Kybernetika

Ivan Kramosil; Zbigniew Zwinogrodzki
Statistical methods for comparing theorem proving algorithms

Kybernetika, Vol. 10 (1974), No. 3, (221)--240

Persistent URL: http://dml.cz/dmlcz/124528

KYBERNETIKA — VOLUME *10* (1974), NUMBER 3

Statistical Methods for Comparing Theorem Proving Algorithms

IVAN KRAMOSIL, ZBIGNIEW ZWINOGRODZKI

In this paper a binary relation on a set of theorem proving algorithms is defined, using some basic notions of statistics and probability theory. This relation is proved to generate a linear ordering in any given set of theorem proving algorithms, it means to serve as an attempt to formalize somehow the often used phrase: "Algorithm A is better than algorithm B", at least in the case of theorem proving algorithms.

A number of assertions describing some basic properties of this relation is stated and proved. The notion apparatus used here is close to that from [1]—[4].

1. INTRODUCTION

The first attempts to use the mathematical machines in order to decide the classes of problems not decidable by algorithms used before were made a long time ago. We can meet with these classes of problems in every field of human activity which is considered to be "creative" one; mathematics being a special case. It is a well-known fact that even for the first order predicate calculus no universal method exists enabling to decide in a finite number of steps, whether a given formula of this calculus is or is not a tautology [5]. Nevertheless, for many mathematical theories based on the first order predicate calculus with equality we can construct the so called *semialgorithms* (this term is used in [6]), enabling to decide in a finite number of steps, for every given tautology of this calculus that it is a tautology. This property is usually called the *completeness* of the semialgorithm. If the tested formula is not a tautology it is possible that the decision process, obtained when the semialgorithm applied to this formula, will not be finite. In the following we shall use the term "algorithm" even when considering a semialgorithm; this terminology is usual in papers of this type.

The first attempt to suggest a theorem proving algorighm which was in some measure successful was presented in [7]. The following years have brought a number of new proposals all of them being interpretable as various forms of improvement

of the so called Gentzen's system, see [8]. Usually the well-known Herbrand's and Craig's theorems are used to this goal. Among the best known results of this type we can mention the Beth's method of semantic tables [10], improved by L. W. Szczerba [11], J. A. Robinson's resolution method [6], [12], [13], S. Ju. Maslov's inverse method, [14], [15]), paramodulation method [16], [17] and some else.

The starting period of this "building movement" in the theory of automatic theorem proving was very optimistic and connected with a firm believe that the next successes were very close. J. A. Robinson wrote in the time (in [6]): "It is not foregone conclusion, at present, that we are going to be outclassed at solving deduction problems by suitably programmed automatic computing devices; but the odds are that we shall be, and overwhelmingly so. If and when this comes to pass, we must all be glad of it, for then there will be so much more time for us to think and to dream. "However, in a short time mathematicians discovered that the organization of a selecting of formulas described in these methods is not sufficient for a following practical application of these methods using even the most powerful and most capable computers being to our disposal at present.

The disappointment following the first failures of the attempts of a practical application of the methods investigated in [10]–[17] was one of the sources of inspiration for a statistical approach to the problem of automatic theorem proving. As far as the authors know, the first suggestion of a statistical method for theorem proving was proposed by A. Špaček [1], [2] for the case of Boolean algebras. This method was significantly modified by I. Kramosil, who avoided some very strong and practically (using computer, e.g.) not realizable conditions, see [3], [4]. The latest attempts to apply some probabilistic notions in order to construct a statistical theorem proving algorithm can be found in [18] and [19].

In the statistical methods of theorem proving the postulate of completeness, mentioned above and satisfied in any "classical" method, is not satisfied. If the procedure, given by a statistical method of theorem proving, says that a formula is a tautology, there is, in general, a positive probability that this decision is not correct. By increasing the number of random extensions, in which the tested formula is investigated, we can reach the situation in which the probability of proclaiming a non-theorem to be a theorem is smaller than an a priori given positive real number. The mechanism of verification used in [3], [4], [18], [19] is close to that known in general as natural induction principle. At present, rather great hopes are connected with these methods because of their greater practical effectivity. However, there is a rather important difference between [3], [4] on one side and [18], [19] on the other side. If these algorithms are to be effective from a practical point of view we must define a priori a number N_0 such that if a decision about the tested formula was not accepted during the first N_0 steps of the decision procedure, we stop the decision procedure. Under this condition we can see that in Maslov's method for an infinite number of theorems the probability of their proclaming to be a theorem is equal to zero, i.e. "they are not given any chance". In the method proposed in [3],

[4] any formula is given a positive probability to be decided correctly, i.e. to be proclaimed to be tautology if it is a tautology and to be proclaimed not to be a tautology if it is not a tautology.

We believe that in our time one of the most important problem is to compare various theorem proving algorithms and to develope a mathematical theory for such a comparing than to add to the list of theorem proving algorithms one more, having only few hope that it will be better than the other ones. The investigation comparing, under some strictly defined conditions and criterions, the various types of theorem proving algorithms gives the possibility of performing the deep analysis of optimization methods for such algorithms and, at the same time, gives us the limits for such an optimization process. The theory of algorithm complexity and computing processes is rather developed in our time, see [20], neverthless, the investigation of this type were only very seldom applied to the theorem proving algorithms. In fact, only three papers, devoted to this subject are known to the authors: [21]–[23]. In the first of these papers only theorem proving algorithms for the propositional calculus based on the so called resolution method are investigated. In [23] a simple method is described how to transform, for an arbitrary formula of the first order predicate calculus, the proof of this formula obtained by Masłov's inverse method (if it is the case) into the proof of this formula based on the Robinson's resolution principle. In [21] some more detailed connections between the two methods are given. In the same paper also various methods of improvement of resolution method as well as the inverse method are described; these improvements consist mainly in the fact that the class of formulas investigated during the testing of a given formula is limited and minimized to be as small as possible using some auxiliary strategies and tactics. The ratio of the lengths of proofs – measured by the number of applications of deduction rules – is investigated (under some conditions).

The statistical method for comparing theorem proving algorithms is of a rather general nature and can be applied also to the algorithms for theorem proving in theories based on the predicate calculi of higher orders. The "classical" methods for comparing theorem proving algorithms allow to obtain some more general results only in case the both investigated algorithms are very similar to each other, which is just the case of Robinson's and Masłov's results. In general, however, the function transforming the proof of a formula obtained by one method into the proof of the same formula obtained by the other method can be rather complicated and the finding of the class of formulas "preferred" by one algorithm and the class of those "prefered" by the other algorithm can be difficult and in different case different methods must be applied.

