KYBERNETIKA — VOLUME 9 (1973), NUMBER 3

A Method for Statistical Testing of an at Random Sampled Formula

(A Method for Random Sampling of Formulas of an Elementary Theory and Statistical Estimation of their Deducibility Equipped by a Program II)

IVAN KRAMOSIL

This paper contains a construction and investigation of a procedure which enables to decide, in the statistical sense, about the deducibility of a formula sampled at random by the procedure described in [4]. The procedure described in this paper tries to find a proof of the investigated formula in the considered theory, or if it is not possible, in some at random sampled extensions of this theory.

This work is an immediate continuation of the paper [4] and contains the other part of the research report mentioned below the title of this paper. Therefore the notions and notations of [4] are used here without a special mentioning. This paper contains a construction and investigation of a procedure which enables to decide about the deducibility of a formula sampled at random by the procedure described in [4]. The procedure described in this paper tries to find a proof of the investigated formula or, if it is not possible, in some at random sampled extensions of this theory. If such a proof can be found in an a priori given number of extensions, the investigated formula is proclaimed to be a theorem. Of course, there is some risk connected with such a decision, the two kinds of risk, to be more correct. We can arrive to an error proclaiming a non-theorem to be a theorem or proclaiming a theorem to be a non-theorem. These two errors are of different importance in various situations and they are not generally comparable, therefore their probabilities will be always investigated separately. In this paper we shall try, above all, to enable the minimization of the probability of proclaiming a non-theorem to be a theorem. The reason for such an effort follows from the fact that when this kind of error occured the set of formulas proclaimed to be theorems could become inconsistent and therefore worthless for a further use. One the other hand, the rejection of a theorem does not menace the previous results.

Roughly speaking, under the notion *algorithm for deducibility testing* (or briefly *algorithm*, as no other types of algorithms are considered here) we shall understand a recursive function defined on a set of formulas and taking the values in an appropriate set of decisions. However, before defining this notion of algorithm in a precise manner we need to define what does it mean "recursive function defined on a set of formulas and taking its values in a set of decisions".

In this paper the following situation will be considered. Let $(\mathscr{A}, T)$ be a formalized theory based on the first-order predicate calculus, $\mathscr{A}$ being the set of all well-formed formulas, T being the set of all theorems. Let $\mathscr{D} = \{d_1, d_2, \ldots, d_n\}$ be a finite abstract set the elements of which are called decisions. Let g be a mapping from the set $\mathscr{A}$ into (not necessarily onto) the set N of all positive integers, let c be a mapping from the set $\mathscr{D}$ into N. This means, that to every formula A an integer $g(A)$ is ascribed and to every d_i an integer $c(d_i)$ is ascribed. The Gödel enumeration can serve as an example of the former mapping, the mapping $c(d_i) = i$ as an example of the latter one. The mappings g, c will be supposed to be fixed during this paper and if some notions depend on g and c, this dependence will not be explicitely expressed. The set of all integers which are ascribed to formulas from $\mathscr{A}$ will be denoted by G.

Definition 1. Let f be a mapping of the set $\mathscr{A}$ of formulas into the set $\mathscr{D}$ of decisions. Let f^* be a function on the set N of integers, taking its values in the same set and defined as follows:

$$\text{If } x \in G,\ x = g(A),\ A \in \mathscr{A},\ \text{then } f^*(x) = c(f(A)),$$
$$\text{if } x \in N - G,\ \text{then } f^*(x) = c(d_1).$$

The mapping f is called a *recursive mapping*, if the function f^* is recursive. For recursive functions see [6], [7].

Definition 2. A mapping f of the set $\mathscr{A}$ of formulas into the set $\mathscr{D} = \{\mathscr{X}, \mathscr{Y}, \mathscr{T}, \mathscr{N}\}$ of decisions is called a $\mathscr{B}$-algorithm (or algorithm with respect to $\mathscr{B}$) where $\mathscr{B}$ is a subset of $\mathscr{A}$ if the following conditions hold:

f is recursive

if $A \in \mathscr{B} \cap T$, then $f(A) = \mathscr{T}$,

if $A \in \mathscr{B} \cap (\mathscr{A} - T)$, then $f(A) = \mathscr{N}$,

if $A \in \mathscr{A} - \mathscr{B}$, then $f(A) = \mathscr{Y}$.

If the equality $f(A) = \mathscr{T}$ is interpreted as proclaiming the formula A to be theorem and the equality $f(A) = \mathscr{N}$ as proclaiming A to be a non-theorem then a $\mathscr{B}$-algorithm can be understood as a formal description of an effective procedure which decides, given a formula, whether this formula belongs to a set $\mathscr{B}$ or not and, if it is the case, whether it is or is not a theorem.

