Kybernetika

RECURSIVE CLASSIFICATION OF PSEUDO-RANDOM SEQUENCES

IVAN KRAMOSIL

ACADEMIA
PRAHA

Some results achieved by Kolmogorov, Chaitin, Solomonoff, Martin-Löf, Schnorr and others proved the high degree of coincidence between the sequences or strings of symbols, which are of high algorithmic complexity and those sequences which are "true-random" in the sense that they satisfy some empirical or theoretical tests of randomness. Some difficulties of this approach are caused by the fact that the algorithmic complexity of a sequence, defined by the length of the shortest program which generates the sequence in question, is not an effectively computable function of this sequence. In the presented paper the definition of algorithmic complexity is modified with respect to a theoretical computer (universal Turing machine) which works within time and space limitations. This modification makes the conditions, which a sequence is to satisfy in order to be taken as pseudo-random, weaker but effectively (recursively) decidable. The aim of this paper is to investigate, in which degree the desirable properties of sequences with high algorithmic complexity are preserved under this modification.

CONTENTS

1. INTRODUCTION – PROBLEM FORMULATION

The papers and works dealing with pseudo-random sequences (strings) of numbers or other symbols can be divided, with respect to their motivation and orientation, into two main groups which are not quite disjoint. The first group is oriented rather technically and practically. It tries to use pseudo-random sequences as an appropriate replacement of physical (true random) generators of random inputs, which are necessary in many methods of statistical estimation of unknown parameters, hypothesis testing of various kinds or approximations of some computational and decision procedures which are not effective in the theoretical or practical sense. The advantage of pseudo-random generators consists in their high speed and low costs, if compared with physical (true random) generators, with which pseudo-random sequences can be generated, including the possibility to use computers for these sakes. The quality of pseudo-random generators is classified just with respect to explicitly utiliary criteria, i.e. with respect to the measure in which the results, obtained when using pseudo-random inputs, satisfy some statistical tests of quality, appropriately deduced from the specific features of the statistical decision or computation problem in question. These papers do not pose and do not solve questions of ontological character, i.e. from the fact that such and such pseudo-random generator has been or can be used with success in order to solve such and such problem no conclusions are drawn as far as the possibility to reduce the notation of randomness to some other, may be more primitive notions, is claimed or refused.

The second group of papers dealing with pseudo-random sequences consists of those ones for which just this ontological aspect is substantial, and which try to penetrate as deeply as possible into the nature of randomness. This effort dates from the very beginnings of mathematical probability theory and mathematical statistics (let us remember, just as an example, the studies of von Mises) and has not been fully abandoned neither after the Kolmogorov axiomatic probability theory having come successfully into scene. This axiomatic approach apriori abandons the problem how to classify particular sequences of results into "random" and "non-random" ones and proclaims such a question to be illegitimate. As a matter of fact, he was Kolmogorov himself who admitted, later, that the question about the randomness of particular sequence is justified and legitimate, moreover, he proposed a way how to construct an appropriate criterion of randomness for finite sequences.

In this paper we uphold the viewpoint which is situated somewhere between the two extremal positions as stated above. The outcome of our explanation is given by a model which has been conceived with the aim to explain the notion of randomness by its reduction or transformation into the notion of high algorithmic complexity. Nevertheless, we shall try to modify this model in such a way that it satisfies some at least theoretical effectivity criteria (i.e. an effective computability of values necessary in order to decide about the randomness of the sequence in question). Moreover, we shall confront the resulting notion of pseudo-random sequence with the possibility of its use in Monte-Carlo methods, i.e. we shall be very near to the position occupied by practical user of pseudo-random generators, as mentioned above.

A number of authors have investigated the idea to measure the complexity of a finite or infinite sequence of symbols by the length of the shortest program, by the use of which an a priori fixed universal Turing machine is able to generate the sequence in question. Then, the sequence is proclaimed to be random, if its complexity is not substantially less than its length, i.e., if these is no substantially shorter way how to define the sequence in question than to write it simply down. It has been shown, that the sequences which are random in this sense can play very well the role of random inputs, when some estimations using the Monte Carlo methods are to be made. Moreover, the results obtained in this way are qualitatively better than in the case of true-random inputs, hence, rather the true random inputs should be taken for a not quite satisfactory approximations of sequences with high algorithmic complexity than vice versa. The difficulties of this approach consist in principal theoretical limitations as far as our possibilities are concerned to decide the validity or nonvalidity of assertions dealing with the algorithmic complexity of sequences. In other words the algorithmic complexity of a sequence is not a recursive function of this sequence (in the case of a finite sequence we may take its Gödel number). Hence, in general we are not able to verify, effectively, the validity of conditions which the complexity of a sequence is to satisfy in order that the sequence in question could be used as a pseudo-random input.

In this paper we shall try to weaken the criterion of randomness, derived from the algorithmic complexity, in such a measure that the predicate of pseudo-randomness would become effectively decidable under the condition that the possibility to use such pseudo-random sequences as inputs in the Monte-Carlo methods were preserved in the most possible degree. The originally defined algorithmic complexity and the concept of pseudo-randomness derived on its base will then play some absolute or limit role. I. e., they will be able to be approached or approximated arbitrarily by an appropriate choose or modification of parameters and by an appropriate increasing of the time and space computational complexity. Our main goal will be to assure that the conditions posed on pseudo-random sequences were effectively decidable at least in the theoretical sense. Even in the cases when we shall obtain some explicit expressions or upper bounds for the time and space complexities of the investigated computational and decision procedures, we shall not study the possibilities of their practical implementations. These questions can be and should be subjected to a further investigation connected with this paper and proceeding in its direction.

Let us start, now, by the building up a formal apparatus necessary for our further considerations.

2. ABSOLUTE AND RELATIVE ALGORITHMIC COMPLEXITY AND PSEUDO-RANDOMNESS

Let us start with a finite set A which will be called *alphabet*; elements of A are called *letters*. Our investigation will be oriented toward finite and infinite sequences (strings) of letters from A. Such sequences can be joined using the *concatenation operation*, denoted by an asterisk (*). The *length* of a sequence is denoted by l and is defined by the number of occurences of letters in the sequence. So we have $A^n = \{x : l(x) = n\}$, $A^\infty = \{x : l(x) = \infty\}$, we set $A^* = \bigcup_{n=0}^{\infty} A^n = \{x : l(x) < \infty\}$, where $A^0 = \{\Lambda\}$ contains just the empty sequence Λ. The inequality $\operatorname{card}(A) = C(A) \geqq 2$ is supposed to be valid throughout this paper.

As an apparatus by the mean of which the sequences of letters will be investigated and handled we shall use the *universal Turing machine* (UTM) over the alphabet A and with one tape, as presented by Davis ([1]) in the case of the binary alphabet $\{0, 1\}$. Machines with more than one tape are not mentioned below. Informally said, a Turing machine of this type consists of one *tape* which is infinite in both the directions and which is divided into an infinite but countable number of boxes, and of a *head*, which is always situated over just one of the boxes and which is able to move to the neighborhood boxes, either to the left or to the right. Finally, Turing machine contains a finite number of *instructions*. By the set of instructions we mean a finite set of quadruples of the form $\langle q_i S_j S_k q_l \rangle$, where q_i, q_l are the so called *internal states* of the Turing machines, S_j is either a letter from A or the symbol B which denotes the *blank* (the box in which no letters is inscribed), finally S_k is either a letter from A or B, or one of the two auxiliary symbols R, L; we always suppose that $A \cap \{B, R, L\} = \emptyset$. The quadruple $\langle q_i S_j S_k q_l \rangle$ represents the following instruction: if the Turing machine is in the internal state q_i and if the box just below the head (i.e. the box which is just being read by the head) contains the symbol S_j (if this box is empty, in case $S_j = B$), then the machine changes its internal position into q_l and executes the following operation: it inscribes S_k into the read box supposing that $S_k \in A$ (S_j is erased); it erases S_j, supposing that $S_k = B$; it changes the position of the head one box to the left (if $S_k = L$) or to the right (if $S_k = R$), leaving S_j unchanged. The Turing machine continues to operate until there is an applicable quadruple in the set of instructions, i.e. such a quadruple $\langle q_i S_j S_k q_l \rangle$ that q_i corresponds to the actual internal state of the machine and S_j corresponds to the symbol just being read. The set of instructions is supposed to be consistent in the sense that there are no two instructions $\langle q_i S_j S_k q_l \rangle$, $\langle q_i' S_j' S_k' q_l' \rangle$ such that $q_i = q_i'$, $S_j = S_j'$, but $S_k \neq S_k'$ or $q_l \neq q_l'$.

As can be shown, there exist so called universal Turing machines which are able to simulate the work of an arbitrary Turing machine supposing that the corresponding input is joined with an appropriate code of the Turing machine which is to be simulated, e.g., with a code of its Gödel number. Clearly, each Turing machine is fully determined by the set of its instructions together with the conventions which is the initial state and from which box the reading begins. Hence, it is, after all, a finite description, the intuitive notions of tape and moving could be eliminated in favour of a more formal, but less intuitive notions of *instantaneous description*. There is an infinite number of universal Turing machines over the alphabet A, let us fix one among them, which will be denoted by U or $U(A)$, when the role of A is to be underlined. We have to accept the unpleasant fact, that the greatest part of the results presented below will be parametrized, more or less, by this choice of $U(A)$.

Let us consider sequences p, $S \in A^*$, $\mathbf{x} \in A^* \cup A^\infty$. The description $U(p, S) = \mathbf{x}$ has the following meaning: if the concatenation $S * p$ (in this order and separated by one blank) is inscribed on the tape of the machine U, if the machine is posed into the initial state (defined by an appropria:e convention), and if the head reads the leftmost symbol in $S * p$, then either the machine stops after a finite number of steps (i.e., after a finite number of applications of instructions) and there will be just the sequence $\mathbf{x}$ inscribed on the tape (it is the case when $\mathbf{x} \in A^*$), or the machine U will never stop, but for every initial segment $x_1 x_2 \dots x_n$ of $\mathbf{x}$ there exists a finite number of steps after execution of which $x_1 x_2 \dots x_n$ will be written on the tape and will not be changed by later steps (it is the case when $\mathbf{x} \in A^\infty$). In what follows the notation $U(p, S) = \mathbf{x}$ will be used almost always for finite sequences $\mathbf{x}$. $\mathcal{N} = \{0, 1, 2, \dots\}$ denotes the set of all non-negative integers.

Definition 1. Let U be a universal Turing machine over a finite alphabet A, let $p, S \in A^*$, $\mathbf{x} \in A^* \cup A^\infty$ be sequences of letters from A. The *absolute algorithmic complexity* $K_{U(A)}(\mathbf{x}/S)$ *of the sequence* x *under the condition* S is defined by the length of the shortest program p, which, concatenated with S, makes the machine U to generate $\mathbf{x}$, i.e.

$$(1) \qquad K_{U(A)}(\mathbf{x}/S) = \min \{l : l \in \mathcal{N}, l = l(p), p \in A^*, U(p, S) = \mathbf{x}\} ,$$

for the empty set $\emptyset$ we set $\min \{\emptyset\} = \infty$. If $S = \Lambda \in A^0$ (the empty word over A), we write $K_{U(A)}(\mathbf{x})$ instead of $K_{U(A(}(\mathbf{x}/\Lambda)$ and omit the expression "under the condition S".

The adjective "absolute" as used in Definition 1 is to distinguish the just introduced complexity measures from its modified variant which will be defined below as a "relative" complexity measure. Both the adjectives will be omitted supposing that no misunderstanding menaces.

Three basic properties of the absolute algorithmic complexity $K_{U(A)}(\mathbf{x}/S)$ are introduced in the following theorem. In spite of the fact that the corresponding proofs can be found in references, they are introduced below as well, as they can serve as an appropriate illustration of the way of reasoning used in the argumentation dealing with the algorithmic complexity.

Theorem 1.

$$(2) \qquad \text{(a)}\ (\exists c \in \mathcal{N})(\forall \mathbf{x} \in A^* \cup A^\infty)(\forall S \in A^*)(K_{U(A)}(\mathbf{x}/S) \leqq l(\mathbf{x}) + c) .$$

(b) If $U_1(A)$, $U_2(A)$ are two universal Turing machines over the alphabet A then

$$(3) \quad (\exists c(U_1, U_2) \in \mathcal{N})(\forall \mathbf{x}, S \in A^*)(|K_{U_1(A)}(\mathbf{x}/S) - K_{U_2(A)}(\mathbf{x}/S)| \leqq c(U_1, U_2)) .$$

$$(4) \qquad \text{(c)}\ (\forall T \in \mathcal{N}, T \geqq 0)(\forall n \in \mathcal{N}, n \geqq T)(\exists \mathbf{x} \in A^n)(K_{U(A)}(\mathbf{x}/l(\mathbf{x})) \geqq n - T) .$$

Proof. (ad a) There exists a program p_1, which works, when concatenated with arbitrary $\mathbf{x}$, $S \in A^*$, in such a way that it erases S and p_1, leaving just $\mathbf{x}$ on the tape. Hence, for $p = p_1 * \mathbf{x}$, $U(p, S) = \mathbf{x}$, so $K_{U(A)}(\mathbf{x}/S) \leqq l(p) = l(p_1) + l(\mathbf{x}) = l(\mathbf{x}) + c$. Clearly, c depends on U, but this will be the case for all other constants occurring below and we shall not always mention this dependence explicitly.

