Ivan Kramosil

Computational complexity of a statistical theoremhood testing procedure for propositional calculus with pseudo-random inputs

Kybernetika, Vol. 17 (1981), No. 5, 359--367

Persistent URL: http://dml.cz/dmlcz/125508

KYBERNETIKA — VOLUME 17 (1981), NUMBER 5

COMPUTATIONAL COMPLEXITY OF A STATISTICAL THEOREMHOOD TESTING PROCEDURE FOR PROPOSITIONAL CALCULUS WITH PSEUDO-RANDOM INPUTS

IVAN KRAMOSIL

In this paper we investigate a statistical verification procedure for propositional formulas under the condition that the corresponding random samples are simulated by deterministic chooses generated by a computer and by extremely long programs .The possibilities as well as the limitations of such an approach are illustrated by several assertions which are proved and briefly discussed. Supposing that the notion of Turing machine is known to the reader, the paper is almost self-explanatory.

1. A STATISTICAL THEOREMHOOD TESTING PROCEDURE FOR PROPOSITIONAL CALCULUS

In this paper we shall study the computational complexity of a statistical verification procedure for propositinal calculus with pseudo-random inputs. The computational complexity of the corresponding pseudo-random number generator is taken as a part of the total complexity of the statistical verification procedure and this total complexity is compared with computational complexity of deterministic decision procedures for propositional calculus.

Let us concern our attention to the classical propositional calculus formalized by the means of one of its usual formalizations. All the usual propositional connectives are supposed to be at our disposal, the propositional indeterminates are denoted by $p_1, p_2, \ldots$ $\mathscr{F}_n$ ($\mathscr{T}_n$, resp.) denotes the set of all formulas (theorems, i.e. tautologies, resp). in which only the indeterminates $p_1. p_2, \ldots, p_n$ may occur, $\mathscr{F} = \bigcup_{n=1}^{\infty} \mathscr{F}_n$, $\mathscr{T} = \bigcup_{n=1}^{\infty} \mathscr{T}_n$. Let $\mathscr{K}_n$ denote the set of all formulas of the form

$$\bigwedge_{i=1}^{2^n} \bigvee_{j=1}^{n} a_{ij}\,, \quad a_{ij} \in \{p_1, \neg p_1, p_2, \neg p_2, \ldots, p_n, \neg p_n\}\,, \tag{1}$$

$\neg p_i$ is the negation of p_i, indeterminates together with their negations are called *literals*. Set $\mathscr{K} = \bigcup_{n=1}^{\infty} \mathscr{K}_n$, $\mathscr{T}\mathscr{K}_n = \mathscr{T} \cap \mathscr{K}_n$. There is a well-known fact that for each formula $A \in \mathscr{T}_n$ there exists $B \in \mathscr{K}_n$ such that the equivalence $A \leftrightarrow B$ belongs to $\mathscr{T}_n$ (B is a conjunctive normal form for A). It is why we shall limit ourselves, in what follows, to $\mathscr{K}$ as the basic set of formulas.

A pair $\{a_i, a_j\}$ of literals is called *complementary*, if a_i is $\neg a_j$ or a_j is $\neg a_i$. A finite sequence of literals is called *closed*, if it contains at least one complementary pair, in the opposite case it is called *open*. As can be easily seen, a formula of the form (1) from $\mathscr{K}_n$ is a theorem (i.e. a propositional tautology), iff all sequences $\langle a_{i1}, a_{i2}, \ldots, a_{in}\rangle$, $i \leqq 2^n$, are closed. Hence, there is a simple deterministic verification procedure for formulas from $\mathscr{K}$: to take rows in (1) considered as a matrix and to check, in each row, all pairs of literals until an open row is found. The maximal number of pairs which must be, eventually, tested for complementarity is $n(n-1)\,2^{n-1}$ for formulas from $\mathscr{K}_n$, as can be easily seen.

Because of the exponential complexity of this deterministic verification procedure we may try to apply the basic idea of the so called probabilistic algorithms – i.e. to reduce the computational complexity of the verification procedure by admitting the possibility of an error under the condition that the complexity savings are "significant" and the probability of error is "acceptable", in a sense. Let $\langle b_1^k, b_2^k, \ldots\rangle$ be, for each $k = 1, 2, \ldots$, a sequence of mutually independent and equally distributed random variables defined on a probability space $\langle \Omega, \mathscr{S}, P\rangle$, taking their values in the set of positive integers and such that

$$(2) \qquad P(\{\omega : \omega \in \Omega,\ b_i^k(\omega) = j\}) = 2^{-k}, \quad j = 1, 2, \ldots, 2^k, \quad i = 1, 2, \ldots,$$
$$k = 1, 2, \ldots.$$

For $A \in \mathscr{K}_k$ (A is supposed to be of the form (1)) and a given positive integer n_0 our statistical procedure runs as follows.