The only goal of the remarks concerning the present stage of the theory of automatic theorem proving is to show the most important reasons leading to an attempt to apply statistics in this field of mathematics – not to persuade the reader that the statistical methods are better than the non-statistical ones. The last and the most important argument in favour of one or the other method will be always the results obtained by this

method. Nowadays only the first attempts are made to exploit the basic notions of something what can be later called the statistical theory of automatic theorem proving, so it is very difficult to do some forecasts dealing with the efficiency of such methods. It will be possible only in case a certain mathematical theory is developed enabling to consider and describe several competetive statistical theorem-proving algorithms at the same time. It is also necessary to perform rather a large number of experiments by computers using these methods.*

The experiments having been made until now, using the computers and following a pattern close to that described in this paper, e.g. the testing of some formulas of number theory, have played a very important role during the process of finding some hypotheses having a high degree of probability that they are valid which were later proved using the "classical" methods. Hence, it seems that just the process of discovering some new hypotheses offers a large field of application for statistical methods described in [3], [4], [18], [19] and that just such an application can be the most useful for mathematician's work. Mathematics were defined as a deductive science and no mathematicians will be satisfied knowing that the hypothesis having been discovered by him has only some probability, even if close to one, that it is valid. He will try to prove the hypothesis using the "classical" means used in the investigated theory and will try to embed this hypothesis into the theory as a new assertion. It is why we ought better to use the term "discovering of theorems" rather than "theorem proving" when discussing the statistical methods mentioned above.

Perhaps we can allow to describe the situation in this field using a rather known slogan of St. Mazur: "What does an user of science expect from statistics.? The same what a drunkard expects from a lamp-post: a prop, not a light."

2. A STATISTICAL LINEAR ORDERING RELATION

In the foregoing part of this paper we discussed the present stage in the field of automatic theorem proving and we saw that they were a few principal ideas in this field and a large number of modifications. Any modification is, for certain formulas and from certain aspect better than preceeding ones, however, for some other formulas and from another point of view it may be worse.

* Some fundamentals of a general theory of automatic theorem proving for algorithms investigated in [7], [8], [10], [11] are presented in [24]. In this paper certain formal system is proposed enabling to define in its frame-work the notions like: Theorem-proving algorithm, process of automatic theorem proving etc. A number of assertions is proved showing the characteristic properties of special classes of automatic theorem-proving systems. However, the authors do not know in which degree this formalism can be used in order to obtain some new results dealing with the degree of complexity of theorem-proving algorithms or at least dealing with the comparing of these algorithms under some criterions and conditions interesting from the practical point of view.

Now, let us consider the following situation. Our aim is not to develope a new theorem proving algorithm or to improve some already known one but we have to choose among some given theorem proving algorithms one, being the most adequate for some reasons staying beyond us. This case occurs, for example, if we construct an automaton, a robot. It is necessary to equipe the automaton memory (computer storage) by some program for automatic theorem proving and so the question arises: which is the best algorithm among those being to our disposal? Or, in other words, how to compare two such algorithms?

There are at least two basic aspects from which it is necessary to judge the qualities of theorem proving algorithms.

(a) The first aspect deals with the set of formulas to which the algorithm can be applied. The larger this set is, the better algorithm.

(b) The second aspect deals with the expenses, time sparing, number of operations and so on connected with a decision. We shall suppose in this paper that there exists a unit by which all these expenses can be uniformly measured. If the algorithm is performed using a computer then, for example, the change of the contents of a bit from 0 to 1 or vice versa is such an elementary operation to which every other operations and instructions can be transformed, at least in principle. So again at least in principle, we can ascribe to any formula decidable by the considered algorithm a natural number giving the number of such elementary $0-1$ operations needed for coming to a decision.The smaller this number is, the better algorithm, of course, with respect to the tested formula.

If we limit ourselves to these two aspects and if we consider, instead of formulas, their Gödelian numbers we can see that every algorithm can be described by a partial function defined in the set of all natural numbers and taking its values in the same set. In the following we shall use this functional representation of theorem proving algorithms in order to define some criteria for comparing of such algorithms and to prove some relations among them. Algorithms will be usually denoted by $A, B, \ldots$, the corresponding functions by $f_A, f_B, \ldots$, domains of these functions by $D(A), D(B), \ldots$, in all cases also with indices if necessary. Relations (only binary relations will be considered here) between algorithms will be denoted by $R, S, R(1), R(2), \ldots$, with other indices if necessary. By F the set of all partial functions defined in the set $N = \{0, 1, 2, \ldots\}$ and taking their values in the same set will be denoted, i.e.

$$F = \bigcup_{X \subset N} (N^X) .$$

The notion of partial ordering is supposed to be known to the reader.

Definition 1. Let f_1, f_2 be functions from F, let $D(f_1)$, $D(f_2)$ be their domains, let $X \subset N$. We define the following three binary relations on F:

$f_1 \, \mathrm{R}(1) \, f_2$ iff $D(f_1) \subset D(f_2)$,

$f_1\, R(2)\, f_2$ iff $D(f_j) \subset D(f_2)$ and, at the same time, $f_1(x) \geqq f_2(x)$ for every $x \in D(f_1)$,

$f_1\, R(3)\,(X)\, f_2$ iff $D(f_1) \cap X \subset D(f_2) \cap X$ and, at the same time, $f_1(x) \geqq f_2(x)$ for every $x \in D(f_1) \cap X$.

Definition 2. Let $\mathscr{S}$ be a set, let R, S be two partial orderings on $\mathscr{S}$. The ordering S is called *consistent* with respect to R (in symbols $R \prec S$) if for every $x, y \in \mathscr{S}$ the implication

$$xRy \to xSy$$

holds.

Corollary. $\prec$ is a partial ordering in the set of all partial orderings on $\mathscr{S}$.

We can easily see that the relations $R(1)$, $R(2)$, $R(3)\,(X)$ correspond to usual critetions according to which algorithms for theorem proving are classified. If A, B are two such algorithms, then $f_A\, R(1)\, f_B$ iff B is applicable to the same or larger set of formulas than A no matter which the expenses connected with the decision may be. It gives that B is better than A iff B has a larger domain of applicability. B is better than A with respect to $R(2)$ iff B has a larger (or at least the same) domain of applicability as A and, moreover, the expenses connected with decisions of formulas which are decidable by both A and B are for B at most the same or smaller than for A. If $X \subset N$, then B is better than A with respect to $R(3)\,(X)$ if it is better than A for the formulas from X in the sense of $R(2)$.

Theorem 1.

(a) $R(2)$ is identical with $R(3)\,(N)$.

(b) $R(1)$, $R(2)$, $R(3)\,(X)$, $X \subset N$, are partial orderings in F.

(c) If $X \subset Y \subset N$, then $R(3)\,(Y) \prec R(3)\,(X)$, $R(2) \prec R(3)\,(Y)$.

(d) If $f_1, f_2 \in F$, $D(f_1) \subset D(f_2)$ and for every $x \in D(f_1)$ the relation $f_1(x) = f_2(x)$ holds, then $f_1\, R(2)\, f_2$ and $f_1\, R(1)\, f_2$ are both valid.

(e) $R(1)$, $R(2)$, $R(3)\,(X)$ are not linear orderings on F supposing X has at least two elements.

Proof. Assertions (a)–(d) are trivial. Let $X \subset N$, $Y \subset N$, $X \cap Y \neq X$, $X \cap Y \neq Y$, let $f_1(x) = 1$ if $x \in X$, f_1 undefined otherwise, $f_2(x) \neq 1$ if $x \in Y$, f_2 undefined otherwise. Then neither $f_1\, R(1)\, f_2$ nor $f_2\, R(1)\, f_1$ hold, i.e. $R(1)$ is not a linear ordering. The same example shows that neither $R(2)$ is a linear ordering.