(ad b) If U_1, U_2 are two universal Turing machines over A, than there exists a fixed program of a fixed length which enables to rewrite programs of U_1 into programs for U_2 which are equivalent as far as the results are concerned, and vice versa. Hence, each program for U_1 can be used as program computing the same sequence on U_2 supposing the program is extended by the mentioned translating program. It follows, that for an appropriate $c = c(U_1, U_2)$ the two inequalities hold:

$$(5) \qquad K_{U_1(A)}(\mathbf{x}/S) \leqq K_{U_2(A)}(\mathbf{x}/S) + c(U_1, U_2) ,$$

$$(6) \qquad K_{U_2(A)}(\mathbf{x}/S) \leqq K_{U_1(A)}(\mathbf{x}/S) + c(U_1, U_2) ,$$

and from this the assertion (b) immediately follows. Hence, this assertion proves that the dependence of the function $K_{U(A)}(\mathbf{x}/S)$ on the choice of U is of limited character and that the results are independent of this choice "if not taking into consideration additive constants".

(ad c) Programs are finite strings of letters, so there are at most $(C(A))^i$ programs of the length i and there are at most

$$(7) \qquad \sum_{i=0}^{n-T-1} (C(A))^i = (C(A)^{n-T} - 1)(C(A) - 1)^{-1} < (C(A))^{n-T}$$

programs of the length shorter than $n - T$. However, there are $(C(A))^n$ sequences of the length n, so there must be at least one $\mathbf{x} \in A^n$ for which, given $l(\mathbf{x})$, no program exists which would be shorter than $n - T$, hence, $K_{U(A)}(\mathbf{x})) \geqq n - T$. □

The assertion (c), which we have just proved, assures the non-triviality of the following definition.

Definition 2. Let $T \in N$,$\mathbf{x} \in A^*$, then the sequence $\mathbf{x}$ is called *T-random.* if $K_{U(A)}(\mathbf{x}/l(\mathbf{x})) \geqq l(\mathbf{x}) - T$.

May be, we could also use the terms "pseudo-random with parameter T" or "absolutely T-random". Of course, the first idea coming into mind is to try, in which measure true-random sequences can br replaced by T-random ones, e.g. as side random inputs in Monte-Carlo methods and such investigations has been actually performed. The object of such investigations was a sequence $\mathbf{x},_1$ $\mathbf{x}_2, \ldots$ of sequences from A^*, satisfying the two following conditions:

(a) for all $i \in \mathcal{N}$, $l(\mathbf{x}_i) = i$, i.e., $\mathbf{x}_i \in A^i$,

(b) for all $i \in \mathcal{N}$ and for a $T \in \mathcal{N}$ a priori fixed, $K_{U(A)}(\mathbf{x}_i/l(\mathbf{x}_i)) \geqq i - T$.

The existence of such a sequence of sequences follows from the assertion (e) of Theorem 1, the well-known assertion proved by Martin-Löf (cf. [2]) yields, that $\mathbf{x}_i$ cannot be initial segments of one infinite sequence (the mentioned Martin-Löf's theorem claims, that there is no infinite sequence $\mathbf{x} = x_1 x_2 x_3 \ldots \in \{0, 1\}^\infty$ and no $T \in \mathcal{N}$ such that the inequality $K_{U(\{0,1\})}(x_1 x_2 \ldots x_n/n) > n - T$ holds for all $n \in \mathcal{N}$ and this result can be easily generalized to the case of another finite alphabet A). The sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ of sequences then possesses the following properties:

(1) If $m \in \mathcal{N}$, $m > 0$, if $a = a_1 a_2 \ldots a_m \in A^m$ is an m-tuple of letters, then the relative frequency of occurrences of a in $\mathbf{x}_i$ tends to $(C(A))^{-m}$, i.e., to the inverted value of the total number of such letters (of course, $\mathbf{x}_i$ must be considered as a sequence of letters from A^m with the overflous letters from A possibly erased). Particularly, for $m = 1$, the relative frequency of occurrence of each letter $a \in A$ in $\mathbf{x}_i$ tends to $(C(A))^{-1}$, in both the cases it is a convergence for $i \to \infty$. Cf. [5] for more details.

(2) Suppose that $E \subset \langle 0, 1 \rangle$ is a Borel measurable set of reals which is a union of semi-open intervals and the Borel measure of which is $\mu(E)$. Then we may use

$\mathbf{x}_i$ with i large enough in order to sample reals from $\langle 0, 1\rangle$ in such a way that the relative frequency of those points which belong to E differs from $\mu(E)$ by a value smaller than an a priori given $\varepsilon > 0$ (of course, the "large enough" i depends on ε). Moreover, let the characteristic function of E be recursive in the sense that the membership of a real $x \in \langle 0, 1\rangle$ to the tested set E can be effectively decided using a finite initial segment of the binary (decadic, $C(A)$-adic) expansion of x. Then for an i large enough the relative frequency obtained by $\mathbf{x}_i$ just equals $\mu(E)$ (the "large enough" i depends on the computational complexity of the algorithm which computes the characteristic function of the set E). Cf. [3] and [4] for more details.

Hence, the absolute T-randomness would seem to be a satisfactory approximation of physical randomness (true randomness) or rather true randomness seems to be an insufficient approximation of T-randomness due to the fact that the limit assertions obtained on the ground of the T-randomness are stronger than the usual laws of large numbers. The problem, however, lies in the fact that because of non-recursivity of the function $K_U(\mathbf{x}/S)$ no algorithm exists which would generate a sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ of sequences with the properties requested above. Even in case an external oracle were able to offer such a sequence, we would not be able to verify algorithmically, that it is a sequence with the demanded properties. Our aim, in what follows, is to weaken the demands inposed to the sequence $\mathbf{x}_1, \mathbf{x}_2, \ldots$ in such a way that such a sequence were, at least in the theoretical sense, effectively constructible. We shall try, meanwhile, to preserve certain continuity with the function $K_{U(A)}(\mathbf{x}/S)$ in the sense that the case based on absolute algorithmic complexity were approchable to as small distance as given a priori, supposing that the time and space complexity of the corresponding computational and decision procedures increases.

Probably the most intuitive idea is to abandon some idealizations which distinguish an abstract universal Turing machine from an actual, technically realizable computer, as a theoretical counterpart of which universal Turing machine was conceived. From the one side, these idealizations enable to abstract from the technical details and parameters of an actual computer, from the other side, however, these idealizations imply the potential non-effectivity of some operations on a universal Turing machine. Namely, we shall abandon the assumption that the machine has at its disposal an infinite tape and that it is allowed to make an unlimited number of operations during a single computation. Hence, we shall suppose that there exists an external oracle $\mathcal{O}$, which watches the work of the universal Turing machine U over an input sequence and which stops the machine if either (a) the number of steps, i.e. the number of applications of not necessarily different operations, exceeds an a priori given $n \in \mathcal{N}$, or (b) if the computation needs more than $m \in \mathcal{N}$ boxes of the tape, not including the boxes occupied by the input sequence at the beginning of the computation, i.e. if the computation needs more than m boxes which were empty at the beginning; again, m is given a priori. If the machine stops before the intervention of the oracle, the oracle does not intervene at all and the work as well as the result of the universal Turing machine does not differ, in this case, from the ideal case as investigated above. By $\mathrm{O}(n, m)$ we shall denote the oracle $\mathcal{O}$ with parameters n, m; the expression $\mathcal{O}_U(n, m ... \mathbf{x}) = 1$ $(=0$, resp.) means that the universal Turing machine is (is not, resp.) stopped by an external intervention of the oracle $\mathcal{O}(n, m)$ when working over an input sequence $\mathbf{x} \in A^* \cup A^\infty$. If $p, S, \mathbf{x} \in A^*$, $n, m \in \mathcal{N} = \{0, 1, 2, \ldots\}$, we write $U(p, S; \langle n, m \rangle) = \mathbf{x}$ as an abbreviation of the conjunction $(U(p, S) = \mathbf{x})\ \&\ (\mathcal{O}_U(n, m, S * p) = 0)$. So it means, that using the input sequence $S * p$ the machine U constructs the sequence $\mathbf{x}$ and needs not more than n applications and not more than m boxes which were empty at the beginning of the computation.

Definition 3. Let $\mathcal{O}$ be an oracle, let p, S, $\mathbf{x}$ be as in Definition 1, let $n, m \in \mathcal{N}$. The *relative algorithmic complexity* $K^*_{U(A)}(\mathbf{x}/S, \langle n, m \rangle)$ *of the sequence* $\mathbf{x}$ *under the condition* S *and with respect to the oracle* $\mathcal{O}(n, m)$ is defined by the length of the shortest program which makes, joined with S, the universal Turing machine U to generate $\mathbf{x}$ without the oracle $\mathcal{O}(n, m)$ intervention, i.e., using at most n steps and at most m boxes not containing the program and S. In symbols,

$$(8) \quad K^*_{U(A)}(\mathbf{x}/S, \langle n, m \rangle) = \min \{l : l \in \mathcal{N}, l = l(p), p \in A^*, U(p, S; \langle n, m \rangle) = \mathbf{x}\},$$

where $\min \{\emptyset\} = \infty$ and for $S = \Lambda$ the same convention holds as in Definition 1.

Definition 4. Let $T, m, n \in \mathcal{N}$, $\mathbf{x} \in A^*$, then the sequence $\mathbf{x}$ is called (T, n, m)-*random* (*relatively* $(T. n, m)$-*random, pseudo-random with parameters* T, n, m), if $K^*_{U(A)}(\mathbf{x}/l(\mathbf{x}); \langle n, m \rangle) \geqq l(\mathbf{x}) - T$.

In the following chapter we shall study some basic properties of the relative complexity and randomness.

3. BASIC PROPERTIES OF RELATIVE COMPLEXITY AND RANDOMNESS

As shown in the foregoing chapter, the main reasons for which we have introduced the notion of relative algorithmic complexity instead of its original absolute version consists in the fact that the function $K_{U(A)}(\mathbf{x}/S)$ is not effectively computable. So it seems to be quite natural to investigate, first of all, whether and in which measure this difficulty is overcome when introducing the relative variant. Let us define, for this sake, the notion of conditional recursiveness of a function with respect to an oracle.

Definition 5. Let $n \in \mathcal{N}$, $n > 0$, let f, g be two functionals, in general partial, defined in $\mathcal{N}^n$ and taking their values in $\mathcal{N}$. We say that the function f is *conditionally partially recursive with respect to the function* g, if f belongs to the minimal class of functions taking $\mathcal{N}^n$ into $\mathcal{N}$, containing all partially recursive functions together with the function g and closed with respect to the oprations of composition, primitive recursion and minimalization (i.e. with respect to the usual operations which define the class of partially recursive functions). If f is defined *on* $\mathcal{N}^n$, it is called *conditionally totally recursive with respect to* g (if g is a partially recursive function, the same is f).

Theorem 2. The relative algorithmic complexity $K^*_{U(A)}(\mathbf{x}/S; \langle n, m \rangle)$ is a conditionally totally recursive function with respect to the oracle $\mathcal{O}$ and takes its values in the set $\mathcal{N} \cup \{\infty\}$. (More correctly said, instead of arguments $\mathbf{x}$ and S we should use their Gödel numbers, using an appropriate one-to-one mapping between A^* and $\mathcal{N}$, instead of $\mathcal{O}$ we use the function $\mathcal{O}_U$).

Proof. The proof will be given in the constructive way, i.e., we construct an algorithm which computes the function $K^*_{U(A)}(\mathbf{x}/S; \langle n, m \rangle)$ given $\mathbf{x}$, S, n and m. The construction will be given in details in order to be useful for further deductions concerning the time and space computational complexities of the function K^*.

