic mean of a very large number of quantities. A deterministic algorithm directs us to add all these quantities and then to divide the sum by the number of terms. A randomized algorithm selects several specimens at random from among the quantities and then operates only with these specimens, which assures an obvious economy in the length of computations. In many important cases, the random result obtained in this way turns out to be close to the true answer with a high probability.

Another well-known example is the Monte Carlo method which has had a forty-year history (since its creation by von Neumann and Ulam).

All of the mentioned algorithms however yield a result which not only *can be* inexact but *is* inexact by its very nature since it only pretends to serve as an approximation to the correct answer, although with a small error and a large probability.

But approximate computations lie beyond the framework of this article. The authors have confined themselves to algorithms that deal only with discrete quantities (it is precisely such algorithms that are studied in the discipline called "theory of algorithms"). The arguments and values of such algorithms may always be regarded as finite chains of letters or, if convenient, words in a suitable alphabet.

Sometimes a probabilistic approach is used to estimate the quality of an algorithm. Our life would turn into a nightmare faster than anything if we did not disregard small probabilities of errors in our practical daily algorithms. In such practical daily algorithms, we are inclined simply to ignore those theoretically possible cases of a problem solvable by us which are rare (and occasionally even those which are disagreeable to us).

Such an approach is customary in the computational sciences. For example, a sorting algorithm (probabilistic or deterministic, it is all the same) is usually acknowledged to be good if it yields a correct answer in a reasonable time for the overwhelming majority of the input. But in order to justify such an approach, we must know something about the probability distribution of the input. In the case of a sorting algorithm, it is usually assumed that all of the $n!$ permutations of the n objects being sorted are equally likely. However such assumptions may prove to be unjustified. It may happen that "difficult" cases permeate the input more often than others in a sorting algorithm.

We shall consider here just the algorithms that lead for any input to a correct answer with a high probability and sufficiently fast. Declaring sorting, search and other important algorithms to be beyond the scope of this article, we are going to concentrate our attention on the following situations.

We wish to evaluate a function φ and compare the performances of deterministic and probabilistic algorithms evaluating it. For the sake of simplifying the presentation, we shall confine ourselves to the case where φ assumes only two values: 0 and 1 or "Yes" and "No". Every such function is called a *predicate.* A predicate φ distinguishes those x for which $\varphi(x)=1$ from those for which $\varphi(x)=0$. One says that φ *recognizes* the set $\{x|\varphi(x)=1\}$ to be a subset of the domain of definition of φ.

A basic questicn is whether there is a probabilistic algorithm that evaluates φ at a faster rate than any deterministic algorithm. In this connection, it is necessary to distinguish whether it is being compared with all conceivable algorithms or just with the known or published algorithms. If, for example, we are interested in Monte Carlo algorithms, then their advantage as to computational speed is established only in comparison to those deterministic algorithms that are known at present.

Among the probabilistic algorithms of the type just considered—algorithms evaluating functions—the most well known are two algorithms for *discerning primality,* i.e., determining whether a given number is a prime. One of them is due to Strassen and Solovay [21] and the other to Rabin [20]. Both start out from the following idea.

In order to show that an integer m is composite, one merely has to find some divisor of m. Therefore any such divisor can be termed a witness to the presence in m of the property of "not being prime," or briefly, a *witness to the fact that m is composite.* If m is a fixed natural number, then the testing of all natural numbers from 1 up to $\sqrt{m}$ with the object of determining its divisors tells us whether m is prime or composite. Such a deterministic algorithm for discerning primality requires however a large number of steps: the number is of the order of an exponential of the notational length of m, i.e., the logarithm of m. If we introduce randomization and select several numbers at random with the hope of finding witnesses, there will be no great advantage in such an action. In fact, no matter what m is, the number of its divisors is comparatively small. Therefore, not finding witnesses after several attempts, we obtain too little information.