Let $X \subset N$, $\operatorname{card} X \geqq 2$, let $x, y \in X$, $x \neq y$, let $f_1(x) = f_2(y) = 0$, $f_1(y) = f_2(x) = 1$, f_1, f_2 defined arbitratily or undefined on $N - \{x, y\}$. Then neither $f_1\, R(3)\,(X)\, f_2$ nor $f_2\, R(3)\,(X)\, f_1$, i.e. $R(3)\,(X)$ is not a linear ordering.

Assertion (e) expresses the fact that two algorithms A, B cannot be compared, at least by method a formal description of which is given in Definition 1, if there exists at least one formula for which A is better than B and, at the same time, if there

exists at least one formula for which B is better than A. Our reason in the following will be to suggest some linear ordering on set F which would be consistent with $R(1)$, $R(2)$, $R(3)(X)$ in the best possible way.

When applying a theorem proving algorithm in practice there are two possible situations. Either we know a priori to which formulas the algorithm will be applied and in this case we try to choose – if possible – an algorithm "good enough" for these formulas. Or we do not know before to which formulas the algorithm will be applied and we have only some idea about the probability that a formula will be tested – some formulas will have a great probability to be submitted to testing, for some other formulas this probability will be very small, still another formulas will never be decided. The probability with which a formula is submitted to be tested depends on the situation in which the algorithm is applied and we are not able, in general, to change this situation or to influence it. Hence, we must take this situation into consideration and we must use such a theorem proving algorithm which is, in some sense "good" or "adequate" for the formulas having a rather great probability to be tested in the considered situation and which is not necessarily too "good" for the other formulas. This way of reasoning offers to us a way how to construct a well-ordering on F using the "statistical" approach.

Definition 3. Let N_0 be a positive integer or $N_0 = \infty$. Let $\boldsymbol{p} = \{p_1, p_2, \ldots\}$ be a sequence of non-negative reals such that $\sum_{i=1}^{\infty} p_i = 1$ (every such a sequence is called distribution or probability distribution). Let A, B be two theorem proving algorithms. Denote

$$c_A(i) = \min\{f_A(i), N_0\} \quad \text{if} \quad f_A(i) \quad \text{defined}, \quad c_A(i) = N_0 \quad \text{otherwise},$$

$$E(\boldsymbol{p}, N_0)(A) = \sum_{i=1}^{\infty} c_A(i) \cdot p_i,$$

analogously for B. We write

$$f_A \, R(4)(\boldsymbol{p}, N_0) \, f_B \quad \text{iff} \quad E(\boldsymbol{p}, N_0)(A) \geqq E(\boldsymbol{p}, N_0)(B).$$

Remark. If α is a random variable defined on some probability space $(\Omega, \mathscr{S}, P)$ and taking its values in the set $N = \{1, 2, \ldots\}$ of integers we shall write $R(4)(N_0, \alpha)$ instead of $R(4)(\boldsymbol{p}, N_0)$ supposing $\boldsymbol{p} = \{p_1, p_2, \ldots\}$ is such that

$$p_i = P(\{\omega : \omega \in \Omega, \alpha(\omega) = i\}).$$

Detailed explanation of all probabilitic and statistical notions and assertions used in this paper can be found, for example, in [25] or [26].

The intuitive sense of the value $E(\boldsymbol{p}, N_0)(A)$ can be seen in the most simple form if $N_0 = \infty$ and $D(f_A) = N$, because in this case $E(\boldsymbol{p}, N_0)(A) = \sum_{i=1}^{\infty} p_i \cdot f_A(i) = Ef_A$,

which is the expected value of the function f_A understood as a random variable defined on some probability space and taking its values in the set of natural numbers, namely taking the value $f_A(i)$ with the probability p_i. If $D(f_A) \neq N$, it means if algorithm A does not give a result for some formulas, we must decide what to do in such a case. Usually we try for some time to obtain the desired result and after some a priori given number of operations or time units we give up our effort, leaving the formula undecided (in such a case the statistical approach to theorem proving would be useful but we do not intend to investigate this matter here). The number N_0 expresses somehow the expenses and the loss connected with this non-resulting searching. We consider the situation that we stop our work after reaching or overreaching the limit N_0 of expenses in every case – even if $f_A(i)$ is defined but greater than N_0 (we suppose that, in general, we do not know whether $f_A(i)$ is defined or not before deciding the formula with the index i). It is why we consider, in the definition of $E(\boldsymbol{p}, N_0)(A)$ the value $\min\{f_A(i), N_0\}$ instead of $f_A(i)$ even if $f_A(i)$ defined.

Theorem 2. (a) For every probability distribution $\boldsymbol{p} = \{p_1, p_2, \ldots\}$, $p_i \geqq 0$, $\sum_{i=1}^{\infty} p_i = 1$ and for every integer N_0

$$R(2) \prec R(4)(\boldsymbol{p}, N_0) .$$

(b) For every $X \subset N$ there exists a probability distribution $\boldsymbol{p}_X$ such that for every N_0

$$R(3)(X) \prec R(4)(\boldsymbol{p}_X, N_0) .$$

(c) For every algorithm A

$$E(\boldsymbol{p}, N_0)(A) \leqq N_0 .$$

(d) If $N_0 = \infty$, $p_i > 0$ for every $i = 1, 2, \ldots$ and if the considered theory is not decidable by algorithm A, then

$$E(\boldsymbol{p}, N_0)(A) = \infty .$$

(e) For every probability distribution $\boldsymbol{p}$ and every N_0 the relation $R(4)(\boldsymbol{p}, N_0)$ is a linear ordering on F with respect to the equality in the set of all reals.

Proof. Let A, B be two algorithms.

(a) Let $f_A \, R(2) \, f_B$. Then $D(f_A) \subset D(f_B)$, $f_A(i) \geqq f_B(i)$, $i \in D(f_A)$. So we have

$$\begin{aligned} E(\boldsymbol{p}, N_0)(A) &= \sum_{i \in N - D(f_A)} (N_0 \cdot p_i) + \sum_{i \in D(f_A)} \min(N_0, f_A(i)) \cdot p_i \geqq \\ &\geqq \sum_{i \in N - D(f_B)} (N_0 \cdot p_i) + \sum_{i \in D(f_B)} \min(N_0, f_B(i)) \cdot p_i = \\ &= E(\boldsymbol{p}, N_0)(B) , \end{aligned}$$

so (a) holds.

(b) Let $X \subset N$, choose a distribution $\boldsymbol{p}_X = \{p_1, p_2, \ldots\}$ such that $p_i = 0$ for $i \in N - X$. Let $f_A\, R(3)\, (X)\, f_B$. Then

$$E(\boldsymbol{p}, N_0)\,(A) = \sum_{i=1}^{\infty} c_A(i) \cdot p_i = \sum_{i \in X} c_A(i) \cdot p_i = \sum_{i \in X - D(f_A)} (N_0 \cdot p_i) +$$

$$+ \sum_{i \in X \cap D(f_A)} \min\,(N_0, f_A(i)) \cdot p_i \geqq \sum_{i \in X - D(f_B)} (N_0 \cdot p_i) +$$

$$+ \sum_{i \in X \cap D(f_B)} \min\,(N_0, f_B(i)) \cdot p_i = E(\boldsymbol{p}, N_0)\,(B)\,,$$

which proves (b).