In this paper we shall always assume that $\mathscr{B}$ contains all formulas which are substitutions into propositional calculus tautologies and all negations of such formulas. If this set is denoted by $\mathscr{B}_0$ this condition gives: $\mathscr{B}_0 \subset \mathscr{B}$. Definition 2 gives immediately that if $f(A) = \mathscr{T}$, then actually $A \in T$, if $f(A) = \mathscr{N}$, then actually $A \in \mathscr{A} - T$.

Definition 3. The expression *semialgorithm for deducibility testing* or briefly *semialgorithm* will be used for a triple $\mathscr{S} = (h, \tilde{f}, \tilde{\mathscr{B}})$, where h is a recursive function mapping the set N of natural numbers into itself, $\tilde{\mathscr{B}} = \{\mathscr{B}_1, \mathscr{B}_2, \ldots\}$ is a sequence of subsets of the set $\mathscr{A}$ such that $\mathscr{B}_0 \subset \bigcap_{i=1}^{\infty} \mathscr{B}_i$, $T \subset \bigcup_{i=1}^{\infty} \mathscr{B}_{h(i)}$, $\tilde{f} = \{f_1, f_2, \ldots\}$ is a recursive sequence such that every f_i is a $\mathscr{B}_i$-algorithm according to Definition 2.

We shall say that a formula A is proclaimed to be a theorem by the semialgorithm $\mathscr{S}$ (notation $\mathscr{S}(A) = \mathscr{T}$ will be used), if there exists an integer n such that $A \in \mathscr{B}_{h(n)} \cap T$, i.e. $f_{h(n)}(A) = \mathscr{T}$. We shall say that A is proclaimed to be a non-theorem by the semialgorithm $\mathscr{S}$ (notation $\mathscr{S}(A) = \mathscr{N}$ will be used), if there exists an integer m such that $A \in \mathscr{B}_{h(m)} \cap (\mathscr{A} - T)$, i.e. $f_{h(m)}(A) = \mathscr{N}$. In both these cases we shall say that the decision was taken at the n-th (m-th, respectively) step.

When $\mathscr{S}(A)$ is defined, then it is defined unambiguously and correctly. Indeed, supposing A were such a formula, that there existed indices m, n such that $f_{h(n)}(A) = \mathscr{T}$, $f_{h(m)}(A) = \mathscr{N}$, then $A \in \mathscr{B}_{h(n)} \cap T$, $A \in \mathscr{B}_{h(m)} \cap (\mathscr{A} - T)$, which leads to a contradiction. The correctness of the decision taken by the semialgorithm $\mathscr{S}$ follows from the fact that this decision is in accordance with a decision of a $\mathscr{B}_i$-algorithm, i.e. if $\mathscr{S}(A) = \mathscr{T}$, then $A \in T$, if $\mathscr{S}(A) = \mathscr{N}$, then $A \in (\mathscr{A} - T)$.

One aspect, in which a semialgorithm differs from a $\mathscr{B}$-algorithm consists in the fact that a semialgorithm represents a non-effective decision procedure. If we ascribe to a formula A an integer $l(A)$ such that

$$l(A) = \min \{n : A \in \mathscr{B}_{h(n)}\}$$

then, in general, $l(A)$ is not defined for some A (for those from $\mathscr{A} - \bigcup_{i=1}^{\infty} \mathscr{B}_{h(i)}$), and, moreover, even if $l(A)$ defined somehow in this case (e.g. $l(A) = 0$) nevertheless the function l^* defined for l in the same manner as f^* for f in Definition 1 will not be, in general, recursive.

A semialgorithm represents the following type of decision procedures: In the n-th step we decide, before all, which decision algorithm will be applied, by the computation of $h(n)$. Then we apply the decision algorithm $f_{h(n)}$ to the tested formula. If $f_{h(n)}$ decides about A the procedure is finished, if it is not the case we compute $h(n + 1)$ and apply $f_{h(n+1)}$ and so on. The non-effectiveness of this procedure consists in the fact that, in general, we do not know a priori, whether we come to a decision following this procedure and even in case we come, the number of necessary steps cannot be determined a priori. The only thing we request is that every theorem is to be, sooner or later, proclaimed to be a theorem.

It follows easily from the foregoing text that a semialgorithm can be used in order to test a formula only when connected with an instruction how and when to stop the testing procedure. This instruction reads usually as follows: The procedure is stopped (if the decision has not been taken yet) in the moment when the time or the expenses connected with the decision process overreach some a priori limits. The expenses can be measured in various manners, e.g. by the computer storage capacity or by the number of some unit operations needed for performing of a decision or a sequence of decisions. However, the question of the appropriate parameters, how to measure them, how to establish the limit values in an actual situation and so on will not be investigated here but we try, in the following definition to describe this idea in a general and formal manner.