Let p_0 be a program of the length c_1 with this property: if the concatenation $S * p_0 * \mathbf{x}$ is written on the tape, then U erases p_0 and S and stops, leaving $\mathbf{x}$ unchanged, hence, $U(S * p_0 * \mathbf{x}) = \mathbf{x}$. To do this, the machine will need a certain number of steps, independent of $\mathbf{x}$, which can be written in the form $c_2 + l(S)$. Moreover, the machine will not need any boxes besides those occupied by S, p_0 and $\mathbf{x}$. Hence, for all $n \geqq c_2 + l(S)$, $m \geqq 0$, we have

$$(9) \qquad K^*_{U(A)}(\mathbf{x}/S; \langle n, m \rangle) \leqq l(p_0) + l(\mathbf{x}) = l(\mathbf{x}) + c_1 > \infty\,, \quad \mathbf{x} \in A^*\,.$$

Because of the fact that the oracle $\mathcal{O}(n, m)$ stops the run of the program after n steps, it suffices, in order to compute $K^*_{U(A)}(\mathbf{x}/S; \langle m, n \rangle)$ under the condition that $n \geqq c_2 + l(S)$, to exhaust all the sequences over A of the lengths $0, 1, 2, \ldots, l(\mathbf{x}) + c'$ as potential candidates to the demanded shortest program for $\mathbf{x}$. The relation (9) assures that such a search will be successful. There are

$$\left(C(A)^{l(\mathbf{x}) + c_1 + 1} - 1\right)\left(C(A) - 1\right)^{-1}$$

sequences over A with the lengths not greater than $l(\mathbf{x}) + c_1$. Using each of these sequences as potential candidate for the shortest one, the machine will not make more than n steps, as it would be stopped by the oracle in the opposite case. Hence, when $n \geqq c_2 + l(S)$, the universal Turing machine needs at most $n\left(C(A)^{l(\mathbf{x}) + c_1 + 1} - 1\right)\left(C(A) - 1\right)^{-1}$ steps in order to compute $K^*_{U(A)}(\mathbf{x}/S; \langle n, m \rangle)$.

Now, let $n < c_2 + l(S)$, let $p \in A^*$ be such that $l(p) > l(\mathbf{x}) + n$. Then $U(p, S; \langle n, m \rangle) \neq \mathbf{x}$, as $l(p * S) > l(\mathbf{x}) + n$, hence, $\mathbf{x}$ differs from $S * p$ in more than n places and no procedure, neither the simple erasing, can give $\mathbf{x}$ from $S * p$ in n steps. Hence, it suffices to overlook exhaustively the sequences $p \in A^*$ with $l(p) \leqq l(\mathbf{x}) + n < l(\mathbf{x}) + l(S) + c_2$. If there is one p which gives, together with S, the sequence $\mathbf{x}$ without an intervention of the oracle $\mathcal{O}(n, m)$, than the length of the shortest p with this property defines $K^*_{U(A)}(\mathbf{x}/S; \langle n, m \rangle)$. If there is no p with this property and such that $l(p) < l(\mathbf{x}) + l(S) + c_2$, we may be sure that $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) = \infty$. Using the same way of reasoning as above we can see that the number of steps necessary to compute $K^*_{U(A)}(\mathbf{x}/S, \langle n, m\rangle)$ is case when $n > c_2 + l(S)$ does not exceed

$$(C(A)^{l(\mathbf{x})+l(S)+c_2} - 1)(C(A) - 1)^{-1}.$$

The condition $n > c_2 + l(S)$ is recursively decidable, and from this fact the conditional recursivity of the function K^* with respect to the oracle $\mathcal{O}(n, m)$ follows. □

Theorem 3. The time complexity tc and the space complexity sc of the computation of the function $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle)$ given the oracle $\mathcal{O}$ satisfy the following relations:

(a) There exists a constant $K_1 \in \mathcal{N}$ such that for all $\mathbf{x}$, $S \in A^*$ and for all $n, m \in \mathcal{N}$,

$$tc(K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle)) \leqq n(C(A)^{l(\mathbf{x})+K_1+l(S)} - 1)(C(A) - 1)^{-1}. \tag{10}$$

(b) There exists a constant $K_2 \in \mathcal{N}$ such that, for all $\mathbf{x}$, $S \in A^*$ and for all $n, m \in \mathcal{N}$,

$$sc(K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle)) \leqq m + l(\mathbf{x}) + l(S) + K_2. \tag{11}$$

Proof. The assertion (a) has been proved, in fact, during the proof of Theorem 2, the only which rests is to set $K_1 = \max(c_1, c_2)$. The longest programs taken into consideration when $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle)$ computed are of the length $l(\mathbf{x}) + c_1$ (if $n \geqq c_2 + l(S)$), or of the length $c_2 + l(S) - 1$ (if $n > c_2 + l(S)$). Hence, setting $K_3 = K_1 = \max(c_1, c_2)$ we obtain, that (11) holds. □

Let us recall the fact that the assertion (9) in the proof of Theorem 2 is nothing else than a relativized version of the relation $K_{U(A)}(\mathbf{x}/S) \leqq l(x) + \text{const}$, which has been proved in Theorem 1 for the case of the absolute algorithmic complexity. The following three theorems show that, and in which sense, $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle)$ plays the role of a monotonneous approximation for $K_{U(A)}(\mathbf{x}/S)$.

Theorem 4. For all $n, n', m, m' \in \mathcal{N}$, $n' \geqq n$, $m' \geqq m$, and for all $\mathbf{x}$, $S \in A^*$ the following holds:

$$K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) \geqq K^*_{U(A)}(\mathbf{x}/S; \langle n', m'\rangle). \tag{12}$$

Proof. If $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle = \infty$, the assertion is trivial. If $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) = l < \infty$, then there exists $p \in A^*$ such that $l(p) = l$, and $U(p, S, \langle n, m\rangle) = \mathbf{x}$. But in such a case also $U(p, S, \langle n', m'\rangle) = \mathbf{x}$, so $p \in \{p' : U(p', S, \langle n', m'\rangle) = \mathbf{x}\}$, hence, $l \geqq \min\{l' : l' = l(p), U(p, S, \langle n', m'\rangle) = \mathbf{x}\} = K^*_{U(A)}(\mathbf{x}/S; \langle n', m'\rangle)$. □

Theorem 5. For all $n, m \in \mathcal{N}$ and for all $\mathbf{x}$, $S \in A^*$ the following holds:

$$K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) \geqq K_{U(A)}(\mathbf{x}/S), \tag{13}$$

so that we could also write that $K_{U(A)}(\mathbf{x}/S) = K^*_{U(A)}(\mathbf{x}/S; \langle\infty, \infty\rangle)$.

Proof. As can be easily seen,

$$\{l : l = l(p), U(p, S) = \mathbf{x}\} = \bigcup_{\langle n,m\rangle \in N \times N} \{l : l(p), U(p, S; \langle n, m\rangle) = \mathbf{x}\}, \tag{14}$$

so

$$K_{U(A)}(\mathbf{x}/S) = \min\{l : l = l(p),\ U(p, S) = \mathbf{x}\} \leqq \min\{l : l = l(p),\ U(p, S, \langle n, m\rangle) = $$
$$= \mathbf{x}\} = K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) \quad \text{for all} \quad n, m \in \mathcal{N}\,. \qquad \square$$

Theorem 6. There exist $n, m \in \mathcal{N}$ such that, for all $n', m' \in \mathcal{N}$, $n' \geqq n$, $m' \geqq m$, the following relation holds:

$$(15) \qquad K_{U(A)}(\mathbf{x}/S) = K^*_{U(A)}(\mathbf{x}/S; \langle n', m'\rangle)\,.$$

Proof. If $K_{U(A)}(\mathbf{x}/S) = \infty$, and this possibility can occur if $\mathbf{x} \in A^\infty$, then clearly $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) = \infty$ for all $n, m \in \mathcal{N}$. If $K_{U(A)}(\mathbf{x}/S) < \infty$, then there is $p \in A^*$ such that $U(p, S) = \mathbf{x}$. Let n be the number of steps performed by the machine U when computing $\mathbf{x}$ from given sequences p and S, let m be the number of boxes on the tape which were used during this computation and which were not occupied by S or p in the initial state. Then $U(p, S; \langle n, m\rangle) = U(p, S) = \mathbf{x}$, hence $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle) = K_{U(A)}(\mathbf{x}/S)$; according to Theorems 4 and 5 this must hold for all $n' \geqq n$, $m' \geqq m$ as well. □

Now, we shall introduce some auxiliary notions in order to be able to state the basic assertion dealing with the possibility to use absolute T-random sequences as pseudo-random inputs. Then we shall formulate and prove an analogy of this basic assertion for the case of the relative T-random sequences.

Let $\mathbf{x} = x_1 x_2 \ldots x_n \in A^n$, let $m \in \mathcal{N}$, $m > 0$. Set

$$(16) \qquad B(m, \mathbf{x}) = \{\{x_1 \ldots x_m\}, \{x_{m+1} \ldots x_{2m}\}, \ldots, \{x_{(k-1)m+1} \ldots x_{km}\}\}\,,$$

where $km \leqq n < (k + 1)\,m$, so $B(m, \mathbf{x}) \in (A^m)^k \subset (A^m)^*$. In other words $B(m, \mathbf{x})$ is a word or string over a new alphabet A^m, obtained by grouping the letters in $\mathbf{x}$ into blocks of the length m (and by neglecting the last $n - km$ letters, if n is not divisible by m). The property of the absolute T-randomness can be defined, using the universal Turing machine U, also for sequences from $(A^m)^*$, as we may define $K_{U(A^m)}(\mathbf{x}/S)$ by $K_{U(A^m)}(B(m, \mathbf{x})/B(m, S))$. As can be shown, the absolute T-randomness is, in a sense, invariant with respect to the replacement of A by A^m, i.e. $\mathbf{x}$ and S by $B(m, \mathbf{x})$ and $B(m, S)$.

Theorem 7. There exists $c_1 \in \mathcal{N}$ such that, for all absolutely T-random $\mathbf{x} \in A^*$ and for all $m \in \mathcal{N}$, $m > 0$, $B(m, \mathbf{x})$ is absolutely T'-random for $T' = T + c_1$, in symbols,

$$(17) \qquad K_{U(A)}(\mathbf{x}/l(\mathbf{x})) \geqq l(\mathbf{x}) - T \Rightarrow K_{U(A^m)}(B(m, \mathbf{x})/l_{A^m}(B(m, \mathbf{x}))) \geqq$$
$$\geqq l_{A^m}(\mathbf{x}) - T - c_1$$

If $l(x)$ is divisible by m, (17) holds for $c_1 = 0$.

Proof. Cf. the proof of Theorem 2 in [5].

Theorem 8. Let $\mathcal{S} = \{S_1, S_2, S_3, \ldots\}$ be an infinite sequence of sequences from A^* such that, for all $i \in \mathcal{N}$, $l(S_i) = i$ and S_i is absolutely T-random, i.e. $K_{U(A)}(S_i/i) \geqq$ $\geqq i - T$. Let $fr^*(a, S_i)$ denote, for $a \in A$, the total number of occurrences of the letter a in S_i, set $fr(a, S_i) = i^{-1} fr^*(a, S_i)$. Then, for all $a \in A$, $\lim\limits_{i \to \infty} fr(a, S_i)$ exists and, moreover,

$$\lim_{i \to \infty} fr(a, S_i) = (card\,(A))^{-1} \,. \tag{18}$$

Proof. In spite of the fact that the proof is given in [5], we repeat it here as well, in a modified form, because of the fact that it contains a construction to which we shall refer several times in the rest of this paper. Write $c = C(A)$ and suppose, in order to arrive at a contradiction, that $\lim\limits_{n \to \infty} fr(a, S_n)$ either does not exist, or its equals to a value $c' \neq c^{-1}$. In the latter case $a \in A$ may be chosen in such a way that $c' < c^{-1}$, or if the relation $\lim\limits_{n \to \infty} fr(a, S_n) = c'_a \geqq c^{-1}$ held for all $a \in A$, with sharp inequality holding for at least one $a \in A$, then we would obtain $\sum\limits_{a \in A} \lim\limits_{n \to \infty} fr(a, S_n) > 1$ and this is not possible. So we may assume that there exist $a \in A$ and $\varepsilon > 0$ such that $0 \leqq fr(a, S_n) < c^{-1} - \varepsilon$ for infinite number of n's from $\mathcal{N}$. Let us fix an $a \in A$, $\varepsilon < c^{-1}$ and one S_n for which this relation holds.

Setting $i = fr^*(a, S_n)$, we obtain $i < c^{-1}n - \varepsilon n$. The sequence S_n can be described by giving these two objects:

(1) a string of the length $n - i$ over the alphabet $A - \{a\}$, it is the element of the set $(A - \{a\})^{n-i}$ which results when all occurrences of a in S_n are erased.

(2) an i-tuple of natural numbers, not exceeding n and giving the indices of the places in S_n where the occurrence of a are situated.

