Tomus 2. Fasciculus 4.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1976

Curat: Universitas Szegediensis de Attila József nominata

2. kötet 4. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1976. július

A Szegedi József Attila Tudományegyetem gondozásában

Tomus 2.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1976

Curat: Universitas Szegediensis de Attila József nominata

INDEX

Tomus 2.

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

Representation of automaton mappings in finite length

By F. GÉCSEG in Turku*)

In [3] we introduced a family of products, and in some cases it was decided for two such products whether one of them is a real generalization of the other one with respect to the homomorphic representation of automata. In this paper we investigate similar problems concerning representations of automaton mappings in finite length.

To make this paper self-contained we recall the notions and notations of automata theory used in our later discussions.

By a *finite automaton* we mean a system $\mathbf{A}=(X, A, Y, \delta, \lambda)$, where X, A and Y are finite (nonvoid) sets, called input, state and output sets, respectively. δ denotes the transition function and λ is the output function of $\mathbf{A}$.

Let $F(X)$ denote the free monoid generated by the input set X. The transition function δ can be extended to $A\times F(X)$ in the following way: for any $p=p'x\in F(X)$ and $a\in A$, $\delta(a,p)=\delta(\delta(a,p'),x)$. In the sequel we use the more convenient notation $ap_{\mathbf{A}}$ for $\delta(a,p)$. If there is no danger of confusion we omit the index $\mathbf{A}$.

Take an $a\in A$. We define a mapping $f_{\mathbf{A},a}\colon F(X)\to F(Y)$ in the following way: for any $p=x_1x_2...x_n\in F(X)$, let $f_{\mathbf{A},a}(p)=y_1y_2...y_n$ where $y_1=\lambda(a,x_1)$, $y_2=\lambda(ax_1,x_2)$, $\ldots, y_n=\lambda(ax_1...x_{n-1},x_n)$. This $f_{\mathbf{A},a}$ is called the *mapping induced by* $\mathbf{A}$ *in the state* a. For convenience, further on we give an automaton in the form $\mathbf{A}=(X, A, a_0, Y, \delta, \lambda)$ if we are interested in $f_{\mathbf{A},a_0}$, and use the notation $f_{\mathbf{A}}$ for $f_{\mathbf{A},a_0}$. In this case it is said that $\mathbf{A}$ is an *initial automaton* with the *initial state* a_0.

A mapping $f\colon F(X)\to F(Y)$ $(|X|,|Y|<\aleph_0)$ is called an *automaton mapping* if there exists a (not necessarily finite) automaton $\mathbf{A}=(X, A, a, Y, \delta, \lambda)$ such that $f=f_{\mathbf{A}}$. Moreover, let n be a natural number. We say that $\mathbf{A}$ *induces* f *in length* n if $f(p)=f_{\mathbf{A}}(p)$ for all $p\in F_n(X)$, where $F_n(X)$ denotes the set of all input words of $\mathbf{A}$ with length nonexceeding n.

If we omit the output set and output function of an automaton $\mathbf{A}=(X, A, Y, \delta, \lambda)$ then we get the *semiautomaton* belonging to $\mathbf{A}$. Thus, a semiautomaton has the form $\mathbf{A}=(X, A, \delta)$. Let n be a natural number, and for an initial semiautomaton $\mathbf{A}=(X, A, a, \delta)$ set $A^{(n)}=\{ap\mid p\in F_n(X)\}$. Take two semiautomata $\mathbf{A}=(X, A, a, \delta)$

*) On leave from the University of Szeged, Hungary.

and $\mathbf{B}=(X, B, b, \delta')$. Then a mapping τ of $A^{(n)}$ onto $B^{(n)}$ is called an *n-homomorphism* of $\mathbf{A}$ *onto* $\mathbf{B}$ if $\tau(ap)=bp$ holds for any $p\in F_n(X)$.

One can easily prove the following:

Lemma 1. Take an automaton $\mathbf{B}=(X, B, b, Y, \delta', \lambda')$ and let $\mathbf{A}'=(X, A, a, \delta)$ be a semiautomaton. Assume that for a natural number n, there exists an n-homomorphism of $\mathbf{A}'$ onto $\mathbf{B}'=(X, B, b, \delta')$. Then there is a mapping λ: $A\times X\to Y$ such that $\mathbf{A}=(X, A, a, Y, \delta, \lambda)$ induces $f_{\mathbf{B}}$ in length $n+1$.

In the sequel by an automaton (semiautomaton) we always mean a finite automaton (semiautomaton).

Let $\mathbf{A}_i=(X_i, A_i, Y_i, \delta_i, \lambda_i)$ $(i=1, \ldots, n)$ be arbitrary automata, X and Y finite (nonvoid) sets. Moreover, take two mappings

$$\varphi\colon A_1\times\ldots\times A_n\times X\to X_1\times\ldots\times X_n$$

and

$$\varphi'\colon A_1\times\ldots\times A_n\times X\to Y.$$

Then it is said that the automaton $\mathbf{A}=(X, A, Y, \delta, \lambda)$ with $A=A_1\times\ldots\times A_n$ is the *(general) product* of $\mathbf{A}_1, \ldots, \mathbf{A}_n$ with respect to X, Y, φ and φ' if

$$\delta((a_1, \ldots, a_n), x) = (\delta_1(a_1, x_1), \ldots, \delta_n(a_n, x_n))$$

and

$$\lambda((a_1, \ldots, a_n), x) = \varphi'(a_1, \ldots, a_n, x)$$

hold for any $(a_1, \ldots, a_n)\in A$ and $x\in X$, where $(x_1, \ldots, x_n)=\varphi(a_1, \ldots, a_n, x)$ (cf. [4]). For this product we shall use the short notation $\mathbf{A}=\prod_{i=1}^{n}\mathbf{A}_i\,[X, Y, \varphi, \varphi']$. The general product of semiautomata can be defined analogously; as it is determined by the input set X and the feedback function completely, we can write $\mathbf{A}=\prod_{i=1}^{n}\mathbf{A}_i[X, \varphi]$ in this case. If $X=X_1\times\ldots\times X_n$ and $\varphi(a, x)=x$ $(a\in A,\ x\in X)$ then we speak of a *direct product*. Moreover, if $\varphi(a, x)$ is independent of a for any $a\in A$ then $\mathbf{A}$ is called a *quasi-direct product*.

Let α be a mapping of the set N of all natural numbers into itself such that $\alpha(i)\geqq i$ for all $i\in N$. A product $\mathbf{A}=\prod_{i=1}^{n}\mathbf{A}_i[X, Y, \varphi, \varphi']$ is an *α-product* if φ can be given in the form

$$\varphi(a_1, \ldots, a_n, x) = (\varphi_1(a_1, \ldots, a_n, x), \ldots, \varphi_n(a_1, \ldots, a_n, x))$$

such that φ_i $(1\leqq i\leqq n)$ is independent of states having indices greater than or equal to $\alpha(i)$. We denote by α_j $(:N\to N; j=0, 1, \ldots)$ the mapping for which $\alpha_j(i)=i+j$ $(i\in N)$. It can be proved (cf. [1]) that the α_0-product is the same as the loop-free composition introduced in [5].

Take a natural number n. An automaton $\mathbf{A}=(X, A, a, Y, \delta, \lambda)$ is called n-free if $ap\neq aq$ for all $p, q\in F_n(X)$ with $p\neq q$.

The following result is obvious.

Lemma 2. Take two semiautomata $\mathbf{A}=(X, A, a, \delta)$ and $\mathbf{B}=(X, B, b, \delta')$. If $\mathbf{A}$ is n-free then there exists an n-homomorphism of $\mathbf{A}$ onto $\mathbf{B}$.

We say that the α_i-product is *metrically equivalent* to the α_j-product (general product) if for any natural number n and system Σ of automata, a mapping f can be induced in length n by an α_i-product of automata from Σ if and only if f can be induced in length n by an α_j-product (general product) of automata from Σ.

Now we are ready to prove the following

Theorem. For all $i=0, 1, \ldots,$ the α_i-product is metrically equivalent to the general product.

Proof. Let Σ be a system of automata. Take a (general) product $\mathbf{A}= =(X, A, a_0, Y, \delta, \lambda)= \prod_{j=1}^{k} \mathbf{A}_j[X, Y, \varphi, \varphi'] \ (\mathbf{A}_j \in \Sigma)$. By Lemma 1, it is enough to show that there exists an α_0-product $\mathbf{B}=(X, B, b, Y, \delta', \lambda')$ of automata from Σ such that for any natural number n, the semiautomaton $\mathbf{B}=(X, B, b, \delta')$ can be mapped n-homomorphically onto $\mathbf{A}'=(X, A, a_0, \delta)$. Thus in the sequel we may confine ourselves to semiautomata, i.e., we can assume that Σ consists of semiautomata.

For a semiautomaton $\mathbf{C}^*=(X, C, \delta_{\mathbf{C}}^*)$ we say that a state $c \in C$ is *ambiguous* if there are $x, x' \in X$ such that $\delta_{\mathbf{C}}^*(c, x) \neq \delta_{\mathbf{C}}^*(c, x')$. Let n be a fixed natural number, and let $u<n$ be the greatest number for which there exist a $\mathbf{C}=(Z, C, \delta_{\mathbf{C}}) \in \Sigma$, $c \in C$ and $p \in F(Z)$ with $|p|=u$ such that cp is ambiguous. ($|p|$ denotes the length of p.) Assume that there exists such a u. Then for all $t \leqq u$ there are states $c_t \in C$ and words $p_t \in F(Z)$ with $|p_t|=t$ such that $c_t p_t$ are ambiguous. (Indeed, c_t and p_t can respectively be chosen as cq_t and q_t', where q_t is the prefix of p with $|q_t|=u-t$ and q_t' is the suffix of p with $|q_t'|=t$.)

First we construct a $(u+1)$-free semiautomaton $\mathbf{D}=(W, D, d, \delta_{\mathbf{D}})$ as an α_0-product of semiautomata from the one-element set $\{\mathbf{C}\}$, where $W=\{w_1, w_2\}$. For each $c_t \in C$ $(t=0, 1, \ldots, u)$ choose two inputs $z_t, z_t' \in Z$ such that $\delta_{\mathbf{C}}(c_t p_t, z_t) \neq \neq \delta_{\mathbf{C}}(c_t p_t, z_t')$. Form the α_0-product $\mathbf{D}_1=(W, D_1, d_1, \delta^{(1)})=\mathbf{C}[W, \varphi^{(1)}]$, where $d_1=c_u$ and for all $d \in D_1$ and $w_r \in W$

$$\varphi^{(1)}(d, w_r) = \begin{cases} z_u & \text{if} \quad r=1, \\ z_u' & \text{if} \quad r=2. \end{cases}$$

It is obvious that $\mathbf{D}_1$ is a 1-free semiautomaton. Now assume that for all $m \leqq s (\leqq u)$ we have constructed an m-free α_0-power $\mathbf{D}_m=(W, D_m, d_m, \delta^{(m)})$ of $\mathbf{C}$. Furthermore, suppose that $p_s=\bar{z}_1 \ldots \bar{z}_s$ and let l be a natural number such that $2^l \geqq 2^{s+1}+s+1$. Take the l-th direct power $C'=(Z', C', c, \delta_{\mathbf{C}}')$ of $\mathbf{C}$, where $c=(c_s, \ldots, c_s)$. Moreover, let $\bar{Z}$ be the subset of Z' consisting of all elements $\bar{z}$ whose each component is either z_s or z_s', and $cp_s\bar{z} \neq cp_s'$ for any prefix p_s' of p. Denote by $\mathbf{D}_{s+1}=(W, D_{s+1}, d_{s+1}, \delta^{(s+1)})$ the α_0-product $(\mathbf{D}_s \times \mathbf{C}')[W, \varphi^{(s+1)}]$, where $d_{s+1}=(d_s, c)$ and for any $p, q \in F(W)$, $w \in W$ and $c', c'' \in C'$,

(i) $\varphi_1^{(s+1)}(d_s p, c', w)=w$,

(ii) $\varphi_2^{(s+1)}(d_s p, c', w)=\bar{z}_{v+1}$ if $|p|=v<s$,

(iii) if $|p|=|q|=s$ then $\varphi_2^{(s+1)}(d_s p, c', w) \in \bar{Z}$ such that

$$\varphi_2^{(s+1)}(d_s p, c', w) \neq \varphi_2^{(s+1)}(d_s q, c'', w') \quad \text{if} \quad (d_s p, w) \neq (d_s q, w').$$

($\varphi_2^{(s+1)}(d_s p, c', w)$ with $|p|=s$ can be chosen in this way, since $|\bar{Z}| \geqq 2^{s+1}$.)

(iv) in all other cases $\varphi^{(s+1)}$ is defined arbitrarily such that the resulting product is an α_0-product.

We prove that $\mathbf{D}_{s+1}$ is an $(s+1)$-free semiautomaton. Take two words $p, q \in F(W)$ with $p \neq q$. Now let us distinguish the following three cases:

1) $|p|, |q| \leqq s$. Then $d_s p \neq d_s q$ since $\mathbf{D}_s$ is an s-free semiautomaton. Therefore, $d_{s+1} p = (d_s, c) p \neq (d_s, c) q = d_{s+1} q$.

2) $|p| = v \leqq s$ and $|q| = s+1$. Let us assume that $q = q'w$ $(w \in W)$. Then by the definition of $\varphi^{(s+1)}$, $(d_s, c)p = (d_s p, c\bar{z}_1 \dots \bar{z}_v)$ and $(d_s, c)q = (d_s q, cp_s \varphi_2^{(s+1)}(d_s q', cp_s, w))$. Again, by the definition of $\varphi^{(s+1)}$, $c\bar{z}_1 \dots \bar{z}_v \neq c_s p_s \varphi_2^{(s+1)}(d_s q', cp_s, w)$.

3) $|p| = |q| = s+1$, $p = p'w$ and $q = q'w'$ $(w, w' \in W)$. Now, by the definition of $\varphi_2^{(s+1)}$, since $\mathbf{D}_s$ is an s-free semiautomaton, thus $\varphi_2^{(s+1)}(d_s p', cp_s, w) \neq \varphi_2^{(s+1)}(d_s q', cp_s, w')$. Therefore, $(d_s, c)p = (d_s p, cp_s \varphi_2^{(s+1)}(d_s p', cp_s, w)) \neq (d_s q, cp_s \varphi_2^{(s+1)}(d_s q', cp_s, w')) = (d_s, c) q$.

Thus we have shown that for all $s \leqq u+1$, $\mathbf{D}_s$ is an s-free semiautomaton. Then $\mathbf{D}$ can be chosen as $\mathbf{D}_{s+1}$.

We now construct a $(u+1)$-free semiautomaton $\mathbf{E} = (X, E, e_0, \delta_{\mathbf{E}})$ as a quasi-direct product of semiautomata from the one-element set $\{\mathbf{D}\}$. Let t be a natural number such that $2^t \geqq |X|$. Moreover, take a one-to-one mapping ψ of X into W^t. We shall prove that $\mathbf{E} = (X, E, e_0, \delta_{\mathbf{E}}) = \mathbf{D}^t[X, \psi]$ with $e_0 = (d, \dots, d)$ is a $(u+1)$-free semiautomaton. (The feed-back function ψ of $\mathbf{E}$ can be given in this form, since for quasi-direct products the feed-back function is independent of states.) Take two words $p, q \in F_{u+1}(X)$ with $p \neq q$. Assume that $p = x_1 \dots x_r$ and $q = x_1' \dots x_s'$. Then there exists an i $(1 \leqq i \leqq t)$ such that $\psi_i(x_1) \dots \psi_i(x_r) \neq \psi_i(x_1') \dots \psi_i(x_s')$. (Note that ψ is given in the form $\psi = (\psi_1, \dots, \psi_t)$.) Therefore, $d\psi_i(x_1) \dots \psi_i(x_r) \neq d\psi_i(x_1') \dots \psi_i(x_s')$ since $\mathbf{D}$ is a $(u+1)$-free semiautomaton. Thus we have got that $e_0 p \neq e_0 q$, showing that $\mathbf{E}$ is a $(u+1)$-free semiautomaton.

Let us now consider the following two cases:

I) $u+1 = n$. In this case, by Lemma 2, $\mathbf{A}'$ is an n-homomorphic image of $\mathbf{E}$.

II) $u+1 < n$. Then take the direct product $\mathbf{G} = (X', G, g_0, \delta_{\mathbf{G}}) = \Pi(\mathbf{A}_j \mid j = 1, \dots, k)$, where $G = A$ and $g_0 = a_0$. Now form the α_0-product $\mathbf{H} = (X, H, h, \delta_{\mathbf{H}}) = (\mathbf{E} \times \mathbf{G})[X, \gamma]$, where $h = (e_0, a_0)$, and for all $x \in X$, $p \in F(X)$, $e \in E$ and $g \in G$,

$$\gamma(e_0 p, g, x) = (x, \varphi(a_0 p_{\mathbf{A}}, x)) \quad \text{if} \quad |p| \leqq u+1$$

and $\gamma(e, g, x) = (x, x')$, where x' is an arbitrary element of X' if e cannot be given in the form $e_0 p$ with $p \in F_{u+1}(X)$.