The probabilistic algorithms—both Solovay and Strassen's and Rabin's—are also based on the notion of witness but only a more delicate one. A witness now is not simply a divisor of m but is rather a number possessing a subtle number-theoretic property. The property can be discerned by an effective procedure in polynomial time (in relation to the notational length of m). If m is prime, then not a single witness exists.

But now if m is composite, then "witnesses ... are abundant. If tests fail many times to produce a witness, then we are provably confident that the number is prime" [20, § 5]. In Rabin's algorithm, for example, if m is composite, then at least three-fourths of the numbers between 1 (inclusively) and m (inclusively) are witnesses [26, Thm. 1] (in Theorem 8 in [20], a weaker lower bound of one-half was established for the fraction of witnesses).

It now turns out to be useful to apply randomization and to select a random number distributed uniformly between 1 and m. Having selected such a b, we test to see whether it is a witness. If the answer is "Yes", then m is composite. If the answer is "No." then we cannot know anything for sure. It will be recalled however that in Rabin's algorithm $\frac{3}{4}$ of all the numbers between 1 and m are witnesses. Therefore after

k independent random selections of b, we have a correct answer with error probability less than $1/2^{2k}$. It is easy to see that to any fixed positive ε, it is necessary to expend polynomial time (in the notational length of m) in order to discern the primality of m with an error probability less than ε.

"Starting with 2^{400} and decreasing each time by 1, every number was subjected to the test. Within a minute 2^{400}-593 was identified as the largest prime below 2^{400}. This number was tested more than 100 times without a change in the conclusion" [20, § 4]. Thus it is possible to say that 2^{400}-593 is a prime with error probability not exceeding 2^{-200} (or even 2^{-2000} if each trial consists of not one random selection but rather ten, which is not clear from the text). Some comments on the nature of such probabilistic assertions may be found on p. 129 in [26].

It should be clearly realized however that we have not seen a randomized algorithm which works faster than a deterministic one. As a matter of fact, it is quite possible that there exists a deterministic algorithm that discerns primality and requires only polynomial computational time. As Miller showed in [19], such an algorithm actually exists whenever the generalised Riemann hypothesis holds.

Nevertheless, there is a sense that the introduction of randomness into algorithms can afford some advantages. The following simple example indicates some possible sources of such advantages if the latter do actually exist.

Suppose that at a distance of one step to the left or to the right of the place where we are situated, there lies a scrap of paper on which some integer is written. We wish to find out if it is even or odd. Beforehand we neither know the answer to this question nor in which direction the note is situated. The best deterministic algorithm requires three steps in the worst case: a step is taken to the left and if the note is not found then two further steps to the right are needed. But if we are willing to obtain a correct answer not with absolute certainty but rather with probability $\frac{3}{4}$, then we can describe a randomized algorithm requiring just one step. Here is the algorithm: flip a coin; if it falls heads, go to the left and if it falls tails go to the right; if there is no note, flip the coin once more, this time with the aim of determining the evenness or oddness of the unknown number.

3.2. Evaluation of a function. Let us repeat what our basic problem is. We are given some two-valued function or predicate φ defined on the set of all words of some alphabet, in particular on the set Ξ of all binary chains. The question is what gains may be achieved in the rate of evaluating the function if a suitable randomized algorithm replaces a deterministic one?

We first state some terminology. Any computational apparatus under consideration will be called a *machine.* A machine operates in steps governed by a *program.* A program is a set of *commands* and exactly one command can be used at each step. This is a description of a *deterministic machine.*

A *probabilistic machine* differs from its deterministic rival by the following feature: at some steps there are several actions, instead of a single specified action, that the machine can perform with given probabilities (it is assumed that all these probabilities are computable real numbers, let us say, rational numbers). It is possible to assume (and all of the facts presented below remain valid) that there are exactly two equally likely actions at each step and a machine selects one of them to use at random. In other words, at the beginning of each step, the machine flips a coin.

