Ján Černý; Jan Vinař

On Simple Stochastic Models

Matematický časopis, Vol. 20 (1970), No. 4, 293--303

Persistent URL: http://dml.cz/dmlcz/126546

Matematický časopis 20 (1970), No. 4

ON SIMPLE STOCHASTIC MODELS

JÁN ČERNÝ, JAN VINAŘ, Košice

This paper consists of two parts. The first part deals with the construction of stochastic transformers using finite deterministic automata. These transformers will be used in the second part to construct models of finite stochastic automata.

1. STOCHASTIC TRANSFORMERS

Gill [1] and Sheng [8] have considered the problem of using a suitably defined deterministic automaton as a stochastic transformer. In this section, some of their results (obtained in part independently by the authors) are presented insofar as they are used later on.

1.1. NOTIONS AND NOTATION

The automaton $\mathscr{A} = (A, X, Y, \delta, \lambda)$ is a finite, deterministic, non-initial Moore automaton whose states (input signals, output signals) form the sets A, X, Y, respectively; δ is the state transition function, λ the output function.

At every moment $t = 0, 1, 2\ldots$ the automation is in the state $a(t)$, is receiving the input signal $x(t)$ and producing the output signal $y(t)$ such that

$$\begin{aligned} a(t+1) &= \delta(a(t), x(t)) \\ y(t) &= \lambda(a(t)). \end{aligned}$$

The transition function $\delta : A \times X \to A$ can be extended to $\mathscr{P}(A) \times \bigcup_{n=0}^{\infty} X^n$, where $\mathscr{P}(A)$ is the set of all subsets of A. Namely, let $B \in \mathscr{P}(A)$, $x \in X$, then

$$\delta(B, x) = \bigcup_{b \in B} \delta(b, x),$$

and if $p \in \bigcup_{n=0}^{\infty} X^n$, then

$$\delta(B, px) = \delta(\delta(B, p), x).$$

Analogously, the function λ can be generalized to a function defined on $\mathscr{P}(A)$ whose values are in $\mathscr{P}(Y)$. Namely, if $B \subset A$, then

$$\lambda(B) = \{\lambda(a) : a \in B\}.$$

For a detailed description of such automata see [3]. Following [5] we shall call such an automaton n-stable if for every $x_1, \ldots, x_n \in X^n$ there exists an $a \in A$ such that $\delta(A, x_1, \ldots, x_n) = \{a\}$. The automaton is weakly n-stable if for every $x_1 \ldots x_n \in X^n$ there exists $y \in Y$ such that $\lambda(\delta(A, x_1 \ldots x_n)) \quad \{y\}$.

A discrete probability space is the ordered triple $(X, \boldsymbol{X}, \mu_0) = \mathfrak{X}_0$, where X is a finite set, μ_0 a probability measure on the σ-algebra $\boldsymbol{X}$ of all subsets of X. An independent source belonging to $\mathfrak{X}_0$ is the probability space $(X^N, \boldsymbol{X}^N, \mu) = \overset{\infty}{\underset{n=1}{\mathsf{X}}} (X, \boldsymbol{X}, \mu_0)$. If $E \in X^N$ is the elementary cylinder determined by the condition that the first n symbols are the letters $x_1, x_2, \ldots, x_n$ in a given order, then $\mu(x_1 \ldots x_n)$ denotes $\mu(E)$.

Remark: All these notations and definitions are in accordance with [4].

Now let $\mathfrak{X} = (X^N, \boldsymbol{X}^N, \mu)$ and $\mathfrak{Y} = (Y^N, \boldsymbol{Y}^N, \nu)$ be independent sources, $\mathscr{A} = (A, X, \overline{Y}, \delta, \lambda)$ a weakly n-stable automaton, $\overline{Y} \subset Y$ and $\varepsilon > 0$. We say that $\mathscr{A}$ transforms $\mathfrak{X}$ into $\mathfrak{Y}$ with the maximum error ε and the slowing factor n if for all $y \in Y$

$$|\nu(y) - \sum \mu(x_1 \ldots x_n)| < \varepsilon,$$

where the sum is taken over the set $\{x_1 \ldots x_n : \lambda(\delta(A, x_1 \ldots x_n)) = y\}$. Clearly by restricting our attention to weakly n-stable automata we evade the necessity of considering the convergence of transition matrices (cf. [1]). If we choose every n-th output symbol, the independence of the output source is ensured automatically by the independence of the input source.