Since for a given $p \in F_{u+1}(X)$ there exists no $q \in F_{u+1}(X)$ such that $p \neq q$ and $e_0 p = e_0 q$, thus γ is well defined.

Let us take a mapping $\tau\colon H^{(n)} \to A^{(n)}$ in the following way: $\tau((e, a)) = a$ $((e, a) \in H^{(n)})$. (Here $A^{(n)}$ is considered in $\mathbf{A}$.) We show that τ is an n-homomorphism of $\mathbf{H}$ onto $\mathbf{A}$. Take an arbitrary word $p \in F_n(X)$ with $|p| = l$. We proceed by induction on the length l of p. For $|p| = 0$, $\tau((e_0, a_0)p) = a_0 p_{\mathbf{A}}$ is obviously valid. Assume that our statement has been proved for all words with length t $(< n)$. Now let $p = p'x$ $(x \in X)$ such that $|p| = j+1$ $(\leqq t+1)$. If $|p'| \leqq u+1$ then

$$(e_0, a_0)p = (e_0 p, a_0 p'_{\mathbf{A}} \varphi(a_0 p'_{\mathbf{A}}, x)) = (e_0 p, a_0 p_{\mathbf{A}}),$$

i.e., $\tau((e_0, a_0)p) = \tau((e_0 p, a_0 p_{\mathbf{A}})) = a_0 p_{\mathbf{A}} = \tau((e_0, a_0)) p_{\mathbf{A}}$.

Now consider the case $n>|p'|>u+1$. Then $(e_0, a_0)p=(e_0, a_0)p'(x, \gamma(e, a, x))= =(ex, a\gamma(e, a, x))$, where $(e, a)=(e_0, a_0)p'$. Observe that $ax_{\mathbf{A}}=ax'_{\mathbf{A}}$ for any $x, x' \in X$, since otherwise there exist an $\mathbf{A}_j$ $(1 \leqq j \leqq k)$, $a_j \in A_j$ and $p_j \in F(X_j)$ with $n>|p_j|>u+1$ such that $a_j p_j$ is ambiguous, contradicting our assumption that u is the greatest number having this property. Thus, taking into consideration the induction hypothesis $a=a_0 p'_{\mathbf{A}}$, we get $(e_0, a_0)p=(e_0 p, a_0 p'_{\mathbf{A}} \gamma(e, a, x))=(e_0 p, a_0 p'_{\mathbf{A}} \varphi(a_0 p'_{\mathbf{A}}, x))=(e_0 p, a_0 p_{\mathbf{A}})$, proving that $\tau((e_0, a_0)p)=a_0 p_{\mathbf{A}}=\tau(e_0, a_0)p_{\mathbf{A}}$. Therefore, we have shown that τ is an n-homomorphism of $\mathbf{H}$ onto $\mathbf{A}$.

If there is no ambiguous state in any semiautomaton from Σ then $\mathbf{A}$ is isomorphic to a quasi-direct product of $\mathbf{A}_1, \ldots, \mathbf{A}_k$.

Since the direct product and quasi-direct product are special cases of the α_0-product, and the α_0-product of α_0-products is also an α_0-product thus $\mathbf{H}$ can be given as an α_0-product of semiautomata from Σ. This ends the proof of the Theorem.

A system Σ of automata is *metrically complete* with respect to the α_i-product (general product) if for any natural number n and automaton mapping f: $F(X) \to F(Y)$ $(|X|, |Y| < \aleph_0)$ there exists an α_i-product (general product) of automata from Σ inducing f in length n. In [2] it was shown that there exists an algorithm to decide for a finite system Σ of automata whether Σ is metrically complete with respect to the α_0-product. Using this result, from our above Theorem we get the following

Corollary. There exists an algorithm to decide for a finite system Σ of automata whether Σ is metrically complete with respect to the general product or any α_i-product $(i=0, 1, \ldots)$.

Представление автоматных отображений в конечном длине

В статье [3] было введено понятие α_i-произведения автоматов $(i=0, 1, \ldots)$. Пусть Σ — произвольное множество конечных автоматов и n — некоторое натуральное число. В настоящей работе доказывается, что автоматное отображение f можно индуцировать в длине n некоторым α_i-произведением автоматов из Σ тогда и только тогда f индуцируется в длине n некоторым произведением автоматов из Σ.

DEPT. OF COMPUTER SCIENCE
A. JÓZSEF UNIVERSITY
H—6720 SZEGED, HUNGARY
SOMOGYI U. 7.

References

[1] GÉCSEG, F., Families of automaton mappings, *Acta Sci. Math. (Szeged)*, v. 28, 1967, pp. 39—54 (Russian).
[2] GÉCSEG, F., Metrically complete systems of automata, *Kibernetika (Kiev)*, v. 3, 1968, pp. 96—98 (Russian).
[3] GÉCSEG, F., Composition of automata, *Proceedings of the 2nd Colloquium on Automata, Languages and Programming*, Saarbrücken, Springer Lecture Notes in Computer Science, v. 14, 1974, pp. 351—363.
[4] GLUŠKOV, V. M., Theory of abstract automata, *Uspehi Mat. Nauk* v. 16, 1961, pp. 3—62 (Russian).
[5] HARTMANIS, J., Loop-free structure of sequential machines, *Information and Control* v. 5, 1962, pp. 24—44.

(Received Nov. 27, 1974)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

An application of truth functions in formalized diagnostics*

By A. ÁDÁM

To Professor Pál Erdős on his sixtieth birthday

§ 1.

In what follows, we shall prove some results concerning truth functions (in §§ 2—4) and apply them to the following problem (in §§ 5—6). There is a set S of objects and there are $n+1$ subsets $Z, X_1, X_2, \ldots, X_n$ of S. Let an object $s(\in S)$ be chosen arbitrarily. We are not able to decide immediately whether or not s belongs to Z; we may observe, however, the validity of any of the n relations $s \in X_i$ and we can infer to the truth of $s \in Z$ if all the relations $s \in X_1, s \in X_2, \ldots, s \in X_n$ are checked. We are interested in deciding, whether $s \in Z$ holds or not, in such a manner that a possibly small number of the relations $s \in X_i$ should be examined (successively, in a straightforward ordering).

§ 2.

Let $f(x_1, x_2, \ldots, x_n)$ be an n-ary truth function. The *rank* $\varrho(f)$ is the number of places where f takes the value $\uparrow$ (true); of course, f takes the value $\downarrow$ (false) at $2^n - \varrho(f)$ places. The *entropy* $\eta(f)$ is defined by

$$\eta(f) = \min\left(\varrho(f), 2^n - \varrho(f)\right).$$

We have $\eta(f) = \eta(\bar{f}) \leqq 2^{n-1}$; furthermore, $\eta(f) = 0$ exactly if f is constant.

Let $\mathfrak{A}$ be an elementary conjunction over the set $\{x_1, x_2, \ldots, x_n\}$. The number of variables occuring in $\mathfrak{A}$ is called the *length* $l(\mathfrak{A})$ of $\mathfrak{A}$.

Suppose that $\mathfrak{A}$ contains (precisely) the variables $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ $\left(l = l(\mathfrak{A})(\geqq 1)\right)$. We denote by $x_{j_1}, x_{j_2}, \ldots, x_{j_{n-l}}$ the elements of the set

$$\{x_1, x_2, \ldots, x_n\} - \{x_{i_1}, x_{i_2}, \ldots, x_{i_l}\}.$$

* The considerations of this paper have been contained in the lecture "On some combinatorial questions" presented on the colloquium "Infinite and finite sets" held at Keszthely, June 1973.

Let $f_{\mathfrak{A}}(x_{j_1}, x_{j_2}, \ldots, x_{j_{n-l}})$ be defined as the function resulting from f if constants are substituted for each of $x_{i_1}, x_{i_2}, \ldots, x_{i_l}$ such that $\mathfrak{A}$ takes the value † with the substitutions prescribed. It is obvious that $\varrho(f_{x_i})+\varrho(f_{\bar{x}_i})=\varrho(f)$. If $\mathfrak{A}$ and $\mathfrak{B}$ are elementary conjunctions (over $\{x_1, x_2, \ldots, x_n\}$) without any variable in common, then clearly $f_{\mathfrak{A}\&\mathfrak{B}}=(f_{\mathfrak{A}})_{\mathfrak{B}}$.

For a truth function f and a variable x_i of it, let the number $\lambda(f, x_i)$ and $\mu(f, x_i)$ be defined by

$$\lambda(f, x_i) = \min(\eta(f_{x_i}), \eta(f_{\bar{x}_i})),$$

$$\mu(f, x_i) = \max(\eta(f_{x_i}), \eta(f_{\bar{x}_i})).$$

It is evident that

$$\lambda(f, x_i)+\mu(f, x_i) = \eta(f_{x_i})+\eta(f_{\bar{x}_i})$$

and that $\lambda(f, x_i)$ is the smallest of the four ranks

$$\varrho(f_{x_i}), \quad \varrho(\bar{f}_{x_i}), \quad \varrho(f_{\bar{x}_i}), \quad \varrho(\bar{f}_{\bar{x}_i}).$$ [1]

Proposition 1. *We have*

$$\lambda(f, x_i) \leqq \frac{\eta(f)}{2}.$$

Proof.

Case 1: $\eta(f)=\varrho(f)$. Then

$$\varrho(f_{x_i})+\varrho(f_{\bar{x}_i}) = \varrho(f) \leqq 2^{n-1},$$

hence

$$\min(\varrho(f_{x_i}), \varrho(f_{\bar{x}_i})) \leqq \frac{\varrho(f)}{2} \leqq 2^{n-2}.$$

This implies the conclusion evidently.

Case 2: $\eta(f)=2^n-\varrho(f)(=\varrho(\bar{f}))$. The inference is analogous to Case 1 (with $\bar{f}$ instead of f).

We say that x_i is a variable *of type* α (or, for the sake of brevity, an α-variable) of the function f if

$$\lambda(f, x_i) \geqq \eta(f)-2^{n-2}.$$

In case

$$\lambda(f, x_i) < \eta(f)-2^{n-2},$$

we call x_i a variable *of type* β (or a β-variable). If $\eta(f)\leqq 2^{n-2}$, then each variable is of type α. [2]

[1] It seems to be advantageous to consider the numbers $\lambda(f, x_i)$ as basic quantities in the subsequent treatment (because the λ's can perhaps be produced in a more natural manner, than the entropies). Another possibility for treating the topics is if one omits the λ's and defines at once the critical variables by their property to be stated in the second sentence of Proposition 8.

[2] It is trivial from this remark that there exist functions all the variables of which are of type α. In case of $n=4$ and $f=x_1x_2x_3 \vee x_1x_4 \vee x_2x_4 \vee x_3x_4$, we have $\eta(f)=8$, $\lambda(f, x_1)=\lambda(f, x_2)=\lambda(f, x_3)=3$ and $\lambda(f, x_4)=1$, hence every variable of f is of type β. In case of $n=3$ and $f=x_1 \vee \bar{x}_2\bar{x}_3$, we have $(\eta f)=3$, $\lambda(f, x_1)=0$ and $\lambda(f, x_2)=\lambda(f, x_3)=1$, thus x_1 is a β-variable and x_2, x_3 are α-variables. We have seen that the three situations, being logically possible, may really occur.

Proposition 2. *If x_i is an α-variable of f, then*

$$\eta(f_{x_i})+\eta(f_{\bar{x}_i}) = \eta(f).$$

Proof.

Case 1: $\eta(f)=\varrho(f)$ and $\varrho(f_{x_i})\leqq\varrho(f_{\bar{x}_i})$. Then

$$2\varrho(f_{x_i}) \leqq \varrho(f_{x_i})+\varrho(f_{\bar{x}_i}) = \varrho(f) = \eta(f) \leqq 2^{n-1},$$

consequently,

$$2^{n-2} \geqq \varrho(f_{x_i}) = \eta(f_{x_i}).$$

Thus

$$\varrho(f_{\bar{x}_i}) = \varrho(f)-\varrho(f_{x_i}) \leqq \eta(f)-\lambda(f, x_i) \leqq 2^{n-2},$$

hence $\eta(f_{\bar{x}_i})=\varrho(f_{\bar{x}_i})$. By summarizing our considerations, we have

$$\eta(f_{x_i})+\eta(f_{\bar{x}_i}) = \varrho(f_{x_i})+\varrho(f_{\bar{x}_i}) = \varrho(f) = \eta(f).$$

We shall now mention the conditions of the remaining three cases; in any of them, the statement can be verified by an analogous inference.

Case 2: $\eta(f)=\varrho(f)$ and $\varrho(f_{\bar{x}_i})\leqq\varrho(f_{x_i})$.

Case 3: $\eta(f)=\varrho(\bar{f})$ and $\varrho(\bar{f}_{x_i})\leqq\varrho(\bar{f}_{\bar{x}_i})$.

Case 4: $\eta(f)=\varrho(\bar{f})$ and $\varrho(\bar{f}_{\bar{x}_i})\leqq\varrho(\bar{f}_{x_i})$.

Proposition 3. *If x_i is a β-variable of f, then*

$$\mu(f, x_i)-\lambda(f, x_i) = 2^{n-1}-\eta(f).$$

Proof. Similarly to the preceding proof, we can distinguish four cases; it suffices by the analogy that we carry out the proof only when $\eta(f)=\varrho(f)$ and $\varrho(f_{x_i})\leqq\varrho(f_{\bar{x}_i})$. The formula

$$2^{n-2} \geqq \varrho(f_{x_i}) = \eta(f_{x_i})$$

is valid as in the former proof.

Our next aim is to verify indirectly that

$$\eta(f_{\bar{x}_i}) = \varrho(\bar{f}_{\bar{x}_i}) < \varrho(f_{\bar{x}_i}).$$

Suppose the contrary, i.e. $\eta(f_{\bar{x}_i})=\varrho(f_{\bar{x}_i})$. Since x_i is of type β, we have

$$2^{n-2} < \varrho(f)-\lambda(f, x_i) = \varrho(f)-\min\left(\varrho(f_{x_i}), \varrho(f_{\bar{x}_i})\right) = \varrho(f)-\varrho(f_{x_i}),$$

hence

$$\varrho(f) > 2^{n-2}+\varrho(f_{x_i}) \geqq 2^{n-1} \geqq \eta(f),$$

this contradicts the supposition $\eta(f)=\varrho(f)$.

The proof (of the case treated in details) is completed by the deduction

$$\mu(f, x_i)-\lambda(f, x_i) = |\eta(f_{x_i})-\eta(f_{\bar{x}_i})| = |\varrho(f_{x_i})-\varrho(\bar{f}_{\bar{x}_i})| =$$
$$= \left|\left(\varrho(f_{x_i})+\varrho(f_{\bar{x}_i})\right)-\left(\varrho(f_{\bar{x}_i})+\varrho(\bar{f}_{\bar{x}_i})\right)\right| =$$
$$= |\varrho(f)-2^{n-1}| = |\eta(f)-2^{n-1}| = 2^{n-1}-\eta(f).$$

Proposition 4. *We have*

$$\eta(f_{x_i})+\eta(f_{\bar{x}_i}) \leqq \eta(f)$$

where equality or strict inequality holds according as x_i is an α-variable or a β-variable, respectively.

Proof. The statement was asserted in Proposition 2 for α-variables. If x_i is a β-variable, then

$$\mu(f, x_i) = 2^{n-1} - \eta(f) + \lambda(f, x_i) < \eta(f) - \lambda(f, x_i)$$

by Proposition 3 and the definition of β-variables.

The next assertion is an obvious consequence of Proposition 2:

Proposition 5. *If both x_i and x_j are α-variables of f, then*

$$\eta(f_{x_i}) + \eta(f_{\bar{x}_i}) = \eta(f_{x_j}) + \eta(f_{\bar{x}_j}).$$

Proposition 6. *Let x_i, x_j be two β-variables of f. If*

$$\lambda(f, x_i) \leqq \lambda(f, x_j),$$

then

$$\mu(f, x_i) \leqq \mu(f, x_j)$$

and

$$\eta(f_{x_i}) + \eta(f_{\bar{x}_i}) \leqq \eta(f_{x_j}) + \eta(f_{\bar{x}_j}).$$

Furthermore, the strict inequality in the hypothesis implies strict inequalities in the conclusion.

Proof. By Proposition 3, we have

$$\mu(f, x_i) = 2^{n-1} - \eta(f) + \lambda(f, x_i) \leqq 2^{n-1} - \eta(f) + \lambda(f, x_j) = \mu(f, x_j),$$

thus also

$$\eta(f_{x_i}) + \eta(f_{\bar{x}_i}) = \lambda(f, x_i) + \mu(f, x_i) \leqq \lambda(f, x_j) + \mu(f, x_j) = \eta(f_{x_j}) + \eta(f_{\bar{x}_j}).$$

It is clear that all of these deductions remain valid with $<$ (instead of $\leqq$) if $\lambda(f, x_i) < \lambda(f, x_j)$ is supposed.

