EXPERIMENTAL EVALUATION OF ONE METHOD OF MINIMIZING THE NUMBER OF STATES OF DISCRETE AUTOMATA

Yu. V. Pottosin

The problem of determining the effectiveness of approximate methods for minimizing the number of states of incompletely specified discrete automata (or, as they are sometimes called, partial automata) is of practical interest. Evaluations of algorithms are necessary, first of all, in choosing the best algorithm for solving some definite class of problems and, in the second place, for verifying the putative improvements borne by algorithms.

To provide analytically an estimate of effectiveness of some method or algorithm is very difficult. It is much easier to determine effectiveness statistically, using an electronic computer for this purpose. A general methodology of statistical investigation is given in [1].

For the statistical evaluation of an algorithm, it is necessary to use it to solve some quantity of examples, and to process the results of these solutions. As the input data, we take the transition and output matrices. In what follows, we shall understand by the transition matrix, $||Z||$, the matrix whose rows correspond to states and whose columns correspond to inputs. At the intersection of a row and a column is placed the value of the transition function, giving the corresponding (next) state and the resulting output. An analogous definition is given to output matrix $||Y||$.

The requisite quantity of examples is obtained by means of an example generator, i.e., a program which randomly constructs transition and output matrices with definite quantitative characteristics. As such characteristics, one can choose $\sigma(X)$, the number of inputs, $\sigma(Z)$, the number of states, $\sigma(Y)$, the number of outputs,* or d, the degree of definiteness.

This latter quantity is defined as the ratio of the number of defined elements of the transition matrix to the number of all its elements, i.e., to the quantity $\sigma(X)\cdot\sigma(Z)$.

In this paper we describe an experiment of a preliminary nature. We investigated an algorithm based on the method of successive reductions of the transition matrix, suggested by Zakrevskii [3].

The nub of the method is as follows. The concept of state compatibility [4] is utilized.

*This symbolism was introduced in [2]. $\sigma(A)$ denotes the cardinality of set A. Our conventions here are that X is the set of inputs, Y is the set of outputs, and Z is the set of states.

V. G. Lazarev et al. (eds.), *Synthesis of Digital Automata / Problemy Sinteza Tsifrovykh Avtomatov /*

The condition of compatibility of two states can itself entail the requirement of pairwise compatibility of a whole series of other states. Therefore, we have to consider the so-called chains of state pairs.

For the given automaton, let there be a chain, C, each pair of which contains mutually compatibile states. We give the name of "one-step" transformation of the given automaton on the basis of chain C" to the transition to another automaton by unifying the states entering into one and the same pair of chain C, with this unification being performed on every pair of the chain.

In the general case, there may be several sets of chains. We eliminate from consideration those chains which would provide no reduction of the number of states if they were used as the bases of transformations. Among the remaining chains, one is selected according to definite criteria. In the algorithm we shall be investigating, the criterion used is that of maximum reduction of number of states. This process of automaton transformation is repeated until no reductions by the given method are possible.

Let us illustrate this method by an example. Suppose we have an automaton whose behavior is described by transition and output matrices of the following form:

$$\| Z \| = \begin{array}{c} {\scriptstyle Z \diagdown X} \\ 0\\1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left|\begin{array}{cccccc} & 2 & 6 & & & \\ & & 3 & 0 & & \\ & & & 2 & 4 & 4 \\ & 4 & & & & \\ & 1 & 1 & & & \\ & & & 6 & & \\ & & & & 7 & 4 \\ 0 & & & & & \end{array}\right| \end{array} ; \quad \| Y \| = \begin{array}{c} {\scriptstyle Z \diagdown X} \\ 0\\1\\2\\3\\4\\5\\6\\7 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left|\begin{array}{cccccc} & 0 & 0 & & & \\ & & 1 & 0 & & \\ & & & 0 & 0 & 0 \\ & 0 & & & & \\ & 1 & 1 & & & \\ & & & 0 & & \\ & & & & 1 & 0 \\ 0 & & & & & \end{array}\right| \end{array}$$