Definition 4. Let $\mathscr{S}$ be a semialgorithm, let $m = m(\mathscr{S})$ be an integer. Let $\xi = \xi(i, j, A, \mathscr{S})$ be a non-negative real function, $i = 1, 2, \ldots, m(\mathscr{S})$, $j = 1, 2, \ldots, \ldots$, $A \in \mathscr{A}$ satisfying the following conditions:

for a fixed $i \leqq m(\mathscr{S})$, $A \in \mathscr{A}$ the function ξ is nondecreasing function of j,

for every $A \in \mathscr{A}$ there exists at least one $i_0 = i_0(A) \leqq m(\mathscr{S})$ such that

$$\lim_{j \to \infty} \xi(i_0, j, A, \mathscr{S}) = \infty .$$

Let $\bar{a} = \{a_1, a_2, \ldots, a_{m(\mathscr{S})})$ be a non-negative real vector.

The expression *restricted semialgorithm* will be used for the triple $\overline{\mathscr{S}} = (\mathscr{S}, \xi, \bar{a})$; this triple will be understood as a mapping of the set $\mathscr{A}$ into the set $\{\mathscr{Y}, \mathscr{T}, \mathscr{N}\}$ defined as follows:

If the semialgorithm $\mathscr{S}$ decides about A in the n-th step and, at the same time, for every $i = 1, 2, \ldots, m(\mathscr{S})$

$$\xi(i, n, A, \mathscr{S}) \leqq a_i$$

we set $\overline{\mathscr{S}}(A) = \mathscr{S}(A)$. In all other cases we set $\overline{\mathscr{S}}(A) = \mathscr{Y}$.

Roughly speaking, $\xi(i, j, A, \mathscr{S})$ represents the i-th component of the expenses connected with performing the first j steps when A decided by the algorithm $\mathscr{S}$. Of course, these expenses are increasing, or at least are not decreasing when the number of steps increases and, moreover, there is quite natural to demand that at least one component of these demands should increase to infinity with the number of steps increasing to infinity (e.g. the time necessary for performing of the first j steps is a component with this property). The equality $\overline{\mathscr{S}}(A) = \mathscr{Y}$ formally expresses the fact that semialgorithm $\mathscr{S}$ is not able to decide about A within the limits given by the vector $\bar{a}$.

In some aspects the restricted semialgorithm stands close to the notion of $\mathscr{B}$-algorithm, however, the formal connections among those two notions will not be investigated here. The following lemma gives an important property of semialgorithms and restricted semialgorithms.

Lemma 1. *Let A be a theorem, let $\mathscr{S}$ be a semialgorithm, let $m(\mathscr{S})$ be an integer, let $\xi = \xi(i, j, A, \mathscr{S})$ be a real function such that the conditions of Definition 4 are satisfied. Then there esists a real vector $\bar{a} = \bar{a}(A) = \{a_1, \ldots, a_{m(\mathscr{S})}\}$ such that the restricted semialgorithm $\overline{\mathscr{S}} = (\mathscr{S}, \xi, \bar{a})$ decides correctly about A, i.e. $\overline{\mathscr{S}}(A) = \mathscr{T}$.*

Proof. Let the conditions hold, let $A \in T$, $\mathscr{S} = (h, \tilde{f}, \tilde{\mathscr{B}})$. As $A \in (\bigcup_{i=1}^{\infty} \mathscr{B}_{h(i)} \cap T)$, then there exists an index j_0 such that $A \in \mathscr{B}_{h(j_0)} \cap T$. Choose a_i, $i = 1, 2, \ldots, m(\mathscr{S})$ in such a way that the inequalities

$$a_i \geqq \xi(i, j_0, A, \mathscr{S}), \quad i = 1, 2, \ldots, m(\mathscr{S})$$

hold. Clearly, according to Definition 4 $\overline{\mathscr{S}}(A) = \mathscr{S}(A) = \mathscr{T}$. Q.E.D.

An analogous statement concerning the non-theorem obviously does not hold, as for a formula $A \in \mathscr{A} - \bigcup_{i=1}^{\infty} \mathscr{B}_{h(i)}$ immediately follows that $\overline{\mathscr{S}}(A) = \mathscr{Y}$ for every $\bar{a}$. If we modified Definition 4 in such a way that instead of $\overline{\mathscr{S}}(A) = \mathscr{Y}$ we set $\overline{\mathscr{S}}(A) = \mathscr{N}$ or if we modified the interpretation of our decision in such a way that $\overline{\mathscr{S}}(A) = \mathscr{Y}$ was interpreted as proclaiming A to be a non-theorem, then Lemma 1 would hold even in the case of non-theorems.