A simple fixed program, the length of which will be denoted by k, constructs then S_n, given the two objects above. There are $(c - 1)^{n-i}$ words of the length $n - i$ over the alphabet $A - \{a\}$, hence, a word of the length $\mathrm{Int}\,(\log_c (c - 1)^{n-i}) + 1$ over A suffices in order to encode the original word of the length $n - i$ over $A - \{a\}$. There are $\binom{n}{i}$ different i-tuples of different positive integers not exceeding n, hence, a word of the length $\mathrm{Int}\left(\log_c \binom{n}{i}\right) + 1$ over A suffices in order to encode such an i-tuple. Combining these results we obtain

$$\begin{aligned} (19) \qquad K_{U(A)}(S_n/n) &\leqq \mathrm{Int}\,(\log_c (c - 1)^{n-i}) + 1 + \mathrm{Int}\left(\log_c \binom{n}{i}\right) + 1 + k \leqq \\ &\leqq \log_c (c - 1)^{n-i} + \log_c \binom{n}{i} + k' \,, \quad k' = k + 2 \,. \end{aligned}$$

In order to close the proof by contradiction it is sufficient, now, to show that the right side of the inequality (19) is smaller than $n - T$ for $i < c^{-1}n - \varepsilon n$ and for n sufficiently large. The relation

$$\log_c (c-1)^{n-i} + \log_c \binom{n}{i} + k' < n - T \tag{20}$$

holds if, and only if:

$$(c-1)^{n-i} \binom{n}{i} c^{k'} < c^n \,.\, c^{-T} \tag{21}$$

and this relation holds, for n sufficiently large, if

$$\lim_{n\to\infty} c^{-n}(c-1)^{n-i} \binom{n}{i} = 0\,. \tag{22}$$

The well-known criterion sounds, that (22) holds, if the ratio of the two subsequent members of the series could be majorized by a value smaller than 1, at least for n large enough. An easy computation yields

$$\frac{(c-1)^{n+1-i}\binom{n+1}{i} c^{-n-1}}{(c-1)^{n-i}\binom{n}{i} c^{-n}} = \left(\frac{c-1}{c}\right) \frac{(n+1)\, n(n-1) \ldots (n-i+2)}{n(n-1)(n-2)\ldots(n-i+1)} =$$

$$= \left(\frac{c-1}{c}\right)\left(1 + \frac{i}{n-i+1}\right).$$

As $i < c^{-1}n - \varepsilon n$ and $\varepsilon < c^{-1}$, we obtain

$$\left(\frac{c-1}{c}\right)\left(1 + \frac{i}{n-i+1}\right) < \left(\frac{c-1}{c}\right)\left(1 + \frac{c^{-1}n - \varepsilon n}{n - c^{-1}n + \varepsilon n + 1}\right) =$$

$$= \left(\frac{c-1}{c}\right)\left(\frac{n+1}{n - c^{-1}n + \varepsilon n + 1}\right) = \left(1 + \frac{c\varepsilon n + 1}{(c-1)(n+1)}\right)^{-1} =$$

$$= \left(1 + \frac{c\varepsilon}{c-1} + \frac{1 - c\varepsilon}{(c+1)(n+1)}\right)^{-1} < \frac{1}{1+\varepsilon} < 1\,.$$

This inequality completes the proof. □

Theorem 9. Let the notations and conditions of Theorem 8 hold, let $m \in \mathcal{N}$, $m > 0$, be given, then for all $\alpha \in A^m$, $\lim_{n\to\infty} fr(\beta, B(m, S_n))$ exists and, moreover,

$$\lim_{n\to\infty} fr(\alpha, B(m, S_n)) = (card\ A)^{-m}\,. \tag{23}$$

Proof. The assertion immediately follows from Theorems 7 and 8. Theorem 7 states that each S_n, which satisfies the conditions of Theorem 8, is an absolutely T'-random sequence over the alphabet A^m and for $T' = T + const.$ Hence, Theorem 9 follows when Theorem 8 applied to the sequence $\{B(m, S_1), B(m, S_2), \ldots\}$ of sequences over the alphabet A^m. □

Now, let us re-formulate and prove the Theorems 8 and 9 to the case of the relative complexity. Let us postpone a discussion concerning the meaning and the importance of such an assertion till the end of its proof.

Theorem 10. Let $\mathscr{S} = \{S_1, S_2, \ldots\}$ be a sequence of sequences from A^* such that $l(S_i) = i$. Then there exist functions $f, g : \mathscr{N} \to \mathscr{N}$ with the following property: if each S_i is relatively $\langle T, n', m' \rangle$-random, i.e. if

$$K^*_{U(A)}(S_i/i; \langle n', m' \rangle) \geqq i - T, \tag{24}$$

for some $n' \geqq f(i)$, $m' \geqq g(i)$, then for each $m \in \mathscr{N}$, $m > 0$ and each $\alpha \in A$, $\lim_{i \to \infty} fr(\alpha, B(m, S_i))$ exists, moreover,

$$\lim_{i \to \infty} fr(\alpha, B(m, S_i)) = (card\, A)^{-m}. \tag{25}$$

Proof. Let $i \in \mathscr{N}$, denote by $p_i \in A^*$ the shortest program which generates S_i given i and using universal Turing machine U (p_i is one of such programs supposing there are more of the same minimal length). So it holds

$$U(p_i, i) = S_i, \; l(p_i) = K_{U(A)}(S_i/i). \tag{26}$$

Denote by $f(i)$ the number of steps performed by the machine U when generating S_i from p_i and i, denote by $g(i)$ the number of boxes on the tape used during this computation but not occupied by p_i or i at the initial state. When defining f and g in this way and when applying Theorem 4, we obtain, that for each $n' \geqq f(i)$, $m' \geqq g(i)$ the relation

$$K^*_{U(A)}(S_i/i; \langle n', m' \rangle) = K_{U(A)}(S_i/i) \tag{27}$$

holds, hence, the condition (24) is equivalent to the condition of the absolute T-randomness of the sequence S_i, $i \in \mathscr{N}$. In this way, the assumptions of Theorems 8 and 9 are satisfied and it is why also their assertions hold (here we present just the generalized variant of Theorem 9, the assertion equivalent to Theorem 8 follows as a special case when setting $m = 1$). □

The consequences deduced in Theorems 9 and 10, are, formally spoken, the same and they are also intuitively acceptable, as the convergence of relative frequencies of particular letters or strings of letters to the uniform distribution is considered to be a necessary condition to admit the sequence of letters in question as a good approximation of the realization of a true random sequence of independent samples from the uniform (equiprobable) distribution over the set A. As far as the premises are considered, both of them are non-effective in the sense that their validity cannot be algorithmically checked. Neither can be algoritmically generated a sequence S satisfying these premises. There is, however, a difference between the condition of absolute T-randomness and the condition (24). When knowing the functions f and g and supposing that they were recursive, the condition (24) would be algoritmically decidable for all $i \in \mathcal{N}$. Moreover, it would be possible to generate arbitrary initial segment of the sequence $\mathcal{S}$ satisfying the conditions of Theorem 10 (in the worst case, by a blind exhaustive searching in the sets A^n with n increasing and by testing whether they are or are not $\langle f(i), g(i), T\rangle$-random where i denotes the length of the investigated sequence). In other words, it is just the fact that the functions f and g are not constructively defined which causes the condition (24) not to be effectively decidable, as we have obtained our definitions for f and g by an argumentation which is substantially based on the axiom of choice. In fact, the relation (27) has been obtained by the process of skolemization applied to the formula

$$(28) \qquad (\forall i)(\exists n)(\exists m)(K^*_{U(A)}(S_i/i; \langle n, m\rangle) = K_{U(A)}(S_i/i)),$$

even if this assertion has not been explicitly mentioned; here f ang g are Skolem functions corresponding to the existential quantifiers in (28). It follows, that if it were possible to define f and g in an effective and constructive way and if they were recursive, it would be possible to verify effectively the validity of premises of Theorem 10. Hence, it would be able to construct an arbitrary initial segment of such a sequence $\mathcal{S} = \{S_1, S_2, \ldots\} \in (A^*)^\infty$, that using $\mathcal{S}$ instead of true-random sequences would assure the convergence of relative frequencies of letters and their strings to the uniform distribution over the corresponding Cartesian product of A. Moreover, the obtained convergence would be that in the usual mathematical sense, i.e. stronger convergence than that offered by statistical laws of large numbers. In what follows we shall try to find, in a constructive way, recursive functions f and g satisfying the demands of Theorem 10.

Consider a sequence $p \in A^*$ which, taken as a program, generates a word $\mathbf{w}' \in A^*$, using the quadruple $\langle i, a, \mathbf{w}, n\rangle$ which satisfies certain conditions, and proceeding as follows. Formally, we can write $U(p * i * a * \mathbf{w} * n) = \mathbf{w}'$ and U can be also considered as a partial mapping which takes the corresponding Cartesian product into A^*.

(1) The program p verifies, first of all, whether

(a) i is the expression for a positive integer, written in the alphabet A,

(b) a is a letter of the alphabet A, i.e., $a \in A$,

(c) $\mathbf{w} \in \bigcup_{j=0}^{i} (A - \{a\})^j$, i.e. $\mathbf{w}$ is a word of the length at most i over the alphabet A which does not contain the letter a,

(d) n is a positive integer which satisfies the inequality $n \leqq \binom{i}{l(\mathbf{w})}$ and which is written in the alphabet A.

All these conditions can be effectively verified, and if at least one of them is not satisfied, then $U(p * i * a * \mathbf{w} * n)$ is not defined. Let the conditions (a)–(d) hold, then the program p proceeds in this way:

(2) It compares i and $l(\mathbf{w})$, if $i = l(\mathbf{w})$, then $U(p * i * a * \mathbf{w} * n) = \mathbf{w}$ and the work of the program p terminates.

(3) Let $l(\mathbf{w}) < i$, then there are $\binom{i}{i - l(\mathbf{w})} = \binom{i}{l(\mathbf{w})}$ possible $(i - l(\mathbf{w}))$-tuples $\langle k_1, k_2, \dots, k_{i-l(\mathbf{w})}\rangle$ of positive integers without repetitions. There exists a uniquely defined ordering of these $(i - l(\mathbf{w}))$-tuples with respect to a supposed and uniquely defined alphabetical ordering of the letters from A. Program p generates (or finds) the nth of these $(i - l(\mathbf{w}))$-tuples, say $\langle k_1, k_2, \dots, k_{i-l(\mathbf{w})}\rangle$, this step can be effectively performed.

(4) Program p generates a sequence $\mathbf{w}' = \langle w'_1, w'_2, \dots, w'_i\rangle$ in this way: if $j \in \{k_1, k_2, \dots, k_{i-l(\mathbf{w})}\}$, then $w'_j = a$, the other $l(\mathbf{w})$ positions in $\mathbf{w}'$ are occupied by the symbols from $\mathbf{w}$ in the same order as in $\mathbf{w}$. In other words, program p interpolates the occurrences of the letter a in the original word $\mathbf{w}$ in such a way that the indices of the occurrences of the letter a in the resulting word $\mathbf{w}'$, $l(\mathbf{w}') = i$, were just $k_1, k_2, \dots, k_{i-l(\mathbf{w})}$.

(5) Program p terminates its work, so $U(p * i * a * \mathbf{w} * n) = \mathbf{w}'$. Clearly, p is the program the existence of which assures the validity of Theorem 8.

Let us denote, now, by $tc(i, a, \mathbf{w}, n)$ the number of steps performed by the machine U when generating the word $\mathbf{w}' = U(p * i * a * \mathbf{w} * n)$ and by $sc(i, a, \mathbf{w}, n)$ the number of boxes on the tape used during this computation and not occupied by the concatenation $p * i * a * \mathbf{w} * n$ in the initial state. Set $tc(i, a, \mathbf{w}, n) = sc(i, a, \mathbf{w}, n) = 0$, if $U(p * i * a * \mathbf{w} * n)$ is not defined. From the fact that the conditions for $\mathbf{w}$ and n, under which $U(p * i * a * \mathbf{w} * n)$ is defined, are recursively decidable, and from the way in which program p proceeds, it follows immediately, that tc and sc are totally recursive functions of their arguments and they could be specified in more details when given the alphabet A and the universal Turing machine U. Set, now, for $i \in \mathcal{N}$,

$$F(i) = \max_{a \in A} \max_{\mathbf{w} \in W(a,i)} \max_{n \leqq K(i,\mathbf{w})} \{tc(i, a, \mathbf{w}, n)\}, \tag{29}$$

$$G(i) = \max_{a \in A} \max_{\mathbf{w} \in W(a,i)} \max_{n \leqq K(i,\mathbf{w})} \{sc(i, a, \mathbf{w}, n)\}, \tag{30}$$

where $W(a, i) = \bigcup_{j=0}^{i} (A - \{a\})^j$, $K(i, \mathbf{w}) = \binom{i}{l(\mathbf{w})}$. For each $i \in \mathcal{N}$ the sets $W(a, i)$ and $\{n : n \leqq K(i, \mathbf{w})\}$ are, clearly, finite because of the finiteness of the set A. Hence, even F and G are totally recursive functions defined on the set of natural numbers and taking it into itself. Let us try to show, now, that the just generated functions F and G can be used as constructive variants of the non-effectively defined Skolem functions f and g occurring in the proof of Theorem 10.