Using our criterion, we take chain $\{<2,3>, <2,4>\}$ as the basis for the one-step transformation. This transformation leads to the following matrices:

$$\| \quad \| = \begin{array}{c} {\scriptstyle Z \diagdown X} \\ 0\\1\\2\\3\\4\\5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left|\begin{array}{cccccc} & 1 & 4 & & & \\ & 2 & 2 & 1 & 1 & 1 \\ & & 0 & 0 & & \\ & & & 4 & & \\ & & & & 5 & 1 \\ 0 & & & & & \end{array}\right| \end{array} ; \quad \| Y \| = \begin{array}{c} {\scriptstyle Z \diagdown X} \\ 0\\1\\2\\3\\4\\5 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left|\begin{array}{cccccc} & 0 & 0 & & & \\ & 1 & 1 & 0 & 0 & 0 \\ & & 1 & 0 & & \\ & & & 0 & & \\ & & & & 1 & 0 \\ 0 & & & & & \end{array}\right| \end{array}$$

State 0 of this new automaton is obtained by unifying states 0 and 3 of the previous automaton, while state 2 is obtained from the unification of previous states 2 and 4. Previous states 1, 5, 6, and 7 go over to the new states 2, 3, 4, and 5.

On the following step, choosing chain $\{<2,3>, <0,4>\}$ as the basis of the transformation, we obtain

$$\| Z \| = \begin{array}{c} {\scriptstyle Z \diagdown X} \\ 0\\1\\2\\3 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left|\begin{array}{cccccc} & 2 & 0 & & 3 & 2 \\ & & 0 & 0 & & \\ & 1 & 1 & 2 & 2 & 2 \\ 0 & & & & & \end{array}\right| \end{array} ; \quad \| Y \| = \begin{array}{c} {\scriptstyle Z \diagdown X} \\ 0\\1\\2\\3 \end{array} \begin{array}{c} \begin{array}{cccccc} 0 & 1 & 2 & 3 & 4 & 5 \end{array} \\ \left|\begin{array}{cccccc} & 0 & 0 & & 1 & 0 \\ & & 1 & 0 & & \\ & 1 & 1 & 0 & 0 & 0 \\ 0 & & & & & \end{array}\right| \end{array}$$

And, finally, unifying states 0 and 3 for the last step, we get

$$\|Z\| = \begin{array}{c|cccccc} Z \diagdown X & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 2 & 0 & & 0 & 2 \\ 1 & & & 0 & 0 & & \\ 2 & & 1 & 1 & 2 & 2 & 2 \end{array}; \qquad \|Y\| = \begin{array}{c|cccccc} Z \diagdown X & 0 & 1 & 2 & 3 & 4 & 5 \\ \hline 0 & 0 & 0 & 0 & & 1 & 0 \\ 1 & & & 1 & 0 & & \\ 2 & & 1 & 1 & 0 & 0 & 0 \end{array}$$

It is easy to see that no further reduction of the number of states is possible. In this example, the initial matrices were obtained by a computer equipped with a pseudo-random code generator. The operator providing the transition and output matrices is presented below.

The algorithm under investigation here is described in the LYaPAS language [2, 5] in the form of an L-operator assigned the code mipsokmas (minimization by the method of successive reduction, using the criterion of maximal reduction for choosing the transformation). In second-level LYaPAS, this operator has the following description:

$$\begin{array}{ll}
\text{mipsokmas} & (272,\ 324,\ 301,\ 314)/\alpha\,\text{п}+,\ \beta\,\text{к},\ \gamma\,\text{п}+,\ \delta\,\text{к},\ \varepsilon\,\text{к},\ \zeta\,\text{к}, \\
 & \eta\,\text{к},\ \vartheta\,\text{к},\ \varkappa\,\text{к},\ \lambda\,\text{к},\ \mu\,\text{к}\,\nu\,\text{п}+,\ \xi\,\text{п}+,\ \pi\,\text{п}+,\ \rho\,\text{п}+,\ \sigma\,\text{п}+/\boldsymbol{u}, \\
 & i/(\zeta\eta\vartheta\varkappa\mu) \\
 & \mathbf{313}\quad \mathbf{205}\quad \mathbf{004}\quad (\zeta\boldsymbol{q}) \\
 & \overline{\bigcirc}\,\boldsymbol{u}\ \gamma - 1 \Rightarrow a\ 0 - c_a \Rightarrow \boldsymbol{q}\ \bigcirc i \\
\S\ 1 & b_\zeta - 1 \Rightarrow b_\varkappa\ \text{ekvioych}\ \delta_\varkappa\ (b_\varkappa)\,\sigma\pi \,\|\, \boldsymbol{u} \circ\!\!\rightarrow 3 \\
\S\ 2 & \gamma - 1 \Rightarrow b_\varkappa \Rightarrow b_\nu \Rightarrow b_\eta \Rightarrow b_\varepsilon\ \text{vykmas}\ 4\varepsilon\nu\pi\sigma\varkappa\vartheta\zeta\eta\boldsymbol{q}\gamma \,\| \\
\text{premav} & \delta\zeta\lambda\gamma \,\|\, \bigcirc\,\boldsymbol{u} \rightarrow 1 \\
\S\ 3 & \gamma \Rightarrow b_\eta\, b_\zeta \Rightarrow b_\mu \Rightarrow b_\xi\ \text{oshpremap}\ \varepsilon\zeta\eta\varkappa\vartheta\beta\xi\nu\alpha\rho\mu\lambda \,\| \\
 & \Delta i b_\zeta \Rightarrow \gamma - 1 \Rightarrow a\ 0 - c_a \Rightarrow \boldsymbol{q} \rightarrow 2 \\
\S\ 4\ . &
\end{array}$$

Operator é k v i v y c h builds binary matrix $\|A\|$ of equivalence relationships with respect to the outputs. Two states are said to be equivalent with respect to outputs if the values of the output functions for these states coincide for all inputs, where both these values are defined. Operator v y k m a s performs the choice of the one-step transformation, operator p r e m a v transforms the output matrix, and operator o s h p r e m a p transforms the transition matrix. All these operators are described in [6]. Also introduced there are the so-called ω-constants, related to the representation of $\|Y\|$ and $\|Z\|$ by arrays δ and ε. Here, the ω-constants are the operands α, ν, ξ, π, ρ, and σ. The number of states is specified by variable γ. The remaining operands, i.e., β, λ, ζ, η, θ, $\varkappa$, and μ, are auxiliary arrays, necessary for the operations of the L-operators.

The program realizing the foregoing operator, and compiled by the LYaPAS programming system for the M-20 computer, contains 843 commands.

We now consider operator m a p e r v y k h which builds the transition and output matrices ($\|Z\|$ and $\|Y\|$) by using the pseudo-random code generator contained in the LYaPAS system. This operator is used in our experiment as a generator of examples.

The number of inputs and the number of states are specified indirectly by means of the ω-constants used in the system of minimization operators.

In the functioning of operator m a p e r v y k h there participate constants ω_5, ω_6, and ω_7, corresponding to operands ζ, α, and β, and defined by the relationships:

$$\begin{aligned}
&\omega_5^i = 1 \leftrightarrow i = p - k, \\
&\omega_6^i = 1 \leftrightarrow i \in \{0,\ k,\ 2k,\ \ldots,\ (\sigma(X) - 1)k\}, \\
&\omega_7^i = 1 \leftrightarrow 0 \leqslant i < k,
\end{aligned}$$

where ω_j^i is the i-th bit of code ω_j; p is the number of bits in the variables and indices; k is the number of bits necessary for the coding of an element of matrix $||Z||$.