Proposition 7. *Let x_i be an α-variable and x_j be a β-variable of f. Then*

$$\lambda(f, x_i) > \lambda(f, x_j)$$

and

$$\eta(f_{x_i}) + \eta(f_{\bar{x}_i}) > \eta(f_{x_j}) + \eta(f_{\bar{x}_j}).$$

Proof. The first inequality follows at once by comparing the definition of α-variables to that of β-variables; the second one is implied by Proposition 4.

§ 3.

We define the *critical variables* of a truth function f by the subsequent two rules (I), (II):

(I) If every variable of f is of type α, then all the variables are critical.

(II) Suppose that f has at least one β-variable. We call a variable x_i critical exactly when

$$\lambda(f, x_i) \leqq \lambda(f, x_j)$$

for each variable x_j of f.

Proposition 8. *Any n-ary function ($n \geqq 1$) has at least one critical variable. Let x_i be a critical variable, we have*

$$\eta(f_{x_i})+\eta(f_{\bar{x}_i}) \leqq \eta(f_{x_j})+\eta(f_{\bar{x}_j})$$

for an arbitrary variable x_j of f; furthermore, equality holds in this formula precisely if x_j is also critical. If f has at least one β-variable, then all the critical variables are of type β.

Proof. If f has α-variables only, then our statements are valid by Proposition 5. Assume that there exists a β-variable of f. Let x_i be a critical variable. Proposition 7 implies that x_i is of type β.

Consider an arbitrary other variable x_j. If $\lambda(f, x_i) \doteq \lambda(f, x_j)$, then x_j is critical, it is of type β and Proposition 6 guarantees

$$\eta(f_{x_i})+\eta(f_{\bar{x}_i}) = \eta(f_{x_j})+\eta(f_{\bar{x}_j}).$$

If $\lambda(f, x_i) < \lambda(f, x_j)$, then

$$\eta(f_{x_i})+\eta(f_{\bar{x}_i}) < \eta(f_{x_j})+\eta(f_{\bar{x}_j})$$

follows from Proposition 7 or Proposition 6 (according as x_j is an α-variable or a β-variable).

§ 4.

In this section, we shall give a method for determining the rank of a truth function f supposing that f is given in some disjunctive normal form. It is required that the reader is familiar with the "principle of inclusion and exclusion". [3]

If $\mathfrak{A}$ is an elementary conjunction over the set $\{x_1, x_2, \ldots, x_n\}$ (considered as an n-ary function), then obviously $\varrho(\mathfrak{A})=2^{n-l(\mathfrak{A})}$.

Let $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_j$ be elementary conjunctions ($j \geqq 1$). Suppose that there exists no variable x_i such that x_i occurs in non-negated form in some $\mathfrak{A}_h$ and negated in an $\mathfrak{A}_{h'}$ (where $1 \leqq h \leqq j$ and $1 \leqq h' \leqq j$). [4] Let $l(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j)$ be defined as the number of *distinct* variables occurring in $\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j$ (i.e. as $l(\mathfrak{B})$ where $\mathfrak{B}$ is the elementary conjunction resulted by the reduction of $\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j$). Since $\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j$ is $\dagger$ exactly when each of $\mathfrak{A}_1, \mathfrak{A}_2, \ldots, \mathfrak{A}_j$ is $\dagger$, we have

$$\varrho(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j) = 2^{n-l(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j)}$$

whenever $l(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_n)$ is defined. [5]

Proposition 9. *If $\mathfrak{A}_1 \vee \mathfrak{A}_2 \vee \ldots \vee \mathfrak{A}_k$ is a disjunctive normal form representing the function $f(x_1, x_2, \ldots, x_n)$, then we have*

$$\varrho(f) = \Sigma 2^{n-l(\mathfrak{A}_i)} - \Sigma 2^{n-l(\mathfrak{A}_{i_1} \& \mathfrak{A}_{i_2})} + \Sigma 2^{n-l(\mathfrak{A}_{i_1} \& \mathfrak{A}_{i_2} \& \mathfrak{A}_{i_3})} -$$
$$- \ldots + (-1)^{j-1} \Sigma 2^{n-l(\mathfrak{A}_{i_1} \& \mathfrak{A}_{i_2} \& \ldots \& \mathfrak{A}_{i_j})} + \ldots$$
$$\ldots + (-1)^{k-1} \Sigma 2^{n-l(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_k)},$$

---

[3] See [3] (p. 282) or [4] (Chapter 3) or [2] (§ 22).

[4] If this supposition is not fulfilled, then we not define $l(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j)$.

[5] If it is undefined, then $\varrho(\mathfrak{A}_1 \& \mathfrak{A}_2 \& \ldots \& \mathfrak{A}_j) = 0$.

where the j th summation is extended to all such j-tuples $(i_1, i_2, \ldots, i_j)$ for which $1 \leqq i_1 < < i_2 < \ldots < i_j \leqq k$ and $l(\mathfrak{A}_{i_1} \& \mathfrak{A}_{i_2} \& \ldots \& \mathfrak{A}_{i_j})$ is defined.

Proof. Let the principle of inclusion and exclusion be applied under such circumstantes that the basic set H is the definition domain of f and, for each $i (1 \leqq i \leqq k)$, H_i is the set of places at which $\mathfrak{A}_i$ takes the value $\uparrow$.

§ 5.

Now we return to our original problem (exposed in § 1). We introduce some notations. For any i, let X_i^* be the difference set $S - X_i$ $(1 \leqq i \leqq n)$. Any set

$$Y = Y_1 \cap Y_2 \cap \ldots \cap Y_n$$

is called an *atom*, where Y_i is either X_i or X_i^*. There exist 2^n atoms (some of them may be empty), any object $s (\in S)$ belongs to exactly one atom.

Postulate. If Y is an arbitrary atom, then either $Y \subseteqq Z$ or $Y \cap Z = \emptyset$.

Next we define the *characteristic* (truth) *function* of the system $\{Z, X_1, X_2, \ldots \ldots, X_n\}$. Let a full elementary conjunction $\mathfrak{A}$ over $\{x_1, x_2, \ldots, x_n\}$ be given. We assign to $\mathfrak{A}$ the atom $\sigma(\mathfrak{A})$ determined in such a way that $Y_i = X_i$ or $Y_i = X_i^*$ according as x_i occurs in $\mathfrak{A}$ without or with negation $(1 \leqq i \leqq n)$. The function value is defined by what follows:

$$f(\mathfrak{A}) = \begin{cases} \uparrow & \text{if} \quad \sigma(\mathfrak{A}) \subseteqq Z \\ \downarrow & \text{if} \quad \sigma(\mathfrak{A}) \cap Z = \emptyset. \end{cases}$$

(When $\sigma(\mathfrak{A})$ is void, then $f(\mathfrak{A})$ is defined arbitrarily. The postulate guarantees that $f(\mathfrak{A})$ is defined at each place $\mathfrak{A}$.)

Algorithm. Step 1. (a) We consider the characteristic function f of the set system $\{Z, X_1, X_2, \ldots, X_n\}$, we form $\eta(f)$ and the minimum of the n values $\lambda(f, x_i)$ (by comparing the $4n$ numbers $\varrho(f_{x_i})$, $\varrho(f_{\bar{x}_i})$, $\varrho(\bar{f}_{x_i})$, $\varrho(\bar{f}_{\bar{x}_i})$, by using Proposition 9).

(b) If this minimum reaches $\eta(f) - 2^{n-2}$, then we choose an arbitrary variable x_i of f. If the minimum is smaller than $\eta(f) - 2^{n-2}$, then we choose such a variable x_i which yields the minimal value of $\lambda(f, x_i)$.

(c) We check whether or not s is contained in X_i. If $s \in X_i$, then we shall perform Step 2 with f_{x_i}. If $s \notin X_i$, then Step 2 will be executed with $f_{\bar{x}_i}$.

...

Step $m (\geqq 2)$. (a) We have produced an $(n-m+1)$-ary function $f_{\mathfrak{A}}$ in Step $m-1$. If $f_{\mathfrak{A}}$ is constantly $\uparrow$, then $s \in Z$ and the algorithm is finished. If $f_{\mathfrak{A}}$ is constantly $\downarrow$, then $s \notin Z$ and the algorithm is also finished. If $f_{\mathfrak{A}}$ is non-constant, then we consider $\eta(f_{\mathfrak{A}})$ and the minimum of the $n-m+1$ values $\lambda(f, x_{j_i})$ (analogously to the part (a) of Step 1).

(b) If this minimum reaches $\eta(\mathfrak{A}) - 2^{n-m-1}$, then we choose an arbitrary variable x_{j_i} of $f_{\mathfrak{A}}$. If the minimum is smaller than $\eta(f_{\mathfrak{A}}) - 2^{n-m-1}$, then we choose such a variable x_{j_i} which yields the minimal value of $\lambda(f_{\mathfrak{A}}, x_{j_i})$.

(c) We check whether or not s is contained in X_{j_i}. If $s \in X_{j_i}$, then Step $m+1$ will be performed with $f_{\mathfrak{A} \& x_{j_i}}$. If $s \notin X_{i_j}$, then we shall execute Step $m+1$ with $f_{\mathfrak{A} \& \bar{x}_{j_i}}$.

§ 6.

This section is devoted to justifying the algorithm. We shall deal with our basic problem (see § 1 and § 5) under such circumstances that the postulate (in § 5) is valid and we know the characteristic function $f(x_1, x_2, ..., x_n)$ but we have no further information (e.g. it is unknown how the elements of S are distributed into the atoms) at beginning the procedure.

It is evident that the algorithm is completed after at most n steps.

The entropy $\eta(f)$ can be viewed as a measure of the uncertainty whether f takes one or other truth value at a randomly chosen place of its domain. Hence we consider $\eta(f)$ as the measure of uncertainty of whether $s \in Z$ or $s \notin Z$ is fulfilled.

We try to proceed towards smaller entropies, as far as possible, by checking the validity of appropriate relations $s \in X_i$ successively. In order to do this, it seems (by Propositions 4, 8) the best strategy to obtain the minimal $\eta(f_{\mathfrak{A}\&x_i}) + \eta(f_{\mathfrak{A}\&\bar{x}_i})$ in each step, i.e. to continue the process with a *critical* variable of the function $f_{\mathfrak{A}}$ (where $\mathfrak{A}$ characterizes the informations being at our disposal after the earlier steps), with respect to that the formulae $s \in X_i$ and $s \notin X_i$ are assumed equiprobable.

§ 7.

The investigations described in the previous parts of the paper seem to admit some generalizations. In this final section, I mention four possibilities of generalizing them (which can be combined with each other). The subsequent list was compiled together with Dr. Gy. Pollák.

(1) More than one membership relations $s \in Z_1, s \in Z_2, ..., s \in Z_w$ should be determined simultaneously (i.e. by the same sequence of observations of whether or not $s \in X_i$).

(2) For any atom Y, we know only the probability $P(s \in Z)$ of that $s (\in Y)$ belongs to Z (possibly lying between 0 and 1), consequently, f is a stochastic truth function (in sense of [1]). We try to achieve that

$$|2P(s \in Z) - 1|$$

should be significant (i.e. larger than a given number $1-\varepsilon$).

(3) For any atom Y, we know the probability of the event that $s (\in S)$ is contained in Y (this probability may differ from $1/2^n$). (The precise goal is also to be determined.)

(4) There is assigned a number (called weight) to each X_i (interpreted as the difficulty of checking of whether or not $s \in X_i$), our aim is to minimize the sum of weights of the observations performed (instead of minimizing the number of observations).

Одно применение функций алгебры логики в формализованной диагностике

Пусть даны подмножества $Z, X_1, X_2, ..., X_n$ некоторого множества S объектов так, что каждый атом

$$Y = Y_1 \cap Y_2 \cap ... \cap Y_n.$$

(где Y_i обозначает либо X_i либо $S-X_i$) удовлетворяет одну из формул $Y \subseteq Z$ и $Y \cap Z = \emptyset$. Предположим, что для произвольного элемента $s(\in S)$ мы можем наблюдать справедливость отношений принадлежности

$$s \in X_1, \quad s \in X_2, \ldots, s \in X_n$$

в зависимом от нас порядке.

Мы интересуемся, что принадлежность $s \in Z$ имеет ли место (где s — произвольно фиксированный элемент множества S). В случае, когда известно, какие атомы являются подмножествами множества Z и какие атомы не пересекают Z (но мы не имеем никакую информацию относительно элемента s специфически), даётся стратегия для целесообразного порядка исполнения наблюдений $s \in X_i$, с целью проверки или опровержения принадлежности $s \in Z$ после (по возможности) меньше чем n наблюдений типа $s \in X_i$.

MATHEMATICAL INSTITUTE OF THE
HUNGARIAN ACADEMY OF SCIENCES
H—1053 BUDAPEST, HUNGARY
REÁLTANODA U. 13–15.

References

[1] ÁDÁM, A., Über stochastische Wahrheitsfunktionen, *Proc. Coll. on Information Theory (Debrecen, 1967)*, 1968, pp. 15—34.
[2] ÁDÁM, A., *Truth functions and the problem of their realization by two-terminal graphs*, Akadémiai Kiadó (Budapest), 1968.
[3] NETTO, E., *Lehrbuch der Combinatorik*, Teubner (Leipzig—Berlin), 1927.
[4] RIORDAN, J., *An introduction to combinatorial analysis*, Wiley (New York), 1958.
[4a] Риордан, Дж., *Введение в комбинаторный анализ*, Изд. ин. лит. (Москва), 1963.

(Received Oct. 24, 1974)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

On two problems of A. Salomaa

By Z. ÉSIK

In this paper we solve two problems raised by A. Salomaa in his book [1]. Namely, we show that all right derivatives of a stochastic language are stochastic. Conversely, if there exists an integer k such that all right derivatives of a language L with respect to all words of length k are stochastic languages then L is stochastic language, too. Furthermore, it is proved that the family of stochastic languages remains unaltered if the components of the output vectors and the cut points are allowed to be arbitrary real numbers. Proving these statements, we give affirmative answers to Problems 3.1 and 5.1 of A. Salomaa.

Before studying these problems we recall some definitions from [1].

By an alphabet I we mean a finite non-empty set. The elements of I are called letters, sometimes input signs. A word over I is a finite string consisting of zero or more letters. The empty word λ is a string consisting of zero letters. If a word P consists of $k(\geqq 0)$ letters then the length of this word is $\lg(P)=k$. The set of all words over I is denoted by $W(I)$. If $P, Q \in W(I)$ then PQ denotes their catenation.

A language L is a subset of $W(I)$. The void language is the language consisting of no words. The union (or sum) of two languages L_1 and L_2 is denoted by $L_1 \vee L_2$ and their catenation is defined by $L_1 L_2 = \{P \mid P = P_1 P_2,\ P_1 \in L_1,\ P_2 \in L_2\}$. If L_2 consists of one word Q only then $L_1 L_2$ is denoted by $L_1 Q$.

If given a word P over I and a language $L \subseteq W(I)$ then the right (left) derivative of L with respect to the word P is defined by $L//P = \{Q \mid QP \in L\}$ ($L \backslash\backslash P = \{Q \mid PQ \in L\}$).

A vector is called stochastic if its each component is nonnegative real number and the sum of its components equals to 1. Moreover, a stochastic matrix is a square matrix whose each row is a stochastic vector.

By a finite probabilistic automaton — or, shortly, probabilistic automaton — over an alphabet I we mean an ordered triple $PA=(S, s_0, M)$, where $S=\{s_1, s_2, \ldots, s_n\}$ is a finite non-empty set, the set of all internal states of PA, s_0 is an n-dimensional stochastic row vector, the initial distribution, whose ith component equals to the probability of PA to be in the state s_i at the beginning of its working; finally, M is a mapping of I into the set of all stochastic matrices of type $n \times n$. For every $x \in I$, $p_{i,j}(x)$ denotes the (i,j)th entry of the matrix $M(x)$. This is the transition probability of PA to go from the state s_i into the state s_j under the input sign x.

We may extend the domain of the function M from I to $W(I)$ by defining: $M(\lambda)=E_n$, $M(Px)=M(P)M(x)$ for every $Px \in W(I)$. (Here E_n is the n-dimensional identity matrix.) The stochastic row vector $s_0M(P)$ is called the distribution of states caused by the word P. Further on this row vector is often denoted by $PA(P)$.

If V_i is the n-dimensional coordinate column vector whose ith component equals to 1 then for every word $P\in W(I)$, $p_i(P)=PA(P)V_i$ is the probability of PA to go into the state s_i under the word P.