1.2. ABSTRACT SYNTHESIS OF STOCHASTIC TRANSFORMERS

For abstract synthesis we shall use suitable decompositions of X^n.

Let $\mathfrak{X} = (X^N, \boldsymbol{X}^N, \mu)$, $\mathfrak{Y} = (Y^N, \boldsymbol{Y}^N, \nu)$ be two independent sources belonging to (X, μ_0) and (Y, ν_0). Further let $\varepsilon > 0$ and $n \in N$ be given. The decomposition $\{E_y\}_{y \in Y}$ of the set X^n is a $(\mathfrak{X}, \mathfrak{Y}, \varepsilon, n)$ — decomposition if for every $y \in Y$

$$\Big| \sum_{x_1 \ldots x_n \in E_y} \mu(x_1 \ldots x_n) - \nu(y) \Big| < \varepsilon.$$

Lemma 1. *If $\{E_y\}_{y \in Y}$ is a $(\mathfrak{X}, \mathfrak{Y}, \varepsilon, n)$ — decomposition, the there exists an n-stable automaton $\mathscr{A} = (A, X, Y, \delta, \lambda)$ which transforms $\mathfrak{X}$ into $\mathfrak{Y}$ with the maximum error ε and the slowing factor n.*

Proof. We use the well-known algorithm of [6] to construct the following n-stable automaton $\mathscr{A} = (A, X, \overline{Y}, \delta, \lambda)$:

$$A = X^n,\ X \text{ as given},\ \overline{Y} = Y,$$

$$\delta((x_1 \dots x_n), x) = (x_2 \dots x_n x) \text{ for all } x_1 x_2 \dots x_n x \in X^{n+1},$$

$$\lambda(x_1 \dots x_n) = y \text{ if and only if } x_1 \dots x_n \in E_y.$$

Clearly $\mathscr{A}$ is n-stable and the rest follows by definition.

Lemma 2. *For every $\mathfrak{X}$, $\mathfrak{Y}$, ε there exists an $n \in N$ and a decomposition $\{E_y\}_{y \in Y}$ of X^n such that $\{E_y\}_{y \in Y}$ is a $(\mathfrak{X}, \mathfrak{Y}, \varepsilon, n)$ —decomposition.*

Proof: Let $X = \{x_1, \dots, x_m\}$, $p = \max \mu_0(x)$ and $n > (\log \varepsilon - \log 2)/\log p$. Then $p^n < \varepsilon/2$ and $\mu(x_1 \dots x_n) \leqslant p^n < \varepsilon/2$. We denote the elements of the (finite) sets $X^n = \{s^1, \dots, s^{m^n}\}$, $Y = \{y^1, \dots, y^h\}$. Evidently we can find numbers $j_0, \dots, j_h$, $1 = j_0 \leqslant j_1 \leqslant \dots \leqslant j_h \leqslant m^n + 1$, such that

$$\left| \sum_{i=j_{k-1}}^{j_k - 1} \mu(s^i) - \nu(y^k) \right| < \varepsilon.$$

Now we put $E_y k = \{s^{j_{k-1}} \dots, s^{j_k - 1}\}$ and $\{E_y k\} k = 1$ is the required decomposition.

Theorem 1. (A consequence of the lemmas 1, 2). *For every $\mathfrak{X} = (X^N, \mathbf{X}^N, \mu)$ and $\mathfrak{Y} = (Y^N, \mathbf{Y}^N, \nu)$, $\varepsilon > 0$, there exists a set $\overline{Y}$, $\overline{Y} \supset Y$, and an n-stable automaton $\mathscr{A} = (A, X, Y, \delta, \lambda)$ which transforms $\mathfrak{X}$ into $\mathfrak{Y}$ with the maximum error ε and the slowing factor n.*

Remark: The estimate for n given by Lemma 2 is rather pessimistic. As a rule we can take n much smaller.