(c) $E(\boldsymbol{p}, N_0)\,(A) = \sum_{i \in D(f_A)} \min\,(N_0, f_A(i)) \cdot p_i + \sum_{i \in N - D(f_A)} (N_0 \cdot p_i) \leqq \sum_{i=1}^{\infty} (N_0 \cdot p_i) =$

$= N_0$, so (c) holds.

(d) Considered theory is not decidable by A, i.e. there exists at least one $i_0 \in N$ such that $c_A(i_0) = N_0 = \infty$, so

$$E(\boldsymbol{p}, N_0)\,(A) = \sum_{i=1}^{\infty} c_A(i) \cdot p_i \geqq c_A(i_0) \cdot p_{i_0} = \infty\,, \quad \text{as} \quad p_{i_0} > 0\,.$$

(e) For every $\boldsymbol{p}$ and every N_0 $E(\boldsymbol{p}, N_0)$ is a mapping from F into $\langle 0, \infty\rangle$ therefore for any two algorithms A, B either $E(\boldsymbol{p}, N_0)\,(A) \geqq E(\boldsymbol{p}, N_0)\,(B)$ or $E(\boldsymbol{p}, N_0)\,(B) \geqq$ $\geqq E(\boldsymbol{p}, N_0)\,(A)$ and (e) is also proved.

There is one weak point of the linear ordering relation defined by $R(4)$ which is not too agreeable, namely its substantial dependence on the chosen probability distribution $\boldsymbol{p}$. This weak point of our construction is expressed explicitely in the following assertion.

Theorem 3. Let $f_A, f_B \in F$ be such that neither $f_A\, R(2)\, f_B$ nor $f_B\, R(2)\, f_A$ holds. Then there exist, for some $N_0 < \infty$, two probability distributions $\boldsymbol{p} = \{p_1, p_2, \ldots\}$, $\boldsymbol{q} = \{q_1, q_2, \ldots\}$ such that the assertions

$$f_A\, R(4)\,(\boldsymbol{p}, N_0)\, f_B\,, \quad f_B\, R(4)\,(\boldsymbol{q}, N_0)\, f_A$$

are simultaneously valid. Moreover, the probability distributions $\boldsymbol{p}$, $\boldsymbol{q}$ can be chosen in such a way that for every $i \in N$ $p_i > 0$, $q_i > 0$ hold.

Proof. $(\neg f_A\, R(2)\, f_B) \,\&\, (\neg f_B\, R(2)\, f_A)$ gives that there exist $i, j \in N$ such that $i \in D(f_A) \cap D(f_B)$, $j \in D(f_A) \cap D(f_B)$ $f_A(i) < f_B(i)$, $f_A(j) > f_B(j)$.

Let $N_0 \geqq \max\,(f_B(i), f_A(j)) + 1$. Define $\boldsymbol{p}$ in such a way that $p_k > 0$, $k = 1, 2, \ldots$

$$1 > p_i \geqq \frac{N_0}{f_B(i) - f_A(i) + N_0}$$

(it can be cleary seen that it is possible). We have

$$E(\boldsymbol{p}, N_0)(B) - E(\boldsymbol{p}, N_0)(A) = \sum_{k=1}^{\infty} c_B(k) \cdot p_k - \sum_{k=1}^{\infty} c_A(k) \cdot p_k =$$

$$= \sum_{k \neq i} (c_B(k) - c_A(k)) \cdot p_k + (c_B(i) - c_A(i)) \cdot p_i =$$

$$= \sum_{k \neq i} (c_B(k) - c_A(k)) \cdot p_k + (f_B(i) - f_A(i)) \cdot p_i \geqq$$

$$\geqq \sum_{k \neq i} (-N_0) \cdot p_k + (f_B(i) - f_A(i)) \cdot \frac{N_0}{f_B(i) - f_A(i) + N_0} \geqq$$

$$\geqq (-N_0), \left(1 - \frac{N_0}{f_B(i) - f_A(i) + N_0}\right) + \frac{(f_B(i) - f_A(i)) \cdot N_0}{f_B(i) - f_A(i) + N_0} = 0$$

so we have $f_B \, R(4)(\boldsymbol{p}, N_0) \, f_A$.

Choosing $q_k > 0$, such that

$$1 > q_j \geqq \frac{N_0}{f_A(j) - f_B(j) + N_0}$$

we can prove analogously that $f_A \, R(4)(\boldsymbol{p}, N_0) \, f_B$, hence our assertion is proved.

Theorem 4. Let A, B be two theorem proving algorithms. Let $\mathscr{P}$ be the set of all one-to-one mappings of the set of all natural numbers into itself. Let for any probability distribution $\boldsymbol{p} = \{p_1, p_2, \ldots\}$ and any $\varphi \in \mathscr{P}$ the symbol $\tilde{\varphi}\boldsymbol{p}$ denote the probability distribution

$$\tilde{\varphi}\boldsymbol{p} = \{\tilde{\varphi}p_1, \tilde{\varphi}p_2, \ldots\} = \{p_{\varphi(1)}, p_{\varphi(2)}, \ldots\} .$$

Consider the three following statements:

(a) There exists a probability distribution $\boldsymbol{p}$ such that for all $\varphi \in \mathscr{P}$ the relation $f_B \, R(4)(\tilde{\varphi}\boldsymbol{p}, N_0) \, f_A$ holds.

(b) There exists an index $n_0 \in N$ such that for all $n \geqq n_0$ the inequality $f_A(n) \leqq \leqq f_B(n)$ holds.

(c) There exists a probability distribution $\boldsymbol{p}$ such that for at most countably many $\varphi \in \mathscr{P}$ the relation $f_B \, R(4)(\tilde{\varphi}\boldsymbol{p}, N_0) \, f_A$ does not hold.

Then (a) and (c) are equivalent and both imply (b). If, moreover the set of all integers i for which $f_A(i) = f_B(i)$ is finite, then all the three conditions are equivalent.

Proof. First, we shall prove that (a) implies (b). For this reason suppose that (b) is not valid. Denote

$$C_0 = \{i : a_i = b_i\} ,$$

$$C_1 = \{i : a_i > b_i\} = \{i : a_i \geqq b_i + 1\} ,$$

$$C_2 = \{i : a_i < b_i\} , \quad \text{where} \quad a_i = f_A(i), \, b_i = f_B(i) .$$