Let PA be the probabilistic automaton defined above and $\bar{S}_1$ an n-dimensional column vector whose each component is either 0 or 1; $\bar{S}_1$ is called output vector. To each such vector $\bar{S}_1$ there corresponds a subset S_1 of S and conversely, where S_1 is given by: $s_i\in S_1$ if and only if the i th component of $\bar{S}_1$ equals to 1. Moreover, let η be a real number such that $0\leqq\eta<1$. The language represented in PA by S_1 and the cut point η is defined by $L(PA, \bar{S}_1, \eta)=\{P|PA(P)\bar{S}_1>\eta\}$. A language L is η-stochastic if and only if for some PA and $\bar{S}_1$, $L=L(PA, \bar{S}_1, \eta)$. Furthermore, a language L is stochastic if and only if for some $\eta\,(0\leqq\eta<1)$, L is η-stochastic. Now we are ready to state

Theorem 1. All right derivatives of a stochastic language with respect to any word are stochastic languages. Conversely, if there is an integer k such that all right derivatives of a language L with respect to all words of length k are stochastic then L is a stochastic language.

Proof. In order to prove the first part of the theorem take an arbitrary stochastic language $L=L(PA, \bar{S}_1, \eta)$ represented in the probabilistic automaton $PA=(S(=\{s_1, s_2, ..., s_n\}), s_0, M)$ over the alphabet $I=\{x_1, x_2, ..., x_r\}$. For any $i=1, 2, ..., n$ and $x\in I$ let $q_i(x)=V_i^*M(x)\bar{S}_1$, where V_i^* is the transpose of the vector V_i. Thus $q_i(x)$ is the probability of PA to go from the state s_i into one of the states of S_1 under the input sign x.

Since our statement is obviously valid for the empty word thus, in the sequel, we may confine ourself to derivatives with respect to words of length exceeding 0. By $L//(x_ix_j)=(L//x_j)//x_i$ $(x_i, x_j\in I)$, it is enough to prove the first statement of Theorem 1 for letters. To make our discussions simplier, further on we shall deal with $L//x_1$ only. Thus $q_i(x_1)$ will simply be denoted by q_i.

If for every $i=1, 2, ..., n$, $q_i=0$ then $L\subseteqq W(I)\{x_2, x_3, ..., x_r\}$, therefore, $L//x_1$ is the void language, which is clearly a stochastic one. Hence we may assume that there is at least one index i with $q_i\neq 0$. Let $i_1, i_2, ..., i_l$ be all different indices such that the product $q_{i_1}q_{i_2}...q_{i_l}\neq 0$ and let $q=q_{i_1}+q_{i_2}+...+q_{i_l}$.

We may assume that $\eta<q$. Indeed, by a theorem of R. Bukharev and P. Turakainen in [1], every stochastic language is η'-stochastic for any η' with $0<\eta'<1$. Furthermore, it can easily be seen that if given a finite probabilistic automaton $PA'=(S', s_0', M')$ then for any language $L'=L(PA', \bar{S}_1', \eta'')$ and η' with $0<\eta'<\eta''$ one can construct a probabilistic automaton PA'' by adding a new state s to the set of the internal states of PA' such that there is no transition from s to S' and from any state of S' to s, moreover, L' can be represented in PA'' with the cut point η' and the same set S_1'.

Now let $S^*=\{s_1, s_2, ..., s_{ln}\}$ and

$$PA^* = \left(S^*, \frac{1}{q}(q_{i_1}s_0, q_{i_2}s_0, ..., q_{i_l}s_0), M^*\right),$$

where

$$M^*(x) = \left\| \begin{matrix} M(x) & & 0 \\ & M(x) & \\ & & \ddots \\ 0 & & M(x) \end{matrix} \right\|$$

for any $x \in I$. Define

$$\bar{S}_1^* = \left\| \begin{matrix} V_{i_1} \\ V_{i_2} \\ \vdots \\ V_{i_l} \end{matrix} \right\|, \quad L^* = L(PA^*, \bar{S}_1^*, \eta/q).$$

Obviously PA^* is a probabilistic automaton and L^* is a stochastic language. We clame that $L//x_1 = L^*$. To prove this statement it is enough to verify that for every word P,

$$q PA^*(P)\bar{S}_1^* = PA(Px_1)\bar{S}_1.$$

Indeed, if $P \in W(I)$ is an arbitrary word then

$$q PA^*(P)\bar{S}_1^* = (q_{i_1}s_0, q_{i_2}s_0, \dots, q_{i_l}s_0) \left\| \begin{matrix} M(P) & & 0 \\ & M(P) & \\ & & \ddots \\ 0 & & M(P) \end{matrix} \right\| \left\| \begin{matrix} V_{i_1} \\ V_{i_2} \\ \vdots \\ V_{i_l} \end{matrix} \right\| =$$

$$= \sum_{j=1}^{l} q_{i_j} s_0 M(P) V_{i_j} = \sum_{i=1}^{n} q_i s_0 M(P) V_i = \sum_{i=1}^{n} p_i(P) q_i = PA(Px_1)\bar{S}_1.$$

The second part of the theorem is also trivial in the case $k=0$. Thus let $k=1$. First we prove that if L is a stochastic language then for every letter x the catenation Lx is stochastic too.

Let again $L = L(PA, \bar{S}_1, \eta)$ be a stochastic language, where $PA = (S(= \{s_1, s_2, \dots, s_n\}), s_0, M)$ is a finite probabilistic automaton over the alphabet $I = \{x_1, x_2, \dots, x_r\}$. Without loss of generality we may assume that $S_1 = \{s_1, s_2, \dots, s_l\}$ for a certain integer $l \leq n$. For arbitrary letter $x \in I$ let $M_i(x)$ denote the i th row of the matrix $M(x)$. For every $i \in \{1, 2, \dots, l\}$ there exists a $j(i) \in \{1, 2, \dots, n\}$ such that $p_{i, j(i)}(x_1) \neq 0$. To every such pair $(i, j(i))$ let us correspond the following probabilistic automaton:

$$PA^i = (S^i(= \{s_1^i, s_2^i, \dots, s_n^i, s_{n+1}^i\}), (s_0, 0), M^i),$$

where

$$M^i(x_1) = \left\| \begin{matrix} & M_1(x_1) & & 0 \\ & \vdots & & \vdots \\ & M_{i-1}(x_1) & & 0 \\ p_{i,1}(x_1) \dots p_{i,j(i)-1}(x_1) & 0 & p_{i,j(i)+1}(x_1) \dots p_{i,n}(x_1) & p_{i,j(i)}(x_1) \\ & M_{i+1}(x_1) & & 0 \\ & \vdots & & \vdots \\ & M_n(x_1) & & 0 \\ & M_{j(i)}(x_1) & & 0 \end{matrix} \right\|$$

if $i \neq j(i)$,

$$M^i(x_1) = \left\| \begin{matrix} & M_1(x_1) & & 0 \\ & \vdots & & \vdots \\ & M_{i-1}(x_1) & & 0 \\ p_{i,1}(x_1) \dots p_{i,j(i)-1}(x_1) & 0 & p_{i,j(i)+1}(x_1) \dots p_{i,n}(x_1) & p_{i,j(i)}(x_1) \\ & M_{i+1}(x_1) & & 0 \\ & \vdots & & \vdots \\ & M_n(x_1) & & 0 \\ p_{j(i),1}(x_1) \dots p_{j(i),j(i)-1}(x_1) & 0 & p_{j(i),j(i)+1}(x_1) \dots p_{j(i),n}(x_1) & p_{j(i),j(i)}(x_1) \end{matrix} \right\|$$

if $i=j(i)$. Moreover, in both cases

$$M^i(x) = \left\| \begin{matrix} M(x) & 0 \\ M_{j(i)}(x) & 0 \end{matrix} \right\|$$

if $x \neq x_1$.

It is clear that $L(PA^i, V_{n+1}, 0) \subseteq W(I)x_1$, where V_{n+1} is the $n+1$-dimensional column vector whose $n+1$th component is 1 and all others are zero. We shall now prove that for every word P,

$$s_0 M(P) V_i = \frac{1}{p_{i,j(i)}(x_1)} (s_0, 0) M^i(Px_1) V_{n+1}.$$

Further on we often use the following notation. If given an arbitrary finite probabilistic automaton $PA' = (S', s_0', M')$ over the alphabet I' and $s'_{i_0} s'_{i_1} \dots s'_{i_{\lg(P)}} \in W(S')$ then

$$p(s'_{i_0} s'_{i_1} \dots s'_{i_{\lg(P)}} | P)$$

denotes the transition probability of PA' to go from the state s'_{i_0} into $s'_{i_{\lg(P)}}$ through the states $s'_{i_1}, \dots, s'_{i_{\lg(P)-1}}$ under the input word P.

Let now $P \in W(I)$ be an arbitrary word. For every $i = 1, 2, \dots, l$ define

$$A_i = \{Qs_i | Q \in W(S), p(Qs_i | P) > 0\},$$

$$B_i = \{Q^i s^i_{n+1} | Q^i \in W(S^i), p(Q^i s^i_{n+1} | Px_1) > 0\}.$$

We say that a $Q \in A_i$ $(i = 1, 2, \dots, l)$ has the property Φ^i_t for some $t \in \{0, 1, \dots, \lg(P) - 1\}$ — in notation $Q \in \Phi^i_t$ — if $Q = s_{i_0} s_{i_1} \dots s_{i_{\lg(P)}}$ such that $i_t = i$, $i_{t+1} = j(i)$, $P = P' x_1 P''$, $\lg(P') = t$. (The fact that Q does not have the property Φ^i_t will be denoted by $Q \notin \Phi^i_t$.)

Let $\varphi_i : A_i \to B_i$ $(i = 1, 2, \dots, l)$ be a mapping given by

$$\varphi_i(s_{i_0} s_{i_1} \dots s_{i_{\lg(P)}}) = s^i_{j_0} s^i_{j_1} \dots s^i_{j_{\lg(P)}} s^i_{n+1},$$

where $j_0 = i_0$ and if $s_{i_0} s_{i_1} \dots s_{i_{\lg(P)}} \in \Phi^i_t$ for certain $t \in \{0, 1, \dots, \lg(P) - 1\}$ then $j_{t+1} = n+1$ otherwise $j_{t+1} = i_{t+1}$. We shall now prove some properties of the mappings φ_i $(i = 1, 2, \dots, l)$.

Assume that $Q = s_{i_0} s_{i_1} \dots s_{i_{\lg(P)}} \in A_i$, $Q' = s_{i'_0} s_{i'_1} \dots s_{i'_{\lg(P)}} \in A_i$ and $Q \neq Q'$. Then there exists an integer t, $-1 \leq t \leq \lg(P) - 1$ such that $i_{t+1} \neq i'_{t+1}$. Let $\varphi_i(Q) = s^i_{j_0} s^i_{j_1} \dots s^i_{j_{\lg(P)}} s^i_{n+1}$ and $\varphi_i(Q') = s^i_{j'_0} s^i_{j'_1} \dots s^i_{j'_{\lg(P)}} s^i_{n+1}$. Now we distinguish three cases.

1. $t=-1$. Then $j_0=i_0\neq i'_0=j'_0$. Thus $\varphi_i(Q)\neq\varphi_i(Q')$.
2. $t\geqq 0$, $Q\notin\Phi^i_t$, $Q'\notin\Phi^i_t$. Then $j_{t+1}=i_{t+1}\neq i'_{t+1}=j'_{t+1}$ and again $\varphi_i(Q)\neq\varphi_i(Q')$.
3. $t\geqq 0$, $Q\in\Phi^i_t$, $Q'\notin\Phi^i_t$. Now $j_{t+1}=n+1, j'_{t+1}\neq n+1$. Thus $\varphi_i(Q)\neq\varphi_i(Q')$.

Since these are all possible cases, we get that φ_i is a one to one mapping for every $i=1, 2, \ldots, l$.

Let $s^i_{j_0}s^i_{j_1}\ldots s^i_{j_{\lg(P)}}s^i_{n+1}\in B_i$ be an arbitrary word. Since this is clearly the image of the word $s_{i_0}s_{i_1}\ldots s_{i_{\lg(P)}}\in A_i$, where $i_0=j_0$ and for every $t\in\{0, 1, \ldots, \lg(P)-1\}$ if $j_{t+1}=n+1$ then $i_{t+1}=f(i)$ otherwise $i_{t+1}=j_{t+1}$ thus we have that φ_i is one to one mapping of A_i onto B_i for every $i=1, 2, \ldots, l$.

Finally, since φ_i is a one to one mapping of A_i onto B_i and

$$p(Q|P) = \frac{1}{p_{i,j(i)}(x_1)}p(\varphi_i(Q)|Px_1)$$

for any $i\in\{1, 2, \ldots, l\}$ and $Q\in A_i$ thus we get:

$$s_0M(P)V_i = \sum_{Q\in A_i}p(Q|P) = \sum_{\varphi_i(Q)\in B_i}p(\varphi_i(Q)|Px_1) = \frac{1}{p_{i,j(i)}(x_1)}(s_0, 0)M^i(Px_1)V_{n+1}.$$

Define

$$S^* = \{s_1, s_2, \ldots, s_{(n+1)l}\},\quad p = \sum_{i=1}^{l}\frac{1}{p_{i,j(i)}(x_1)},$$

$$s^*_0 = \frac{1}{p}\left(\frac{1}{p_{1,j(1)}(x_1)}(s_0, 0), \frac{1}{p_{2,j(2)}(x_1)}(s_0, 0), \ldots, \frac{1}{p_{l,j(l)}(x_1)}(s_0, 0)\right)$$

and for every $x\in I$ take

$$M^*(x) = \begin{Vmatrix} M^1(x) & & & 0 \\ & M^2(x) & & \\ & & \ddots & \\ 0 & & & M^l(x)\end{Vmatrix},\qquad S^*_1 = \begin{Vmatrix} V_{n+1} \\ V_{n+1} \\ \vdots \\ V_{n+1}\end{Vmatrix}.$$

Moreover, consider the stochastic language $L^*=L(PA^*, S^*_1, \eta/p)$, where $PA^*=(S^*, s^*_0, M^*)$ is obviously a probabilistic automaton over the alphabet I. In order to prove that $L^*=Lx_1$ it is enough to show, by $L^*\subseteq W(I)x_1$, that $Px_1\in Lx_1$ if and only if $Px_1\in L^*$ for arbitrary $P\in W(I)$. But this can be seen immediately because

$$pPA^*(Px_1)\bar{S}^*_1 =$$

$$= \left(\frac{(s_0, 0)}{p_{1,j(1)}(x_1)}, \frac{(s_0, 0)}{p_{2,j(2)}(x_1)}, \ldots, \frac{(s_0, 0)}{p_{l,j(l)}(x_1)}\right)\begin{Vmatrix} M^1(Px_1) & & & 0 \\ & M^2(Px_1) & & \\ & & \ddots & \\ 0 & & & M^l(Px_1)\end{Vmatrix}\begin{Vmatrix} V_{n+1} \\ V_{n+1} \\ \vdots \\ V_{n+1}\end{Vmatrix} =$$

$$= \sum_{i=1}^{l}\frac{(s_0, 0)}{p_{i,j(i)}(x_1)}M^i(Px_1)V_{n+1} = \sum_{i=1}^{l}s_0M(P)V_i = PA(P)\bar{S}_1.$$

Since x_1 is an arbitrary letter thus the language Lx is stochastic for any $x\in I$.

Now let L be a language over I such that all the languages $L//x_1, L//x_2, \ldots, L//x_r$ are stochastic. Thus the languages $(L//x_1)x_1, (L//x_2)x_2, \ldots, (L//x_r)x_r$ are also stochastic. They can be represented, respectively, in the probabilistic automata $PA_{x_1}=(S_{x_1}, (s_0)_{x_1}, M_{x_1})$, $PA_{x_2}=(S_{x_2}, (s_0)_{x_2}, M_{x_2})$, ..., $PA_{x_r}=(S_{x_r}, (s_0)_{x_r}, M_{x_r})$ by the sets $S_{x_{1_1}}, S_{x_{2_1}}, \ldots, S_{x_{r_1}}$ and the cut points $\eta_1, \eta_2, \ldots, \eta_r$, where every automaton PA_{x_i} is constructed in a way analogous to the construction of PA^*. It follows from our discussions above that $L(PA_{x_i}, \bar{S}_{x_{i_1}}, 0) \subseteq W(I)x_i$ for every $i=1, 2, \ldots, r$.