1.3. STRUCTURAL SYNTHESIS

In constructing the required automaton, it is possible to use one-tact delay elements and standard logical elements with two inputs (though there may be some advantages in using threshold logic as Sheng [8] pointed out). The memory (cf. [3]) of the automaton will consist of n delay elements joined serially, while the combination part will realize the function

$$f(x_1, \dots, x_n) = y \text{ if and only if } x_1, \dots, x_n \in E_y.$$

The automaton must furthermore contain a clock, i. e. a counter mod n which lets out only every n-th output symbol. The complete structure of the automaton is shown in Fig. 1.

Example 1. Let $X = Y = \{0, 1\}$; $\mu_0(0) = 0.25$, $\mu_0(1) = 0.755$, $v_0(0) = 0.535$, $v_0(1)$ 0.465, $e = 0.01$.

Lemma 2. gives $n = 18$. Let us, however, compute directly the measures of elements of X, X^2, X^3:

X: $\mu(0) = 0.25$, $\mu(1) = 0.75$.
X^2: $\mu(00) = 0.0625$, $\mu(01) = \mu(10) = 0.1875$, $\mu(11) = 0.5625$.
X^3: $\mu(000) = 0.015625$, $\mu(001) = \mu(010) = \mu(100) = 0.046875$, $\mu(011) = \mu(101) = \mu(110) = 0.140625$, $\mu(111) = 0.421875$.

We see immediately that there exists a suitable decomposition of X^3, namely, $E_1 = \{111, 001\}$, $E_0 = X^3 - E_1$. Then $\mu(E_0) = 0.531$, $\mu(E_1) = 0.469$ (to 3 decimals). $\{E_0, E_1\}$ is therefore a $(\mathfrak{X}, \mathfrak{Y}; 0.01; 3)$ — decomposition of X^3 and the combination part must be constructed to realize the function

$$f(x_1, x_2, x_3) = (x_1x_2x_3 \vee \bar{x}_1\bar{x}_2x_3) = (x_1x_2 \vee \bar{x}_1\bar{x}_2)x_3.$$

The automaton — without the clock — is shown in fig. 2.

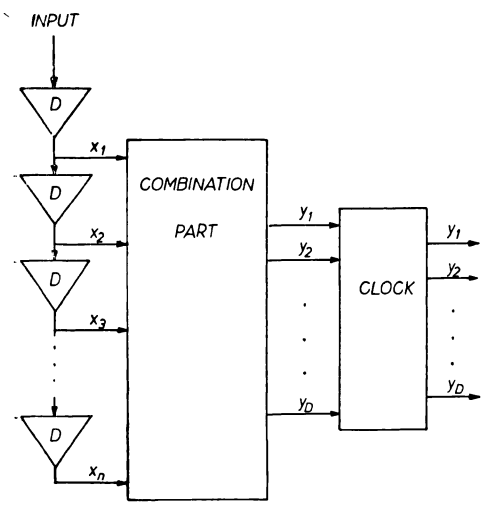

Fig. 1

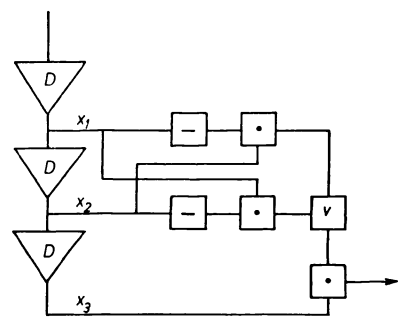

Fig. 2

The notion of a Moore automaton can be generalized in the following way: Let $Y^* = \bigcup_{n=0}^{\infty} Y^n$ be the set of all output strings (Y^0 is the set consisting of the empty string e). We shall call the system $\mathscr{G} = |A, X, Y, \delta, \lambda|$, where A, X, Y and δ have the same meaning as in the definition of an automaton and λ is a function defined on A with values in Y^*, a non-initial generalized sequential machine (ngsm).

Remark: This definition is different from that used by Ginsburg and Rose [2]; they speak of a gsm which is not non-initial since it has a distinguished initial state. Moreover, they use a function $\lambda : A \times X \to Y^*$ which is a generalization of the Mealy-rather than Moore — automaton.

In other words the ngsm differs from our automaton in that to one input symbol there may correspond a sequence (empty or not) of output signals. This means that input and output of a ngsm have, in general, a different time scale.