We now assign a meaning to the statement: *a machine computes a function φ in a (running) time T with a probability p.* By definition, here is what this sentence signifies. When a machine processes an arbitrary argument a supplied to the input, the probability of the following event is at least p: "the machine stops after at most $T(a)$ steps with the result $\varphi(a)$."

Henceforth, we shall always assume that $p > \frac{1}{2}$. This probability p has a rather small effect on the computational time. Indeed, it is easy to convert a machine that evaluates a function with a probability p into a machine that evaluates it with a pre-assigned probability arbitrarily close to one, with a resulting insignificant increase in computational time, namely, it is multiplied by a constant. The new machine repeats several times the evaluation of the initial machine and then chooses the final result to be the one that has been obtained in a majority of the trials. It should be noted that also for a deterministic machine the computational time can really only be evaluated up to a multiplicative constant.

However one may not be interested at all in the computational time and one may consider a machine that evaluates φ with a specified probability p but with no time restrictions, i.e., with an arbitrary computational time. There is an important (and chronologically the very first) theorem on probabilistic machines proved by Claude Shannon and his co-authors thirty years ago [4]: if a function φ is computable on some probabilistic machine with some probability p, then this function is also computable on some deterministic machine. Thus if probabilistic machines do possess advantages over deterministic machines, these advantages are not manifested by the mere existence of an algorithm.

An analysis of the proof of the theorem just stated shows that if a probabilistic machine computes a function φ in time T, then a corresponding deterministic machine can do it in a time not exceeding c^T for some constant c. Thus, the largest acceleration achievable at the expense of using randomization in algorithms involves merely a reduction of the time T to $\log T$.

At present the question remains unanswered as to whether it is possible to gain something in computational time by replacing a deterministic machine by a probabilistic one. Of course, no benefit will be derived if the function subject to evaluation is extremely simple (for example, is a constant). As Shen' pointed out recently, one also cannot hope to derive any benefit from evaluating extremely complicated functions.

The question in its fully general form thus remains an open one. However, Freivald gave a positive answer for one limited class of well-known simple computers. He produced a function requiring $O(n^2)$ computational steps when using deterministic single-tape single-head Turing machines and $O(n \log n)$ steps when using probabilistic machines of the same type.

Thus Freivald has proved some unique facts about the advantages of evaluating a function probabilistically even if only for a very limited computational model.

We now formulate the basic facts in a more precise manner. Our simplest Turing machine has one tape which is infinite in both directions and is divided into storage cells. At most one word in a pre-assigned alphabet can be written in each storage cell. The machine head can run along the tape, read it, print out and also label itself from a pre-assigned finite set of labels.

Before the computation begins, the argument is written down on the tape as a finite chain of letters. When the machine finally stops, one of the digits zero or one is in front of the head which it can read; and what the head reads at this final moment is then the computational result.

We now give a simple example which seems to be due to Freivald. Let

$$\varphi(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y. \end{cases}$$

Thus, φ evaluates the truth of the statement whether x and y coincide. In order to compute φ on a Turing machine, we must represent the input pair $\langle x, y\rangle$ in the form of a chain $x * y$ provided that x and y have been expressed in binary, decimal or any other reasonable system of notation; here $*$ is a new letter. Then the chain $x * y$ is placed on the tape. We agree to denote the length of the chain $x * y$ by n.

Using the technique of Barzdin' [6], we can prove that the best deterministic algorithm for computing φ on a single-tape single-head Turing machine requires computational time at least $O(n^2)$. (Of course, for any given algorithm, the computational time depends on the whole input $x * y$ and not just its length n. Therefore, in saying "requires a time which is not less than $O(n^2)$", we are thinking of the worst case—caused by the "worst" input of length n—which could only happen in computing φ by our best algorithm.) Freivald proposed a method in [18], [24] with which it is possible to prove that if one allows random steps in a computation, then the computational time can be reduced down to $O(n \log n)$—and then, as was already stated, a correct answer will occur with arbitrarily high probability. In addition, Freivald proved that it is impossible to lower the computational time any further.