First we deal with the case $\eta_1\eta_2\ldots\eta_r \neq 0$. Then, as it was noted in the proof of the first part of Theorem 1, we may assume that $\eta_1=\eta_2=\ldots=\eta_r=\eta$. Define

$$n = \sum_{i=1}^{r} \operatorname{card}(S_{x_i}), \quad PA = (S(=\{s_1, s_2, \ldots, s_n\}), s_0, M),$$

where

$$s_0 = \frac{1}{r}((s_0)_{x_1}, (s_0)_{x_2}, \ldots, (s_0)_{x_r}), \quad M(x) = \begin{Vmatrix} M_{x_1}(x) & & & 0 \\ & M_{x_2}(x) & & \\ & & \ddots & \\ 0 & & & M_{x_r}(x) \end{Vmatrix}$$

for arbitrary $x \in I$. Let

$$\bar{S}_1 = \begin{Vmatrix} \bar{S}_{x_{1_1}} \\ \bar{S}_{x_{2_1}} \\ \vdots \\ \bar{S}_{x_{r_1}} \end{Vmatrix}, \quad L^* = L(PA, \bar{S}_1, \eta/r).$$

It follows immediately that $L^* = \bigvee_{i=1}^{r} (L//x_i)x_i$ because for every word $P \in W(I)$ and $x_i \in I$,

$$rPA(Px_i)\bar{S}_1 = \sum_{j=1}^{r} PAx_j(Px_i)\bar{S}_{x_{j_1}} = PA_{x_i}(Px_i)\bar{S}_{x_{i_1}}.$$

If there is at least one index i such that $\eta_i=0$ we may assume, without loss of generality, that $\eta_1=\eta_2=\ldots=\eta_j=0$ but the product $\eta_{j+1}\eta_{j+2}\ldots\eta_r \neq 0$ for an integer $j \leqq r$. Since by a theorem in [1] every 0-stochastic language is regular, the language $\bigvee_{i=1}^{j} (L//x_i)x_i$ is regular. Moreover, in the same way as it was done in the previous case, it can be proved that the language $\bigvee_{i=j+1}^{r} (L//x_i)x_i$ is stochastic. Thus, using a theorem of P. Turakainen (see [1]) by which the sum of a stochastic and a regular language is stochastic, we get that $L^* = \bigvee_{i=1}^{r} (L//x_i)x_i$ is a stochastic language.

Finally, since $L=L^*$ or $L=L^* \vee \{\lambda\}$ we have that L is stochastic.

We continue our proof by induction. Assume that the second part of the theorem holds true for a certain integer $k \geqq 1$, and assume that for every word xP of length $k+1$ the language $L//xP$ is stochastic. Since $L//xP=(L//P)//x$ thus by our result for the case $k=1$ and the inductive hypothesis we get that L is stochastic.

We now prove

Theorem 2. The family of stochastic languages remains unaltered if the components of $\bar{S}_1$ as well as η are allowed to be arbitrary real numbers.

Proof. We distinguish two cases.

1. The components of $\bar{S}_1$ are arbitrary nonnegative reals.

Assume that $PA=(S(=\{s_1, s_2, ..., s_n\}), s_0, M)$ is a probabilistic automaton over the alphabet $I=\{x_1, x_2, ..., x_r\}$ and consider the language $L=L(PA, \bar{S}_1, \eta)=$ $=\{P\in W(I)|PA(P)\bar{S}_1>\eta\}$, where the components of $\bar{S}_1$ are arbitrary nonnegative numbers and η is an arbitrary real number. Let

$$\bar{S}_1 = \begin{Vmatrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{Vmatrix}$$

and $v=\max\{v_1, v_2, ..., v_n\}$. Since Theorem 2 is trivial if $v=0$, therefore, we shall deal with the case $v>0$ only. Moreover, we may assume that $0\leqq\eta<v$ because if $\eta\geqq v$ then L is void and if $\eta<0$ then clearly $L=L(PA, \bar{S}_1', 0)$, where a component of $\bar{S}_1'$ equals to 0 or 1 depending on whether the same component of $\bar{S}_1$ is 0 or positive. Thus in both cases L is stochastic.

Define $S^*=\{s_1, s_2, ..., s_{n+2}\}$, $s_0^*=(s_0, 0, 0)$ and let $PA^*=(S^*, s_0^*, M^*)$ be a probabilistic automaton over the alphabet $I^*=\{x_1, x_2, ..., x_{r+1}\}$, where

$$M^*(x) = \begin{Vmatrix} & & & 0 & 0 \\ & & & 0 & 0 \\ & M(x) & & \vdots & \vdots \\ & & & 0 & 0 \\ 0 & 0 & \dots\ 0 & 0 & 1 \\ 0 & 0 & \dots\ 0 & 0 & 1 \end{Vmatrix}$$

for every $x\in I$ and

$$M^*(x_{r+1}) = \begin{Vmatrix} & & & v_1/v & 1-v_1/v \\ & & & v_2/v & 1-v_2/v \\ & 0 & & \vdots & \vdots \\ & & & v_n/v & 1-v_n/v \\ 0 & 0 & \dots\ 0 & 0 & 1 \\ 0 & 0 & \dots\ 0 & 0 & 1 \end{Vmatrix}.$$

Let $\bar{S}_1^*$ denote the $n+2$-dimensional column vector whose $(n+1)$th component is 1 and the others are zero. Define $L^*=L(PA^*, \bar{S}_1^*, \eta/v)$. L^* is stochastic because $0\leqq\eta/v<1$. Our purpose is to show that $L^*=Lx_{r+1}$. Thus, by Theorem 1, it follows that $L=(Lx_{r+1})//x_{r+1}$ is a stochastic language. Since $L^*\subseteqq W(I)x_{r+1}$, therefore in order to prove this equation it is enough to verify that for every word $P\in W(I)$,

$$vPA^*(Px_{r+1})\bar{S}_1^* = PA(P)\bar{S}_1.$$

Indeed,

$$vPA^*(Px_{r+1})\bar{S}_1^* = v(s_0,0,0)\,M^*(P)\left\|\begin{matrix} & & & & v_1/v & 1-v_1/v\\ & & & & v_2/v & 1-v_2/v \\ & & 0 & & \vdots & \vdots \\ & & & & v_n/v & 1-v_n/v \\ 0 & 0 & \dots & 0 & 0 & 1 \\ 0 & 0 & \dots & 0 & 0 & 1\end{matrix}\right\| \left\|\begin{matrix}0\\0\\\vdots\\0\\1\\0\end{matrix}\right\| =$$

$$= (s_0,0,0)\left\|\begin{matrix} & & & 0 & 0\\ & & & 0 & 0\\ & M(P) & & \vdots & \vdots\\ & & & 0 & 0\\ 0 & 0 & \dots\; 0 & 0 & 1\\ 0 & 0 & \dots\; 0 & 0 & 1\end{matrix}\right\| \left\|\begin{matrix}v_1\\v_2\\\vdots\\v_n\\0\\0\end{matrix}\right\| = s_0M(P)\bar{S}_1 = PA(P)\bar{S}_1.$$

2. There exists at least one negative number among the components of $\bar{S}_1$. This case is traceable to the previous one by adding to η and to each component of $\bar{S}_1$ a number which is not smaller than the absolut value of the minimum of the components of $\bar{S}_1$.

After having written the article the author obtained knowledge of the fact, that among others the same problems had been solved in a different way by P. Turakainen in [2].

О двух проблемах А. Саломаа

В этой статьи мы решили две проблемы, поставленные А. Саломаа в [1]. Именно покажем, что правосторонные частные стохастические языки, образованные с любыми цепочками, являются стохастическими, наоборот, если имеется такой целое число k, что у одного языка все правосторонные частные, образованные всеми цепочками длиной k, стохастические, тогда он сам является стохастическим. Далее покажем, что семейство стохастических языков не расширяется, если компоненты выходного вектора любые действительные числа.

DEPT. OF COMPUTER SCIENCE
A. JÓZSEF UNIVERSITY
H—6720 SZEGED, HUNGARY
SOMOGYI U. 7.

Literature

[1] Salomaa, A., *Theory of Automata*, Pergamon Press, 1969.
[2] Turakainen, P., Generalized automata and stochastic languages, *Proc. Amer. Math. Soc.* v. 21, 1969, pp. 303—309.

(Received August 24, 1973)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

О сравнении предельных логик при моделировании в них конечно-значных логик

J. Demetrovics

В настоящей работе мы будем рассматривать предельные логики [1] с точки зрения их возможностей моделировать конечно-значные логики. Это свойство естественным образом индуцирует на множестве всех предельных логик некоторое отношение частичного порядка. Именно, каждой предельной логике ставится в соответствие некоторое покрытие натурального ряда бесконечной системой конечных множеств. Отношение порядка на множестве предельных логик индуцируется некоторым уточняемым ниже отношением порядка на множестве таких покрытий. Будет показано существование максимального и минимального элементов в этом частичном порядке и относительно естественно возникающей при этом эквивалентности предельных логик будет установлена континуальность классов эквивалентности.

Автор выражает свою благодарность С. В. Яблонскому, под руководством которого выполнена работа.

1°. $P_{\aleph_0}$ обозначает множество всех функций, переменные которых определены на множестве $E_{\aleph_0}$ мощности $\aleph_0$ и сами функции принимают значения из этого же множества. В качестве $E_{\aleph_0}$ возьмем множество всех целых неотрицательных чисел $\{0, 1, 2, \dots\}$. Множество $P_{\aleph_0}$ называется счетнозначной логикой. Обычно, $P_{\aleph_0}$ задается путём фиксации алфавита переменных $X = \{x_1, x_2, \dots\}$ и имен функций $f_\nu(x_{i_1}, \dots, x_{i_n})$, где ν пробегает некоторое континуальное множество индексов (обозначение: $P_{\aleph_0}(X)$).

Функция $g(z_1, \dots, z_i, \dots, z_n)$ называется функцией k — значной логики $P_k (k \geqq 2)$, если ее аргументы определены на множестве $E_k = \{0, 1, \dots, k-1\}$ и любое значение $g(a_1, \dots, a_i, \dots, a_n)$ принадлежит множеству E_k при $a_i \in E_k (1 \leqq i \leqq n)$.

Обычным образом на множествах $P_{\aleph_0}$ и P_k, $k \geqq 2$ определяется суперпозиция функций, понятие замыкания и замкнутого относительно суперпозиции класса функций [2, 3].

Множество функций $\mathfrak{A} \subseteqq P_{\aleph_0}(X)$ гомоморфно отображается на множество функций $\mathfrak{B} \subseteqq P_{\aleph_0}(Y)$, если

1. существует взаимно-однозначное соответствие между переменными $x_i \leftrightarrow y_i$;
2. каждой функции f из $\mathfrak{A}$ однозначно отвечает функция g из $\mathfrak{B}$, зависящая от соответствующих переменных;
3. всякой суперпозиции функций из $\mathfrak{A}$, принадлежащей $\mathfrak{A}$, отвечает анало-

гичная суперпозиция соответствующих функций системы $\mathfrak{B}$, которая также принадлежит $\mathfrak{B}$.

Если при этом отображении гомоморфизм имеет место в обе стороны, то говорят, что системы $\mathfrak{A}$ и $\mathfrak{B}$ изоморфны.

Замкнутый класс $P \subset P_{\aleph_0}$ называется предельной логикой, если

1. P состоит из счетного числа функций;

2. P содержит гомоморфные прообразы k — значных логик $P_k (k \geqq 2)$, т. е. для всякого натурального числа k, $k \geqq 2$ существует множество A_k из P, которое гомоморфно отображается на множество всех функций k — значной логики P_k.

2^0. Пусть $\mathfrak{A}$ — некоторая предельная логика, ε — подмножество множества $E_{\aleph_0}$ и пусть функция $f(x_1, \ldots, x_n) \in \mathfrak{A}$. Обозначим через $\bar{f}_\varepsilon(x_1, \ldots, x_n)$ функцию, определенную на множестве $\varepsilon \times \ldots \times \varepsilon$ и совпадающую на этом множестве с функцией $f(x_1, \ldots, x_n)$. Функцию $\bar{f}_\varepsilon(x_1, \ldots, x_n)$ назовем сужением функции $f(x_1, \ldots, x_n)$ на множество ε. Сужением множества функций $\mathfrak{M}$, $\mathfrak{M} \subseteq \mathfrak{A}$ на множество ε назовем множество $\bar{\mathfrak{M}}_\varepsilon$ сужений функций из $\mathfrak{M}$ на множество ε.

Скажем, что замкнутое множество функций $A_k \subset P_{\aleph_0}$ моделирует k — значную логику P_k на множестве $\varepsilon_k = \{e_0, e_1, \ldots, e_{k-1}\}$, $e_j < e_{j+1}$, $j = 0, 1, \ldots, k-2$, $\varepsilon_k \subset E_{\aleph_0}$, если существует функция $f(x_1, x_2)$, принадлежащая множеству A_k такая, что

$$\bar{f}_{\varepsilon_k}(x_1, x_2) = \begin{cases} e_{i+1}, & \text{если } (x_1, x_2) \in \varepsilon_k \times \varepsilon_k \text{ и } \max(x_1, x_2) = e_i, \text{ где } 0 \leqq i \leqq k-2; \\ e_0, & \text{если } (x_1, x_2) \in \varepsilon_k \times \varepsilon_k \text{ и либо } x_1 = e_{k-1}, \text{ либо } x_2 = e_{k-1}. \end{cases}$$

Заметим, что если множество функций A_k моделирует k — значную логику на множестве ε_k, то в нем существует подмножество, сужение которого на ε_k изоморфно P_k. В самом деле в качестве такого подмножества следует взять $[\{f(x_1, x_2)\}]$. Тогда $\bar{f}_{\varepsilon_k}(x_1, x_2) \leftrightarrow W_k(z_1, z_2)$, где $W_k(z_1, z_2)$ — функция Вебба [4], дает нужный нам изоморфизм.

Пусть дан замкнутый класс A, $A \subset P_{\aleph_0}$. Введем понятие таблицы класса T_A следующим образом.

Определение. Систему T_A конечных подмножеств множества $E_{\aleph_0}$ назовем таблицей класса A, если множество $\varepsilon_k = \{e_0, e_1, \ldots, e_{k-1}\}$ принадлежит T_A тогда и только тогда, когда класс A моделирует k — значную логику P_k на множестве ε_k. Посредством $E^A_{\aleph_0}$ обозначим множество $\bigcup_{\varepsilon \in T_A} \varepsilon$.

Легко видеть, что для каждого каласса A существует единственная таблица T_A.

Из определения таблицы T_A следует лемма.

Лемма 1. Класс функций A является предельной логикой тогда и только тогда, когда в его таблице T_A содержатся множества с любым числом элементов.

Предельные логики можно классифицировать по способу моделирования k — значных логик P_k.

Определение. Предельную логику P назовём возрастающей, если её таблица T_P содержит бесконечную последовательность конечных множеств $\Pi = \{\varepsilon^1, \varepsilon^2, \ldots, \varepsilon^i, \ldots\}$, такую что $\varepsilon^i \subset \varepsilon^{i+1}$, $i = 1, 2, 3, \ldots$. Обозначим через $E'^P_{\aleph_0} = \bigcup_{\varepsilon^i \in \Pi} \varepsilon^i$.

Предельную логику P назовём ящичной, если её таблица содержит бесконечную последовательность конечных множеств $\varepsilon^1_{n_1}, \varepsilon^2_{n_2}, \ldots, \varepsilon^i_{n_i}, \ldots,$ такую что

1. для любых i, j $(i \neq j)$ $\varepsilon^i_{n_i} \cap \varepsilon^j_{n_j} = \emptyset$;
2. если $\varepsilon \in T_p$, то существует i такое, что $\varepsilon \subseteqq \varepsilon^i_{n_i}$.

Определение. Замкнутый класс $\mathfrak{A}$ моделируется в замкнутом классе $\mathfrak{B}$, если существует однозначное отображение δ множества $E^{\mathfrak{A}}_{\aleph_0}$ в множества $E^{\mathfrak{B}}_{\aleph_0}$ такое, что

1. если $e_i \neq e_j$ $(e_i, e_j \in E^{\mathfrak{A}}_{\aleph_0})$, то $\delta(e_i) \neq \delta(e_j)$;
2. если $\varepsilon \in T_{\mathfrak{A}}$, то $\delta(\varepsilon) \in T_{\mathfrak{B}}$.

Это обстоятельство обозначим через $\mathfrak{B} \succeq \mathfrak{A}$ и скажем, что $\mathfrak{B}$ и $\mathfrak{A}$ сравнимы. Скажем, что логики $\mathfrak{A}$ и $\mathfrak{B}$ подобны (обозначение: $\mathfrak{A} \asymp \mathfrak{B}$), если $\mathfrak{B} \succeq \mathfrak{A}$ и $\mathfrak{A} \succeq \mathfrak{B}$.

Определение. Минимальной предельной логикой называется та логика, которая моделируется в любой предельной логике. Максимальной предельной логикой называется та логика, в которой моделируется любая предельная логика.

Лемма 2. В любой возрастающей предельной логике моделируется любая предельная логика.

Доказательство. Пусть P — возрастающая предельная логика, $\mathfrak{A}$ — любая предельная логика, и пусть Π — бесконечная последовательность из определения возрастающей логики и $E'^{P}_{\aleph_0} = \bigcup\limits_{\varepsilon^i \in \Pi} \varepsilon^i$. Тогда существует отображение $\delta\colon E^{\mathfrak{A}}_{\aleph_0}$ в $E'^{P}_{\aleph_0}$, удовлетворяющее условию: если $e_i \neq e_j$ $(e_i, e_j \in E^{\mathfrak{A}}_{\aleph_0})$, то $\delta(e_i) \neq \delta(e_j)$. Заметим, что из определения таблицы следует, что если множество $\varepsilon \in T_P$, то и любое подмножество ε, содержащее не менее двух элементов, также принадлежит T_P. Поэтому из свойства последовательности Π следует, что отображение δ удовлетворяет условию: если $\varepsilon \in T_{\mathfrak{A}}$, то $\delta(\varepsilon) \in T_P$. Лемма доказана.