Now let us consider the system Moore automaton — clock as defined above. The clock is a finite automaton whose states can be divided into two groups: „good" states which permit the passage of the output signal from the Moore automaton, and „bad" states which block this signal. A composition of these two automata will have the set of states $A \times C$, where C is the set of the clock states. The definition of the transition function is trivial. Let us define a ngsm with the above mentioned set of states and transition function. Its input (output) signal sets will be identical to those of the Moore automaton. Its output function will be defined as follows for $a \in A$, $c \in C$,

$$\lambda([a, c]) = \begin{cases} \lambda(a), & \text{if } c \text{ is a „good" state,} \\ e, & \text{if } c \text{ is a „bad" state.} \end{cases}$$

Then obviously the output of this ngsm will exhibit the characteristics of a random source with the desired signal probabilities. We can therefore use a ngsm to transform any source into an ε — approximation of any other source. The fact that we do not require the number of input signals to be equal the number of output signals permits us to evade considerations of the slowing factor.

Remark: Let us consider the Moore automaton with the state and input signal sets the same as in the ngsm just mentioned. Its output set $\bar{Y}$ will contain, in addition to all output signals of the Moore automaton, the natural zero signal Λ and the output function will be

$$\begin{aligned} \lambda(a, c]) &= \lambda(a) \quad \text{if } c \text{ is a „good" state,} \\ \lambda(a, c]) &= \Lambda \quad \text{if } c \text{ is a „bad" state.} \end{aligned}$$

This construction gives us an abstract automaton which „approximates" the given source.

1.4. GLOBAL PROPERTIES OF ε-APPROXIMATIONS

Let us consider the following.

Example 2: In some random source, let $y \in Y$, $\mu_0(y) = 1$. Then $\mu(y^n) = 1$ regardless of n. If in another source $\nu_0(y) = 1 - \varepsilon$, then for every $\varepsilon > 0$ $\lim_{n \to \infty} \nu(y^n) = 0$.

This example shows that even a very good „symbol — by — symbol" ε-approximation can exhibit a behaviour strikingly different from that of the original source if we consider the set of all strings emitted by these sources. This problem will now be investigated in greater detail.

Let $0 \leqslant q < p < 1$. Then

$$p^n - q^n = (p - q)(p^{n-1} + p^{n-2}q + \ldots q^{n-1}) \leqslant (p - q)np^{n-1}.$$

The function $f(x) = xp^{x-1}$, $x \in (0, \infty)$, has an absolute maximum for $x = -\frac{1}{\ln p}$ and therefore $\frac{p^n - q^n}{p - q} \leqslant -\frac{1}{\ln p} \cdot p^{\frac{1+\ln p}{\ln p}} = \varphi(p)$.

Consider two sources with the same output alphabet $y_1, \ldots, y_m$. Let $p_1 p_2 \ldots p_m$ be the probabilities of these symbols in the first source, $q_1 q_2 \ldots q_m$ those for the second source. Let $\varepsilon = \max_i |p_i - q_i|$

$$p = \max_i \max (p_i, q_i).$$

Consider the string $y_{k_1} y_{k_2} \ldots y_{k_n}$. Its probability for the first source is $p_{k_1} p_{k_2} \ldots p_{k_n}$, for the second source it is $q_{k_1} q_{k_2} \ldots q_{k_n}$. We shall now find an upper bound for

$$W = |p_{k_1} \ldots p_{k_n} - q_{k_1} \ldots q_{k_n}|.$$

Let $r_{k_i} = \max (p_{k_i}, q_{k_i})$, $s_{k_i} = \min (p_{k_i}, q_{k_i})$. Then

$$W \leqslant r_{k_1} \ldots r_{k_n} - s_{k_1} \ldots s_{k_n}.$$

Of course the relations $r_{k_i} \leqslant p$, $|r_{k_i} - s_{k_i}| \leqslant \varepsilon$ hold. In the sequel we shall use the following assertion:

Let $a_i \geqslant b_i \geqslant 0$, $\varkappa_i \geqslant 0$, $(i = 1, 2, \ldots, n)$. Then

$$\prod_{i=1}^{n} a_i - \prod_{i=1}^{n} b_i \leqslant \prod_{i=1}^{n} (a_i + \varkappa_i) - \prod_{i=1}^{n} (b_i + \varkappa_i).$$

Now $r_{k_i} \leqslant p$. Let $t_{k_i} = s_{k_i} + (p - r_{k_i})$.