Theorem 11. Let $\mathcal{S} = \{S_1, S_2, \dots\}$ be a sequence of sequences from A^* such that $l(S_i) = i$ and each S_i is relatively $\langle T, F(i), G(i)\rangle$-random, hence,

$$K^*_{U(A)}(S_i/i; \langle F(i), G(i)\rangle) \geqq i - T. \tag{31}$$

Then for all $m \in \mathscr{N}$, $m > 0$ and all $\alpha \in A^m$, $\lim_{i \to \infty} fr(\alpha, B(m, S_i))$ exists, moreover,

$$\lim_{i \to \infty} fr(\alpha, B(m, S_i)) = (card\,(A))^{-m}\,. \tag{32}$$

Proof. Let us investigate only the case $m = 1$, as the generalization to $m > 1$ is the same as in the case of Theorem 9. Let for some $a' \in A$ the relation $\lim_{i \to \infty} fr(a', S_i) = c^{-1}$ does not hold, where $c = card\,(A)$. Then there is $\varepsilon' > 0$ such that for infinitely many values $i \in \mathscr{N}$ the inequality $|fr(a', S_i) - c^{-1}| > \varepsilon'$ holds. As $\sum_{a \in A} fr(a, S_i) = 1$ for all $i \in \mathscr{N}$, then there must exist $a \in A$ and $\varepsilon > 0$ such that $fr(a, S_i) < c^{-1} - \varepsilon$ for infinitely many i's from $\mathscr{N}$. For each $i \in \mathscr{N}$ there exist $\mathbf{w}_i \in W(a, i)$ and $n_i \leqq K(i, \mathbf{w}_i)$ such that $U(p * i * a * \mathbf{w}_i * n_i) = S_i$; $\mathbf{w}_i$ is nothing else than S_i with all occurrences of $a \in A$ erased and n_i is the number of the $(i - l(\mathbf{w}))$-tuple of erased indices with respect to the supposed alphabetical ordering of such $(i - l(\mathbf{w}))$-tuples. According to the way, how the functions F and G are defined, we have also that $U(p * i * a * \mathbf{w}_1 * n_i;\ \langle F(i), G(i)\rangle) = S_i$, as $F(i)$ abd $G(i)$ are the upper bounds of the time and space complexity of the program p over all $a \in A$, $\mathbf{w}_i \in W(a, i)$, and $n_i \leqq K(i, \mathbf{w}_i)$. Hence,

$$K^*_{U(A)}(S_i/i;\ \langle F(i), G(i)\rangle) \leqq l(p * a * \mathbf{w}_i * n_i) \tag{33}$$

for infinitely many i's from $\mathscr{N}$ and using the same argumentation as in the proof of Theorem 8, the right-hand side in (33) is majorized by i-T for all $i \geqq i_0$, if i_0 is appropriately chosen. This contradicts the assumption (31), so there exists, for each $\varepsilon > 0$, such an $i_1 \in \mathscr{N}$, that $|fr(a, S_i) - c^{-1}| > \varepsilon$ holds for all $a \in A$, $i \geqq i_1$, hence, $\lim_{i \to \infty} fr(a, S_i) = c^{-1}$. As already said, the generalization to $m > 1$ is straightforward, of course, the functions F and G must be replaced by functions F_m and G_m corresponding to the alphabet A^m. □

Before closing this chapter let us mention one alternative definition of relative algorithmical complexity which seems to be more general and appropriate, together with introducing the reasons for which we have preferred, nevertheless, the definition presented above. In the accepted approach we interpreted the expression $U(p, S;\ \langle n, m\rangle) = \mathbf{x}$ in such a way that the universal Turing machine U stops itself its work over $S * p$, before the oracle's intervention forces it to do so. It would be possible to extend the sense of the expression $U(p, S;\ \langle n, m\rangle) = \mathbf{x}$ also to the case when $\mathbf{x}$ is written on the tape in the moment when the oracle $\mathcal{O}(n, m)$ stops the work of the machine U over the input string $S * p$ (and when this $\mathbf{x}$ would be modified and changed by the further actions of the machine U supposing the oracle let it run). Under such an interpretation of the relation $U(p, S;\ \langle n, m\rangle) = \mathbf{x}$, however, the monotonous character of the function $K^*_{U(A)}(\mathbf{x}/S;\ \langle n, m\rangle)$ with respect to the parameters n and m would be seriously threaten; this monotonicity is explicitly stated in Theorems 4 and 5 and substantially used several times above. As an example, consider a digital ($A = \{0, 1, \ldots, 9\}$) or binary ($A = \{0, 1\}$) alphabet and take the sequence $\mathbf{x} = 00\ldots0$, $l(\mathbf{x}) = n$ (or $\mathbf{x} = 0^n$, in other way). Then there exist $n', m' \in \mathcal{N}$ such that $K^*_{U(A)}(\mathbf{x}/A; \langle n', m'\rangle) = const$, where *const* does not depend on n, on the other hand, $K^*_{U(A)}(\mathbf{x}/A; \langle n'', m''\rangle) \geqq const + \log_c n$ for $n'' \neq n'$, $m'' \neq m'$, here $c = card\,(A) = 2$ or 10. Or, if n', m' appropriately chosen, then $\mathbf{x}$ can be constructed using the simple program, independent of n, which generates an infinite sequence of zeros; it is just the oracle which stops the work after having generated the nth zero. In other cases, the instruction to stop after having generated 0^n must be incorporated into the program, so the c-adic expression for the number n must be a part of the program, and this needs at least $\log_c n$ boxes on the tape. Even if it might be interesting and useful to study in more details the mentioned above extension of the relation $U(p, S; \langle n, m\rangle) = \mathbf{x}$, for the sake of simplicity and monotonically conservation we have preferred the variant adopted here.

4. RELATIVE ALGORITHMICAL RANDOMNESS AND MONTE-CARLO METHODS

In its most general form the notion of the Monte-Carlo method covers each computational or decision method which takes profit of the statistical laws of large numbers in such a way that the unknown expected values (probabilities) are replaced by arithmetical average values (relative frequencies) and the risks following from such a replacement are accepted. Let $\{X_1, X_2, \ldots\}$ be a sequence of independent and identically distributed random variables with a finite expected value $\mathsf{E}X$ and finite dispersion D^2X; the random variables are defined on a probability space $\langle \Omega, \mathscr{S}, \mathsf{P}\rangle$ and take their values in the Borel line $\langle E, \mathscr{B}\rangle$, $E = (-\infty, \infty)$. Then

$$(34) \qquad \mathsf{P}(\{\omega : \omega \in \Omega, \lim_{n\to\infty} n^{-1} \sum_{i=1}^{n} X_i(\omega) = \mathsf{E}X\}) = 1\,,$$

so the arithmetical average value tends almost surely to the expected value. Hence, it is reasonable, in a sense, to take the value $n^{-1} \sum_{i=1}^{n} X_i(\omega)$, for n large enough, as an acceptable approximation of the value $\mathsf{E}X$. The well-known Tchebyshev inequality describes in a quantitative way, in which sense the arithmetical average value approximates the expected value, giving an upper bound for the probability with which both the values differ from each other by more than an $\varepsilon > 0$. Precisely,

$$(35) \qquad \mathsf{P}(\{\omega : \omega \in \Omega, |n^{-1} \sum_{i=1}^{n} X_i(\omega) - \mathsf{E}X| > \varepsilon\}) < \mathsf{D}^2X(n\varepsilon^2)^{-1}\,.$$

There exist a number of improved variants for this inequality, however, we shall not introduce them here, neither we shall investigate here the character of the convergence occurring in the laws of large numbers and the consequences for the philosophical and metodological justification of Monte-Carlo methods. As special cases we may consider the two-valued random variables X_i, for which $X_i(\omega) = 1$ with. a probability p and $X_i(\omega) = 0$ with a complementary probability $1 - p$. Clearly. these random variables can be interpreted within the framework of the Bernoulli schema, i.e. as occurences $(X_i(\omega) = 1)$ or absences $(X_i(\omega) = 0)$ of a random event in a series of independent trials. Then $n^{-1} \sum_{i=1}^{n} X_i(\omega)$ is just the relative frequency $r(A, p, n, \omega)$ of the occurrences of a random event A in n independent trials, $\mathsf{E}X = p$, $\mathsf{D}^2X = p(1 - p)$. In this notation

$$(36) \qquad P(\{\omega : \omega \in \Omega, \lim_{n \to \infty} r(A, p, n, \omega) = p\}) = 1\,,$$

$$(37) \qquad P(\{\omega : \omega \in \Omega, |r(A, p, n, \omega) - p| < \varepsilon\}) > p(1 - p)(n\varepsilon^2)^{-1} \leqq (4n\varepsilon^2)^{-1}\,,$$

as the value $p(1 - p)$ takes its maximum $\frac{1}{4}$ for $p = 1 - p = \frac{1}{2}$.

Monte-Carlo methods became attractive just with the computers coming into scene when the computations connected with large samples (i.e., for n large enough) became technically accessible, enabling to obtain estimates of high correctness and realibility. However, the limiting factor of implementation of Monte-Carlo methods on computers consisted in the fact that the used true-random generators were rather slow and expensive. This led to the idea of the so called pseudo-random numbers, i.e. sequences of numbers which are generated deterministically (as a rule, by an appropriate computer program), but which can replace the true-random generators for the Monte-Carlo methods. As said in the introductory part of this paper, from this stage of reasoning there is a direct path to the idea to use sequences of high absolute or relative algorithmic complexity to these purposes, in the introduced terms, to use absolutely or relatively T-random sequences. As already mentioned, the aspirations of T-random sequences are justifiable, even the obtained limit results are, in the case of absolutely T-random sequences, qualitatively better than in the case of the true-random ones. Even the relatively T-random sequences were proved, in the last chapter, to satisfy, supposing the time and space limitations are large enough, one of the basic condition of an independent random sample, i.e. the stability of the relative frequencies of occurrences of particular letters and strings of letters together with the convergence of these relative frequencies to the uniform distribution. Let us try to find, now, whether relatively T-random sequences can be used also in order to generate random samples necessary for Monte-Carlo methods.

Consider the following most simple model. Let $M = \{a_1, a_2, \ldots\}$ be a finite or infinite countable set, let V be a formula of an appropriate first-order predicate language with a single variable, which is interpreted as ranging over thr set M. Hence, V is a predicate which can be attributed to each element of the set M. We shall suppose, that for each $a_i \in M$ the validity or non-validity of $V(a_i)$ can be decided effectively, quickly and within low expenses. The *measure* $\mu(V, M)$ *of the property* V

in the set M is defined by

$$(38) \qquad \mu(V, M) = (card\,(M))^{-1}\, card\,\{i : i \leqq card\,(M),\, V(a_i)\}\,,$$

if $card\,(M) < \infty$, or by

$$(39) \qquad \mu(V, M) = \lim_{n \to \infty} n^{-1}\, card\,\{i : i \leqq n,\, V(a_i)\}\,,$$

supposing that $card\,(M) = \infty$ and that the introduced limit exists. Our aim is to obtain the value $\mu(V, M)$, however, we shall mostly investigate the situations where this is not immediately possible because of theoretical (infinite set M) or practical (finite, but very large set M) reasons. In such a situation the Monte-Carlo method, and the statistically based estimate which it offers, seem to be an acceptable outcome. When we are satisfied just with an appropriate approximation of the value $\mu(V, M)$, we may limit ourselves to the case when the set M is finite. If M is not finite, we may approximate the value $\mu(V, M)$, if defined. by a value $\mu(V, M_n)$, with $M_n = \{a_1, a_2, \dots \dots, a_n\} \subset M$ and with $n \in \mathcal{N}$ large enough. The difference $|\mu(V, M) - \mu(V, M_n)|$ may be done as small as demanded.