Из леммы 2 вытекает

Следствие. Возрастающие предельные логики подобны между собой.

Лемма 3. В любой предельной логике моделируется любая ящичная логика.

Доказательство. Пусть $\mathfrak{A}$ — некоторая предельная логика, P — ящичная предельная логика, и пусть $\varepsilon^1_{n_1}, \varepsilon^2_{n_2}, \ldots, \varepsilon^i_{n_i}, \ldots$ — последовательность множеств из определения ящичной логики. Очевидно, что тогда $E^P_{\aleph_0} = \bigcup\limits_{i=1}^{\infty} \varepsilon^i_{n_i}$. Строим отображение $\delta\colon E^P_{\aleph_0}$ в $E^{\mathfrak{A}}_{\aleph_0}$ следующим образом. Пусть ε_{n_1} — некоторый элемент таблицы $T_{\mathfrak{A}}$, содержащий n_1 элементов. В качестве δ возьмем любое взаимно-однозначное соответствие между $\varepsilon^1_{n_1}$ и ε_{n_1}. Пусть отображение δ уже определено на множествах $\varepsilon^1_{n_1}, \varepsilon^2_{n_2}, \ldots, \varepsilon^i_{n_i}$. Пусть $n = n_1 + n_2 + \ldots + n_i + n_{i+1}$ и пусть ε_n — произвольный элемент таблицы $T_{\mathfrak{A}}$, содержащий n элементов. В нем можно выбрать подмножество $\varepsilon_{n_{i+1}}$, содержащее n_{i+1} элементов, такое что $\varepsilon_{n_{i+1}} \cap \delta(\varepsilon^k_{n_k}) = \emptyset$ $(k = 1, 2, \ldots, i)$. Очевидно, что $\varepsilon_{n_{i+1}} \in T_{\mathfrak{A}}$ и между $\varepsilon^i_{n_{i+1}}$ и $\varepsilon_{n_{i+1}}$ можно установить взаимно-однозначное соответствие. Любое из этих соответствий можно взять в качестве δ на множестве $\varepsilon^{i+1}_{n_{i+1}}$.

Лемма доказана.

Следствие. Из леммы 3 немедленно вытекает, что все ящичные предельные логики подобны между собой.

Из лемм 2 и 3 непосредственно вытекает справедливость следующей теоремы:

Теорема 1. Максимальной предельной логикой является возрастающая предельная логика. Минимальной предельной логикой является ящичная предельная логика.

Следствие. Предельная логика является максимальной (соответственно, минимальной) тогда и только тогда, когда она возрастающая (соотчетственно, ящичная).

3^0. Отношение $\succsim$ разбивает множество предельных логик на классы эквивалентности. Мы рассмотрим вопрос о мощности множества этих классов.

Пусть $\alpha=\{\alpha_1, \alpha_2, \alpha_3, \dots\}$ — бесконечная последовательность, состоящая из 0 и 1.

Определение. Последовательности $\alpha=\{\alpha_1, \alpha_2, \alpha_3, \dots\}$ и $\beta=\{\beta_1, \beta_2, \beta_3, \dots\}$ назовем эквивалентными, если существует такое k, $k\geqq 0$, что либо для всех i имеет место $\beta_i=\alpha_{i+k}$, либо для всех i справедливо $\alpha_i=\beta_{i+k}$, где $i\geqq 1$.

Другими словами, последовательности α и β назовем эквивалентными, если либо последовательность α является концом последовательности β, либо последовательность β является концом последовательности α.

Легко видеть, что справедлива следующая лемма:

Лемма 4. Максимальная мощность множества попарно неэквивалентных последовательностей равна континууму.

Каждой бесконечной последовательности $\alpha=\{\alpha_1, \alpha_2, \dots, \alpha_i, \dots\}$ можно поставить в соответствие бесконечную последовательность конечных множеств $\varepsilon_\alpha=\{\varepsilon_{\alpha_0}, \varepsilon_{\alpha_1}, \varepsilon_{\alpha_2}, \dots, \varepsilon_{\alpha_i}, \dots\}$, где $\varepsilon_{\alpha_0}=\{2, 4, 6, 8\}$ и при $i\geqq 1$

$$\varepsilon_{\alpha_i}=\{2+b_i, 4+b_i, 6+b_i, 8+b_i\},$$

где $b_i=2i-2a_i$ и $a_i=\sum_{j=1}^{i}\alpha_j$.

Лемма 5. Если в последовательности α i-ый член равен 1 (соответственно, $\alpha_i=0$) то в последовательности ε_α множества $\varepsilon_{\alpha_{i-1}}$ и ε_{α_i} имеют два (соответственно, один) общих члена.

Доказательство. Пусть $\varepsilon_{\alpha_{i-1}}=\{2+b_{i-1}, 4+b_{i-1}, 6+b_{i-1}, 8+b_{i-1}\}$ и

$$\varepsilon_{\alpha_i} = \{2+b_i, 4+b_i, 6+b_i, 8+b_i\},$$

и пусть $\alpha_i=1$. Тогда имеем, что $b_{i-1}=6(i-1)-2(a_i-1)=b_i-4$. Отсюда получим, что $6+b_{i-1}=b_i+2$ и $8+b_{i-1}=b_i+4$.

А если $\alpha_i=0$, то $b_{i-1}=6(i-1)-2a_i=b_i-6$. Отсюда немедленно получим, что $b_{i-1}+8=b_i+2$. Лемма доказана.

По последовательности ε_α построим замкнутый класс функций

$$A_\alpha = \left[\bigcup_{i=0}^{\infty} \{\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2)\}\right],$$

где

$$\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2) = \begin{cases} \max(x_1, x_2)+2, & \text{если } (x_1, x_2)\in\varepsilon_{\alpha_i}\times\varepsilon_{\alpha_i} \text{ и } \\ & x_1\neq 8+b_i \text{ и } x_2\neq 8+b_i; \\ 2+b_i, & \text{если } (x_1, x_2)\in\varepsilon_{\alpha_i}\times\varepsilon_{\alpha_i} \text{ и } \\ & \text{либо } x_1=8+b_i, \text{ либо } x_2=8+b_i; \\ 0 & \text{в остальных случаях.} \end{cases}$$

Лемма 6. Четырехэлементное множество ε принадлежит T_{A_α} тогда и только тогда, когда при некотором i имеет место $\varepsilon=\varepsilon_{\alpha_i}$.

Доказательство

а) Пусть $\varepsilon=\varepsilon_{\alpha_i}$. Возьмем функцию $\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2)$ из A_α. В силу определения этой функции $\varepsilon_{\alpha_i}\in T_{A_\alpha}$.

б) Пусть $\varepsilon\in T_{A_\alpha}$, и функция $f(x_1, x_2)\in A_\alpha$ на множестве ε моделирует P_4. Так как $f(x_1, x_2)\in A_\alpha$, то $f(x_1, x_2)$ может быть получена из $\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2)$ путем суперпозиции. Если в этой суперпозиции участвуют функции $\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2)$ и $\varphi_{\varepsilon_{\alpha_{i+j}}}(x_1, x_2)$ $(j\geqq 2)$, то $f(x_1, x_2)\equiv 0$. Следовательно, в этой суперпозиции участвуют только $\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2)$ и $\varphi_{\varepsilon_{\alpha_{i+1}}}(x_1, x_2)$ для некоторого i. По лемме 5 ε_{α_i} и $\varepsilon_{\alpha_{i+1}}$, имеют не более двух общих элементов отличных от нуля e_1 и e_2. Поэтому $f(a_1, a_2)\neq 0$, если только $a_i\in\{e_1, e_2\}$ $(i=1, 2)$. Следовательно, в суперпозиции может принимать участие только одна функция $\varphi_{\varepsilon_{\alpha_i}}(x_1, x_2)$ для некоторого i. Но это означает, что $\varepsilon=\varepsilon_{\alpha_i}$. Лемма доказана.

Лемма 7. Если последовательности α и β не эквивалентны, то A_α и A_β несравнимы.

Доказательство. Пусть α и β — неэквивалентные последовательности. Допустим, что A_α и A_β сравнимы. Пусть, например, $A_\beta\succsim A_\alpha$. Тогда в силу определения отношения $\succsim$ существует такое отображение $\delta:\varepsilon_\alpha$ в ε_β, что:

1. если $e_i\neq e_j$, то $\delta(e_i)\neq\delta(e_j)$;
2. если $\varepsilon\in T_{A_\alpha}$, то $\delta(\varepsilon)\in T_{A_\beta}$.

Рассмотрим последовательность множеств $\varepsilon_{\beta_0}, \varepsilon_{\beta_1}, \varepsilon_{\beta_2}, \ldots$. Все они суть элементы T_{A_β}. Поэтому $\delta(\varepsilon_{\alpha_0}), \delta(\varepsilon_{\alpha_1}), \delta(\varepsilon_{\alpha_2}), \ldots$ в силу 2. должны быть элементами T_{A_β}. По лемме 6 $\delta(\varepsilon_{\alpha_i})$, $i=0, 1, 2, \ldots$ есть одно из множеств ε_{β_j}. Так как ε_{α_i} и $\varepsilon_{\alpha_{i+1}}$ пересекаются, то $\delta(\varepsilon_{\alpha_i})$ и $\delta(\varepsilon_{\alpha_{i+1}})$ также должны пересекаться. Но это возможно только в том случае, если $\delta(\varepsilon_{\alpha_i})$ есть некоторое ε_{β_j}, а $\delta(\varepsilon_{\alpha_{i+1}})$ есть $\varepsilon_{\beta_{j+1}}$. Отсюда и из 1. немедленно следует, что если $\delta(\varepsilon_{\alpha_0})=\varepsilon_{\beta_k}$, то $\delta(\varepsilon_{\alpha_1})=\varepsilon_{\beta_{k+1}},\ldots$, $\delta(\varepsilon_{\alpha_n})=\varepsilon_{\beta_{k+n}}, \ldots$. В силу построения по последовательностям α и β множеств ε_{α_i} и ε_{β_i} имеем: $\alpha_i=\beta_{k+i}$, $i=0, 1, 2, \ldots$, т. е. α и β — эквивалентные последовательности. Это противоречит условию леммы.

Лемма доказана.

Рассмотрим следующее разбиение множества $E_{\aleph_0}$: $E_{\aleph_0}=\varepsilon_0\cup\varepsilon_1$, где $\varepsilon_0= \{0, 1, 3, \ldots, 2r+1, \ldots\}$ и $\varepsilon_1 = \{2, 4, 6, \ldots, 2r, \ldots\}$, $r\geqq 0$. Множество ε_0 представим в виде $\bigcup\limits_{k=1}^{\infty}\varepsilon_k^1$, где $\varepsilon_1^1=\{0\}$ и при $k\geqq 2$ $\varepsilon_k^1=\{k(k-1)-1, k(k-1)+1, k(k-1)+3, \ldots, k(k-1)+2k-3\}$.

Определим функцию $\varphi_k(x_1, x_2)$, $k\geqq 2$, областью определения которой является множество $E_{\aleph_0}\times E_{\aleph_0}$, а областью значений множество $\varepsilon_k^1\cup\{0\}$:

$$\varphi_k(x_1,x_2)=\begin{cases}\max(x_1,x_2)+2, & \text{если } (x_1,x_2)\in\varepsilon_k^1\times\varepsilon_k^1 \text{ и } \\ & x_1\neq k(k-1)+2k-3 \text{ и } x_2\neq k(k-1)+2k-3;\\ k(k-1)-1, & \text{если } (x_1,x_2)\in\varepsilon_k^1\times\varepsilon_k^1 \text{ и либо } x_1=k(k-1)+2k-3,\\ & \text{либо } x_2=k(k-1)+2k-3;\\ 0 & \text{в остальных случаях.}\end{cases}$$

Через R обозначим замыкание множества $\bigcup_{k=1}^{\infty}[\{\varphi_k(x_1,x_2)\}]$, где $\varphi_1(x_1,x_2)\equiv 0$. Легко видеть, что класс функций R является предельной логкой.

Теорема 2. Максимальная мощность множества попарнонесравниыхм предельных логик равна континууму. е

Доказательство. Заметим, что множество функций $[A_\alpha\cup R]$ явл яется пр-и дельной логикой, так как R — предельная логика, а мощность класса A_α счетна.

Легко видеть, что предельная логика R и класс функций A_α несравнимы. Из леммы 7 и из того, что любая суперпозиция, содержащая функции из R и из $A_\alpha(A_\beta)$, тождественно равна нулю, вытекает, что предельные логики $[A_\alpha\cup R]$ и $[A_\beta\cup R]$ несравнимы. А континуальность семейства предельных логик $\{[A_\alpha\cup R]\}$ вытекает из леммы 4.

Теорема доказана.

Автор выражает благодарность В. Б. Кудрявцеву за ценные советы и помощь, оказанную при работе.

On the comparison of limit logics by the simulation of finite-valued logic

A class of functional systems is considered known by the name "limit logics". In this class there is introduced a relation of partial ordering, which allows the comparison of different limit logics in the respect of their ability of simulating finite-valued logics. The existence of maximal and minimal elements with respect to this ordering is proved.

It is also proved, that the family of the equivalence classes naturally generated by this ordering has the cardinality of continuum.

COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES
H—1502 BUDAPEST. HUNGARY

Литература

[1] Яблонский, С. В., О предельных логиках. *ДАН СССР*, том 118, вып. 3, 1958, стр. 657—660.

[2] Яблонский, С. В., Функциональные построения в k —значной логике, *Труды мат. ин-та им. В. А. Стеклова*, том 51, 1958, стр. 5—142.

[3] Гаврилов, Г. П., О функциональной полноте в счетно-значной логике, *Сб. Проблемы кибернетики*, вып. 15, 1965, стр, 5—64.

[4] Webb, D., Generation of any n-valued logic by one binary operator, *Proc. Nat. Acad. Sci. U.S.A.*, v. 21, 1935, pp. 252—254.

(Поступило 24-ого января 1974 г.)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

Endomorphisms of group-type quasi-automata

By I. Babcsányi

In this paper the endomorphisms of group-type quasi-automata are investigated using the concept of the generating system of quasi-automata. For the notions and notations which are not defined here, we refer the reader to [4] or [5].

Let the characteristic semigroup $\bar{F}=F/\varrho_A$ of an arbitrary quasi-automaton $\mathbf{A}=(A, F, \delta)$ be a monoid, and let $\bar{e}$ $(e\in F)$ be the identity element of $\bar{F}$. Take the subset $A'=\langle\delta(a,f)|\ a\in A; f\in F\rangle$ of A and the A-sub-quasi-automaton $\mathbf{A}'=(A', F, \delta')$ of $\mathbf{A}$. It is easy to see that $a\in A'$ if and only if $\delta(a, e)=a$ for an arbitrary state a of $\mathbf{A}$. Furthermore, the characteristic semigroup of $\mathbf{A}'$ is equal to that of $\mathbf{A}$. Assume that the set $A\setminus A'$ is non-empty. Let V be an arbitrary (non-empty) subset of $A\setminus A'$, and let π denote a mapping of V into $A\setminus A'$. Moreover, let α' be an endomorphism of $\mathbf{A}'$. The following holds:

Theorem 1. *The mapping* α: $A\to A$, *defined by*

$$\alpha(a)=\begin{cases}\alpha'(a) & \text{if}\quad a\in A',\\ \alpha'(\delta(a,e)) & \text{if}\quad a\in A\setminus A',\end{cases} \tag{1}$$

is an endomorphism of $\mathbf{A}$. *The mapping* α_π: $A\to A$, *for which*

$$\alpha_\pi(a)=\begin{cases}\alpha'(a) & \text{if}\quad a\in A',\\ \pi(a) & \text{if}\quad a\in V,\\ \alpha'(\delta(a,e)) & \text{if}\quad a\in (A\setminus A')\setminus V\end{cases} \tag{2}$$

holds, is an endomorphism of $\mathbf{A}$ *if and only if*

$$\alpha'(\delta(a,e))=\delta(\pi(a),e) \tag{3}$$

holds for every $a(\in V)$. *Furthermore, if* β *is an endomorphism of* $\mathbf{A}$, *then* β *is a mapping of type* (1) *or* (2).

Proof. α and α_π are well-defined. It can immediately be seen that α is an endomorphism of $\mathbf{A}$. Now let $a(\in V)$, $b(\in (A\setminus A')\setminus V)$ and $f(\in F)$ be arbitrary elements. Assume that the condition (3) holds. Then

$$\alpha_\pi(\delta(a,f))=\alpha'(\delta(a,f))=\alpha'(\delta(a,ef))=\alpha'(\delta(\delta(a,e),f))=$$
$$=\delta(\alpha'(\delta(a,e)),f)=\delta(\delta(\pi(a),e),f)=\delta(\pi(a),ef)=\delta(\alpha_\pi(a),f),$$

and

$$\alpha_\pi(\delta(b,f)) = \alpha'(\delta(b,f)) = \alpha'(\delta(b, ef)) =$$
$$= \alpha'(\delta(\delta(b, e),f)) = \delta(\alpha'(\delta(b, e)),f) = \delta(\alpha_\pi(b),f).$$

These mean that α_π is an endomorphism of **A**. Conversely, if (2) is an endomorphism of **A**, then for every $a(\in V)$ we get

$$\alpha'(\delta(a, e)) = \alpha_\pi(\delta(a, e)) = \delta(\alpha_\pi(a), e) = \delta(\pi(a), e),$$

that is, (3) holds.