Obviously for $i = 1, \ldots, n$

$$t_{k_i} \geqslant p - \varepsilon.$$

Finally,

$$W \leqslant r_{k_1} r_{k_2} \ldots r_{k_n} - s_{k_1} s_{k_2} \ldots s_{k_n} \leqslant p^n - t_{k_1} \ldots t_{k_n} \leqslant p^n - (p - \varepsilon)^n \leqslant \varepsilon \cdot \varphi(p).$$

Thus the change in the probability of any string caused by taking an ε-approximation of the original source does not exceed $\varepsilon \cdot \varphi(p)$, provided $p < 1$. For $p = 1$, $\lim_{n \to \infty} p^n - (p - \varepsilon)^n = 1$ and, as we have shown in example 2, we can construct a string whose probabilities in the two sources differ by a number arbitrarily close to 1.

Calculation shows that $\varphi(p)$ is an increasing function of p for $0 \leqslant p \leqslant 1$. It tends to zero as $p \to 0$ and to infinity as $p \to 1$. The following table illustrates the behaviour of $\varphi(p)$:

p	$= 0.5$	0.6	0.7	0.8	0.9	0.95	0.99
$\varphi(p)$	$= 1{,}06$	1.20	1.86	2.06	3.89	7.37	36.60.

Therefore if we want to get a „string" ε-approximation, all we have to do is to construct a „symbol — by — symbol" $\varepsilon/\varphi(p)$-approximation where p is the greatest probability in our problem.

Example 3: We have the same sources as in example 1. Here $p = 0.535$ and $\varphi(p) \leqslant 1{,}20$. But $0.535 - 0.531 = 0.004$ and $0.469 - 0.465 = 0.004$. Therefore our automaton is also a „string" 0.01 — approximation.

2. MODELS OF STOCHASTIC AUTOMATA

We shall define a stochastic automaton $\mathscr{S} = (A, X, F)$ to be a finite Moore stochastic automaton (cf. [7]) with its states (input signals) forming the sets $A(X)$ with $n(p)$ elements respectively, and with the transiton mapping F. The set Y of the output signals will be identical with A, and the (deterministic) M-mapping will be such as to generate an identical mapping of A into Y.

The function — valued mapping $F[a, x]$, defined on $A \times X$, maps $A \times X$ into a set of probability distributions on A. If at the moment t the automaton is in the state a_0 and accepts subsequently the inputs $x(t), \ldots, x(t + k - 1)$, then in $t + 1, \ldots, t + k$ its states will be $a_1, \ldots, a_k$ with the probability $\prod_{i=0}^{k-1} F[a_i, x(t + l)]\,(a_{i+1})$.

Note: The automaton just described is not the most general device of this nature. There are, however, two good reasons for investigating this particular automaton: a) while it is particularly simple, it can be, in a certain sense, equivalent to any other stochastic automaton ([7]), b) all other stochastic automata can be described by matrix schemes essentially similar to the one used here.

2.1. A DETERMINISTIC MODEL

We shall now show that we can build a model of a stochastic automaton using a deterministic automaton and some kind of independent random signal sources.

Let a stochastic automaton $\mathscr{S} = (A, X, F)$ be given. The set $A \times X$ is evidently finite and its element can be numbered. Let $h(a, x)$ $(a \in A,\ x \in X)$ be the ordinal number of the element $[a, x]$ in some ordering of this set. We construct a set of independent signal sources $\boldsymbol{a}_1, \ldots, \boldsymbol{a}_{np}$, where $\boldsymbol{a}_{h(a,x)}$ generates the symbol $a' \in A$ with the probability $F[a, x]\,(a')$. Now let us construct a deterministic Moore automaton $\mathscr{A}' = (A', X', Y', \delta, \lambda)$, where $A' = A$, $X' = X \times A^{pn}$, $Y' = A'$, $\lambda(a) = a$ for every $a \in A'$ and $\delta(a, [x, a_1, \ldots, a_{np}]) = a_{h(a,x)}$. Evidently, if we consider only the first component x of the input

signal, then the probability of the state transition from a to a' is $F[a, x]\,(a')$.