The number of outputs is specified by the value of operand δ, whose i-th bit is determined as follows:

$$\delta^i = 1 \leftrightarrow i \in \{0, k-l, k-l+1, \ldots, k, 2k-l, 2k-l+1, \ldots, 2k, \ldots, mk-l, mk-l+1, \ldots, mk-1\},$$

where $l = \log_2 \sigma(Y)$; $m = \sigma(X) - 1$.

Matrices $||Z||$ and $||Y||$ are represented by arrays γ and ε, respectively:

$$\text{mapervykh}\ \alpha\,\text{п}+,\ \beta\,\text{п}+,\ \gamma\,\text{к},\ \delta\,\text{п}+,\ \varepsilon\,\text{к},\ \zeta\,\text{п}+/\boldsymbol{a}, a$$
$$\mathbf{337 \quad 43 \quad 1}$$
$$\bigcirc\, a$$
$$\S\ 1 \qquad \text{я} \Rightarrow \boldsymbol{a} \wedge \alpha \circ\!\!\rightarrow 1\, \bar{x}\, \beta < 1 \wedge \boldsymbol{a} \Rightarrow \gamma_a \wedge \delta \Rightarrow \varepsilon_a\, \Delta a - \zeta \circ\!\!\rightarrow 1.$$

The numbers of outputs and states for the matrices obtained by means of this operator are integral powers of two. The degree of definiteness, d, can deviate from its average value:

$$d_{av} = \frac{1}{2\left(1 - 2^{-\sigma(X)}\right)}.$$

These deviations are due to the randomness in the construction of matrix $||Z||$. The values of the elements of the transition matrix, obtained by means of operator mapervykh, are distributed uniformly, i.e., if an element is defined, then its value can, with identical probability, be the ordinal number of any state.

All told, ten series of examples were solved on the M-20 computer. Each series contained no fewer than 25 examples with identical values of $\sigma(X)$, $\sigma(Y)$, and $\sigma(Z)$. The solution time for one example from the series with characteristics $\sigma(X) = 5$, $\sigma(Y) = 4$, and $\sigma(Z) = 32$ was no more than 30 seconds.

Parallel with the solution of each example, the number of steps, n, was counted while, at the conclusion, the number of states, $\sigma(Z')$, obtained as the result of minimization was determined. For the character of the distribution of these quantities within one series, see Fig. 1, which gives the histograms of the values of number of states (Fig. 1a) and number of steps (Fig. 1b) for 60 examples with the characteristics $\sigma(X) = 6$, $\sigma(Y) = 4$, and $\sigma(Z) = 16$.

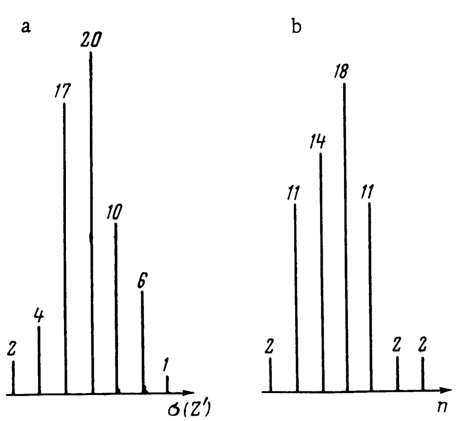

Fig. 1

For the other series, the forms of these histograms were qualitatively almost the same, so we shall not give them here. For each series of examples we computed the mean values of n, of $\sigma(Z')$, and of the magnitude of reduction per step, p. The results for n_{av} and $\sigma_{av}(Z')$ are shown graphically on Fig. 2. The average magnitude of reduction per step for different series of examples is given graphically on Fig. 3. When $\sigma(X) = 6$, $\sigma(Y) = 4$, and $\sigma(Z) = 16$, this quantity was 1.13, while for the remaining series of examples it oscillated between 1.38 and 1.84.