Freivald's method, which makes it possible to lower the deterministic estimate $O(n^2)$ down to the probabilistic estimate $O(n \log n)$, is reminiscent of the method of witnesses for the algorithm of primality developed in § 3.1. Any prime p which is not a divisor of the difference $x-y$ is a witness to the noncoinciding of x and y. Starting from this, Freivald's probabilistic algorithm works as follows. It generates in random fashion just one prime p altogether (but nevertheless by means of a special procedure) and then checks whether x is congruent to y modulo p. If $x \neq y \pmod p$, then the answer to the main question "Is $x = y$?" is "No" and it is a correct answer. If $x \equiv y \pmod p$, the answer is "Yes, they coincide" and though this answer need not be correct, the probability of an error is small. The lowering of the computational time achievable by using a single-tape single-head computer is due to the following. There is no need (in the new probabilistic algorithm) to transfer *all* of the information from the portion of the tape where x is written to the portion of the tape where y is written; it suffices now to transfer just a *part* of this information (namely, the remainder on dividing by p—and, indeed, including p itself).

It might be surmised that all the advantages of probabilistic machines are associated with the potential errors. It turns out that this is not so. Freivald constructed a function ψ in [25], [44] requiring essentially less time in its computation on a suitable probabilistic *errorless* (single-tape single-head) Turing machine than in its computation on any kind of single-tape single-head deterministic Turing machine. The word "errorless" means that every result given by the machine is correct. This function ψ shows that at least some of the advantage in using randomized algorithms is not in allowing little likely errors but in the random bifurcation of the computational process.

It is unknown whether to any probabilistic Turing machine M that evaluates some function, there is an errorless machine evaluating the same function just as fast as M. No one knows either how to distinguish those problems in which the application of randomized algorithms leads to demonstrable benefits from those problems where no such benefits will arise.

3.3. Polynomial computational time. The preceding facts concern only single-tape single-head Turing machines and cannot be carried over immediately to other computational models, say, multi-tape and multi-head Turing machines and so forth. So that the results of the previous section talk not so much about the properties of the considered functions as about the properties of computational models. And in fact, the predicate of an equality, say, can easily be computed (deterministically) in many computational models—and randomization introduces no benefits here.

Of course, the computational time for a specified function may change when we change a computational model. Fortunately, such time changes are not too radical. For a fixed computational model, consider the class $\mathcal{P}$ of all functions computable on machines of this model in polynomial time (polynomial in the input length). It turns out that $\mathcal{P}$ is independent of the model selected: this class is unique for all standard computational models. Any function in $\mathcal{P}$ is said to be *computable in deterministic polynomial time.*

One can define in a completely analogous way the class $\mathcal{BPP}$ of all functions computable in probabilistic polynomial time. By virtue of what was said earlier in § 3.2, there is no need to indicate any specific value for a probability in defining the class $\mathcal{BPP}$.

The main question which has drawn a lot of attention is whether $\mathcal{P}$ and $\mathcal{BPP}$ coincide. It is only known that

$$\mathcal{P} \subset \mathcal{BPP} \subset \mathcal{P}\text{ Space.}$$

$\mathcal{P}$ Space in this denotes the class of all (two-valued) functions computable on the polynomial range on deterministic or probabilistic machines, which one does not matter in this case, as shown by J. Simon's theorem published, for example, in [28].

Although the question "Is the class $\mathcal{BPP}$ equal to the class $\mathcal{P}$?" is still open, there are a number of theorems furnishing a partial answer in the affirmative direction. We are thinking here of the Adleman-Bennet-Gill theorem and Gács' theorem.

The first one establishes that

$$\mathcal{BPP} \subset \mathcal{NUP},$$

where the class $\mathcal{NUP}$ is in a certain sense similar to $\mathcal{P}$: it is the so-called "non-uniform $\mathcal{P}$-class".