Take an arbitrary endomorphism β of **A**. We prove the following implications:

$$a\in A' \Rightarrow \beta(a)\in A',$$
$$a\in A\setminus A' \Rightarrow \beta(a)\in A\setminus A' \quad \text{or} \quad \beta(a) = \beta(\delta(a, e)).$$

If $a\in A'$, there are $b(\in A)$ and $f(\in F)$ such that $\delta(b,f)=a$. Then

$$\beta(a) = \beta(\delta(b,f)) = \delta(\beta(b),f)\in A'.$$

If $a\in A\setminus A'$ and $\beta(a)\in A'$, there are $b(\in A)$ and $f(\in F)$ such that $\beta(a)=\delta(b,f)$. That is,

$$\beta(a) = \delta(b,f) = \delta(b,fe) = \delta(\delta(b,f), e) = \delta(\beta(a), e) = \beta(\delta(a, e)).$$

Let β be an arbitrary endomorphism of **A** and let β' be an endomorphism of **A'** for which $\beta'(a)=\beta(a)$ $(a\in A')$. If $V=\langle a|\beta(a)\in A\setminus A'\rangle$ is a non-empty set, then β is a mapping (2). If V is the empty set, then β is a mapping (1).

Consequently, we can give the endomorphisms of **A**, if we know the endomorphisms of **A'**. In Theorem 3 we give all of the endomorphisms of **A'**, if **A'** is a group-type quasi-automaton.

A non-empty subset B of the state set A of a quasi-automaton $\mathbf{A}=(A, F, \delta)$ is called a *generating system* of **A** if for each state $a(\in A)$ there exists a state $b(\in B)$ and a $f(\in F)$ such that $\delta(b,f)=a$. A generating system B of **A** is *minimal* if none of the proper subset of B is a generating system of **A**. A quasi-automaton is said to be *(finitely) generated* if it has a (finite) generating system. (We note that a quasi-automaton is called *cyclic* if it has an one-element generating system.)

Let the characteristic semigroup $\bar{F}$ of a quasi-automaton $\mathbf{A}=(A, F, \delta)$ be again a monoid and let $\bar{e}$ $(e\in F)$ be the identity element of $\bar{F}$. It can easily be proved that the quasi-automaton **A** has a generating system if and only if

$$\forall_{a\in A}\ a[\delta(a, e) = a]. \tag{4}$$

In the following lemma the theorem of Yu. I. Sorkin [7] concerning finitely generated automata are generalised on generated quasi-automata.

Lemma 1. *If G_1 and G_2 are two minimal generating systems of a generated quasi-automaton $\mathbf{A}=(A, F, \delta)$ then $|G_1|=|G_2|$.* [1]

[1] $|A|$ is the cardinal number of the set A.

Proof. Let G_1 and G_2 be two minimal generating systems of **A**. For every $a_2(\in G_2)$, there exist $a_1(\in G_1)$ and $f(\in F)$ such that $\delta(a_1,f)=a_2$ holds. It can easily be seen that the set

$$G_{12} = \langle a \mid a\in G_1 \quad \text{and} \quad \underset{f\in F}{\exists} f[\delta(a,f)\in G_2]\rangle$$

is also a generating system of **A**. Since $G_{12}\subseteqq G_1$ and G_1 is a minimal generating system of **A**, thus $G_{12}=G_1$. Assume that $\delta(a_1,f)$, $\delta(a_1,h)\in G_2$ $(a_1\in G_1; f, h\in F)$. There exists a $k(\in F)$ such that $\delta(a_1,fk)=\delta(\delta(a_1,f),k)\in G_1$. Since G_1 is a minimal generating system of **A**, thus $\delta(a_1,fk)=a_1$, that is,

$$\delta(\delta(a_1,f),kh) = \delta(\delta(a_1,fk),h) = \delta(a_1,h)\in G_2.$$

Since G_2 is also a minimal generating system of **A**, we get that $\delta(a_1,h)=$ $=\delta(\delta(a_1,f),kh)=\delta(a_1,f)$. Furthermore, if $\delta(a_1,f)=\delta(a_1',g)\in G_2(a_1'\in G_1, g\in F)$, then $a_1=\delta(a_1,fk)=\delta(a_1',gk)$, that is, $a_1=a_1'$. Consequently, the mapping $\varphi\colon G_1\to G_2$, for which

$$\varphi(a_1) = a_2 \Leftrightarrow \underset{f\in F}{\exists} f[\delta(a_1,f) = a_2]$$

holds, is an one-to-one mapping of G_1 onto G_2.

We define the following relation ϱ on A:

$$a\varrho b\,(a,b\in A) \Leftrightarrow \underset{c\in A;\,f,g\in F}{\exists} (c,f,g)\,[\delta(c,f) = a,\ \delta(c,g) = b]. \tag{5}$$

If the quasi-automaton $\mathbf{A}=(A, F, \delta)$ is generated then ϱ is a reflexive and symmetric relation. If the quasi-automaton **A** is generated and the characteristic semigroup $\bar{F}$ of **A** is a group then ϱ is an equivalence relation.

A non-empty subset E of the state set A of a quasi-automaton $\mathbf{A}=(A, F, \delta)$ is called a *strongly connected subset* of A, if for every a, b $(\in E)$ there exists an $f(\in F)$ such that $\delta(a,f)=b$. A partition C of A is called *strongly connected,* if $C(a)$ is a strongly connected subset of A for every $a(\in A)$ ($C(a)$ denotes the class of C containing the element a).

Lemma 2. *If the characteristic semigroup of a generated quasi-automaton* $\mathbf{A}=(A, F, \delta)$ *is a group, then* C_ϱ *is a strongly connected partition of* A*, where* C_ϱ *is the partition on* A *induced by* ϱ.

Proof. Let $a, b\in C_\varrho(c)$ $(c\in A)$, then there exist $f\in F$ and $g\in F$ such that $\delta(c,f)=a$ and $\delta(c,g)=b$. Since $\bar{F}$ is a group, there exists an $h(\in F)$ such that $\bar{f}\bar{h}=\bar{g}$, therefore,

$$\delta(a,h) = \delta(\delta(c,f),h) = \delta(c,fh) = \delta(c,g) = b,$$

that is, $C_\varrho(c)$ is a strongly connected subset of A.

Assume that the conditions of this Lemma are satisfied. It can easily be seen that $C_\varrho(a)=\langle\delta(a,f)\mid f\in F\rangle$ holds for every $a(\in A)$. Thus $\mathbf{C}_\varrho(a)=(C_\varrho(a), F, \delta_a)$ is a strongly connected sub-quasi-automaton of **A** for every $a(\in A)$ (cf. CH. A. TRAUTH [6]).

Lemma 3. *If the characteristic semigroup of a generated quasi-automaton* $\mathbf{A}=$ $=(A, F, \delta)$ *is a group, then* **A** *has a minimal generating system.*

Proof. By Lemma 2, C_ϱ is a strongly connected partition of A. Let $G(\subseteq A)$ such that $A=\bigcup_{a\in G} C_\varrho(a)$ and if $a\neq b$ $(\in G)$ then $C_\varrho(a)\neq C_\varrho(b)$. We can easily prove that G is a minimal generating system of **A**.

We note that if G is a minimal generating system of **A** then $A=\bigcup_{a\in G} C_\varrho(a)$ and if $a\neq b(\in G)$ then $C_\varrho(a)\neq C_\varrho(b)$.

It is possible that C_ϱ is a strongly connected partition of A if the characteristic semigroup of **A** is not a group. Take the following example:

A	1	2	3	4	5
x	2	1	1	5	4
y	3	2	2	4	5

$C_\varrho(1)=\langle 1, 2, 3\rangle$ and $C_\varrho(4)=\langle 4, 5\rangle$ are strongly connected subsets of A, $C_\varrho(1)\cup C_\varrho(4)=A$ and $C_\varrho(1)\cap C_\varrho(4)=\emptyset$. But $\overline{F(X)}$ is not a group. ($F(X)$ denotes the free semigroup with out identity element generated by $X=\langle x, y\rangle$.) Note that $G=\langle 1, 4\rangle$ is a minimal generating system of **A**.

Theorem 2. *If a quasi-automaton* $\mathbf{A}=(A, F, \delta)$ *is finitely generated and* C_ϱ *is a strongly connected partition of* A *then*

$$o(E(A)) \geqq \prod_{i=1}^{k} o(E(C_\varrho(a_i))), \tag{6}$$

where $G=\langle a_1, \ldots, a_k\rangle$ *is a minimal generating system of* **A**.

Proof. $E(A)$ and $E(C_\varrho(a_i))$ denote the endomorphism semigroups of the quasi-automaton $\mathbf{A}=(A, F, \delta)$ and $\mathbf{C}_\varrho(a_i)=(C_\varrho(a_i), F, \delta_{a_i})$ $(a_i\in G)$, respectively. Denote by $\alpha=\bigcup_{a_i\in G}\alpha_{a_i}$ the following mapping of A into itself:

$$\alpha(a)=\alpha_{a_i}(a), \quad \text{if} \quad a\in C_\varrho(a_i) \tag{7}$$

where $\alpha_{a_i}\in E(C_\varrho(a_i))$. It can easily be proved that $\alpha\in E(A)$. Furthermore, if

$$\alpha=\bigcup_{a_i\in G}\alpha_{a_i}(\alpha_{a_i}\in E(C_\varrho(a_i))) \quad \text{and} \quad \beta=\bigcup_{a_i\in G}\beta_{a_i}(\beta_{a_i}\in E(C_\varrho(a_i)))$$

such that $\alpha=\beta$, then $\alpha_{a_i}=\beta_{a_i}$ for every a_i $(\in G)$.

Lemma 4. *If a group-type quasi-automaton* $\mathbf{A}=(A, F, \delta)$ *is generated, then the sub-quasi-automaton* $\mathbf{C}_\varrho(a)$ *is quasi-perfect and the characteristic group of* $\mathbf{C}_\varrho(a)$ *is equal to the characteristic group of* **A** *for every* a $(\in A)$. *Moreover* $\mathbf{C}_\varrho(a)\cong\mathbf{C}_\varrho(b)$ *for every pair* a, b $(\in A)$.

Proof. Let a $(\in A)$ and f, g $(\in F)$ such that

$$\forall_{h\in F}\, h[\delta(a, hf)=\delta(\delta(a, h), f)=\delta(\delta(a, h), g)=\delta(a, hg)].$$

Since **A** is state-independent, thus $\bar{h}\bar{f}=\bar{h}\bar{g}$. Let $\bar{h}=\bar{e}$, where $\bar{e}$ is the identity element of the characteristic group of **A**, then $\bar{f}=\bar{g}$. Consequently, the characteristic group of $\mathbf{C}_\varrho(a)$ is equal to the characteristic group of **A**. A sub-quasi-automaton of a state-independent quasi-automaton is also state-independent, therefore, by Lemma 2, $\mathbf{C}_\varrho(a)$ is quasi-perfect. Let $a, b\ (\in A)$ be arbitrary states. It is clear that the mapping $\delta(a,f)\to\delta(b,f)\ (f\in F)$ is an isomorphic mapping of $\mathbf{C}_\varrho(a)$ onto $\mathbf{C}_\varrho(b)$.

Corollary 1. *If a group-type A-finite quasi-automaton* $\mathbf{A}=(A, F, \delta)$ *is generated, then* $O(\bar{F})\,|\,|A|$.[2]

Proof. From Lemma 4 and Theorem 7 of Ch. A. Trauth [6] we get that $|C_\varrho(a)|=O(\bar{F})$ for every $a\ (\in A)$. $|C_\varrho(a)|=|C_\varrho(b)|$ follows also from Lemma 4 for every pair $a, b\ (\in A)$. Thus $O(\bar{F})=|C_\varrho(a)|\,|\,|A|$.

Corollary 2. *If an A-finite group-type quasi-automaton* $\mathbf{A}=(A,F,\delta)$ *is generated and* $|A|$ *is a prime number, then either* $\bar{F}$ *has only one element or* $\mathbf{A}$ *is quasi-perfect.*

Proof. By Corollary 1, if $|A|$ is a prime number, then either $O(\bar{F})=1$ or $|C_\varrho(a)|= =O(\bar{F})=|A|\ (a\in A)$. If $|A|=|C_\varrho(a)|\ (a\in A)$, then $\mathbf{A}$ is a cyclic quasi-automaton. Cyclic group-type quasi-automaton is quasi-perfect (Ch. A. Trauth [6]).

Theorem 3. *If a group-type quasi-automaton* $\mathbf{A}=(A, F, \delta)$ *is generated, then there exist a subsemigroup* T *and two subgroups* H *and* P *of the endomorphism semigroup* $E(A)$ *of* $\mathbf{A}$ *such that*

$$E(A) = TH, \quad G(A) = PH = HP, \quad T\cap H = \{\iota\}, \quad P \subseteqq T$$

hold, where ι *is the identity element of* $E(A)$.[3]

Proof. Let the group-type quasi-automaton $\mathbf{A}=(A, F, \delta)$ be generated. By Lemma 3, there exists a minimal generating system G of $\mathbf{A}$. Let H denote the set of all endomorphisms (7). By Lemma 4 and Theorem 4 of I. Babcsányi [1], the endomorphisms (7) are automorphisms of $\mathbf{A}$. H is a subgroup of the automorphism group $G(A)$ of $\mathbf{A}$ under the usual multiplication of mappings.

Let π be an arbitrary mapping of G into itself. We define the mapping φ_π: $A\to A$ by

$$\varphi_\pi(\delta(c,f)) = \delta(\pi(c),f) \quad (c\in G,\ f\in F). \tag{8}$$

We show that φ_π is an endomorphism of $\mathbf{A}$. Let a be an arbitrary state of $\mathbf{A}$ and let $c\in G$ and $f, g\in F$ such that $a=\delta(c,f)=\delta(c, g)$. Since $\mathbf{A}$ is state-independent, thus $\delta(\pi(c),f)=\delta(\pi(c), g)$, that is, φ_π is well-defined. If $a=\delta(c, h)\ (c\in G, h\in F)$ and $f\in F$ then

$$\varphi_\pi(\delta(a,f)) = \varphi_\pi(\delta(\delta(c, h),f)) = \varphi_\pi(\delta(c, hf)) = \delta(\pi(c), hf) =$$
$$= \delta(\delta(\pi(c), h),f) = \delta(\varphi_\pi(\delta(c, h)),f) = \delta(\varphi_\pi(a),f),$$

that is, $\varphi_\pi\in E(A)$. Let T denote the set of all mappings (8). T is a subsemigroup of $E(A)$. Namely, if $\varphi_\pi, \varphi_{\pi'}\in T$ and $a=\delta(c, h)$, then

$$\varphi_\pi\varphi_{\pi'}(a) = \varphi_\pi\varphi_{\pi'}(\delta(c, h)) = \varphi_\pi(\delta(\pi'(c), h)) =$$
$$= \delta(\pi\pi'(c), h) = \varphi_{\pi\pi'}(\delta(c, h)) = \varphi_{\pi\pi'}(a)$$

that is, $\varphi_\pi\varphi_{\pi'}=\varphi_{\pi\pi'}\in T$.

[2] If n and k are natural numbers then $k|n$ means that n can be divided by k.

[3] $TH=\langle\varphi\alpha|\varphi\in T, \alpha\in H\rangle$.

If π is a permutation of G and $\varphi_\pi(a)=\varphi_\pi(b)$ $(a, b\in A)$ then there exist $c, d\in G$ and $h, k\in F$ such that $\delta(c, h)=a$ and $\delta(d, k)=b$, therefore

$$\delta(\pi(c), h) = \varphi_\pi(\delta(c, h)) = \varphi_\pi(a) = \varphi_\pi(b) = \varphi_\pi(\delta(d, k)) = \delta(\pi(d), k).$$

Let $k'\in\bar{k}^{-1}$, then $\delta(\pi(c), hk')=\delta(\pi(d), kk')=\pi(d)$. Since $\pi(c), \pi(d)\in G$ and G is a minimal generating system of **A**, thus $\pi(c)=\pi(d)$, that is, $c=d$ and $\bar{h}=\bar{k}$. Therefore $a=b$, that is, φ_π is an one-to-one mapping. Now let a be an arbitrary state of **A**, then there exist $d\in G$ and $f\in F$ such that $\delta(d,f)=a$. Furthermore, there exists a $c\in G$ such that $\pi(c)=d$, because π is a permutation of G. Thus

$$\varphi_\pi(\delta(c,f)) = \delta(\pi(c),f) = \delta(d,f) = a,$$

that is, φ_π is onto. Consequently, if π is a permutation of G, then $\varphi_\pi\in G(A)$. Denote by P the set of this automorphisms φ_π. It is obvious, that P is a subgroup of $G(A)$. It can easily be seen that $T\cap H=\{\iota\}$, $P\subseteq T$, $TH\subseteq E(A)$ and PH, $HP\subseteq G(A)$ hold.