Theorem 12. Let $M = \{a_1, a_2, \dots, a_n\}$ be a finite nonempty set, let $\mu(V, M)$ be defined by (38). Let $A = \{\bar{a}_1, \bar{a}_2, \dots, \bar{a}_n\}$ be such an alphabet that each letter $\bar{a}_i$ is the number of the element $a_i \in M$. Let $\mathcal{S} = \langle S_1, S_2, \dots \rangle$ be such a sequence of finite sequences over A, that for a given $T \in \mathcal{N}$ and for each $i \in \mathcal{N}$

(a) $l(S_i) = i$

(b) $K^*_{U(A)}(S_i/i;\, \langle F(i), G(i) \rangle) \geqq i - T$,

where F and G are the functions defined by (29) and (30). Set, for each $i \in \mathcal{N}$, $S_i = x_{i1}x_{i2} \dots x_{ii}$,

$$(40) \qquad \bar{\mu}(V, S_i) = i^{-1} \,.\, card\,\{j : j \in \mathcal{N}, j \leqq i, V(f\,(x_{ij}))\}\,,$$

where f is the mapping which ascribes to each $\bar{a}_j \in A$ the element $a_j \in M$, which is labelled or enumerated by $\bar{a}_j$. Then

$$(41) \qquad \lim_{i \to \infty} \bar{\mu}(V, S_i) = \mu(V, M)\,.$$

Proof. Let $\mathbf{x} \in A^*$, $j \leqq n$. Denote by $r(\bar{a}_j, \mathbf{x})$ the number of occurrences of $\bar{a}_j$ in $\mathbf{x}$, denote by $M_V \subset M$ the subset of all elements from M which possess the property V, and denote by $A_V \subset A$ the set of letters corrrsponding to M_V. Then

$$(42) \qquad \mu(V, M) = card\,(M_V)\,(card\,(M))^{-1} = card\,(A_V)\,(card\,(A))^{-1}\,.$$

Hence,

$$(43) \qquad \lim_{i \to \infty} \bar{\mu}(V, S) = \lim_{i \to \infty} i^{-1} \sum_{\bar{a}_j \in A_V} r(\bar{a}_j, S_i) = card\,(A_V)\,(card\,(A))^{-1} = \mu(V, M)\,,$$

as $\lim_{i \to \infty} i^{-1}\, r(\bar{a}_j, S_i) = (card\,(A))^{-1}$, according to Theorem 11, for each $\bar{a}_j \in A$. □

The assumption that $card(M) = card(A)$ may seem to be, from the first sight, rather strong and limiting, because it implies the existence of a one-to-one isomorphism between M and A. As we have already showed, however, the sequence $\mathcal{S}$ of sequences can be seen not only as sequence of sequences over A, but also over a product alphabet A^n with n taken in such a way that $(card(A))^n = (card\ A^n) \geqq card\ M$. Replace the condition (b) of Theorem 12 by a new condition

$$\text{(b')} \qquad K^*_{U(A^n)}(B(n, S_i)/l_{A^n}(S_i)\ ;\ \langle F_n(i), G_n(i)\rangle) \geqq l_{A^n}(S_i) - T\,.$$

We assure the validity of Theorem 11 with respect to the alphabet A^n, clearly, $B(n, S_i)$ is the sequence S_i taken as a sequence of strings from A^n, $l_{A^n}(S_i)$ is the length of $B(n, S_i)$ with respect to A^n, and the functions F_n, G_n are defined, with respect to the alphabet A^n, in the same way as F and G were with respect to A. When $card(A^n) >$ $> card(M)$, the letters from A^n, to which no element from M corresponds are not taken into consideration, when $\bar{\mu}(V, S_i)$ computed. A more detailed analysis of the proof of Theorem 12 shows that this assertion holds even in this case.

Relatively T-random sequences may be used even in order to estimate probabilities or measures defined on infinite spaces, when appropriately using the diagonalization method. Let us demonstrate this claim in the case of some Borel measurable sets. Let us limit ourselves to subsets of the unit interval $\langle 0, 1)$, as the generalization to subsets of other finite intervals will be straighforward and follows immediately from the constructions presented below.

Let $E \subset I = \langle 0, 1)$ be a finite union of semi-open intervals, i.e. a special case of Borel set in I. Let $\mu(E)$ be its Borel measure, hence, $\mu(E)$ is the sum of the lengths of disjoint intervals the union of which is E. Denote, for $j, n \in \mathcal{N}$, $j \leqq n$, by $I(j, n)$ the semi-open interval $\langle (j-1)\,n^{-1}, jn^{-1})$, so $I = \bigcup\limits_{j=1}^{n} I(j, n)$. Set

$$(44) \qquad \mu^*(n, E) = n^{-1}\ card\ \{j : j \leqq n, I(j, n) \cap E \neq \emptyset\}\,,$$

hence, μ^* is something like an "outer measure" of the set E generated by the intervals $I(j, n)$. Clearly, for each interval $\langle a, b) \subset I$ the relation

$$(45) \qquad \lim_{n\to\infty} \mu^*(n, \langle a, b)) = b - a = \mu(\langle a, b))\,,$$

holds, so the same must hold for E as well, so $\lim\limits_{n\to\infty} \mu^*(n, E) = \mu(E)$. The value $\mu^*(n, E)$, is completely defined by the fact, which of the finite number of intervals $I(1, n)$, $I(2, n), \ldots, I(n, n)$ possess the property V, i.e. the property of having a nonempty intersection with the set E. Hence, $\mu^*(n, E)$ can be approximated by using appropriate relatively T-random sequences according to Theorem 12.

Consider a sequence $\mathcal{S}^* = \{\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \ldots\}$ of infinite sequences with the following properties: Each $\mathcal{S}_k$ is a sequence $\{S_{k1}, S_{k2}, \ldots\}$ of finite sequences of elements from A_k, i.e. of k-tuples of letters of the original alphabet A. Moreover,

$$(46) \qquad l_{A^k}(S_{ki}) = i\,, \quad \text{for all} \quad k, i \in \mathcal{N}\,, \quad \text{where} \quad l_{A^k} \ \text{is defined by the number}$$
$$\text{of } k\text{-tuples in } S_{ki}, \text{ i.e. } l(S_{ki}) = l_{A^1}(S_{ki}) = k\,.\,l_{A^k}(S_{ki})\,;$$

$$K^*_{U(A^k)}(S_{ki}/i\,;\ \langle F_k(i), G_k(i)\rangle) \geqq i - T\,, \tag{47}$$

with the function F_k, G_k being effectively constructed, with respect to the alphabet A^k, in the same way as F and G were with respect to the original alphabet A.

Let us divide, for each $k \in \mathcal{N}$, the interval $I = \langle 0, 1)$ into subintervals $I(1, c^k)$, $I(2, c^k), \ldots, I(c^k, c^k)$, with $c = card\,(A)$, so that to each letter from A^k just one interval $I(j, c^k)$ corresponds, let us denote it immediately $I(x, c^k)$ for $x \in A^k$. Let us define, given $k, i \in \mathcal{N}$, the value $\bar{\mu}(S_{ki}, E)$ in this way:

$$\bar{\mu}(S_{ki}, E) = i^{-1}\, card\,\{j : j \leqq i, I(x_{kj}, c^k) \cap E \neq \emptyset\}\,, \tag{48}$$

where $S_{ki} = \{x_{ki}, x_{k2}, \ldots, x_{ki}\}$, hence $\bar{\mu}(S_{ki}, E)$ is the relative frequency of occurrences of those elements from A^k, whose corresponding intervals of the length c^{-k} have a nonempty intersection with the tested set E. Because of the fixed one-to-one correspondence between the intervals $I(j, c^k)$ and the letters from A^k we are allowed to consider the sequence S_{ki} as a sequence of intervals $I(j, c^k)$ and we shall often do so in what follows.

Theorem 12 implies the following assertion.

Theorem 13. Let E, $\mathcal{S}^*$, $\mathcal{S}_k$, S_{ki}, $\bar{\mu}$ and μ^* satisfy the conditions introduced above, then

$$\lim_{i\to\infty} \bar{\mu}(S_{ki}, E) = \mu^*(c^k, E)\,, \tag{49}$$

for all $k, i \in \mathcal{N}$.

Proof. Denote

$$B_k(E) = \{j : j \leqq c^k, I(j, c^k) \cap E \neq \emptyset\}\,, \tag{50}$$

so that, according to (44), $\mu^*(c^k, E) = c^{-k} \,.\, card\,(B_k(E))$. In the same time we may write

$$\bar{\mu}(S_{ki}, E) = i^{-1} \sum_{l \in B_k(E)} card\,\{j : j \leqq i, x_j = l\} = \sum_{l \in B_k(E)} fr(l, S_{ki})\,, \tag{51}$$

so that, in the limit case

$$\lim_{i\to\infty} \bar{\mu}(S_{ki}, E) = \lim_{i\to\infty} \sum_{l \in B_k(E)} fr(l, S_{ki}) = \sum_{l \in B_k(E)} \lim_{i\to\infty} fr(l, S_{ki})\,, \tag{52}$$

as the number of summands is finite (at most c^k) and independent of i. According to Theorem 12, $\lim_{i\to\infty} fr(l, S_{ki}) = c^{-k}$ for each $l \in A^k$, i.e., $l \leqq c^k$. Hence,

$$\lim_{i\to\infty} \mu(S_{ki}, E) = \sum_{l \in B_k(E)} c^{-k} = c^{-k} \,.\, card\,(B_k(E)) = \mu^*(c^k, E)\,. \qquad \square \tag{53}$$

The relation (53) clearly implies that there exists, for each $k \in \mathcal{N}$, an $i(k) \in \mathcal{N}$ such that $|\bar{\mu}(S_{ki(k)}, E) - \mu^*(c^k, E)| \leqq 1/k$, and because of the fact that $\mu^*(c^k, E)$ tends to $\mu(E)$ with k increasing, also $\lim_{k\to\infty} \mu(S_{ki(k)}, E) = \mu(E)$, A constructive variant of this limit assertion can be obtained as follows.

When proving Theorem 13, we used the fact that for each $l \in A^k$, $\lim\limits_{i \to \infty} fr(l, S_{ki}) =$ $= c^{-k}$, the corresponding proof being presented in Theorem 12. Hence, for each $k \in \mathscr{N}$ there exists $i(k) \in \mathscr{N}$ such that, for all $i \geqq i(k)$ and all $l \in A^k$, the relative frequency of occurrences of l in S_{ki} differs from c^{-k} by less than $(c + 1)^{-k}$, hence, for $i \geqq i(k)$, $l \in A^k$,

$$fr(l, S_{ki}) \in (c^{-k} - (c + 1)^{-k}, c^{-k} + (c + 1)^{-k}) . \tag{54}$$

By a detailed analysis of the proof of Theorem 12 we shall find, that such an $i(k)$ can be effectively found given $k \in \mathscr{N}$. Using the sequence $\mathscr{S}^*$, we shall define a new sequence

$$\mathscr{S} = \{S_{1i(1)}, S_{2i(2)}, S_{3i(3)}, \ldots\} \tag{55}$$

of finite sequences; for all $k \in \mathscr{N}$,

$$l_{A^k}(S_{ki(k)}) = i(k) , \tag{56}$$

$$K^*_{U(A^k)}(S_{ki(k)}/i(k)\,;\,\langle F^*(k), G^*(k)\rangle) \geqq i(k) - T , \tag{57}$$

where $F^*(k) = F_k(i(k))$, $G^*(k) = G_k(i(k))$ are recursive functions according to the recursivity of the functions F_k, G_k and $i(k)$.

Theorem 14. Let E, $\mathscr{S}^*$, $\mathscr{S}$, $\bar{\mu}$ and μ^* satisfy the conditions introduced above, then

$$\lim_{k \to \infty} \bar{\mu}(S_{ki(k)}, E) = \mu(E) , \tag{58}$$

Proof. A simple computation yields that

$$0 \leqq \lim_{k \to \infty} |\bar{\mu}(S_{ki(k)}, E) - \mu(E)| \leqq \lim_{k \to \infty} (|\bar{\mu}(S_{ki(k)}, E) - \mu^*(c^k, E)| + |\mu^*(c^k, E) - \mu(E)|) =$$

$$= \lim_{k \to \infty} |\bar{\mu}(S_{ki(k)}, E) - \mu^*(c^k, E)| + \lim_{k \to \infty} |\mu^*(c^k, E) - \mu(E)| =$$

$$= \lim_{k \to \infty} |\bar{\mu}(S_{ki(k)}, E) - \mu^*(c^k, E)| .$$

Relations (52) and (54) imply that

$$\bar{\mu}(S_{ki(k)}, E) = \sum_{l \in B_k(E)} fr(l, S_{ki(k)}) < \sum_{l \in B_k(E)} (c^{-k} + (c + 1)^{-k}) =$$

$$= c^{-k} \,.\, card\,(B_k(E)) + (c + 1)^{-k}\, card\, B_k(E) \leqq \mu^*(c^k, E) + (c + 1)^{-k}\, c^k ,$$

as $B_k(E) \subset A^k$, so $card\,(B_k(E)) \leqq c^k$.

Using analogously the other side of the relation (54) we obtain, that

$$\bar{\mu}(S_{ki(k)}, E) \geqq \mu^*(c^k, E) - (c + 1)^{-k}\, c^k , \tag{59}$$

so that

$$\lim_{k \to \infty} |\bar{\mu}(S_{ki(k)}, E) - \mu(E)| = \lim_{k \to \infty} |\bar{\mu}(S_{ki(k)}, E) - \mu^*(c^k, E)| \leqq \tag{60}$$

$$\leqq \lim_{k \to \infty} 2 \,.\, (c + 1)^{-k} \,.\, c^k = 0 . \qquad \square$$

Hence, Theorem 14 improves the results obtained in [3] and [4] in the sense that it enables an arbitrarily close approximation of the unknown value $\mu(E)$ by Monte-Carlo methods, using as a pseudo-random input sequence a sequence of *relatively* T-random sequences with appropriate recursive time and space limitations, hence, there is no need of *absolutely* T-random sequences, as it claimed the premises of assertions proved in [3] and [4]. Each finite initial segment of the defined above "diagonal" sequence $\mathscr{S}$ can be effectively, i.e. recursively, constructed, or it is possible to decide effectively, given a finite sequence of sequences of letters from A, whether it is an initial segment of an appropriate sequence $\mathscr{S}$ or not. The problem how large would be the computational complexity of such a constructive or decision procedure will be postponed to one of the following chapters, as well as the question of its practical use. Let us just remember, that in the case of the absolute T-randomness the initial segments of the corresponsing sequence $\mathscr{S}$ are principally non-constructive and the predicate of being an initial segment of $\mathscr{S}$ is algorithmically undecidable (because of the general undecidability of the halting problem for the universal Turing machine cf. [1] in general, [3], [4] in the context of the problems solved here).