In order to clarify what $\mathcal{NUP}$ is, we recall the definition of $\mathcal{P}$. A predicate or two-valued function φ defined on the set Ξ of all binary chains belongs to $\mathcal{P}$ if and only if there is a polynomial Q and a deterministic machine $\mathfrak{A}$ such that, for any n and each chain x of length n, the machine $\mathfrak{A}$ computes $\varphi(x)$ in at most $Q(n)$ steps. The definition of the class $\mathcal{NUP}$ differs from that of $\mathcal{P}$ in just the following detail: the condition of existence of a unique deterministic machine $\mathfrak{A}$ changes to the weaker condition of the existence of some sequence $\mathfrak{A}_0, \mathfrak{A}_1, \cdots$ of such machines. The precise formulation is as follows: by definition, $\varphi \in \mathcal{NUP}$ if and only if there exists a polynomial Q and a sequence of deterministic machines $\mathfrak{A}_0, \mathfrak{A}_1, \cdots$ such that, for any n and each chain x of length n, the machine $\mathfrak{A}_n$ evaluates $\varphi(x)$ in at most $Q(n)$ steps, the length of the program of $\mathfrak{A}_n$ also not exceeding $Q(n)$. Observe that this definition requires not only no polynomial computability but even no simple computability of the sequence $\langle\mathfrak{A}_n\rangle$.

The definitive form of the inclusion $\mathcal{BPP} \subset \mathcal{NUP}$ was proved by Bennet and Gill [27]. Adleman [23] was the first to state a theorem about the inclusion but he considered only randomized machines of a special form.

Gács theorem was published in [36]. It expresses an arbitrary predicate $G \in \mathcal{BPP}$ in terms of a suitable predicate $S \in \mathcal{P}$ and polynomially restricted quantifiers. Here is an exact formulation: any predicate G belonging to $\mathcal{BPP}$ can be represented in the form

$$\begin{gathered} G(x) \Leftrightarrow \exists y \quad \forall z \quad S(x, y, z) \\ |y| < Q(|x|), \qquad |z| < Q(|x|), \end{gathered}$$

where Q is a polynomial, $|x|$, $|y|$ and $|z|$ are as usual the lengths of chains x, y and z, and $S \in \mathcal{P}$. The notation "$S \in \mathcal{P}$" means that $S(x, y, z)$ is computable by some deterministic machine in a time not exceeding a polynomial in $|x|+|y|+|z|$.

The mathematical and philosophical importance of the question as to whether the classes $\mathcal{P}$ and $\mathcal{BPP}$ coincide warrants no comment.

4. Randomized Algorithms: Application to the Foundations of Probability Theory. It is highly remarkable that the study of randomized algorithms finds application in the foundations of probability theory and the theory of complexity of finite objects.

4.1. A priori probability of a binary chain. In this section, the term "machine" will denote any representative of a fixed family of computers, for definiteness say, all single-tape single-head Turing machines. We choose some *probabilistic* machine $\mathfrak{A}$ from this family and feed 0 into the input of the machine. If and when the machine stops after processing the input, a binary chain $\xi \in \Xi$ is read from the tape. Once a machine $\mathfrak{A}$ is given, each chain $\xi \in \Xi$ has its own probability $p(\xi)$ of appearing on the tape at the conclusion of a computation starting from zero. The sum $s = \sum_{\xi \in \Xi} p(\xi)$ does not exceed one so that the function p can be called a *semi-distribution.* Clearly, $1-s$ is the probability of the event "the machine never stops if its input is 0". It turns out that there is among all such semi-distributions an *almost maximal* one which we denote by $\bar{p}$. This means that there is a machine $\bar{\mathfrak{A}}$ with the following property: if $\bar{p}$ is the semi-distribution corresponding to $\bar{\mathfrak{A}}$ and p is any semi-distribution corresponding to any other machine $\mathfrak{A}$, then

$$c\bar{p}(\xi) \geqq p(\xi)$$

for any $\xi \in \Xi$, where c is a constant depending just on $\mathfrak{A}$ but not on ξ. Any two almost maximal semi-distributions $\bar{p}_1$ and $\bar{p}_2$ are related by the relation

$$c_1\bar{p}_1(\xi) \geqq \bar{p}_2(\xi), \qquad c_2\bar{p}_2(\xi) \geqq \bar{p}_1(\xi),$$

for some constants c_1 and c_2.