To construct this deterministic automaton we use the canonical structural synthesis algorithm of [3]. According to this algorithm, the automaton consists of two parts:

1. The memory element — this is a deterministic Moore automaton with the set of states $B \supset A$ and a complete transition system.

This means that a function $\mu(a, a') : A \times A \to X$ can be defined on all $A \times A$ in such a way that $d(a, \mu(a, a')) = a'$.

2. The combinatory circuit which has the following function: upon accepting the input signal x and the signal a denoting the state of the automaton, it feeds into the memory element input the signal $\mu(a, \delta(a, x)) = \varrho(a, x)$. In this case $\varrho(a, x) = \mu(a, \delta(a, [x, a_1, \ldots, a_{np}])) = \mu(a, a_{h(a,x)})$.

Note: The variables $a_1, \ldots, a_{np}$ are considered a part of the system and therefore do not appear in the input.

2.2. APPROXIMATION OF STOCHASTIC AUTOMATA

Definition: *Consider the stochastic automata* $\mathscr{S} = (A, X, F)$ *and* $\mathscr{S}' = |A, X, F'|$. *The latter is an* ε*-approximation of the former if* $|F[a, x]\,(a') - F'[a, x]\,(a')| < \varepsilon$ *for all* $a \in A$, $a' \in A$, $x \in X$.

We have now at our disposal all the means necessary to construct an ε-approximation of any stochastic automaton. Consider especially a stochastic automaton with only one input signal. This automaton will behave like a Markov source of multiplicity 1. This solves the question of approximating (in the above sense) such sources. The whole process will be best illustrated by an example.

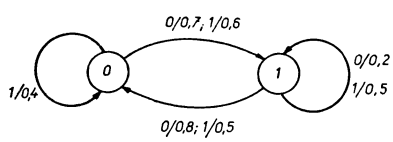

Fig. 3

Example 4. Consider a stochastic trigger (Fig. 3). To construct a 0.01-approximation, we use a battery of random signal sources in Fig. 4a. It is left to the reader to verify that it has the characteristics of Fig. 4b. (Provided of course that we take only every fourth output signal to insure independence).

These generators we shall use to construct the automaton of Fig. 5 which is obviously a 0,01 — approximation of the previous automaton. A comparison with Tab. 1 gives us

$$h(0.0) = 1,\ h(0.1) = 2,\ h(1.0) = 3,\ h(1.1) = 4$$

Our memory element need have only two states: 0 and 1. We can use, for instance, a delay element in which $\mu(a, b) = b$. Then our combination circuit must realize the function