5. ABSOLUTE AND RELATIVE T-RANDOMNESS IN RELATION TO RECURSIVE CHOOSING RULES

It was proved, in the third chapter of this work, that when accepting "high enough" time and space limitations and with these limitations increasing quickly enough when the length of the tested sequence increases, the relative T-randomness of such sequences may serve as a sufficient condition for the stability of relative frequencies of occurrences of particular letters and strings of letters. Such a stability is one of the demands usually imposed to a sequence in order to consider it for a useful simulation of an independent and equally distributed random sample from the set A of letters. However, usually more is requested, namely the condition of stability and convergence of relative frequencies is demanded to hold not only for the sequence in question but also for some of its subsequences, at least for those of them when it is just the index of the occurrence which decides about the belonging of this occurrence of a letter to the subsequence in question. Let us investigate, in this chapter, whether, and in which measure, absolutely and relatively T-random sequences possess this property.

Let $f : N \rightarrow N$ be a total (i.e. always defined) recursive function such that, for all $i, j \in \mathscr{N}$, if $i < j$, then $f(i) < f(j)$, let us call such functions *monotonously increasing* (or simply *monotonous*, as no monotonously decreasing function in this sense exists). Hence, $f(1), f(2), \ldots$ is a monotonously increasing sequence of natural numbers. Denote, for $S = \langle x_1, x_2, \ldots, x_i \rangle \in A^i$,

$$S^f = \langle x_{f(1)}, x_{f(2)}, x_{f(3)}, \ldots, x_{f(k(i))} \rangle , \tag{61}$$

where

$$(62) \qquad k(i) = \max\{j : j \in \mathcal{N}, f(j) \leqq i\}\,.$$

By $\bar{S}^f$ we shall denote the sequence obtained from S when the occurrences $x_{f(1)}, x_{f(2)}, \ldots, x_{f(k(i))}$ are erased, so $l(S^f) + l(\bar{S}^f) = l(S)$.

Theorem 15. Let $\mathscr{S} = \{S_1, S_2, \ldots\}$ be a sequence of absolutely T-random sequences of increasing lengths over a finite, and at least binary, alphabet A. Hence, for a given $T \in \mathcal{N}$,

$$(63) \qquad l(S_i) = i\,,$$

$$(64) \qquad K_{U(A)}(S_i/i) \geqq i - T\,.$$

Let $f : \mathcal{N} \to \mathcal{N}$ be a monotonously increasing recursive function. Then there exists $T' = T'(T, f)$ such that $\mathscr{S}^f = \{S_1^f, S_2^f, \ldots\}$ is a sequence of absolutely T'-random sequences of non-decreasing lengths, i.e.

$$(65) \qquad i \leqq j \Rightarrow l(S_i^f) \leqq l(S_j^f)\,,$$

$$(66) \qquad K_{U(A)}(S_i^f/i) \geqq l(S_i^f) - T' = k(i) - T'\,.$$

Proof. Let us prove the assertion by contradiction, supposing that there are, for each $T' \in \mathcal{N}$, infinitely many i's in $\mathcal{N}$, for which

$$(67) \qquad K_{U(A)}(S_i/i) < l(S_i^f) - T'\,.$$

Now, consider three objects:

(a) A $k(i)$-tuple $\langle f(1), f(2), \ldots, f(k(i))\rangle$ of natural numbers; such a $k(i)$-tuple is completely defined, given i, by the program which defines the recursive function f. If this program is of the length $c_1(U)$, we have

$$(68) \qquad K_{U(A)}(\langle f(1), f(2), \ldots, f(k(i))\rangle/i) \leqq c_1(U)\,.$$

(b) A $k(i)$-tuple S_i^f of letters over A; as far as the complexity $K_{U(A)}(S_i^f/i)$ is concerned, we suppose, for the sake of this proof, (67) to hold.

(c) An $(i - k(i))$-tuple $\bar{S}_i^f$ of letters over A; the general assertion (cf. Theorem 1) implies that there is a constant $c_2(U)$ such that

$$(69) \qquad K_U(\bar{S}_i^f/i) < l(\bar{S}_i^f) + c_2(U) = i - k(i) + c_2(U)\,.$$

There exists a program P of the length, say, $c_3(U)$, independent of $i, S_i^f, \bar{S}_i^f, f$, which proceeds as follows:

(a) constructs the sequence of i zeros,

(b) calls a subprogram for computing the function f, which computes $k(i), f(1), f(2), \ldots, f(k(i))$.

(c) calls a subprogram which generates S_i^f,

(d) calls a subprogram which generates $\bar{S}_i^f$,

(e) goes through the zero sequence from the left to the right; if a zero is the $f(j)$th one from the left for $j = 1, 2, \ldots, k(i)$, it is replaced by the jth symbol from $S_i{}^f$ If it is not the case, the zero is replaced by the first (from the left) occurrence of a symbol in $\bar{S}_i^f$, not yet used by P. Clearly, the work of P terminates by generating S_i.

According to the assumption (67) we obtain, for each $T' \in \mathcal{N}$ and for infinitely many i's from $\mathcal{N}$, that

$$(70) \qquad K_{U(A)}(S_i/i) < c_1(U) + l(\bar{S}_i^f) + c_2(U) + l(S_i^f) - T' + c_3(U) = l(S_i) + c_4(U) - T' \,.$$

Hence, for each $T'' \in \mathcal{N}$ there are infinitely many i's in $\mathcal{N}$ such that

$$(71) \qquad K_{U(A)}(S_i/i) < l(S_i) + c_4(U) - T' = l(S_i) - T'' \,,$$

setting $T' = T'' + c_4(U)$; however, this contradicts the assumption that the sequences is $\mathcal{S}$ are absolutely T-random, hence, (66) must hold. □

Applying Theorem 9 and the other results obtained in Chapter 3 we get immediately that also the subsequences $S_1^f, S_2^f, \ldots$ of absolutely T-random sequences $S_1, S_2, \ldots$ satisfy the condition of stability and convergence to the equiprobable distribution for the relative frequencies of occurrences of letters and strings of letters, supposing f is a monotonously increasing recursive choosing rule. Because of the importance and easy interpretability of this result let us formulate it as a particular theorem.

Theorem 16. Let the notations and conditions of Theorem 15 hold, then for each $m \in \mathcal{N}$, $m > 0$, and each $\alpha \in A^m$,

$$(72) \qquad \lim_{i \to \infty} fr(\alpha, B(m, S_i)) = (card\,(A))^{-m} \,.$$

Proof. An immediate consequence of Theorems 9 and 15. □

Because of the fact that the absolutely T-randomness of subsequences $S_1^f, S_2^f, \ldots$ in the proof of Theorem 15 is demonstrated by "reduction ad absurdum" and this proof is of constructive character consisting in giving an appropriate program P, we may try to apply the same idea as in Theorem 11, i.e. to replace the demand of absolute T-randomness by the relative T-randomness with respect to time and space limitations large enough to be able to apply the program P. Denote by $tc(i, f, S_i^f, \bar{S}_i^f)$ the number of steps made by the universal Turing machine U when working over the inputs $\langle i, f, S_i^f, \bar{S}_i^f \rangle$, denote by $sc(i, f, S_i^f, \bar{S}_i^f)$ the number of used boxes on the tape, not taking into account the boxes occupied by the input data in the initial state.

There are $(card\,(A))^i$ sequences of the length i over the alphabet A. Each decomposition of S_i into S_i^f and $\bar{S}_i^f$ is uniquely determined by a subset of the set $\{1, 2, \ldots, i\}$ of integers. There are just 2^i of such subsets, hence, there are at most $2^i \,.\, (\text{card}\,(A))^i = (2\, card\,(A))^i$ pairs $\langle S_i^f, \bar{S}_i^f \rangle$. So we may define, for F and G being given by (29) and (30),

$$(73)\qquad F_0(f, i) = \max_{\langle S_i{}^f, \bar{S}_i{}^f\rangle} \{tc(i, f, S_i^f, \bar{S}_i^f)\}, F_1(f, i) = \max\{F_0(f, i), F(i)\}\,,$$

$$(74)\qquad G_0(f, i) = \max_{\langle S_i{}^f, \bar{S}_i{}^f\rangle} \{sc(i, f, S_i^f, \bar{S}_i^f),\}\; G_1(f, i) = \max\{G_0(f, i), G(i)\}\,.$$

To be able to perform the work of the program P described above within some time and space limitations, and this will be the basic idea of the proof of the following theorem, these limitations must be great enough to allow to compute S_i^f using a program for f.

Denote by $tc(\mathbf{x}, S_i^f)$ the number of steps used by our fixed universal Turing machine U in order to compute S_i^f given $\mathbf{x} \in A^*$, $i \in \mathcal{N}$. If $U(\mathbf{x}, i) \neq S_i^f$, then $tc(\mathbf{x}, S_i^f)$ is not defined. Set

$$(75)\qquad c(U) = \min\{c : c \in \mathcal{N}, K_{U(A)}(\mathbf{x}/S) \leqq l(x) + c \quad \text{for all} \quad \mathbf{x}, S \in A^*\}\,,$$

the set over which the minimum is taken is nonempty because of Theorem 1. Set, moreover,

$$(76)\qquad TC(i) = \max\{tc(\mathbf{x}, S_i^f) : \langle S_i^f, \mathbf{x}\rangle \in A^* \times A^*, l(S_i^f) \leqq i, l(\mathbf{x}) \leqq$$
$$\leqq i + c(U), U(\mathbf{x}, i) = S_i^f\}\,.$$

Again, the set over which the maximum is taken is nonempty, as follows from Theorem 1, it is also finite, as there are only finitely many pairs $\langle S_i^f, \mathbf{x}\rangle$ with $l(S_i^f) \leqq i$, $l(\mathbf{x}) \leqq i + c(U)$. Hence, $TC(i)$ is an always defined finite value, let us define $SC(i)$ in an analogous way using the space complexity $sc(\mathbf{x}, S_i^f)$, and set, finally

$$(77)\qquad F^*(f, i) = F_1^*(f, i) + TC(i)\,, \quad G^*(f, i) = G_1^*(f, i) + SC(i)\,.$$

Combining these considerations with the results already obtained we arrive at the next theorem which may be seen as a "relative" variant of Theorems 15 and 16.

Theorem 17. Let $f : \mathcal{N} \rightarrow \mathcal{N}$ be a monotonous recursive function, let the functions $F^*(f, i)$, $G^*(f, i)$ be defined by (77), let $\mathcal{S} = \{S_1, S_2, \ldots\}$ be a sequence of relatively $\langle T, F^*(f, i), G^*(f, i)\rangle$ – random sequences over a finite, and at least binary, alphabet A. Hence, for a given $T \in \mathcal{N}$,

$$(78)\qquad l(S_i) = i\,,$$

$$(79)\qquad K^*_{U(A)}(S_i/i; \langle F^*(f, i), G^*(f, i)\rangle) \geqq i - T\,.$$

Then for all $m \in \mathcal{N}$, $m > 0$, and all $\alpha \in A^m$ the relation (72) holds.

Proof. The relation (79) yields, that there exists $T' \in \mathcal{N}$ such that

$$(80)\qquad K^*_{U(A)}(S_i^f/i; \langle F^*(f, i), G^*(f, i)\rangle) \geqq l(S_i^f) - T'\,.$$

Or, if (80) were not valid, it would be possible to prove, using program P described in the proof of Theorem 15, that for each $T \in \mathcal{N}$ there are infinitely many i's in $\mathcal{N}$

such that (79) does not hold. If (72) were not valid, it would be possible to construct S_i^f using a program shorter than $l(S_i^f) - T'$, as the accepted time and space limitations permit to use programs defined in Chapter 3 and constructing S_i on the ground of the knowledge of the indices of the extremely frequented (the minimally or the maximally frequented) letter and of the knowledge of the rest of the word S_i resulting when these occurrences are erased. Hence, (72) must hold for $m = 1$, applying this result to the alphabet A^m for $m \geqq 2$ we prove (72) in all the generality. □

Clearly, the results of Chapter 3 dealing with the convergence of relative frequencies of occurrences of letters and their strings to the equiprobable (uniform) distribution can be seen as special cases of the assertions obtained here, using the most trivial identical choosing function $f(i) = i$, $i \in \mathcal{N}$. However, if we wanted to interprete the last statement in the sense that the results of Chaptes 5 can be seen as generalizations of the results of Chapter 3, it would be necessary to precise, first of all, in which sense and measure we may speak about a generalization. The limits $F^*(f, i)$ and $G^*(f, i)$ of time and space complexity are not effectively computable, as they request to know the computational complexity of all programs which are not longer than an upper bound and which generate S_i^f. Each program, which would be able to compute the values $F^*(f, i)$, $G^*(f, i)$, given f and $i \in \mathcal{N}$, would contain a sub-program deciding, for each $\mathbf{x} \in A^*$, $l(\mathbf{x}) \leqq l(S_i^f) + c(U)$, whether $U(\mathbf{x}, i) = S_i^f$ or not; if the answer were positive, the program would have to compute $tc(\mathbf{x}, S_i^f)$. The demand of a general existence of such a program contradicts the undecidability of the halting problem for the universal Turing machine, so there does not exist a general program to compute $F^*(f, i)$ and $G^*(f, i)$, given f and $i \in \mathcal{N}$. On the other hand, in the particular case of $f(i) = i$, as investigated in Chapter 3, there exist recursively computable time and space limitations $F(i)$ and $G(i)$.