If $\bar{p}$ is any almost maximal distribution, then $\bar{p}(\xi)$ can be treated as *the a priori probability* of the chain ξ. Each conception of a priori probability may in turn be treated as a measure of complexity in that simpler events have higher subjective or a priori probability. It is better to take as a measure of complexity not $\bar{p}(\xi)$ itself but its logarithm, or more precisely $-\log_2 \bar{p}(\xi)$. If we want our measure of complexity to be an integer, then this logarithmic quantity can be replaced by its closest integral value.

This approach, which goes back to [15], is closely related to the ideas of § 1.4. In fact, a binary sequence $\langle a_0, a_1, \cdots\rangle \in \Omega$ is random relative to the uniform Bernoulli distribution if and only if the inequality

$$-\log_2 \bar{p}(a_0, a_1, \cdots, a_{n-1}) \geqq n - c$$

holds for all n and some c not depending on n. This assertion of the equivalence of the two definitions, and also for a slightly different notion of a priori probability, can be deduced from [29]: it is an immediate consequence of Corollary 3.2 and Theorem 2.3.

4.2. A priori frequency of a binary chain. Another development of the topic being discussed in this chapter was suggested recently by Muchnik [43]. Consider some computable sequence of binary chains $\langle \xi_0, \xi_1, \cdots\rangle$, $\xi_i \in \Xi$, $i = 0, 1, \cdots$. For ξ any chain in Ξ, we can compute its *lower frequency* $q(\xi)$ given by

$$q(\xi) = \varliminf_{n\to\infty} |\{i | i < n \ \& \ \xi_i = \xi\}|/n,$$

where $|\cdot|$ denotes cardinality of a set. It turns out that among the computable sequences there is an *almost maximal* sequence $\langle \bar{\xi}_0, \bar{\xi}_1, \cdots \rangle$ (in fact even many such sequences) possessing the following property: if $\bar{q}$ is the lower frequency for this computable sequence and q is the lower frequency for any other computable sequence, then

$$c\bar{q}(\xi) \geqq q(\xi).$$

In this, ξ is an arbitrary chain in Ξ and c depends only on the sequence $\langle \xi_0, \xi_1, \cdots \rangle$ as a whole and not on ξ.

For ξ any chain in Ξ, the quantity $\bar{q}(\xi)$ can be treated as *the a priori frequency* of ξ and thereby as some measure of its complexity.

Thus, in this chapter we have described two measures of complexity of a word ξ: its a priori probability $\bar{p}(\xi)$ and its a priori frequency $\bar{q}(\xi)$. It would probably be more correct to call them measures of simplicity rather than measures of complexity. These two measures do not coincide (one can only speak here about coincidence up to at most a multiplicative constant). Nevertheless, they have, indeed, a lot in common.

More precisely $\bar{q}$ coincides with $\bar{p}'$ (which means that $0 < a \leqq \bar{p}'(\xi)/\bar{q}(\xi) \leqq b$ for some constants a and b), where $\bar{p}'$ is the so-called relativized a priori probability with respect to $\mathbf{0}'$. The quantity $\bar{p}'$ is defined in a similar way to $\bar{p}$ with the only difference now being that instead of ordinary randomized Turing machines, one uses randomized machines that in some way know an answer to any question out of a series forming an algorithmically unsolvable lump problem, as say, whether a Turing machine with a program z would ever come to a halt in processing an input 0.

REFERENCES

[1] R. VON MISES, *Grundlagen der Wahrscheinlichtkeitsrechnung*, Math. Z., 5 (1919), pp. 52-99.