We sample at random $b_1^k(\omega)$ and test the sequence $\langle a_{b_1^k(\omega),1}, a_{b_1^k(\omega),2}, \ldots, a_{b_1^k(\omega),k}\rangle$ for closure. If it is open we proclaim A to be a non-theorem and finish the procedure. Clearly, this decision is always correct. In the opposite case we sample at random $b_2^k(\omega)$ and repeat the test and decision procedure for the $b_2^k(\omega)$-th disjunction in A and so on. If also the $b_{n_0}^k(\omega)$-th disjunction is closed, we finish the procedure by proclaiming A to be a theorem. Of course, in this case the possibility of an error is not avoided. More precisely, denoting by $S(A)$ the number of closed disjunctions in a formula A of the form (1) and setting, for $A \in \mathscr{K}_k$, $R(A) = S(A) \cdot 2^{-k}$, we may easily show that with the probability $(R(A))^{n_0}$ the formula A will be proclaimed for theorem. Hence, each theorem A will be always proclaimed to be theorem (as in this case $S(A) = 2^k$ and $R(A) = 1$), for non-theorems the expression $(R(A))^{n_0}$ expresses the probability with which they may be wrongly proclaimed to be theorems.

2. PSEUDO-RANDOM SAMPLING

Define, for $A \in \mathscr{K}_n$, a binary sequence $w(A) = \langle w(j, A)\rangle_{j=1}^{2^n}$ such that $w(j, A) = 1$, if the j-th disjunction in A is open, $w(j, A) = 0$ otherwise, set $Ded\,(A) = 1$ if $A \in \mathscr{T} \cap \cap \mathscr{K}_n$, $Ded\,(A) = 0$ otherwise and denote, for $j = 0, 1$ by $fr\,(j, w(A))$ the relative frequency of j's in $w(A)$ (hence, $R(A) = fr\,(0, w(A))$, $1 - R(A) = fr\,(1, w(A))$). Clearly,

$$Ded\,(A) = 1 \quad \text{iff} \quad fr\,(1, w(A)) = 0\,, \tag{3}$$

$$Ded\,(A) = 0 \quad \text{iff} \quad fr\,(1, w(A)) > 0\,.$$

Hence, we have converted deducibility testing to the problem, whether the relative frequency of an event in a large set (sampling space) equals 0, or whether it is positive. This problem can be solved in the statistical sense and by statistical means and we have already suggested such a procedure. It is nothing else than the test of a simple hypothesis $fr\,(1, w(A)) = 0$ again the composed alternative $fr\,(1, w(A)) \geqq 2^{-n}$. The well-known Neymann-Pearson theorem proves the decision rule introduced above to be rational in the sense that it minimizes the probability of error connected with this rule (only the trivial decision rules consisting in uniform acceptation or uniform refutation of the tested formulas without any more detailed investigation may compete with the introduced non-trivial rule).

Let $u^k = \langle u(j, k)\rangle_{j=1}^{k} \in \{0, 1\}^k$, $u(j, k) = 1$ for $j = i_1 \leqq i_2 \leqq \ldots \leqq i_m$, $u(j, k) = 0$ otherwise. Define, for $A \in \mathscr{K}_n$, $k = 2^n$,

$$w(A) * u^{2^n} = \langle w(i_1, A), w(i_2, A), \ldots, w(i_m, A)\rangle\,, \tag{4}$$

$$g(A, u^{2^n}) = \chi_{\{0\}}(fr\,(1, w(A) * u^{2^n}))\,, \tag{5}$$

where $\chi_{\{0\}}$ is the characteristic function of the singleton $\{0\}$. Hence, g is a primitively recursive function of two arguments A (the Gödelian number of A, more precisely), and u^{2^n} such that

$$g(A, u^{2^n}) = 1\,, \quad \text{iff} \quad fr\,(1, w(A) * u^{2^n}) = m^{-1} \sum_{j=1}^{m} w(i_j, A) = 0\,, \tag{6}$$

$$g(A, u^{2^n}) = 0\,, \quad \text{iff} \quad fr\,(1, w(A) * u^{2^n}) > 0\,.$$

If u^{2^n} is considered as a random number generator which samples statistically independently and with the same probability 2^{-n} numbers from $\{1, 2, \ldots, 2^n\}$, then the computation of $w(A) * u^{2^n}$ can be taken as the verification of closedness for disjunctions with indices $i_1, i_2, \ldots, i_m$ and the function g can be seen as a formalization of the statistical theoremhood testing procedure as explained above.