Now, we are in a position to introduce the main notion of this paper.

Definition 5. Let us consider a probability space $(\Omega, \mathscr{F}, P)$ two integers $1 \leqq m_1 \leqq \leqq n_1$, random variables $\alpha_1, \alpha_2, \ldots, \alpha_{n_1}$ defined on $(\Omega, \mathscr{F}, P)$ and taking their values in the set $\mathscr{A}$ of formulas of the considered theory $(\mathscr{A}, T)$. Let R be a $\mathscr{B}$-algorithm, $\mathscr{B} \subset \mathscr{A}$, let $\overline{\mathscr{S}} = (\mathscr{S}, \xi, \bar{a})$ be a restricted semialgorithm. Then the sequence

$$\Sigma = (R, \overline{\mathscr{S}}, (\Omega, \mathscr{F}, P), \{\alpha_1, \ldots, \alpha_{n_1}\}, m_1, n_1)$$

will be called a *statistical algorithm for deducibility testing in at random sampled extansions* (briefly *statistical test* or *statistical testing algorithm*). A statistical test is understood as a random variable, defined on the Cartesian product $\Omega \times \mathscr{A}$ and taking its values in the set $\{\mathscr{T}, \mathscr{N}\}$. This random variable is defined as follows:

$$\text{If } R(A) \in \{\mathscr{T}, \mathscr{N}\}, \text{ then } \Sigma(A) = R(A).$$

In other case:

$$\text{If } \overline{\mathscr{S}}(A) \in \{\mathscr{T}, \mathscr{N}\}, \text{ then } \Sigma(A) = \overline{\mathscr{S}}(A).$$

In other case:

Sample at random formulas $\alpha_1(\omega), \ldots, \alpha_{n_1}(\omega)$ and denote by $[\![\alpha_1(\omega)]\!], \ldots, [\![\alpha_{n_1}(\omega)]\!]$ their universal closures. Set $\beta_i = 1$, if $R([\![\alpha_i(\omega)]\!] \rightarrow \bar{A}) = \mathscr{T}$ or $R(\alpha_i(\omega) \rightarrow A) = \mathscr{T}$ and

$\alpha_i(\omega) \to A$ is a propositional calculus tautology or if $\overline{\mathscr{S}}([\![\alpha_i(\omega)]\!] \to \bar{A}) = \mathscr{T}$. Set $\beta_i = 0$ is other cases. If $\sum_{i=1}^{n_1} \beta_i \geqq m_1$, set $\Sigma(A) = \mathscr{T}$, if $\sum_{i=1}^{n_1} \beta_i < m_1$, set $\Sigma(A) = \mathscr{N}$. The values $\mathscr{T}$ and $\mathscr{N}$ are interpreted as before.

Having introduced some theoretical notions dealing with the deducibility testing it seems to be useful and reasonable to mention some actual examples among the already existing and developed testing procedures.

The class of procedures corresponding to $\mathscr{B}$-algorithms is rather numerous. There exists, e.g., a number of procedures for propositional calculus formulas testing starting from the classical zero-one procedure and finishing with some modern variants based on the Gentzen calculus of sequences (e.g. the algorithm P in [3]). However, there exist decision methods also in the case, when $\mathscr{B}$ is the set of all A – E formulas; see the detail explanation and the procedures Qp, Qr again in [3]. Such methods exist even for another sets $\mathscr{B}$ of formulas having a more complicated structure. These procedures are usually based on the Herbrand theorem and some of its modifications. This theorem enables, roughly speaking, to transform the deducibility problem in the first-order functional calculus on the problem of generating of the so called substitution instances and deciding whether an appropriate alternative of such substitution instances is or is not a propositional calculus tautology.

What we have just said about the Herbrand theorem leads to the question of semialgorithms, as a great part of procedures, having the properties of semialgorithms is based again on the Herbrand theorem and on the substitution instances method. As an example can serve the procedure Q from [3], but there exist even some modifications and at present this approach seems to be the most perspective one in the field of theorem proving.

The $\mathscr{B}$-algorithms enable to decide, in an effective way, about formulas from some decidable set of formulas and the decision is always correct. The fact that such algorithms play a specific role in our definition of the general notion of statistical testing algorithm is caused by the following fact: in some situations a class of formulas may be of a very important significance and it may be desirable to ensure the correct deciding about these formulas even in case the decision is more demanding. If these "important" formulas form a decidable set we can satisfy this demand using an appropriate $\mathscr{B}$-algorithm.