It means, we suppose that C_1 is not finite. Let $\boldsymbol{p}$ be any probability distribution. Take an index n_0 such that

$$\sum_{i=n_0+1}^{\infty} p_i < \frac{1}{N_0+2}.$$

There exists a mapping $\varphi \in \mathscr{P}$ such that for all $i \in C_0 \cup C_2$ is $\varphi(i) > n_0$, i.e. if $\varphi(i) \leqq n_0$, then $i \in C_1$ (in fact, there exist uncountably many mappings having this property). Now, we have:

$$\begin{aligned}
\sum_{i=1}^{\infty} a_i \cdot \tilde{\varphi} p_i = \sum_{i=1}^{\infty} a_i \cdot p_{\varphi(i)} &= \sum_{i \in C_0} a_i \cdot p_{\varphi(i)} + \sum_{i \in C_1} a_i \cdot p_{\varphi(i)} + \sum_{i \in C_2} a_i \cdot p_{\varphi(i)} \geqq \\
&\geqq \sum_{i \in C_0} b_i \cdot p_{\varphi(i)} + \sum_{i \in C_1} (b_i + 1) \cdot p_{\varphi(i)} + \sum_{i \in C_2} (b_i - N_0) \cdot p_{\varphi(i)} = \\
&= \sum_{i=1}^{\infty} b_i \cdot p_{\varphi(i)} + \sum_{i \in C_1} p_{\varphi(i)} - N_0 \cdot \sum_{i \in C_2} p_{\varphi(i)} \geqq \\
&\geqq \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi} p_i + \sum_{i=1}^{n_0} p_i - N_0 \cdot \sum_{i=n_0+1}^{\infty} p_i \geqq \\
&\geqq \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi} p_i + 1 - \frac{1}{N_0+2} - N_0 \cdot \frac{1}{N_0+2} = \\
&= \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi} p_i + \frac{1}{N_0+2} \geqq \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi} p_i ,
\end{aligned}$$

hence $f_B\, R(4)\, (\tilde{\varphi}\boldsymbol{p}, N_0)\, f_A$ is not valid. This gives that (a) implies (b).

Second, we prove that (c) and (a) are equivalent in such a way that we prove: if there exists one $\varphi_0 \in \mathscr{P}$ such that $f_B\, R(4)\, (\tilde{\varphi}_0 \boldsymbol{p}, N_0)\, f_A$ does not hold, then there exist uncountably many $\varphi \in \mathscr{P}$ with the same property.

Let $f_B\, R(4)\, (\tilde{\varphi}_0 \boldsymbol{p}, N_0)\, f_A$ does not hold, i.e.

$$\sum_{i=1}^{\infty} a_i \cdot \tilde{\varphi}_0 p_i > \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi}_0 p_i .$$

Let n_0 be such an index that

$$\sum_{i=n_0}^{\infty} \tilde{\varphi}_0 p_i \leqq \Big(\sum_{i=1}^{\infty} a_i \cdot \tilde{\varphi}_0 p_i - \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi}_0 p_i\Big) \cdot \frac{1}{2N_0} .$$

Let $\mathscr{P}' \subset \mathscr{P}$ be the set of all permutations φ such that $\varphi(i) = \varphi_0(i)$ for all $i < n_0$. Clearly, $\mathscr{P}'$ is uncountable. However, $\varphi \in \mathscr{P}'$ implies

$$\sum_{i=1}^{\infty} a_i \cdot \tilde{\varphi} p_i - \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi} p_i = \Big(\sum_{i=1}^{\infty} a_i \cdot \tilde{\varphi}_0 p_i - \sum_{i=1}^{\infty} b_i \cdot \tilde{\varphi}_0 p_i\Big) +$$

$$+\sum_{i=n_0}^{\infty}(a_i - b_i)\,\tilde{\varphi}p_i - \sum_{i=n_0}^{\infty}(a_i - b_i)\,.\,\tilde{\varphi}_0 p_i =$$

$$= -\sum_{i=n_0}^{\infty}(a_i - b_i)\,.\,\tilde{\varphi}_0 p_i + \Big(\sum_{i=1}^{\infty} a_i\,.\,\tilde{\varphi}_0 p_i - \sum_{i=1}^{\infty} b_i\,.\,\tilde{\varphi}_0 p_i\Big) +$$

$$+\sum_{i=n_0}^{\infty}\big(a_{\varphi^{-1}(i)} - b_{\varphi^{-1}(i)}\big)\,.\,p_i = \Big(\sum_{i=1}^{\infty} a_i\,.\,\tilde{\varphi}_0 p_i - \sum_{i=1}^{\infty} b_i\,.\,\tilde{\varphi}_0 p_i\Big) +$$

$$+\sum_{i=n_0}^{\infty}\big(a_{(\varphi_0\varphi^{-1})(i)} - b_{(\varphi_0\varphi^{-1})(i)}\big)\,.\,\tilde{\varphi}_0 p_i -$$

$$-\sum_{i=n_0}^{\infty}(a_i - b_i)\,.\,\tilde{\varphi}_0 p_i \geqq \Big(\sum_{i=1}^{\infty} a_i\,.\,\tilde{\varphi}_0 p_i - \sum_{i=1}^{\infty} b_i\,.\,\tilde{\varphi}_0 p_i\Big) -$$

$$-2N_0\,.\sum_{i=n_0}^{\infty}\tilde{\varphi}_0 p_i \geqq 0\,,$$

according to our assumptions. Here $\tilde{\varphi}p_i$ denotes $p_{\varphi(i)}$, $\tilde{\varphi}_0 p_i$ denotes $p_{\varphi 0(i)}$, φ^{-1} is the inverse mapping with respect to φ and finally $\varphi_0\varphi^{-1}$ denotes the composed mapping, i.e. $(\varphi_0\varphi^{-1})(i) = \varphi_0(\varphi^{-1}(i))$. So $f_B\, R(4)\,(\tilde{\varphi}\boldsymbol{p}, N_0)\, f_A$ does not hold for any $\varphi \in \mathscr{P}'$ and equivalence of (a) and (c) is proved.

Third, let $\{i\colon a_i = b_i\}$ be finite and let (b) hold. This gives that there exists such an index n_1 that for all $n \geqq n_1$ $a_n < b_n$, i.e. $b_n - a_n \geqq 1$. Choose such a distribution $\boldsymbol{p}$ that

$$\max_{1 \leqq i < \infty} p_i < \frac{1}{n_1\,.\,N_0}\,.\sum_{i=n_1+1}^{\infty} p_i\,.$$

Such a distribution always exists, e.g. the distribution with $p_i = 1/(3\,.\,n_1\,.\,N_0)$ for $i \leqq 3\,.\,n_1\,.\,N_0$ and $p_i = 0$ otherwise satisfies the condition, as

$$\max_{1 \leqq i < \infty} p_i = \frac{1}{3\,.\,n_1\,.\,N_0} \leqq \frac{1}{3} < \frac{2}{3} \leqq 1 - \frac{1}{3\,.\,N_0} =$$

$$= (3\,.\,n_1\,.\,N_0 - n_1)\,.\,\frac{1}{3\,.\,n_1\,.\,N_0} = \sum_{i=n_1+1}^{\infty} p_i.$$

As $\sum_{i=1}^{\infty} p_i = 1$, the $\max_{1 \leqq i < \infty} p_i$ for every probability distribution clearly exists. Now

$$\sum_{i=1}^{\infty}(b_i - a_i)\,.\,p_i = \sum_{i=1}^{n_1}(b_i - a_i)\,.\,p_i + \sum_{i=n_1+1}^{\infty}(b_i - a_i)\,.\,p_i \geqq$$

$$\geqq -N_0\,.\sum_{i=1}^{n_1} p_i + \sum_{i=n_1+1}^{\infty} p_i \geqq -N_0\,.\,n_1\,.\max_{1 \leqq i < \infty} p_i + \sum_{i=n_1+1}^{\infty} p_i \geqq 0$$

according to the assumptions. So $f_B\, R(4)\,(\boldsymbol{p}, N_0)\, f_A$ holds. As $\max_{1 \leqq i < \infty} p_i = \max_{1 \leqq i < \infty} \tilde{\varphi}p_i$

for every permutation $\varphi \in \mathscr{P}$ and only $\max_{1 \leqq i < \infty} p_i$ occurs in the above derivation, we have that $f_B\, R(4)\,(\tilde{\varphi}\boldsymbol{p}, N_0)\, f_A$ is valid for every $\varphi \in \mathscr{P}$. So under the mentioned condition (b) implies (a) which gives, that (a), (b) and (c) are equivalent. The theorem is proved.