The character of the distribution of $\sigma(Z')$ and n (Fig. 1) shows that the mean values of these

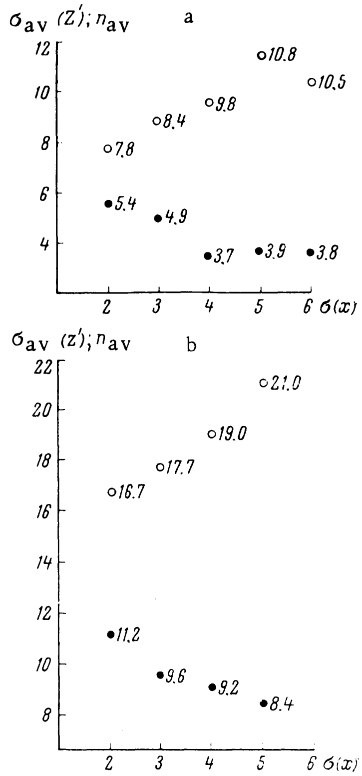

Fig. 2.

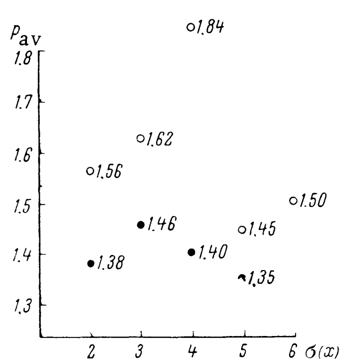

Fig. 3

quantities do not differ by much from the most probable values. The same may be said of P_{av}, since it is related to $\sigma(Z')$ and n. Therefore, by taking the previously adduced data into account, one can say that the magnitude of reduction of the number of states per step is most frequently 1 or 2, despite the criterion, used in the given algorithm, of choosing the one-step transformations providing the maximal reduction on each step. This fact can be used in the following manner.

Consider the operator for chain construction, contained in operator vykmas. Chain C is represented by the binary matrix $||C||$, called the chain matrix. An element of this matrix $c_{ij} = 1$ if and only if $\langle i,j\rangle \in C$ and $i < j$. We shall say that pair $\langle i,j\rangle$ implies pair $\langle k,l\rangle$ if there exists a column of matrix $||Z||$ with ordinal number m such that $\langle z_{im}, z_{jm}\rangle = \langle k,l\rangle$. The process of chain construction reduces to that of finding all the pairs implied by each pair entering into this chain [4]. For the construction of matrix $||C||$, we construct an auxiliary matrix, $||C'||$, for which $c'_{ij} = 1$ if and only if $\langle i,j\rangle \in C$; the set of all pairs implied by pair $\langle i,j\rangle$ has still not been found, and $i < j$. The presence of at least one unit element in matrix $||C'||$ indicates that construction of the chain is still incomplete. Using the coordinates of such elements, one finds the corresponding rows of matrix $||Z||$ and completes matrix $||C||$.

For finding the unit elements of matrix $||C'||$, we use operator nakedél [6], which does this by scanning all the rows of the matrix. However, the unit elements in matrix $||C'||$ cannot be more numerous than the pairs of chains and as was stated earlier, this number is always small. Therefore, by orienting ourselves to the most frequent case in the set of automata in question, we do well to introduce a new operator which would allow us to shorten the process of finding the unit elements. We give this operator the code élemed (search for unit elements of a matrix). For it, we must specify the set of nonzero rows, which we represent by variable ε. The matrix itself is represented by array β, while the coordinates of a unit element are given by indices γ and δ. If all the elements of the given matrix equal zero, a transfer is executed to the external program from the supplementary output terminal α:

elemed αч, βк, γи, δи /—, —
335 27 2
§ 1 ε $\mathfrak{X}$ $\alpha\gamma$
§ 2 β_γ $\mathfrak{X}$ 1δ $c_\gamma \vee \varepsilon \Rightarrow \varepsilon$.

It is convenient to use this operator only in those cases when the number of unit elements in the matrix is small. In the other cases, it is more effective to use nakedél.