Now, having to our disposal a semialgorithm $\mathscr{S}$ we can transform it into a restricted semialgorithm $\overline{\mathscr{S}}$ e.g. in such a way that we set $m = m(\mathscr{S}) = 2$ and denote by $\xi(1, j, A, \mathscr{S})$ the number of time units (e.g. seconds) necessary for performing the first j steps of the semialgorithm $\mathscr{S}$, denote by $\xi(2, j, A, \mathscr{S})$ the number of bytes in a computer storage necessary for saving all the data and results connected with these first j steps of $\mathscr{S}$. The conditions of Definition 4 are then clearly satisfied. Of course, we can choose $m(\mathscr{S}) > 2$ and investigate the particular components of the demands connected with $\mathscr{S}$ in more detailed form.

The notion of statistical deducibility testing algorithm as introduced above is a formal and rather simplified description of the statistical procedure investigated and constructed in [5]. Let us remark, that the statistical testing algorithm can be defined even in another way. As an example of a rather different approach to the introduction of statistical aspects and notion into theorem proving can serve the van Westrhenen's approach (see [8], [9]).

Let $H1$ be a $\mathscr{B}$-algorithm with the property that $\mathscr{B}$ contains, beside the propositional calculus tautologies and their negations even all the A – E formulas (and perhaps another formulas). Let $H2 = (\mathscr{S}, \xi, \bar{a})$ be a restricted semialgorithm with the property that the semialgorithm $\mathscr{S}$ is identical or equivalent with the procedure Q from [3]. Let $(\Omega, \mathscr{F}, P)$ be a probability space, $m_1 \leqq n_1$ integers, $\alpha_1, \alpha_2, \dots, \alpha_{n_1}$ random variables defined on $(\Omega, \mathscr{F}, P)$ such that the conditions of Definition 5 as well as the condition

$$(*) \qquad P(\{\omega : \omega \in \Omega, \alpha_i(\omega) = A\}) > 0$$

for every $A \in \mathscr{A}$, $i \leqq n_1$ hold.

Let us denote by Σ_1 the statistical testing algorithm

$$(H1, H2, (\Omega, \mathscr{F}, P), \{\alpha_1, \dots, \alpha_{n_1}\}, m_1, n_1)\,.$$

This statistical testing algorithm will be investigated in all the rest of this paper. We always assume that when a formula A is to be tested, the test is applied to the formula $A_0 \rightarrow A$ (or $[A_0]\,[A]$ in the notation of [4]) where A_0 is $AX_1 \,\&\, AX_2 \,\&\, \dots \,\&\, \&\, AX_s$, AX_i, $i = 1, 2, \dots, S$ are the specific axioms of the considered theory $(\mathscr{A}, T)$.

2. SOME PROPERTIES OF STATISTICAL ALGORITHMS FOR DEDUCIBILITY TESTING

Theorem 1. *Let us consider a statistical algorithm Σ. Let T denote the set of all theorems of our theory. Let A be an at random sampled formula, obtained by the algorithm explained in* [4]. *Let the conditions of Theorem* 3 *in* [4] *hold. Let P_1 denote the conditional probability*

$$P(\{\omega : \Sigma(A(\omega)) = \mathscr{T}\} \mid \{\omega : A(\omega) \in T\})\,.$$

Then the following inequality holds:

$$P_1 \geqq \left(\frac{M_0 + N_1}{N_2}\right)\left(\frac{N_1}{N_1 - 1}\right)\left(\frac{K_0 + N_0 + N_1 - 1}{M_0 + K_0 + 1}\right).$$

$$\cdot\left(1 - \left(\frac{K_0 + N_0 + N_1 - 1}{N_2}\right)^{Z_1 - 1}\right) - \left(\frac{M_0 + N_1}{N_2}\right)\left(\frac{N_1}{N_1 - 1}\right)\left(\frac{K_0 + N_0}{M_0 + K_0 + N_1}\right).$$

$$\cdot\left(1 - \left(\frac{K_0 + N_0}{N_2}\right)^{Z_1 - 1}\right) + \frac{N_2}{N_1}\frac{M_0^2}{N_2^2}\left(\frac{K_0 + N_0 + N_1}{M_0 + K_0}\right).$$

$$\cdot\left(1-\left(\frac{K_0+N_0+N_1}{N_2}\right)^{Z_1-1}\right)-\frac{N_2}{N_1}\frac{M_0^2}{N_2^2}\left(\frac{K_0+N_0}{M_0+K_0+N_1}\right)\left(1-\left(\frac{K_0+N_0}{N_2}\right)^{Z_1-1}\right).$$

(For the meaning of the symbols M_0, N_0, N_1, N_2, K_0, Z_1 see [4].)