The theorem which we have just proved is, in some sense, a counterpart of the foregoing one. Theorem 3 expresses the substantial dependence of the criterion $R(4)\,(\boldsymbol{p}, N_0)$ on the probability distribution $\boldsymbol{p}$. The first idea how to minimize or to eliminate such a dependence would be, probably, to require that the condition $f_A\, R(4)\,.\,(\boldsymbol{p}, N_0)\, f_B$ be satisfied not only for one probability distribution $\boldsymbol{p}$ but for all class of distribution however, Theorem 4 shows the limits of this approach, namely: if we require that $f_A\, R(4)\,(\boldsymbol{p}, N_0) f_B$ should be valid for all probability distributions obtainable from $\boldsymbol{p}$ by permutations, then the result is equivalent with the criterion: "B is better than A if B is better than A for almost all formulas". This criterion is not of statistical nature, however, it does not induce a linear ordering. It follows, that wanting to have a linear ordering on the set of all theorem proving algorithms we must admit some degree of dependence on the used probability distribution.

3. A STATISTICAL METHOD FOR COMPARING OF TWO THEOREM PROVING ALGORITHMS WITH RESPECT TO A GIVEN STATISTICAL CRITERION

In the foregoing chapter we suggested a possibility how to order, from statistical point of view, theorem proving algorithms. Namely, we defined the criterion $R(4)$ and we proved that it was a linear ordering in the set of all theorem proving algorithms. It means, that for any given random variable α, integer N_0 and any two algorithms A, B at least one of the relations

$$f_A\, R(4)\,(N_0, \alpha)\, f_B\,, \quad f_B\, R(4)\,(N_0, \alpha)\, f_A$$

is valid. Naturally, the following question now arises: how to decide, in any effective way, which of these two possibilities holds?

This problem is far from being trivial. If we consider some actual theorem proving algorithm it is very difficult for us to express the function f_A in any easy form so that we were able to manipulate with it and, moreover, so that we were able to express in an explicit way the series

$$\sum_{i=1}^{\infty} c_A(i)\,.\,P(\{\omega : \alpha(\omega) = i\})\,.$$

In this chapter we want to suggest some other approach to the problem. Since we have already once used the probabilistic and statistical approach we use probability and statistics still once more and we try to develop a statistical test for testing which relation among the two ones mentioned above is valid. We shall consider only a very

simple statistical testing procedure leaving open the problem of introducing some more developed statistical methods (see [27], [28]) into this field.

The basic idea of our statistical test is rather simple. We sample at random, step by step, a number of formulas and we apply both algorithms A and B to these formulas. Having decided these formulas (or having stopped after N_0 useless steps) we compute the average number of steps needed for deciding by each algorithm. The algorithm for which this average value is smaller we proclaim to be better. Of course, we undertake, in this case, some risk of an error, some risk that our decision has not been correct. In the following we try to formalize this way of reasoning and to express somehow the probability of error connected with it.

Let $\alpha_1, \alpha_2, \alpha_3, \ldots$ be a sequence of random variables defined on the probability space $(\Omega, \mathscr{S}, P)$, taking their values in the set of all natural numbers, mutually independent, equally distributed and such that for every $i = 1, 2, \ldots$

$$P(\{\omega\colon \alpha_j(\omega) = i\}) = P(\{\omega\colon \alpha(\omega) = i\}), \quad j = 1, 2, \ldots$$

where α is an a priori given random variable (i.e. α_j are something like copies of this random variable α). Let N_0 be a given positive integer, let for a partially recursive function f the numbers $c_f(i)$ be defined as above, i.e.

$$c_f(i) = \min(f(i), N_0) \quad \text{if} \quad f(i) \quad \text{defined},$$

$$c_f(i) = N_0 \quad \text{otherwise}.$$

We shall also consider the composed random variables $c_f(\alpha_n(\omega))$, $n = 1, 2, \ldots$

Theorem 5. Let A, B be two theorem proving algorithms. If $f_B\, R(4)\,(N_0, \alpha)\, f_A$ and not $f_A\, R(4)\,(N_0, \alpha)\, f_B$, then

$$P\left(\left\{\omega\colon \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} c_{f_A}(\alpha_i(\omega)) < \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} c_{f_B}(\alpha_i(\omega))\right\}\right) = 1.$$

Proof. Our assertion follows immediately from the so called strong law of large numbers (see, e.g. [25], [26]) according to which

$$P\left(\left\{\omega\colon \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} c_{f_A}(\alpha_i(\omega)) = \sum_{j=1}^{\infty} [c_{f_A}(j) \cdot P(\{\omega\colon \alpha(\omega) = j\})]\right\}\right) = 1,$$

$$P\left(\left\{\omega\colon \lim_{n\to\infty} \frac{1}{n} \cdot \sum_{i=1}^{n} c_{f_B}(\alpha_i(\omega)) = \sum_{j=1}^{\infty} [c_{f_B}(j) \cdot P(\{\omega\colon \alpha(\omega) = j\})]\right\}\right) = 1$$

and from the fact that

$$f_B\, R(4)\,(N_0, \alpha)\, f_A, \quad \text{non}\ f_A\, R(4)\,(N_0, \alpha)\, f_B$$

imply

$$\sum_{j=1}^{\infty} c_{f_A}(j) \cdot P(\{\omega\colon \alpha(\omega) = j\}) < \sum_{j=1}^{\infty} c_{f_B}(j) \cdot P(\{\omega\colon \alpha(\omega) = j\}) .$$

Corollary. Denote

$$H_A(n, \omega) = \frac{1}{n} \cdot \sum_{i=1}^{n} c_{f_A}(\alpha_i(\omega)) , \quad H_B(n, \omega) = \frac{1}{n} \cdot \sum_{i=1}^{n} c_{f_B}(\alpha_i(\omega)) .$$

Then the foregoing theorem implies that with probability one there exists such an index $n_0 = n_0(\omega) = n_0(A, B, \omega)$ that for all $n \geqq n_0$

$$H_A(n, \omega) < H_B(n, \omega) .$$

The foregoing theorem and its corollary offers to us the following way how to define our test:

1. Fix an integer n and compute $H_A(n, \omega)$, $H_B(n, \omega)$.
2. If $H_A(n, \omega) < H_B(n, \omega)$, proclaim:
   "f_B $R(4)$ (N_0, α) f_A *is valid*".
3. If $H_B(n, \omega) \leqq H_A(n, \omega)$, proclaim:
   "f_A $R(4)$ (N_0, α) f_B *is valid*".