[2] ———, *Wahrscheinlichkeit, Statistik und Wahrheit*, Springer, Vienna, 1928.

[3] A. CHURCH, *On the concept of a random sequence*, Bull. Amer. Math. Soc., 46 (1940), pp. 130-135.

[4] K. DE LEEUW, E. F. MOORE, C. E. SHANNON AND N. SHAPIRO, *Computability by probabilistic machines*, in Automata Studies, C. E. Shannon and J. McCarthy, eds., Princeton Univ. Press, Princeton, 1956, pp. 183-212.

[5] A. N. KOLMOGOROV, *On tables of random numbers*, Sankhya, Ser. A, 25 (1963), pp. 369-376.

[6] YA. M. BARZDIN', *Complexity in detecting symmetry on Turing machines*, Problemy Kibernet., 15 (1965), pp. 245-248. (In Russian.)

[7] A. N. KOLMOGOROV, *Three approaches to defining "quantity of information"*, Problems Inform. Transmission, 1 (1965), pp. 1-7.

[8] P. MARTIN-LÖF, *On the concept of a random sequence*, Theory Prob. Appl., 11 (1966), pp. 177-179.

[9] ———, *The definition of random sequences*, Inform. and Control, 9 (1966), pp. 602-619.

[10] D. LOVELAND, *A new interpretation of von Mises' concept of random sequence*, Z. Math. Logik Grundl. Math., 12 (1966), pp. 279-294.

[11] ———, *The Kleene hierarchy classification of recursively random sequences*, Trans. Amer. Math. Soc., 125 (1966), pp. 497-510.

[12] A. N. KOLMOGOROV, *On the logical foundations of information theory and probability theory*, Problems Inform. Transmission, 5 (1969), pp. 3-4.

[13] K. JACOBS, *Turing-Maschinen und züfallige 0-1-Folgen*, in Selecta Mathematica, Vol. 2, Springer, Berlin, 1970, pp. 141-167.

[14] P. MARTIN-LÖF, *Notes on Constructive Mathematics*, Almqvist and Wiksell, Stockholm, 1970.

[15] A. K. ZVONKIN AND L. A. LEVIN, *The complexity of finite objects and development of the concepts of information and randomness by means of the theory of algorithms*, Russian Math. Surveys, 25 (1970), pp. 83-124.

[16] L. A. LEVIN, *On the notion of a random sequence*, Soviet Math. Dokl., 14 (1973), pp. 1413-1416.

[17] C. P. SCHNORR, *Process complexity and effective random tests*, J. Comput. System Sci., 7 (1973), pp. 376-378.

[18] R. V. FREIVALD, *Fast computations on probabilistic Turing machines*, Uchen. Zap. Latviisk. Gos. Inst., 233 (1975), pp. 201-205. (In Russian.)

[19] G. L. Miller, *Riemann's hypothesis and tests for primality*, J. Comput. System Sci., 13 (1976), pp. 300-317.

[20] M. O. Rabin, *Probabilistic algorithms*, in Algorithms and Complexity: New Direction and Recent Results, J. Traub, ed., Academic Press, New York, 1976, pp. 21-39.

[21] R. Solovay and V. Strassen, *A fast Monte-Carlo test for primality*, SIAM J. Comput., 6 (1977), pp. 84-54; erratum, 7 (1978), p. 118.

[22] C. P. Schnorr, *A survey of the theory of random sequences*, in Basic Problems in Methodology and Linguistics, R. E. Butts and J. Hintikka, eds., D. Reidel, Dordrecht, 1977, pp. 193-211.

[23] L. Adleman, *Two theorems on random polynomial time*, in Nineteenth Symposium on Foundations of Computer Science (Ann Arbor, Mich.), IEEE, Long Beach, 1978, pp. 75-83.

[24] R. V. Freivald, *Accelerating detection of certain sets by use of a random number generator*, Problemy Kibernet., 36 (1979), pp. 209-224. (In Russian.)