Definition 1. A sequence $u^{2^n} = \langle u(j, 2^n)\rangle_{j=1}^{2^n} \in \{0, 1\}^{2^n}$ is called *adequate* with respect to (the decision problem on the theoremhood of) a formula $A \in \mathscr{K}_n$, if

$$(\neg\, Ded\,(A)) \Rightarrow (\exists j \leqq 2^n)\,(w(j, A) = u(j, 2^n) = 1)\,. \tag{7}$$

Hence, for $A \in \mathscr{T} \cap \mathscr{K}_n$, each $u^{2^n} \in \{0, 1\}^{2^n}$ is adequate. Denote by $\mathscr{A}(A)$ the set of all sequences from $\{0, 1\}^{2^n}$ adequate for a given $A \in \mathscr{K}_n$.

Theorem 1. For each $A \in \mathscr{K}_n$

$$(u^{2^n} \in \mathscr{A}(A)) \Rightarrow (g(A, u^{2^n}) = Ded\,(A)) \,. \tag{8}$$

Proof. If $A \in \mathscr{T} \cap \mathscr{K}_n$ then each disjunction in A is closed and $w(j, A) \equiv 0$, $j \leqq 2^n$. So, for each $u^{2^n} \in \{0, 1\}^{2^n}$, either $w(A) * u^{2^n} = \Lambda$ (the empty sequence, when $u^{2^n} \in \{0\}^{2^n}$), or $w(A) * u^{2^n}$ contains only zeros, hence, $fr\,(1, w(A) * u^{2^n}) = 0$ and $g(A, u^{2^n}) = 1 = Ded\,(A)$. If $A \in \mathscr{K}_n - \mathscr{T}$, and if $u^{2^n} \in \mathscr{A}(A)$, then there is $j_0 \leqq 2^n$ such that $w(j_0, A) = u(j_0, 2^n) = 1$, so $w(j_0, A)$ occurs in $w(A) * u^{2^n}$. However, then $fr\,(1, w(A) * u^{2^n}) \geqq 2^{-n} > 0$, so $g(A, u^{2^n}) = 0 = Ded\,(A)$. □

For an adequate u^{2^n}, the number of pairs of literals which will be, eventually, tested for complementarity, can be majorized by $\frac{1}{2}n(n-1) \,.\, (\sum_{j=1}^{2^n} u(j, 2^n))$. It is why we would like to construct a two-input Turing machine with one input giving A in the form (1) and the other (oracle) giving an adequate u^{2^n} with a small number of units. Let us characterize, at least partially, the adequate sequences in terms of their algorithmic complexity.

3. ALGORITHMIC COMPLEXITY OF PSEUDO-RANDOM INPUTS

Definition 2. Let U be a universal Turing machine (UTM), let p, S, u be finite binary sequences, i.e. $p, S, u \in \{0, 1\}^* = \bigcup_{n=0}^{\infty} \{0, 1\}^n$, $\{0, 1\}^0 = \{\Lambda\}$. The size of a $p \in \{0, 1\}^*$ is defined by the length $l(p)$, so $l(p) = n$ iff $p \in \{0, 1\}^n$. UTM U computes (or yields) u using (the program)p under the condition S when, having inscribed the ordered pair $\langle p, s\rangle$ on the input tape and having settled U into the initial state, U eventually finishes its work giving u as output, in symbols, $U(p, s) = u$. The *conditional algorithmic complexity of u under the condition S and with respect to U*, $K_U(u \mid S)$, in symbols, is defined by

$$K_U(u \mid S) = \min \{l : l \in N,\ l = l(p),\ U(p, S) = u\} \,. \tag{9}$$

When $S = \Lambda$ we write $K_U(u)$ instead of $K_U(u \mid \Lambda)$ and omit the adjective "conditional".