$$\varrho(a, x) = a_{h(a,x)},$$

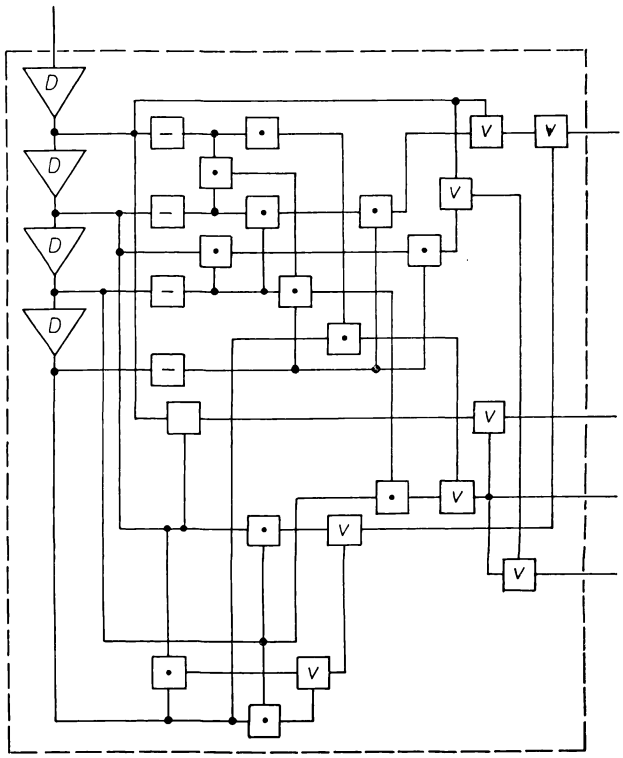

Fig. 4a

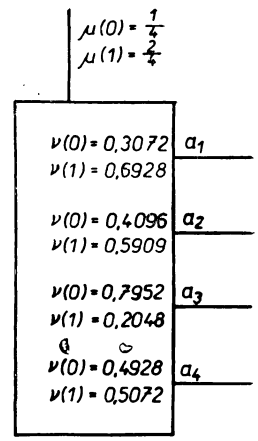

Fig. 4b

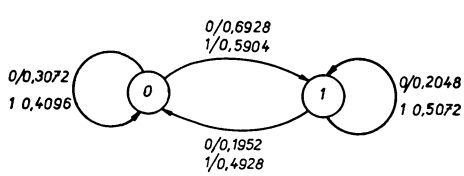

Fig. 5

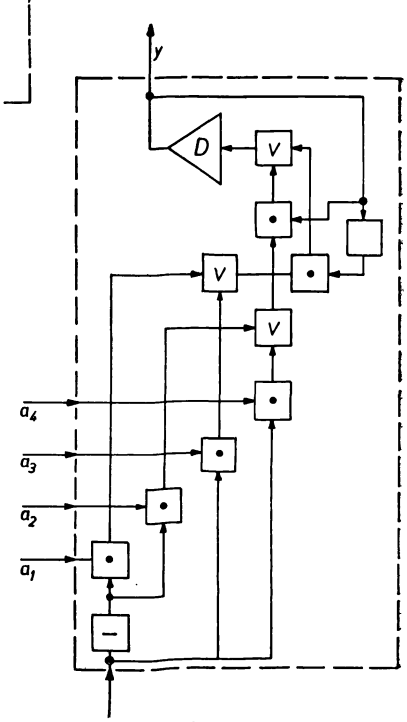

Fig. 6

which gives the following table of its activity (Tab.1) Here the $\sim$'s denote the „don't care" elements of the function. The required function may be

$$\varrho(a, x, a_1, a_2, a_3, a_4) = \bar{a}\bar{x}a_1 \vee \bar{a}xa_2 \vee a\bar{x}a_3 \vee axa_4 = \bar{a}(\bar{x}a_1 \vee xa_2 \vee a(\bar{x}a_3 \vee xa_4)$$

and Fig. 6 shows the combination circuit.

Tab. 1.

a	x	a_1	a_2	a_3	a_4	ϱ
0	0	0	$\sim$	$\sim$	$\sim$	0
0	0	1	$\sim$	$\sim$	$\sim$	1
0	1	$\sim$	0	$\sim$	$\sim$	0
0	1	$\sim$	1	$\sim$	$\sim$	1
1	0	$\sim$	$\sim$	0	$\sim$	0
1	0	$\sim$	$\sim$	1	$\sim$	1
1	1	$\sim$	$\sim$	$\sim$	0	0
1	1	$\sim$	$\sim$	$\sim$	1	1

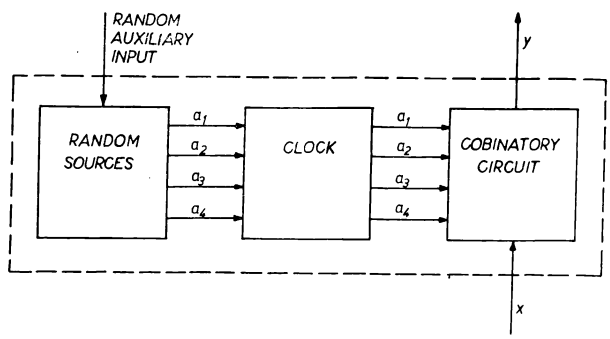

Fig. 7

Now if we connect the source battery, the clock element and the combination circuit as in Fig. 7, we get a system behaving exactly as the automaton of Fig. 5. All it needs to function is a random independent signal source whose output is 0 or 1 with the probabilities 1/4 and 3/4, respectively. This is connected to an auxiliary input which constitutes the only difference (externally observable) between our system and the automaton in Fig. 5.

Note. Similarly as in § 1.3 we could construct a single deterministic abstract automaton equivalent to the system in Fig. 7.