In this description, clearly, "to compute $H_A(n, \omega)$" is nothing else than to sample at random and mutually independently n formulas, using the same random generation in every case, to test them or to stop the testing procedure after N_0-th step supposing the decision was not reached and then compute the average number of steps used for one formula. Denoting, for abbreviation, the decision taken in 2. by $\mathscr{D}_1$ and the decision taken in 3 by $\mathscr{D}_2$ we have that our test is a mapping T from the Cartesian product $N \times \Omega$ into the set $\{\mathscr{D}_1, \mathscr{D}_2\}$ (here $N = \{1, 2, 3, \ldots\}$) such that

$$T(n, \omega) = \mathscr{D}_1 \quad \text{iff} \quad H_A(n, \omega) < H_B(n, \omega) ,$$

$$T(n, \omega) = \mathscr{D}_2 \quad \text{iff} \quad H_A(n, \omega) \geqq H_B(n, \omega) .$$

This test is, of course, of statistical nature and it is why the decision taken by this test is not necessarily correct, there is, in general, some probability of error. In the following we give some upper bounds for this probability in order to judge the quality of our test.

We denote

$$E_A = E_A(N_0, \alpha) = \sum_{i=1}^{\infty} c_{f_A}(i) \cdot P(\{\omega\colon \alpha(\omega) = i\}) ,$$

$$D_A = D_A(N_0, \alpha) = \sum_{i=1}^{\infty} (c_{f_A}(i) - E_A)^2 \cdot P(\{\omega\colon \alpha(\omega) = i\}) =$$

$$= \left(\sum_{i=1}^{\infty} (c_{f_A}(i))^2 \cdot P(\{\omega\colon \alpha(\omega) = i\})\right) - (E_A)^2 .$$

Theorem 6. Let A be a theorem proving algorithm. Then for every n, N_0 integers, $\varepsilon > 0$ real the following inequality holds:

$$P(\{\omega: |H_A(n, \omega) - E_A| \geqq \varepsilon\}) \leqq \frac{1}{n \cdot \varepsilon^2} \cdot D_A \leqq \frac{1}{n \cdot \varepsilon^2} \cdot \frac{(N_0 - 1)^2}{2}.$$

Proof. We know that $H_A(n, \omega) = (1/n) \cdot \sum_{i=1}^{n} c_{f_A}(\alpha_i(\omega))$, where $c_{f_A}(\alpha_i(\cdot))$ are mutually independent and equally distributed random variables the expected value of which is

$$E(c_{f_A}(\alpha_i(\cdot))) = \sum_{j=1}^{\infty} c_{f_A}(j) \cdot P(\{\omega: \alpha_i(\omega) = j\}) = E_A .$$

So we can apply the well-know Tchebysheff inequality and we obtain

$$P(\{\omega: |H_A(n, \omega) - E_A|\} \geqq \varepsilon) \leqq \frac{1}{\varepsilon^2} \cdot D(H_A(n, \omega))$$

where $D(H_A(n, \omega))$ (dispersion of the random variable $H_A(n, \cdot)$) is defined by

$$D(H_A(n, \cdot)) = \int_\Omega \left(H_A(n, \omega) - \int_\Omega H_A(n, \omega)\, dP \right)^2 dP = \frac{D_A}{n}$$

according to the Tchebysheff theorem (for Tchebysheff inequality or Tchebysheff theorem see [25] or [26]). From this fact the first part of our inequality immediately follows.

Looking for an upper bound for D_A we remember that the random variables $c_f(\alpha_j(\cdot))$ take only the values from the set $\{1, 2, \ldots, N_0\}$ of integers (we suppose that for every formula at least one step is necessary in order to derive it and, according to our intentions, we stop the decision procedure after N_0-th step supposing the decision was not reached before). It gives that the dispersion D_A is limited by the dispersion of some other random variable ξ taking both the extremal values 1 and N_0 with the probability $1/2$. So we have

$$D_A \leqq D_\xi = \frac{1}{2} \cdot (N_0 - E_\xi)^2 + \frac{1}{2} \cdot (E_\xi - 1)^2 =$$

$$= \frac{1}{2} \cdot \left[\left(N_0 - \frac{N_0 + 1}{2} \right)^2 + \left(\frac{N_0 + 1}{2} - 1 \right)^2 \right] = \frac{1}{2} \cdot (N_0 - 1)^2$$

which gives the last part of our inequality.

Now, we use the obtained inequality in order to estimate somehow the probability of error connected with the test T.

Theorem 7. The following inequalities for conditional probabilities are valid:

$$P(\{\omega\colon \mathscr{D}_1 \text{ is valid}\}/\{\omega\colon T(n,\omega) = \mathscr{D}_1\}) =$$
$$= P(\{\omega\colon f_B\, R(4)\,(N_0, \alpha)\, f_A\}/\{\omega\colon H_A(n,\omega) < H_B(n,\omega)\}) \geqq$$
$$\geqq 1 - \frac{4}{n} \cdot \left(\frac{N_0 - 1}{H_B(n,\omega) - H_A(n,\omega)}\right)^2.$$

If a real $\varepsilon > 0$ is given and if $n > \frac{4}{\varepsilon} \cdot \left(\frac{N_0 - 1}{E_A - E_B}\right)^2$, then

$$P(\{\omega\colon \mathscr{D}_1 \text{ is valid}\}/\{\omega\colon T(n,\omega) = D_1\}) \geqq 1 - \varepsilon\,.$$

Remark. The first inequality gives a lower bound for the probability that the decision taken, according to our test, after having tested n formulas will be correct. There is an advantage of this inequality that it does not depend on E_A, E_B, so we can judge the quality of our test only on the basis of the result $H_A(n,\omega)$, $H_B(n,\omega)$. The inequality enables to minimize the probability of error below an apriori given $\varepsilon > 0$, at the cost of "sufficiently large" n but in order to determine this n it would be necessary for us to know E_A and E_B. This inequality can be useful in such a peculiar situation: we are given two algorithms, we know that one of them is A and the other is B, we know also the functions f_A and f_B, but we do not know, which algorithm is A and which is B and we want to decide this question applying both algorithms to a finite sequence of formulas sampled at random.

Proof. Let the decision $\mathscr{D}_1$ be taken wrongly, i.e. let $H_A(n,\omega) < H_B(n,\omega)$ and, at the same time, $f_A\, R(4)\,(N_0, \alpha)\, f_B$ which means $E_B \leqq E_A$. It follows that either

$$|H_A(n,\omega) - E_A| \geqq \tfrac{1}{2} \cdot (H_B - H_A)$$

or

$$|H_B(n,\omega) - E_B| \geqq \tfrac{1}{2} \cdot (H_B - H_A)\,.$$

However, the foregoing theorem gives

$$P(\{\omega\colon |H_A(n,\omega) - E_A| \geqq \tfrac{1}{2} \cdot (H_B - H_A)\}) < \frac{1}{2n} \cdot (N_0 - 1)^2 \cdot \left(\frac{2}{H_B - H_A}\right)^2,$$

the same inequality being valid for B. So we have

$$P(\mathscr{D}_1 \text{ is valid}/T(n,\omega) = \mathscr{D}_1) \geqq$$
$$\geqq 1 - P(\{\omega\colon |H_A(n,\omega) - E_A| \geqq \tfrac{1}{2} \cdot (H_B - H_A)\} \cup$$
$$\cup \{\omega\colon |H_B(n,\omega) - E_B| \geqq \tfrac{1}{2} \cdot (H_B - H_A)\}) \geqq$$
$$\geqq 1 - [P(\{\omega\colon |H_A(n,\omega) - E_A| \geqq \tfrac{1}{2} \cdot (H_B - H_A)\}) + P(\{\omega\colon |H_B(n,\omega) - E_B)| \geqq$$
$$\geqq \tfrac{1}{2} \cdot (H_B - H_A)\})] \geqq 1 - \frac{4}{n} \cdot \left(\frac{N_0 - 1}{H_B - H_A}\right)^2$$