In order to resume our notation: we use the term *input size* for tested formulas and define it by the length of disjunctions in (1), i.e. by the number of different indeterminates in the tested formula, so the size of formulas from $\mathscr{K}_n$ is n. The term *program size* denotes the lengths of binary sequences used as programs. *Algorithmic complexity* of a sequence is given by the program size of the shortest program

generating this sequence and *computational complexity* is given by the number of steps in a computation.

Clearly, as $u \in \{0, 1\}^*$ can be generated by a program consisting of u and an instruction of length c_U, independent of u, which orders to rewrite u on the output tape without any changes, then for each u, $S \in \{0, 1\}^*$

$$K_U(u \mid S) \leqq l(u) + c_U \,. \tag{10}$$

Theorem 2. There exists a natural number $c_1 = c_1(U)$, independent of n and such that, for each formula $A \in \mathscr{K}_n$ and each $u^{2^n} \in \{0, 1\}^{2^n}$,

$$(K_U(u^{2^n} \mid w(A)) \geqq S(A) + c_1) \Rightarrow (u^{2^n} \in \mathscr{A}(A)) \tag{11}$$

(let us recall that $S(A)$ denotes the number of closed disjunctions in A).

Proof. For $A \in \mathscr{T} \cap \mathscr{K}_n$ we have $\mathscr{A}(A) = \{0, 1\}^{2^n}$ and the assertion is trivial. If $A \in \mathscr{K}_n - \mathscr{T}$, then $S(A) < 2^n$, hence, there are $2^n - S(A) > 0$ units in $w(A)$. Each non-adequate u^{2^n} must have zeros on the places of units in $w(A)$, so each non-adequate $u^{2^n} = \langle u(j, 2^n)\rangle_{j=1}^{2^n}$ can be obtained from $w(A)$ and some binary sequence $v^{S(A)} = \langle v(j, S(A))\rangle_{j=1}^{S(A)}$ in this way: $u(j, 2^n) = 0$ if $w(j, A) = 1$, $u(j, 2^n) = v(k, S(A))$, if j is the k-th integer, in the increasing order, for which $w(j, A) = 0$. Each $v^{S(A)}$ can be generated by a program shorter than $S(A) + c_U$, let $c_2(U)$ denote the length of the program taking $w(A)$ and $v^{S(A)}$ into u^{2^n} and described verbally above, $c_2(U)$ does not depend on A. Hence, for each non-adequate u^{2^n}, $K_U(u^{2^n} \mid w(A)) < S(A) + c_U + c_2(U)$, taking $c_1 = c_U + c_2(U)$ we obtain that the reverse inequality $K_U(u^{2^n} \mid w(A)) \geqq S(A) + c_1$ assures the adequacy of u^{2^n}. □

We shall always suppose that c_U, c_1, c_2 and similar constants, originally connected with existential quantifiers, are fixed by the minimal values satisfying the corresponding existential assertions. Moreover, the lower bound $S(A) + c_1$ in (11) can be shown to be the minimal possible in order to assure the adequacy of u^{2^n}, at least for $A \in \mathscr{K}_n - \mathscr{T}$. In fact, take $a \leqq S(A) + c_2(U) - 1$, then there is a non-adequate sequence u^{2^n} with $K_U(u^{2^n} \mid w(A)) > a$. Or, there are $S(A)$ closed disjunctions in A, so there are just $2^{S(A)}$ non-adequate sequences of the length 2^n, each of them corresponding to just one sequence from $\{0, 1\}^{S(A)}$. Programs generating these sequences are also finite binary sequences, hence, there is at most one program of the length 0 (i.e. Λ), at most two programs of the length 1, etc., and at most 2^a programs of the length a. Summarizing, there are at most $\sum_{j=0}^{a} 2^j = 2^{a+1} - 1 < 2^{S(A)+c_2(U)}$ programs with the lengths not exceeding a, hence, there exists $v_0^{S(A)}$ which cannot be generated by such a program. As $c_2(U)$ is the minimal length of a program converting $v_0^{S(A)}$ into a non-adequate sequence, the non-adequate sequence corresponding to $v_0^{S(A)}$ cannot be generated by a program shorter than $a + 1$. The constant c_U can be, in particular cases, zero, e.g. when the input and output tapes of the UTM U are identi-

cal and so giving an input with no further instructions is nothing else than giving the output as well.