To mention another important circumstance, the limits $F^*(f, i)$ and $G^*(f, i)$ substantially depend on the particular recursive choosing rule f and cannot be replaced by some functions $\bar{F}(i)$ and $\bar{G}(i)$ universal for all recursive choosing rules. When implementing some overfluous cycles not influencing the final result we may define each recursive function by a program the computational complexity of which is, given the argument $i \in \mathcal{N}$ and a function $t(i)$, greater than $t(i)$. However, it is not possible to decide effectively, in general, whether a universal Turing machine computes the same function or not, given two different programs ("the same" function in the sense of identifying the function f with the set of ordered pairs $\langle x, f(x) \rangle$ with x ranging over the domain Dom (f) of f). Hence, recursive functions cannot be effectively distinguished in other way than by identifying them with corresponding programs; from this fact the non-existence of universal limitations $\bar{F}(i)$ and $\bar{G}(i)$ immediately follows. Hence, there does not exist a sequence of *relatively* T-random sequences which would simulate one important property of "true random" infinite sequences of independent and equally distributed random samples: the convergence to equiprobable distribution for the relative frequencies of occurrences of letters and strings in *all* recursively sampled subsequences. As we have seen, in the case of *absolutely* T-random sequences such a simulation was possible. Hence, the results of this chapter show the limits of possibilities when true-random sequences are to be simulated by sequences with effectively decidable properties, and this was the aim of this work. In the next chapter we shall briefly investigate the computational complexity of constructive procedures which generate initial segments of sequences of relatively T-random sequences of letters over a given finite alphabet.

6. COMPUTATIONAL COMPLEXITY OF GENERATORS OF RELATIVELY T-RANDOM SEQUENCES

Using som: simple combinatorial reasonings we have already derived simple upper bounds for the time complexity $tc(K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle))$ and space complexity $sc(K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle))$ connected with the computation of the relative algorithmic complexity $K^*_{U(A)}(\mathbf{x}/S; \langle n, m\rangle)$, cf. Theorem 3 and the relations (10) and (11) above. Hence, due to (10), the time complexity is bounded by an exponential function of the lengths $l(\mathbf{x})$ and $l(S)$, which could be expected, because of the fact that the algorithm works on the ground of a blind exhaustive searching in the set of combinatorial objects and the cardinality of this set (of strings) increases exponentially with their lengths increasing. On the other hand, the use of a more sophisticated algorithm may be justified just on the ground of an a priori information about the particular problem. Hence, using the idea of the worst case classification we are not allowed to omit the blind exhaustive search as a possible candidate. Let us investigate, now, how rapidly increases the function $tc(K^*_{U(A)}(\mathbf{x}/l(\mathbf{x}); \langle F(l(\mathbf{x})), G(l(\mathbf{x}))\rangle)$ with $l(\mathbf{x})$ increasing in the case of effectively computable functions F and G defined by the relations (29) and (30).

Theorem 18. Let $\mathcal{S} = \{S_1, S_2, \ldots\}$ be a sequence of sequences from A^* which satisfies the conditions of Theorem 11, let the functions $F, G : \mathcal{N} \to \mathcal{N}$ be defined by (29) and (30). Then there exist constants $K_2, K_3 \in \mathcal{N}$, which depend only on the used universal Turing machine U and on the cardinality $c(A)$ of the alphabet A, such that

$$(81) \qquad tc(K^*_{U(A)}(S_i/i; \langle F(i), G(i)\rangle)) < K_2 i^2 c^i + K_3 i^2 (2c)^i .$$

Hence, the time complexity of the computation of the relative $\langle T, F(i), G(i)\rangle$-algorithmic complexity of S_i may be majorized by an exponential function of i.

Proof. When writing $K_{U(A)}(\mathbf{x}/l(\mathbf{x}))$ or $K_{U(A)}(S_i/i)$, we mean by i the description of the natural number in the alphabet A, hence, $l(i) \leqq (\log_c i) + 1$. So, using (10) we obtain

$$(82) \qquad tc(K^*_{U(A)}(S_i/i; \langle F(i), G(i)\rangle)) \leqq F(i) \,.\, (c-1)^{-1} \,.\, (c^{i+K_1+\log_c i+1} - 1) <$$
$$< F(i) \,.\, c^{K_1+1}(c-1)^{-1} \,.\, ic^i .$$

Now, we have to obtain an upper bound for $F(i)$, which would be of at most exponential character. (29) yields that

$$(83) \qquad F(i) = \max_{a \in A} \max_{\mathbf{w} \in W(a,i)} \max_{n < K(i,\mathbf{w})} \{tc(i, a, \mathbf{w}, n)\},$$

where $tc(i, a, \mathbf{w}, n)$ is the time computational complexity of the program P defined in Theorem 11. Hence, we have a letter $a \in A$, a word $\mathbf{w} \in W(a, i)$ of the length at most i over the alphabet $A - \{a\}$, i.e., $\mathbf{w}$ does not contain a, and we have a natural number $n \leqq K(i, \mathbf{w}) = \binom{i}{l(w)}$. All i-tuples of zeros and units containing just $l(\mathbf{w})$ zeros and $i - l(\mathbf{w})$ units are supposed to be alphabetically ordered in the sense that unit follows zero, hence, the first i-tuple is $00 \dots 011 \dots 1$, the last being $11 \dots \dots 100 \dots 0$. There are just $K(i, \mathbf{w})$ of such i-tuples which can be numbered using the numbers $0, 1, \dots, K(i, \mathbf{w}) - 1$ according to the mentioned ordering.

Consider an algorithm which finds, given $n < K(i, \mathbf{w})$, the corresponding i-tuple with $i - l(\mathbf{w})$ units in such a way that it constructs first n such i-tuples. Every i-tuple can be generated, using the preceding one, by a fixed number of checking of this preceeding i-tuple from the left to the right, hence, an upper bound of the time complexity of the transformation of the $(j - 1)$th i-tuple into the jth one can be given as a linear function of i, say $c_1 i$. Hence, the time complexity of our trivial algorithm for finding the n-th i-tuple with $i - l(\mathbf{w})$ units is, given i, $l(\mathbf{w})$ and $n < K(i, \mathbf{w})$, majorized by the expression $c_1 i \binom{i}{k} < c_1 i\, 2^i$.

Having a word $\mathbf{w}$ of the length at most i over the alphabet $A - \{a\}$, and hawing an i-tuple of zeros and units with just $(i - l(\mathbf{w}))$ units, we are able, going just once through this i-tuple from the left to the right, i.e., using $c_2 i$ steps for an appropriate c_2, to generate the sequence S_i as already described (units replaced by symbols from $\mathbf{w}$, zeros replaced by a's). If c_1 and c_2 depend on $a \in A$, we take their maxima values over A. Hence, setting into (83), we have

$$(84) \qquad F(i) \leqq c_2 i + c_1 i\, 2^i,$$

combining with (82), we obtain

$$(85) \qquad tc(K^*_{U(A)}(S_i/i; \langle F(i), G(i)\rangle)) < (c_2 i + c_1 i\, 2^i)(c^{K_1+1}(c-1)^{-1}\, ic^i) = \\ = K_2 i^2 c^i + K_3 i^2 (2c)^i,$$

setting

$$(86) \qquad K_2 = c_2 c^{K_1+1}(c-1)^{-1}, \quad K_3 = c_1 c^{K_1+1}(c-1)^{-1}.$$

The assertion is proved. □

Let us try to find an upper bound of the computational complexity of an algorithm which generates initial segments of the sequence $\mathcal{S} = \{S_1, S_2, \dots\}$, satisfying the demands of Theorem 11.

Theorem 19. There exists a totally recursive function $\mathscr{F} : \mathscr{N} \rightarrow A^*$ with the following properties

(a) $l(\mathscr{F}(i)) = i$ for all $i \in \mathscr{N}$,

(b) $K^*_{U(A)}(\mathscr{F}(i)/i;\ \langle F(i), G(i)\rangle) \geqq i - T$ for all $i \in \mathscr{N}$, given $T \in \mathscr{N}$, functions F and G are defined by (29) and (30).

(c) $$tc(\mathscr{F}(i)) < K_2 i^2 (2c)^i + K_3 i^2 (3c)^i ,$$

where K_2, K_3 and c are the same as in Theorem 18.

Proof. The assertion is satisfied by the trivial blind exhaustive search algorithm which, given $i \in \mathscr{N}$, takes one i-tuple over A after another, computes their relative algorithmical complexities $K^*_{U(A)}(\mathbf{x}/i, \langle F(i), G(i)\rangle)$ and stops its work when finding that this complexity is at least $i - T$. The corresponding string from A^i is then proclaimed to be $\mathscr{F}(i)$, Theorem 4 assures that there is at least one such string. So, for $i \in \mathscr{N}$ and $\mathbf{x} \in A^i$,

$$(87) \qquad tc(\mathscr{F}(i)) \leqq c^i\, tc(K^*_{U(A)}(\mathbf{x}/i;\ \langle F(i), G(i)\rangle)) < c^i(K_2 i^2 c^i + K_3 i^2 (2c)^i) = \\ = K_2 i^2 (2c)^i + K_3 i^2 (3c)^i ,$$

by a substitution into (81). □

Theorem 20. There exists a recursive function $\mathscr{F} : \mathscr{N} \rightarrow A^*$, with an exponential upper bound for the time computational complexity, such that the sequence $\mathscr{S}_{\mathscr{F}} = \{\mathscr{F}(1), \mathscr{F}(2), \ldots\}$ can be used as a pseudorandom input in the Monte-Carlo methods in the sense that it satisfies the conditions of Theorems 12, 13 and 14.

Proof. An immediate consequence of Theorems 11 and 19. □

7. CONCLUSIVE REMARKS

Let us close this paper by a very brief re-consideration of the obtained results. We have declared as our goal to investigate, whether and in which measure are preserved these good properties of sequences of high algorithmic complexity, which enable them to serve as good approximations of true-random sequences, even in case when the algorithmic complexity is defined and tested with respect to universal Turing machines with time and space limitations. We have proved that if the time and space limitations grow up "quickly enough" with the lengths of the tested sequences, the stability of relative frequencies of occurrences of letters and strings of letters as well as their convergence to the equiprobable distribution are preserved. Due to this fact, even within the appropriate time and space limitations, the usefulness of relatively pseudorandom sequences in order to estimate unknown probabilities by Monte-Carlo methods is preserved as well. Choosing appropriately, and in a way which is not algorithmizable in general, the time and space limitations, the mentioned above stability and convergence of relative frequencies may be preserved to hold also in subsequences chosen by a recursive rule, or by one of a finite set of such rules. On the other hand, because of principial reasons it is not possible to satisfy this property for *all* recursive rules, no matter which the time and space limitations may be, supposing they are finite. Let us recall that the von Mises conception of a "collective" and the related conceptions emphasize just this aspect. Hence, if the stability and convergence of the relative frequencies of occurrences of letters or strings in *all* recursively chosen subsequences in considered as a necessary condition for a sequence to be taken as random, then the notion of random sequence is principially non-effective and non-recursive and can be described and handled only within the apparatus of universal Turing machines without time and space limitations. Because of the fact that this work has been conceived as a mathematical one in its nature, we shall not investigate here the methodological and may be even philosophical consequences of the obtained results for a deeper penetration into the nature of the relations between complexity, algorithmizability and randomness; let us postpone such considerations till another study.

REFERENCES

[1] M. Davis: Computability and Unsolvability. McGraw Hill Book Company, New York—Toronto—London 1958.

[2] T. L. Fine: Theories of Probability. An Examination of Foundations. Academic Press, New York—London 1973.

[3] I. Kramosil: Monte-Carlo methods from the point of view of algorithmic complexity. In: Transactions of the Ninth Prague Conference on Information Theory, Statistical Decision Functions and Random Processes, Academia, Prague 1983, pp. 39—51.

[4] I. Kramosil: Pseudo-random Monte-Carlo methods. Ann. Soc. Math. Polon. Ser. IV. Fund. Inform. *5* (1982), 3—4, 301—312.

[5] I. Kramosil: On pseudo-random sequences over finite alphabets. Ann. Soc. Math. Polon. Ser. IV. Fund. Inform. (to appear).