and the first inequality is proved. Now, if

$$n > \frac{4}{\varepsilon} \cdot \left(\frac{N_0 - 1}{E_B - E_A}\right)^2,$$

then

$$P\left(\left\{\omega: |H_A(n, \omega) - E_A| \geqq \left|\frac{E_B - E_A}{2}\right|\right\}\right) \leqq$$

$$\leqq \left[\frac{4}{\varepsilon} \cdot \left(\frac{N_0 - 1}{E_B - E_A}\right)^2\right] \cdot \left[\frac{E_B - E_A}{2}\right]^{-2} \cdot \frac{(N_0 - 1)^2}{2} = \frac{\varepsilon}{2}$$

and analogously for B. However,

$$P(\mathscr{D}_1 \text{ is correct} | T(n, \omega) = \mathscr{D}_1) \geqq$$

$$\geqq 1 - P\left(\left\{\omega: |H_A(n, \omega) - E_A| \geqq \frac{|E_B - E_A|}{2}\right\} \cup\right.$$

$$\left. \cup \left\{\omega: |H_B(n, \omega) - E_B| \geqq \frac{|E_B - E_A|}{2}\right\}\right) \geqq 1 - \varepsilon$$

and the theorem is proved.

By this theorem we finish our treatment of the statistical criterions for comparing of theorem proving algorithms leaving for some future times the application of some more advanced statistical techniques in this field.

Closing this paper we would like to mention one thing which is, in our opinion, of rather great importance. We have developed our method for statistical comparing theorem-proving algorithms in such a way that we expressed, first, a theorem-proving algorithm in the form of a partially recursive function and, second, we defined a linear ordering relation on the set of all partially recursive functions. Finally, combining these two results, we obtained a linear ordering on the set of all theorem--proving algorithms. However, it can be easily seen that a procedure like this is not limited only to the theorem-proving algorithms. It is possible to generalize our notions, methods and results without any substantial change to any algorithms (e.g. algorithms finding the roots of equations or algorithms solving the diophantic equations etc.) just under the condition that such algorithms can be expressed in or transformed into the form of partially recursive functions.

(Received November 15, 1973.)

[1] A. Špaček: Statistical Estimation of Probability in Boolean Logics. In: Transactions of the Second Prague Conference on Information Theory. Prague 1960, 609—626.

[2] A. Špaček: Statistical Estimation of Semantic Probability. In: Proceedings of the fifth Berkeley Symposium on Mathematical Statistics, 1960, vol. 1, 655—688.

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

[4] I. Kramosil: A Method for Random Sampling of Well-Formed Formulas. Kybernetika *8* (1972), 2, 133—148.

[5] A. Chirch: A note on the Entscheidungsproblem. The Journal of Symbolic Logic *1*, (1936), 40—41.

[6] J. A. Robinson: Theorem-proving on the Computer. Journal of the Assoc. for Comput. Mach. *10* (1963), 163—174.

[7] H. Wang: Toward Mechanical Mathematics. IBM J. Res. Develop. *4* (1960), 2—22.

[8] Gentzen G.: Untersuchungen über das logische Schliessen. Math. Zeit. *39* (1934—35), 176—210.

[9] W. Craig: Linear Reasoning — a New Form of Herbrand — Gentzen Theorem. The Journal of Symbolic Logic *22* (1957), 250—268.

[10] Beth E. W.: Formal Methods. D. Reidel Publishing Company, Dordrecht, 1962.

[11] S. W. Szczerba: Semantic Method of Proving Theorems. Bull. de l'Academie Polonaise des Sciences. Serie des sciences math., astr. et phys. *18* (1970), 9, 507—512.

[12] J. A. Robinson: A Machine Oriented Logic Based on the Resolution Principle J. Assoc. Comput. Mach. *12* (1965), 23—41.

[13] J. A. Robinson: The Generalized Resolution Principle. Machine Intelligence 3, Edinburgh University Press, 1968, 77—93.

[14] S. Ju. Maslov: The Inverse Method for Establishing Deducibility for Logical Calculi. Trudy Matem. Inst. Steklov. Translation: Proc. Steklov Inst. Math. *98*, (1968), 26—87.

[15] Maslov S. Ju.: An Inverse Method of Establishing Deducibility of Non-prenex Formulas of the Predicate Calculus. Translation: Soviet Math. Dokl. *8* (1967), 1, 16—19.

[16] R. Kowalski: An Exposition of Paramodulation with Refinements. Department of Computational Logic, University of Edinburgh, 1968.

[17] J. A. Robinson, S. Wos: Paramodulation and Theorem-Proving in First-Order Theories with Equality. Machine Intelligence 4 (1969), Edinburgh University Press, 135—150.

[18] S. C. van Westrhenen: Statistical Studies of Theoremhood in Classical Propositional and First-Order Predicate Calculus. J. Assoc. Comp. Mach. *19* (1972), 2, 347—365.

[19] S. Ju. Maslov, E. D. Rusakov: Probabilistic Canonical Systems. Translation: Seminars in Mathematics V. A. Steklov Math. Inst., Leningrad, *32* (1972), 66—76.

[20] B. A. Trachtenbrot: Složnosť algoritmov i vyčislenij. Novosibirsk 1967.

[21] G. S. Tseitin: On the complexity of Derivation in Propositional Calculus. Seminars in Mathematics V. A. Steklov Math. Inst., Leningrad, *8* (1970), 115—125.

[22] S. Ju. Maslov: Relationship Between Tactics of the Inverse Method and the Resolution Method. Ibid, *16* (1971), 69—73.

[23] D. G. Kuehner D. G.: A Note on the Relation Between Resolution and Maslov's Inverse Method. Machine Intelligence 6, Edinburgh University Press, 1971, 73—76.

[24] Orłowska E.: Theorem-proving systems. Dissertationes Mathematicae. PWN, Warszawa 1973.

[25] M. Loève: Probability Theory. Second Edition. D. van Nostrand Comp., Princeton, New Jersey—Toronto—New York—London 1961.

[26] B. V. Gnedenko: Kurs teoriji verojatnostej. Third Edition, Firmatgiz, Moskva 1961.
[27] A. Walde: Sequential Analysis. Russian Translation: Fizmatgiz, Moskva 1960.
[28] E. L. Lehman: Statistical Hypotheses Testing. Russian Translation. Nauka, Moskva 1964.

Dr. Ivan Kramosil, CSc.; Ústav teorie informace a automatizace ČSAV (Institute of Information Theory and Automation — Czechoslovak Academy of Sciences), Pod vodárenskou věží 4, 180 76 Praha 8. Czechoslovakia.

Dr. Zbigniew Zwinogrodzki; Instytut Filozofii i Socjologii PAN (Institute of Philosophy and Sociology — Polish Academy of Sciences), ul. Kanonicza 14/25, 31-002 Kraków. Poland.