Theoretically, Theorem 2 can be used in order to compute the function *Ded*, as

$$(12) \qquad (K_U(u^{2^n} \mid w(A)) \geqq S(A) + c_1) \Rightarrow (g(A, u^{2^n}) = Ded\,(A)) .$$

However, from the practical point of view this way is useless, as it requires to know $w(A)$ and to submit it to a transformation, on the other hand, the knowledge of $w(A)$ already solves the deducibility (theoremhood) problem for A. It is why we shall try to define $\mathscr{A}(A)$, at least partially, in the terms of unconditional algorithmic complexity $K_U(u^{2^n})$.

Definition 3. Let t, m, $m > t$ be integers, $m > 0$, a sequence $u^m \in \{0, 1\}^m$ is called *t-random*, if $K_U(u^m) > m - t$.

Clearly, no t-random sequence exists for $t \leqq -c_U$. Let us define, for $A \in \mathscr{K}_n$, the *adequacy coefficient* $\varrho(A, t)$:

$$(13) \qquad \varrho(A, t) = \frac{card\,(\{u : u \in \mathscr{A}(A),\ K_U(u) > 2^n - t\})}{card\,(\{u : u \in \{0, 1\}^{2^n},\ K_U(u) > 2^n - t\})} .$$

Theorem 3. For each $A \in \mathscr{K}_n$ and each $t > 1$,

$$(14) \qquad \varrho(A, t) > 1 - (2^{c_1(U)+1} \cdot 2^{S(A)} \cdot 2^{-(2^n)}) .$$

Proof. Theorem 2 yields

$$\varrho(A, t) \geqq \frac{card\,(\{u^{2^n} : K_U(u^{2^n} \mid w(A)) \geqq S(A) + c_1,\ K_U(u^{2^n}) > 2^n - t\})}{card\,(\{u^{2^n} : K_U(u^{2^n}) > 2^n - t\})} \geqq$$

$$\geqq \frac{card\,(\{u^{2^n} : K_U(u^{2^n}) > 2^n - t\}) - card\,(\{u^{2^n} : K_U(u^{2^n} \mid w(A)) < S(A) + c_1\})}{card\,(\{u^{2^n} : K_U(u^{2^n}) > 2^n - t\})} =$$

$$= 1 - \frac{card\,(\{u^{2^n} : K_U(u^{2^n} \mid w(A)) < S(A) + c_1\})}{card\,(\{u^{2^n} : K_U(u^{2^n}) > 2^n - t\})} .$$

There are at most $2^{S(A)+c_1} - 1$ sequences with $K_U(u^{2^n} \mid w(A)) < S(A) + c_1$ and there are at least $2^{2^n} - 2^{2^n - t + 1} - 1$ sequences u^{2^n} with $K_U(u^{2^n}) > 2^n - t$ (cf. the argumentation preceeding this theorem). Hence,

$$(15) \qquad \varrho(A, t) \geqq 1 - \frac{2^{S(A)+c_1} - 1}{2^{2^n} - 2^{2^n - t + 1} - 1} > 1 - \frac{2^{c_1} \cdot 2^{S(A)}}{2^{2^n}(1 - 2^{-t-1})} ,$$

so $t > 1$ implies $\varrho(A, t) > 1 - 2^{c_1 + 1 + S(A) - (2^n)}$. □

The value of this adequacy coefficient depends on A through the number $S(A)$ of closed disjunction in A, but "in average" $\varrho(A, t)$ tends to 1 very quickly, in the super-exponential order. E.g., if $S(A) = \frac{1}{2}\,2^n = 2^{n-1}$, i.e. if just one half of disjunctions in A are closed, then $\varrho(A, t) > 1 - 2^{c_1+1} \cdot (\frac{1}{2})^{2^{n-1}}$, as can be derived from (14)

by an easy substitution and calculation. In order to consider the adequacy coefficients $\varrho(A, t)$ in a more global sense, i.e. with respect to all the class $\mathscr{K}_n$ of formulas and not only with respect to particular formulas we may suppose that the tested formula A is sampled from $\mathscr{K}_n$ by a random mechanism (here we take into consideration the physical randomness, not the randomness defined by algorithmic complexity). Let $\langle \Omega, \mathscr{S}, P\rangle$ be a probability space, let $\langle \alpha_{ij}^n\rangle$, $i = 1, \ldots, 2^n$, $j = 1, \ldots, n$, $n = 1, \ldots$ be a system of random variables defined on $\langle \Omega, \mathscr{S}, P\rangle$ and taking their values in the set $\{p_1, \neg p_1, p_2, \neg p_2, \ldots, p_n, \neg p_n\}$ of literals. Hence, denoting this system of random variables by α^n, we may take α^n as a random variable defined on $\langle \Omega, \mathscr{S}, P\rangle$ and taking its values in the set $\mathscr{K}_n$ of formulas. For the sake of simplicity we shall suppose that