Proof. Let us consider the sets $T(n, m, k)$ and $S(n, m)$ from the proof of Theorem 3 in [4]. By $G(\beta_1(\omega), \dots, \beta_{Z_1}(\omega))$ the random variable, defined on $(\Omega, \mathscr{F}, P)$, taking its values in the set of all formulas and defined in [4] is denoted.

We proved in the mentioned proof that if $(\beta_1(\omega), \dots, \beta_{Z_1}(\omega))$ belongs to some of the sets $T(n, m, k)$ or $S(n, m)$, then $G(\beta_1(\omega), \dots, \beta_{Z_1}(\omega))$ is a theorem of the form $[B]\,[F]\,[A]\,[E_k]$ or $[B]\,[E_k]\,[A]\,[E_k]$. In every case all the l.s. from $\mathcal{O}(G(\beta_1(\omega), \dots, \beta_{Z_1}(\omega))$ ale a.l.s., therefore the theorem $G(\beta_1(\omega), \dots, \beta_{Z_1}(\omega))$ is proclaimed to be a theorem. It follows:

$$P_1 \geqq P(\{\omega: (\beta_1(\omega), \beta_2(\omega), \dots, \beta_{Z_1}(\omega)) \in$$
$$\in (U_{n=1}^{Z_1} U_{m=n+1}^{Z_1} U_{k=1}^{N_1}\ T(n, m, k)) \cap (U_{n=1}^{Z_1-1} U_{m=n+1}^{Z_1}\ S(n, m))\}) .$$

The fact that this expression is greater than or equal to the lower bound given in this theorem was proved in Theorem 3 in [4]. Q.E.D.

The lower bound for P_1 given in Theorem 1 may seem to be rather low, e.g. in the example investigated in [4] it would be about 60%. There are several reasons for this fact, namely:

a) The estimation just derived holds for any values of m_1 and n_1 in Σ, i.e. even in the case $m_1 = n_1$, $n_1 \to \infty$ which is the most undesirable from the point of view of proclaiming a theorem to be a theorem (the demands for proclaiming a theorem to be a theorem are increasing). On the other hand, in the case $m_1 = 1$, $n_1 \gg 1$ much more pleasant estimation can be obtained, in fact $P_1 \to 1$ if $n_1 \to \infty$ and m_1 is fixed. However, this fact is joined with the increasing of the probability of adopting a non-theorem. For every statistical algorithm for deducibility testing (at least in the sense how we have defined this notion) there exists some lower bound for the sum of the both probabilities of errors. By choosing appropriate values of m_1 and n_1 we can only control the ratio of those two probabilities and so we can minimize one of them under an a priori given value. This sum of probabilities could be minimized under an a priori given value only in case we abandoned the demand on the unambiguity of the decision, i.e. in case we took into consideration, besides the abstract values $\mathscr{T}$, $\mathscr{N}$, some third value the semantic interpretation of which would be "I cannot decide on the tested formula with a sufficient confidence".

b) The considered estimation holds for every formalized theory based on the first-order functional calculus and for every statistical algorithm for deducibility testing (in the defined sense) only under the condition the algorithm for random sampling of well-formed formulas from [4] plays the role of $\alpha_1, \dots, \alpha_{n_1}$. The estimation does not depend on the axioms of the considered theory and on the properties of the procedures R, $\overline{\mathscr{S}}$ in Σ. Let us describe briefly an example showing how at least partial use of the axioms can improve the estimation for P_1.

Let us consider the equality theory described in [4] and enriched by a new binary logical constant N' (inequality, $\neq$) and by the following axioms:

AX_5: $[x_1[x_2[N'x_1x_2][[Ix_1x_2][F]]]]$,

AX_6: $[x_1[x_2[[Ix_1x_2][F]][N'x_1x_2]]]$

(the conjunction of these two axioms is equivalent to the definition axiom $x_1 \neq x_2 \underset{\mathrm{df}}{\Leftrightarrow} \text{non}\,(x_1 = x_2)$ in the usual notation).

AX_7: $[x_1[N'x_1x_1][F]]$ (i.e. $\forall_{x_1}(\text{non}\,(x_1 \neq x_1)))$.

Let us sample at random a formula (using the algorithm from [4] and let us check this formula using the statistical algorithm Σ_1.

Let us consider a sequence of symbols containing an occurence of an elementary formula Ix_ix_i not followed by another occurence of an elementary formula or F. Such a sequence will be transformed into a formula of the form $[A][Ix_ix_i]$. Hence, during its testing the formula $[x_1[Ix_1x_1]][x_i[A][x_ix_i]]$ (maybe with some other quantifiers following the $[x_i)$ will be investigated, as its antecedent is one of the axioms, $i \neq 1$. After substituting the positive indeterminate x_i at the place of negative indeterminate x_1 and after erasing the quantifiers we obtain the formula $[Ix_ix_i][A][Ix_ix_i]$ which is a substitution into the propositional calculus tautology $p \to (g \to p)$. So the formula $[A][Ix_ix_i]$, being evidently a theorem is correctly proclaimed to be a theorem.