The new operator t s e c h (construction of chains of state pairs for partial automata) is described by the following expression:

tsech $(317,\ 335)/\alpha\,\text{ч},\ \beta\,\text{к}+,\ \gamma\,\text{и},\ \delta\,\text{и},\ \varepsilon\,\text{к},\ \zeta\,\text{п}+,\ \eta\,\text{п}+,\ \vartheta\,\text{п}+,$
$\varkappa\,\text{к},\ \lambda\,\text{п}+/\boldsymbol{d},\ d/\ (\varepsilon\varkappa\mu)$
$\mathbf{321\quad 120\quad 4}\quad (\varepsilon c \boldsymbol{d})$
$\bigcirc\, \boldsymbol{d}$ impach $1\beta\gamma\delta\lambda\zeta\eta\vartheta ab//$ élemed $4\varepsilon\gamma\delta\boldsymbol{d}//$
$\rightarrow$ impach 1
§ 1 $c_a \Rightarrow \boldsymbol{c} \wedge \varkappa_b | \rightarrow$ impach $3 \rightarrow \alpha$
§ 2 $\boldsymbol{c} \vee \varepsilon_b \Rightarrow \varepsilon_b \vee \varkappa_b \Rightarrow \varkappa_b\ c_b \vee \boldsymbol{d} \Rightarrow \boldsymbol{d}$
§ 3 $\rightarrow$ impach 3
§ 4 .

Here, operator i m p a c h [6] finds pairs of states implied by a given pair. Matrix ||Z|| is represented by array β. Indices γ and δ correspond to the ordinal numbers of the states of the given pair which generates the chain. Arrays ε and $\varkappa$ represent matrices ||C'|| and ||C||, respectively. λ represents the number of states in the automaton, while ζ, η, and θ are ω-constants.

Due to the changes in operator t s e c h, operator v y k m a s must also undergo certain changes. The new expression for operator v y k m a s takes the form:

vykmas $(321,\ 335)\ /\alpha\,\text{ч},\ \beta\,\text{к}+,\ \gamma\,\text{п}+,\ \delta\,\text{п}+,\ \varepsilon\,\text{п}+,\ \zeta\,\text{к}+,\ \eta\,\text{к},\ \vartheta\,\text{к}$
$\varkappa\,\text{к},\ \lambda\,\text{п}+,\ \mu\,\text{п}+/\ \boldsymbol{p},\ f/\ (\zeta\eta\vartheta\varkappa\lambda)$
$\mathbf{324\ 412\ 12}\ (\zeta \boldsymbol{c f i l n})$
$\bigcirc e\ \bigcirc \boldsymbol{e}\ \bigcirc \boldsymbol{f}\ \overline{\bigcirc} \boldsymbol{k}\ \zeta_e \Rightarrow \boldsymbol{l}\ \bigcirc a$
§ 1 $\bigcirc \vartheta_a\ \bigcirc \varkappa_a\ \Delta a - b_\zeta \circ\!\!\rightarrow 1$
§ 2 $\boldsymbol{l}\,\mathfrak{X}\,7\ f \Rightarrow d\ e \rightarrow c\ c_f \Rightarrow \vartheta_e \vee c_e \Rightarrow \boldsymbol{i}\ 1 \Rightarrow \boldsymbol{bf}\ \bigcirc \underline{\boldsymbol{n}}$
tsech $5\beta dc\ \varkappa\gamma\delta\varepsilon\nu\mu\ //\bigcirc a\ \boldsymbol{i} \nabla \Rightarrow \boldsymbol{m}\ \boldsymbol{j} - \boldsymbol{m} | \rightarrow 1\ \ 2\bar{x}\boldsymbol{j} - \boldsymbol{m} \Rightarrow \boldsymbol{d}$
$\boldsymbol{m} - \boldsymbol{j} \Rightarrow \boldsymbol{h} - \boldsymbol{e} \circ\!\!\rightarrow 1 | \rightarrow 3\ \ \boldsymbol{d} - \boldsymbol{k} | \rightarrow 1$
§ 3 $\boldsymbol{d} \Rightarrow \boldsymbol{k}\ \ \boldsymbol{i} \Rightarrow \boldsymbol{f}\ \ \boldsymbol{h} \Rightarrow \boldsymbol{e}\ \bigcirc a\ \ \boldsymbol{n} \Rightarrow \boldsymbol{p}$
§ 4 $\nu_a \Rightarrow \eta_a\ \bigcirc \nu_a\ \Delta a - b_\zeta \circ\!\!\rightarrow 4 \rightarrow 2$
§ 5 $\boldsymbol{c} \wedge \zeta_b \circ\!\!\rightarrow 6\ \vee \boldsymbol{n} \Rightarrow \boldsymbol{n} \vee\ c\ \vee\ \boldsymbol{i} \Rightarrow \boldsymbol{i}\Delta \boldsymbol{j} \rightarrow$ tsech 2
§ 6 $c_c\ \urcorner \wedge \zeta_d \Rightarrow \zeta_d c_f\ \urcorner \wedge \zeta_e \Rightarrow \zeta_e\ \bigcirc a \rightarrow 1$
§ 7 $\Delta e - b_\zeta | \rightarrow 10\ \ \zeta_e \Rightarrow \boldsymbol{l} \rightarrow 2$
§ 10 $\boldsymbol{f} \circ\!\!\rightarrow \alpha\ \bigcirc c\ \boldsymbol{f} \oplus \lambda \Rightarrow \boldsymbol{a}$ élemed 11 $\eta ab\boldsymbol{p}//c_a \vee c_b \Rightarrow \nu_c \Delta c$
$\rightarrow$ élemed 2
§ 11 $\boldsymbol{a}\,\mathfrak{X}\,12\,a\ c_a \Rightarrow \dot{\nu}_c \Delta c \rightarrow 11$
§ 12 $c \Rightarrow b_\nu$.