(16) the random vectors $\langle \alpha_{i1}^n, \ldots, \alpha_{in}^n\rangle$, $\langle \alpha_{j1}^n, \ldots, \alpha_{jn}^n\rangle$ are statistically independent for each n and each $i, j \leqq 2^n$, $i \neq j$.

(17) $P(\{\omega : \omega \in \Omega, \langle \alpha_{i1}^n(\omega), \alpha_{i2}^n(\omega), \ldots, \alpha_{in}^n(\omega)\rangle \text{ is closed}\}) = pb(n)$ for each n and each $i \leqq 2^n$.

Theorem 4. Consider the random variable α defined above and suppose that α satisfies (16) and (17). Then the expected value $E(\varrho(\alpha^n(\cdot), t))$ of the random variable $\varrho(\alpha^n(\cdot), t)$ satisfies, for $t > 1$,

$$E(\varrho(\alpha^n(\cdot), t)) > 1 - 2^{c_1+1}((1 + pb(n))/2)^{2^n}. \tag{18}$$

Proof. The assumptions imposed on α^n yield that the number $S(\alpha^n(\omega))$ of closed disjunctions in $\alpha^n(\omega)$ can be seen to be a random variable which is governed by the binomic distributions with parameters $pb(n)$ and 2^n, i.e.

$$P(\{\omega : \omega \in \Omega, S(\alpha^n(\omega)) = k\}) = \binom{2^n}{k} (pb(n))^k (1 - pb(n))^{2^n - k}, \tag{19}$$

and an easy calculation gives, using (14)

$$E(\varrho(\alpha^n(\cdot), t)) \geqq \sum_{j=0}^{2^n} \binom{2^n}{j} (pb(n))^j (1 - pb(n))^{2^n - j} (1 - 2^{c_1+1} \cdot 2^j \cdot 2^{-(2^n)}) =$$

$$= 1 - 2^{c_1+1} \cdot 2^{-(2^n)} \left(\sum_{j=0}^{2^n} \binom{2^n}{j} (pb(n))^j \cdot (1 - pb(n))^{2^n - j} \cdot 2^j \right) =$$

$$= 1 - 2^{c_1+1} \cdot 2^{-(2^n)} \left(\sum_{j=0}^{2^n} \binom{2^n}{j} (2\, pb(n))^j (1 - pb(n))^{2^n - j} \right) =$$

$$= 1 - 2^{c_1+1} \cdot 2^{-(2^n)} (1 + pb(n))^{2^n} = 1 - 2^{c_1+1}((1 + pb(n))/2)^{2^n}. \qquad \square$$

Theorem 5. For $A \in \mathscr{K}_n$, the computational complexity of the function $g(A, u^{2^n})$ (defined by (5) and (6)) connected with a pseudo-random generator of t-random sequences ($t > 0$) of the length 2^n is of the class at least $\mathcal{O}(n^2 \cdot 2^n)$, i.e. the same as for the deterministic computation of the function *Ded* based on the sequence $w(A)$.

Proof. The length of each program generating a t-random sequence u^m must be, by the definition, at least $m - t + 1$. All the program must be used and read (in the opposite case a shorther program would generate u^m as well, but this contradicts the t-randomness of u^m), so at least $m - t + 1$-times the operation of reading from the input tape must be applied. If one such application takes K units of computational complexity, then we need at least $K(m - t + 1) = \mathcal{O}(m)$ for all the program. Moreover, there is no function $f(m)$ of the class $o(m)$ such that $\sum_{i=1}^{m} u(i, m) < f(m)$ for t-random sequences u^m. If such an $f(m)$ existed, it would be able to define t-random sequences by a program of the length $f(m) \,.\, (\log_2 m + 1) + \text{const}$, which is of the class $o(m)$; such a hypothetic program would consist in a binary coding of all those at most $f(m)$ values $j \leqq m$, for which $u(j, m) = 1$ and each such code requests $\log_2 m + 1$ binary digits. Hence, for $m = 2^n$, the number of disjunctions in $A \in \mathcal{K}_n$ which must be, eventually, tested for closure using a t-random u^{2^n} is of the class $\mathcal{O}(2^n)$. Checking a disjunction from A for closure, we have to compare, in the worst case, $\frac{1}{2}n(n-1)$, i.e. $\mathcal{O}(n^2)$ pairs of literals for complementarity. Summarizing, the computational complexity of the function $g(A, u^{2^n})$ is of the class $\mathcal{O}(n^2 \,.\, 2^n)$. □