We shall come to a similar result considering a sequence of symbols in which an elementary formula $N'x_ix_i$ occurs, followed by at least one occurence of an elementary formula or F but not followed by an occurrence of the right bracket. Such a sequence will be transformed into a formula of the form $[A][Nx_ix_i][B]$. Hence, during its testing the formula

$$[x_1[Nx_1x_1][F]][x_i[A][Nx_ix_i][B]]$$

(maybe with some other quantifiers following the $[x_i)$ – will be investigated as its antecedent is one of the axioms, $i \neq 1$. After substituting the positive indeterminate x_i at the place of negative indeterminate x_1 and after erasing the quantifiers we obtain the formula $[[Nx_ix_i][F]][A][Nx_ix_i][B]$ which is a substitution into the propositional calculus tautology non $p \to (S \to (p \to g))$. Therefore also the formula $[A][Nx_ix_i][B]$, being a theorem, is correctly proclaimed to be a theorem.

Denoting by γ the random variable choosing the particular symbols during the procedure for random sampling of formulas, denoting by $K = K(K_0, M_0, N_0, N_1, Z_1)$ the lower bound for P_1 in Theorem 1, setting

$$f = P(\{\omega\colon \gamma(\omega) \text{ is an elementary formula of the form } N'x_ix_i\}),$$

$$g = P(\{\omega\colon \gamma(\omega) \text{ is an elementary formula of the form } Ix_ix_i\})$$

and computing the probability of sampling a sequence of symbols from one of the sets $T(n, m, k)$, $S(n, m)$ or of the sets described above we obtain the following improved estimation for P_1:

$$P_1 \geqq 1 - (1 - K)\left(1 - \frac{f}{a + e} K\right)\left(1 - \frac{g}{a + c} K\right) = K'(K_0, M_0, N_0, N_1, Z_1) > K$$

as $f > 0$, $g > 0$. The symbols a, e, c denote the same numbers as in [4].

Example:

$$K_0 = 10\,, \quad M_0 = 10\,, \quad N_0 = 5\,, \quad Z_1 = 15 : K = 0{\cdot}46225\,, \quad K' = 0{\cdot}67867\,;$$

$$K_0 = 6\,, \quad M_0 = 10\,, \quad N_0 = 5\,, \quad Z_1 = 15 : K = 0{\cdot}79615\,, \quad K' = 0{\cdot}87364\,.$$

From these numbers the improvement obtained by partial considering the axioms can be easily seen.

Theorem 2. *Let the conditions of Theorem* 1 *hold. Let* $m_1 = n_1$, *let* β *be a real number satisfying the condition*

$$0 \leqq \beta \leqq \mathsf{P}(\{\omega\colon A(\omega) \in T\})\,.$$

Let P_2 *denote the probability of proclaiming an at random sampled formula to be a theorem under the condition it is not a theorem, i.e.*

$$P_2 = \mathsf{P}(\{\omega\colon \Sigma(A(\omega)) = \mathscr{T}\} \mid \{\omega\colon A(\omega) \notin T\})\,.$$

Then the inequality

$$P_2 \leqq (1 - \beta)^{n_1}$$

holds, supposing the random variables $\{\alpha_i\}_n^{n_1}$ *are mutually independent.*

Proof. When considering the properties of statistical testing algorithms we can see that no non-theorem can be proclaimed to be a theorem when the randomized part of the algorithm is not used. Such an error can occur only if $A(\omega)$ is a non-theorem and at the same time such non-theorems $\alpha_1(\omega), \alpha_2(\omega), \ldots, \alpha_{n_1}(\omega)$ were sampled that for every $i = 1, 2, \ldots, n_1$ the formulas $[\alpha_i(\omega)]\,[A(\omega)]$ were by R or $\overline{\mathscr{S}}$ decidable

Table 1.

β \ $P_2 \leqq$	0·25	0·1	0·05	0·01	0·005	0·001	0·0005
0·46	3	4	5	8	9	12	13
0·57	2	3	4	6	7	9	10
0·68	2	3	3	5	5	7	7
0·87	1	2	2	3	3	4	4

theorems. According to the supposed independence of the random variables $\alpha_1, \alpha_2, \ldots$ $\ldots, \alpha_{n_1}$ the number $(1 - \beta)^{n_1}$ represents an upper bound for the probability of such an event. Q.E.D.