The initial data for this operator are matrices ||A|| and ||Z||, represented by arrays ζ and β. Variables γ, ε, δ, and λ represent ω-constants, while variable μ represents the number of states. Matrices ||C|| and ||C'|| are represented by arrays η and $\varkappa$. The results of the operations of operator v y k m a s are given in the form of transformation matrix ||G||. This is a binary matrix whose rows correspond to the new states, while its columns correspond to the old ones, where $g_{ij} = 1$ if and only if old state j is transformed to new state i. Matrix ||G|| is represented by array θ. If there exists no chain on the basis of which a transformation would lead to a reduction in the number of states, a transfer from the operator is executed from the supplementary output terminal α.

On Fig. 2 one can clearly discern the dependence of the average minimal number of states obtainable by minimization with the given method, as well as the mean number of steps, on the number of automaton inputs. It can be assumed that, if the number of examples in each series is increased, one will obtain more precise curves and, if the number of series is increased, these dependencies might be expressed analytically. This, in its turn, would make it possible to extrapolate the capabilities of the algorithm to more complicated cases, where the setting up of similar experiments would require a significant expenditure of machine time.

LITERATURE CITED

1. V. I. Ostrovskii, "Development of methods of evaluating algorithms on the basis of statistical experiments on digital computer," in: Account of Topic 6.1 of the Plan for International Cooperation within the Borders of SEV, Tr. Sib. Fizikotekhn. Inst., Tomsk (1965).
2. Logical Language for the Representation of Algorithms for the Synthesis of Relay Devices, Izd. "Nauka" (1965).
3. A. D Zakrevskii, "On synthesizing sequential automata," Tr. Sib. Fizikotekhn. Inst., Vol. 40 (1961).
4. M. C. Paull and S. H. Unger, "Minimizing the number of states in incompletely specified sequential switching functions," IRE Trans., EC-8, No. 3 (1959).
5. A. D. Zakrevskii, "Automation of the synthesis of discrete automata on the basis of algorithmic language LYaPAS," in: Computing Systems, Vol. 18, Novosibirsk (1965).
6. Yu. V. Pottosin, "Algorithms for minimizing the number of states of a discrete automaton," in: Logical Language for the Representation of Algorithms for the Synthesis of Relay Devices, Izd. "Nauka" (1966).