4. COMMENTS AND CONCLUSIVE REMARKS

The statistical theoremhood testing procedure as explained in Chapter 1 is a probabilistic algorithm with random input (cf. the classification of probabilistic algorithms in [4]). Let us remark that an assertion similar to Theorem 5 will be valid even when we re-formulate our algorithm as an algorithm with random steps. In fact, let $A \in \mathcal{K}_n$, let p_A, $0 < p_A < 1$, be given and let us realize, for each $j \leqq 2^n$, statistically independently, an experiment whose probability of success is just p_A. When the experiment is successful for a $j \leqq n$, we verify the j-th disjuntion in A for closure. The algorithm terminates in three cases: when an open disjunction is found in A (in this case A is ultimately a non-theorem), when the experiment for $j = 2^n$ was not successful or when the 2^n-th disjunction in A was closed (in both the last cases we proclaim A to be a theorem). The probability of error of this modified algorithm is majorized by $(1 - p_A)^{2^n - S(A)}$ as for at least all the open disjunctions in A the auxiliary experiment must fail in order to come to the error, and the mean value of the corresponding computational algorithm is of the class $\mathcal{O}(2^n \,.\, p_A \,.\, n^2)$. Hence, the same arguments as those used in the proof of Theorem 5 show that if we use t-random sequences u^{2^n} as simulations for the random sampling of steps in our modified algorithm, it is not possible to take $p_A = p_A(n)$ of the class $\mathcal{O}(n)$, hence, $\mathcal{O}(2^n \,.\, p_A \,.\, n^2)$ is identical with $\mathcal{O}(2^n \,.\, n^2)$ and the situation is similar to the case of algorithm with random input.

Hence, if we take t-random sequences as the only acceptable source of randomness in our statistical theoremhood testing algorithm and if we try to generate these sequences by a computer, the result of Theorem 5 is rather pesimistic. Our decision

to consider just t-random sequences was motivated by the following fact. There exists a rather developed theory of pseudo-random numbers which suggest various tests for answering the question whether a binary sequence (generated by a computer, say) is random or not. As can be shown (cf., e.g. [1], [2], [3]) all the existing as well as the hypothetical tests of randomness can be joined into a universal test of randomness. And there exists a rather high coincidence between t-random sequences and those sequences (of the same length) which satisfy the universal test. More precisely, for each fixed $t > 1$ and $m \to \infty$ the relative frequency of t-random u^m's among those u^m's which satisfy the universal test tends to 1 and vice versa.

Using the same way of reasoning as in the proof of Theorem 5 we can see that random samples of a limited extend m from a sampling space of cardinality $n \gg m$ can be simulated by t-random sequences of the length $m(\text{Int}\,(\log_2 n) + 1)$ even if those sequences do not correspond to t-random sequences of the length n. This fact leads to the idea that the demand of "universal randomness" is too strong for the inputs simulating the random sample in our statistical theoremhood testing procedure or in other probabilistic algorithm. In other words, considering a random number generator satisfying not the universal, but only a weaker test of randomness we may obtain satisfactory results in the sense that the probability of error is kept below an appropriate treshold value and the computational complexity is qualitatively lower than for deterministic algorithms. The notion of a "relative randomness" seems to be an interesting subject for further studies.

(Received March 25, 1981.)

REFERENCES

[1] G. J. Chaitin: Information-theoretic limitations of formal systems. J. Assoc. Comput. Mach. *21* (1974), 3, 403—424.

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

[3] C. P. Schnorr: Zufälligkeit und Wahrscheinlichkeit (Lecture Notes in Math. 218). Springer-Verlag, Berlin—New York 1971.

[4] J. Wiedermann: Pravdepodobnostné algoritmy (Probabilistic algorithms — in Slovak). Informačné systémy *3* (1980), 245—257.

RNDr. Ivan Kramosil, CSc., Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 182 08 Praha 8. Czechoslovakia.