[25] ———, *On operating time on deterministic and nondeterministic Turing machines*, in Latv. Mat. Ezhegodnik, 23, Zinatne, Riga, 1979, pp. 158-165. (In Russian.)

[26] M. O. Rabin, *Probabilistic algorithms for testing primality*, J. Number Theory, 12 (1980), pp. 128-138.

[27] C. H. Bennett and J. Gill, *Relative to a random oracle A, $p^A \neq NP^A \neq$ co-NP^A with probability* 1, SIAM J. Comput., 10 (1981), pp. 96-113.

[28] H. Jung, *Relationships between probabilistic and deterministic tape complexity*, Lecture Notes Comput. Sci., 118 (1981), pp. 339-346.

[29] V. V. V'yugin, *Algorithmic entropy (complexity) of finite objects and its application to defining randomness and quantity of information*, Semiotika i Informatika, VINITI, Moscow, 16 (1981), pp. 14-43. (In Russian.)

[30] V. A. Uspensky and A. L. Semenov, *What are the gains of the theory of algorithms: basic developments connected with the concept of algorithm and with its applications in mathematics*, Lecture Notes, Comput. Sci., 122 (1981), pp. 100-234.

[31] ———, *Algorithmic theory: its basic discoveries and applications*, in Algorithms in Contemporary Mathematics and Their Applications, A. P. Ershov, D. Knut, eds., Part 1, BTs SO Akad. Nauk SSSR, Novosibirsk, 1982, pp. 99-342. (In Russian.)

[32] A. Kh. Shen', *The frequency approach to defining a random sequence*, in Semiotika i Informatika, 19 (1982), VINITI, Moscow, pp. 14-42. (In Russian.)

[33] ———, *The concept of Kolmogorov (α, β)-stochasticity and its properties*, *Soviet Math. Dokl.*, 28 (1983), pp. 295-299.

[34] A. N. Kolmogorov, *Combinatorial foundations of information theory and calculus of probabilities*, Russian Math. Surveys, 38 (1983), pp. 29-40.

[35] ———, *On logical foundations of probability theory*, Lecture Notes Math., 1021, Springer, New York, 1983, pp. 1-5.

[36] M. Sipser, *A complexity-theoretic approach to randomness*, in Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing, Boston, MA, 1983, ACM, New York, 1983, pp. 330-335.

[37] V. G. Vovk, *Algorithmic information theory and prediction problems*, in Complexity Problems of Mathematical Logic, M. I. Kanovich et al., eds., Kalininsk. Gos. Univ., Kalinin, 1985, pp. 21-24. (In Russian.)

[38] E. A. Asarin and A. V. Pokrovskii, *Application of Kolmogorov complexity to analyzing the dynamics of controlled systems*, Automat. and Telemekh., 1 (1986), pp. 25-33. (In Russian.)

[39] E. A. Asarin, *Individual random continuous functions*, in Summaries of Reports of the First All-World Congress of the Bernoulli Society of Mathematical Statistics and Probability Theory, Yu. V. Prokhorov, ed.-in-chief, Vol. 1, Nauka, Moscow, 1986, p. 450. (In Russian.)

[40] ———, *Some properties of Kolmogorov Δ-random finite sequences*, Theory Probab. Appl., 32 (1987), pp. 507-508.

[41] G. G. Vovk, *The law of the iterated logarithm for random Kolmogorov, or chaotic, sequences*, Theory Probab. Appl., 32 (1987), pp. 413-425.

[42] V. V. Vyugin, *On the defect of randomness of a finite object with respect to measures with given complexity bounds*, Theory Probab. Appl., 32 (1987), pp. 508-512.

[43] An. A. Muchnik, *Lower limits of frequencies in computable sequences and relativized a priori probability*, Theory Probab. Appl., 32 (1987), pp. 513-514.

[44] R. V. Freivald. *On the operating time of errorless Turing machines*, Theory Probab. Appl., 32 (1987), pp. 514-516.