A great advantage of the estimation for P_2 just obtained lies in the fact, that it does not depend on the specific axioms and constants of the investigated theory. The Table 1 gives the minimal values for n_1 necessary for satisfying the condition $P_2 \leqq 0{\cdot}25$; $0{\cdot}1$; $0{\cdot}05$ and so on. The values of β are borrowed from the examples investigated in [4].

Theorem 3. *For every formalized theory based on the first-order predicate calculus the following holds:*

a) *For every theorem there exists a statistical algorithm for deducibility testing which proclaims this theorem to be a theorem with probability one.*

b) *Under the condition* $(*)$ *for every formula A and every statistical algorithm Σ there is a positive probability for the decision $\sum(A)$ to be correct.*

Proof. a) Let A be a theorem. In case A belongs to a decidable set of formulas we choose an appropriate alggorithm R. If it is not the case we can choose an appropriate restricted semialgorithm $\overline{\mathscr{S}}(a_1, a_2, \ldots, a_{m(\mathscr{S})})$ according to Lemma 1. So A will be decided without using randomized extensions: it follows that the decision will be correct with probability 1.

b) Supposing the tested formula A is decided without using randomized extensions this decision is correct with probability 1. Let us consider the opposite case. Every formula is given a positive probability to be sampled by $\alpha_1, \alpha_2, \ldots, \alpha_{n_1}$. If A is a theorem, then there is a positive probability to obtain at least m_1 formulas $\alpha_{i_1}, \alpha_{i_2}, \ldots, \alpha_{i_{m_1}}$ such that $[\alpha_{ij}]$ $[A]$ is, for every $j = 1, 2, \ldots, m_1$ by R or $\overline{\mathscr{S}}(a_1, \ldots, a_m)$ correctly decidable theorem, hence $P(\{\omega\colon \Sigma(A(\omega)) = \mathscr{T}\}) > 0$. If A is a non-theorem, then it is sufficient for its refusing to choose at least $n_1 - m_1 + 1$ theorems among $\alpha_1, \alpha_2, \ldots$ $\ldots, \alpha_{m_1}$, hence again $P(\{\omega\colon \Sigma(A(\omega)) = \mathscr{N}\}) > 0$. Q.E.D.

This theorem intends to express some connections between the statistical algorithms and those non-statistical (i.e. deterministic) ones. Its first assertion means, roughly speaking, that in the case of theorems the assertion of Lemma 1 can be extended even to the statistical algorithms. It is not possible, however, to extend it to every formula because of the fact that there may exist a non-theorem with the property that not semialgorithm ascribes the value $\mathscr{N}$ to it. So this non-theorem will be in every case checked in randomized extensions and therefore there is, in general, a positive probability of its wrong accepting. The other assertion shows that if a statistical algorithm is used, then every formula is given a "chance" to be decided correctly. This is a principal advantage when comparing with the non-statistical algorithms, which do not give such a "chance" to a formula which cannot be decided in the deterministic way.

Because of its more general significance we introduce this Theorem 3 at the end of this paper.

The author would like to express his sincere thanks to Dr. Petr Hájek (Mathematical Institute of the Czechoslovak Academy of Sciences) for his valuable remarks.

(Received February 28, 1972.)

REFERENCES

[1] Church Alonzo: Introduction to Mathematical Logic, Part. I. Princeton University Press, Princeton 1956.

[2] Gentzen Gerhardt: Untersuchungen über das logische Schliessen. Mathem. Zeitschrift *39* (1934—35), 176—210, 405—431.

[3] Hao Wang: A Survey of Symbolic Logic. Science Press, Peking; North Holland Publishing Co., Amsterdam 1962.

[4] Kramosil Ivan: A Method for Random Sampling of Well-Formed Formulas. Kybernetika *8* (1972), 2, 135—148.

[5] Kramosil Ivan: Statistical Estimation of Deducibility in Polyadic Algebras. Kybernetika *7* (1971), 3, 181—200.

[6] Davis Martin: Conputability and unsolvability. McGraw Hill Book Company, New York, Toronto, London 1958.

[7] Kleene Stephen Cole: Introduction to Metamathematics, D. van Nostrand Company, New York, Toronto 1952.

[8] van Westrhenen S. C.: A probabilistic machine for the estimation of Probability in the first order predicate calculus. Zeitschr. für Math. Logik and Grundlagen der Math. *15* (1969), 291—297.

[9] van Westrhenen S. C.: Statistical Studies of Theoremhood in Classical Propositional and First-Order Predicate Calculus. Journal of the Association for Computing Machinery *19* (1972), 2, 347—365.

Dr. Ivan Kramosil, CSc.; Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Vyšehradská 49, 128 48 Praha 2. Czechoslovakia.