Now, we prove that $E(A)\subseteq TH$. Let $\beta\in E(A)$ and $a\in A$. There exist states $c, d(\in G)$ such that $a\in C_\varrho(c)$ and $\beta(a)\in C_\varrho(d)$. Take the mapping π of G into itself such that $\pi(c)=d$. We show that π is well-defined. Let $b\in C_\varrho(c)$ and suppose that $\beta(b)\in C_\varrho(d')$ $(d'\in G)$. There exist $h, h'\in F$ for which $\delta(c, h)=a$ and $\delta(c, h')=b$ hold. Thus $\beta(a)=$ $=\beta(\delta(c, h))=\delta(\beta(c), h)$ and $\beta(b)=\beta(\delta(c, h'))=\delta(\beta(c), h')$, that is, $C_\varrho(d)=C_\varrho(d')$, thus $d=d'$. We define φ_π as in (8). If $\beta(a)=\delta(d, k)$ $(k\in F)$, then let α_c be an automorphism of $\mathbf{C}_\varrho(c)$ such that $\alpha_c(a)=\alpha_c(\delta(c, h))=\delta(c, k)$. (Since $\mathbf{C}_\varrho(c)$ is quasi-perfect, therefore the automorphism group of $\mathbf{C}_\varrho(c)$ is transitive, thus α_c exists (CH. A. TRAUTH [6]).) We prove that α_c depends only on β. Let $b\in C_\varrho(c)$ and $\delta(c, h')=b$ $(h'\in F)$, furthermore $\bar{h}'=\bar{h}l$ $(l\in F)$. Then

$$b = \delta(c, h') = \delta(c, hl) = \delta(\delta(c, h), l) = \delta(a, l).$$

Thus, if $\beta(b)=\delta(d, k')$ $(k'\in F)$, then

$$\delta(d, k') = \beta(b) = \beta(\delta(a, l)) = \delta(\beta(a), l) = \delta(\delta(d, k), l) = \delta(d, kl).$$

Since **A** is state-independent, thus $\bar{k}'=\bar{k}l$, that is,

$$\alpha_c(b) = \alpha_c(\delta(c, h')) = \alpha_c(\delta(c, hl)) = \alpha_c(\delta(\delta(c, h), l)) =$$

$$= \delta(\alpha_c(\delta(c, h)), l) = \delta(\delta(c, k), l) = \delta(c, kl) = \delta(c, k').$$

Thus

$$\varphi_\pi\alpha_c(a) = \varphi_\pi\alpha_c(\delta(c, h)) = \varphi_\pi(\delta(c, k)) = \delta(d, k) = \beta(a).$$

Take this α_c for every c $(\in G)$ and let $\alpha= \bigcup_{c\in G} \alpha_c$. It is clear that $\beta=\varphi_\pi\alpha$, that is, $\beta\in TH$, since $\varphi_\pi\in T$ and $\alpha\in H$. Therefore $E(A)\subseteq TH$, thus $E(A)=TH$.

Suppose that $\beta=\varphi_\pi\alpha\in G(A)$. Since $\alpha\in G(A)$, therefore $\varphi_\pi=\beta\alpha^{-1}\in G(A)$. If $a\in G$ then $\varphi_\pi(a)=\pi(a)$, that is, π is a permutation of G, thus $\varphi_\pi\in P$. We get that $G(A)=$ $=PH$. Finally, we shall show that $PH=HP$. Let $\varphi_\pi(\in P)$ and $\alpha(\in H)$ be arbitrary endomorphisms. Furthermore, let $a=\delta(c, h)$ $(c\in G, h\in F)$ be an arbitrary state of **A** and let $\alpha(a)=\delta(c, k)$ $(k\in F)$. Take the automorphism $\alpha_{\pi(c)}$ of $\mathbf{C}_\varrho(\pi(c))$ such that

$$\alpha_{\pi(c)}(\delta(\pi(c), h)) = \delta(\pi(c), k).$$

It can easily be seen that $\alpha_{\pi(c)}$ depends only on α. Since π is a permutation of G, therefore the mapping $\alpha_{\pi(c)} \to C_\varrho(\pi(c))$ is one-to-one and $\alpha' = \bigcup_{c \in G} \alpha_{\pi(c)} \in H$. Thus

$$\alpha' \varphi_\pi(a) = \alpha' \varphi_\pi(\delta(c, h)) = \alpha'(\delta(\pi(c), h)) = \delta(\pi(c), k) =$$
$$= \varphi_\pi(\delta(c, k)) = \varphi_\pi \alpha(\delta(c, h)) = \varphi_\pi \alpha(a),$$

that is, $\alpha' \varphi_\pi = \varphi_\pi \alpha$. Thus $G(A) = PH \subseteq HP$, therefore $PH = HP$.

Corollary 3. *If a group-type quasi-automaton* $\mathbf{A} = (A, F, \delta)$ *is generated, then*

$$\varphi\alpha = \psi\beta \Rightarrow \varphi = \psi \quad \textit{and} \quad \alpha = \beta,$$

where $\varphi, \psi \in T$ *and* $\alpha, \beta \in H$.

Proof. Let $\varphi, \psi \in T$ and $\alpha, \beta \in H$ such that $\varphi\alpha = \psi\beta$, then $\varphi\alpha\beta^{-1} = \psi$. Let G be a minimal generating system of $\mathbf{A}$ and $c \in G$, then $\varphi(\alpha\beta^{-1}(c)) = \psi(c)$. Since $\alpha\beta^{-1}(c) \in \in C_\varrho(c)$, there exists $f \in F$ such that $\alpha\beta^{-1}(c) = \delta(c, f)$, that is,

$$\psi(c) = \varphi(\alpha\beta^{-1}(c)) = \varphi(\delta(c, f)) = \delta(\varphi(c), f).$$

Since $\varphi(c), \psi(c) \in G$, thus $\varphi(c) = \psi(c)$ $(c \in G)$ and $\bar{f} = \bar{e}$, where $\bar{e}$ is the identity element of $\bar{F}$. We get that $\varphi = \psi$ and $\alpha\beta^{-1}(c) = \delta(c, f) = \delta(c, e) = c$, that is $\alpha = \beta$.

Corollary 4. *Let a group-type quasi-automaton* $\mathbf{A} = (A, F, \delta)$ *be generated. If* $O(\bar{F}) > 1$, *then* P *is isomorphic to a subgroup of the automorphism group of* H. *If* $O(\bar{F}) = 1$ *then* $H = \{\iota\}$.

Proof. Let $\varphi \in P$. We define the following mapping ω_φ of H into itself:

$$\omega_\varphi(\alpha) = \alpha' \Leftrightarrow \varphi\alpha = \alpha'\varphi. \tag{9}$$

ω_φ is one-to-one and onto. Let $\alpha_1, \alpha_2 \in H$ then

$$(\alpha_1\alpha_2)'\varphi = \varphi(\alpha_1\alpha_2) = (\varphi\alpha_1)\alpha_2 = (\alpha_1'\varphi)\alpha_2 = \alpha_1'(\varphi\alpha_2) = \alpha_1'(\alpha_2'\varphi) = (\alpha_1'\alpha_2')\varphi,$$

that is, $(\alpha_1\alpha_2)' = \alpha_1'\alpha_2'$, thus ω_φ is an automorphism of H. Suppose that $\omega_\varphi = \omega_\psi$ $(\varphi, \psi \in P)$, that is,

$$\varphi\alpha = \alpha'\varphi \Leftrightarrow \psi\alpha = \alpha'\psi.$$

Let $\varphi\alpha = \alpha'\varphi$ and $\psi\alpha = \alpha'\psi$, then $\alpha' = \psi\alpha\psi^{-1}$ thus $\varphi\alpha = \psi\alpha\psi^{-1}\varphi$, that is $\psi^{-1}\varphi\alpha = \alpha\psi^{-1}\varphi$ $(\alpha \in H)$. Let $O(\bar{F}) > 1$. Let $\alpha \in H$ such that $\alpha(a) = \delta(a, f)$ and $\alpha(\psi^{-1}\varphi(a)) = \delta(\psi^{-1}\varphi(a), g)$ $(a \in A)$, where $\bar{f} \neq \bar{g}$ $(\in \bar{F})$. α exists if $C_\varrho(a) \neq C_\varrho(\psi^{-1}\varphi(a))$. Then

$$\delta(\psi^{-1}\varphi(a), f) = \psi^{-1}\varphi(\delta(a, f)) = \psi^{-1}\varphi\alpha(a) = \alpha\psi^{-1}\varphi(a) = \delta(\psi^{-1}\varphi(a), g),$$

that is $\bar{f} = \bar{g}$, since $\mathbf{A}$ is state-independent. It is a contradiction. Thus $C_\varrho(a) = C_\varrho(\psi^{-1}\varphi(a))$, that is $\psi^{-1}\varphi = \iota$ and $\varphi = \psi$. Therefore the mapping $\varphi \to \omega_\varphi$ is one-to-one. We prove that this mapping is isomorphism. Let $\varphi, \psi \in P$ and $\alpha \in H$ then $\omega_\varphi\omega_\psi(\alpha) = \omega_\varphi(\alpha_1) = \alpha_2$, where $\psi\alpha = \alpha_1\psi$ and $\varphi\alpha_1 = \alpha_2\varphi$. Then

$$(\varphi\psi)\alpha = \varphi(\psi\alpha) = \varphi(\alpha_1\psi) = (\varphi\alpha_1)\psi = (\alpha_2\varphi)\psi = \alpha_2(\varphi\psi),$$

that is $\omega_{\varphi\psi}(\alpha) = \alpha_2$, thus $\omega_\varphi\omega_\psi = \omega_{\varphi\psi}$.

If $O(\bar{F}) = 1$, then $|C_\varrho(c)| = 1$ $(c \in G)$, that is $H = \{\iota\}$. (In this case $G = A$, $E(A) = T$ and $G(A) = P$.)

Let G and G' be two minimal generating systems of a group-type generated quasi-automaton $\mathbf{A}=(A, F, \delta)$. Let T, P and T', P' be sets which are defined in Theorem 3.

Corollary 5. *$T'=\alpha T\alpha^{-1}$, $P'=\alpha P\alpha^{-1}$ where $\alpha\in H$ and $\alpha(G)=G'$.*[4] *Furthermore $T'\cong T$, $P'\cong P$.*

Proof. Let π be a mapping of G into itself and let π' be a mapping of G' into itself such that

$$\alpha(\pi(c)) = \pi'(\alpha(c)) \quad (c\in G) \tag{10}$$

holds, where $\alpha\in H$ and $\alpha(G)=G'$. The mapping $\pi\to\pi'$ is one-to-one, thus the mapping $\varkappa\colon \varphi_\pi\to\varphi_{\pi'}$ is one-to-one also. Let $a\in A$, then

$$\alpha\varphi_\pi(a) = \alpha\varphi_\pi(\delta(c, h)) = \alpha(\delta(\pi(c), h)) = \delta(\alpha(\pi(c)), h) =$$

$$= \delta(\pi'(\alpha(c)), h) = \varphi_{\pi'}(\delta(\alpha(c), h)) = \varphi_{\pi'}\alpha(\delta(c, h)) = \varphi_{\pi'}\alpha(a)$$

$(c\in G, h\in F)$, that is, $\alpha\varphi_\pi=\varphi_{\pi'}\alpha$ thus $\varphi_{\pi'}=\alpha\varphi_\pi\alpha^{-1}$. It can easily be seen, that the mapping $\varkappa$ is onto, that is $T'=\alpha T\alpha^{-1}$.

$$\varkappa(\varphi_{\pi_1}\varphi_{\pi_2}) = \varkappa(\varphi_{\pi_1\pi_2}) = \alpha\varphi_{\pi_1}\varphi_{\pi_2}\alpha^{-1} = \alpha\varphi_{\pi_1}\alpha^{-1}\alpha\varphi_{\pi_2}\alpha^{-1} = \varkappa(\varphi_{\pi_1})\cdot\varkappa(\varphi_{\pi_2})$$

$(\varphi_{\pi_1}, \varphi_{\pi_2}\in T)$ therefore $T\cong T'$. It is evident that $P'=\alpha P\alpha^{-1}$ and $P\cong P'$.

We note, if G is a minimal generating system of a group-type generated quasi-automaton $\mathbf{A}$ and $\alpha\in H$, then $\alpha(G)$ is also a minimal generating system of $\mathbf{A}$. If $\alpha\neq\beta\in H$ then $\alpha(G)\neq\beta(G)$. Furthermore, if G and G' are two minimal generating systems of $\mathbf{A}$, then there exists $\alpha\in H$ such that $\alpha(G)=G'$ holds. Therefore, the cardinality of the set of all minimal generating systems of $\mathbf{A}$ is equal to $O(H)$.

Theorem 4. *If an A-finite group-type quasi-automaton $\mathbf{A}=(A, F, \delta)$ is generated, $|A|=n$ and $|G|=k$ then*

$$O(G(A)) = k!\cdot\left(\frac{n}{k}\right)^k \quad \text{and} \quad O(E(A)) = n^k,$$

where G is a minimal generating system of $\mathbf{A}$.

Proof. If $|A|=n$ and $|G|=k$, where G is a minimal generating system of $\mathbf{A}$, then $O(\bar{F})=\frac{n}{k}$. By Lemmas 2 and 4, $|C_\varrho(c)|=\frac{n}{k}$ $(c\in G)$. Since $\mathbf{C}_\varrho(c)$ is quasi-perfect, therefore $O(E(C_\varrho(c)))=|C_\varrho(c)|=\frac{n}{k}$. The number of sets $C_\varrho(c)$ $(c\in G)$ is equal to k, thus $O(H)=\left(\frac{n}{k}\right)^k$. By Theorem 3, $O(P)$ is equal to the number of the permutations of G, that is $O(P)=k!$. By Theorem 3 and Corollary 3, $O(G(A))=O(P)\cdot O(H)=k!\cdot\left(\frac{n}{k}\right)^k$ and $O(E(A))=O(T)\cdot O(H)=k^k\cdot\left(\frac{n}{k}\right)^k=n^k$.

---

[4] $\alpha(G)=\langle\alpha(c)\,|\,c\in G\rangle$.

Example:

A	1	2	3	4
e	1	2	3	4
f	2	1	4	3

($F=\{e,f\}$ is the Abelian group of degree two, where e is the identity element of F.) Let $abcd$ $(a, b, c, d=1, 2, 3, 4)$ denote the mapping $\varphi\colon A\to A$ such that $\varphi(1)=a$, $\varphi(2)=b$, $\varphi(3)=c$ and $\varphi(4)=d$. It is clear that

$H=\{1234;\ 1243;\ 2134;\ 2143\}$

$T=\{1234;\ 3412;\ 1212;\ 3434\}$

$P=\{1234;\ 3412\}$.

In this example $n=4$ and $k=2$, that is $O(G(A))=2!\cdot 2^2=8$ and $O(E(A))=4^2=16$. But $HT\neq TH=E(A)$, since $|HT|=12$.

We can more easily determine the endomorphisms of a group-type quasi-automaton $\mathbf{A}=(A, F, \delta)$ by means of the following:

Let G be a minimal generating system of $\mathbf{A}'$ (see page 1). Let

$$B_c=\langle b\,|\,b\in A \text{ and } \underset{f\in F}{\exists}\ f[\delta(b,f)=c]\rangle$$

where $c\in G$. It is evident that this is a partition of A. Furthermore, $C_\varrho(c)\subseteqq B_c$ $(c\in G)$.

Lemma 5. *If α is an arbitrary endomorphism of the group-type quasi-automaton $\mathbf{A}=(A, F, \delta)$, then for every $c(\in G)$, there exists a $d(\in G)$ such that $\alpha(B_c)\subseteqq B_d$.*

Proof. Let $\alpha\in E(A)$ and $a\in B_c$ $(c\in G)$, then there exists an $f(\in F)$ such that $\delta(a,f)=$ $=c$, thus $\delta(\alpha(a),f)=\alpha(c)$. It is obvious, that there exists a $d(\in G)$ such that $\alpha(c)\in B_d$. If $h\in F$ such that $\delta(\alpha(c), h)=d$, then $\delta(\alpha(a),fh)=\delta(\alpha(c), h)=d$, that is, $\alpha(a)\in B_d$.

Эндоморфизмы группа — типных квази-автоматов

В этой работе рассматриваем эндоморфизмы группа-типных квази-автоматов (см. Ch. A. Trauth [6]) при помощи системы образующих квази-автоматов.

Пусть $\mathbf{A}=(A, F, \delta)$ произвольный квази-автомат и $A'=\langle\delta(a,f)|a\in A, f\in F\rangle$. В теореме 1 получаем эндоморфизмы квази-автомата $\mathbf{A}$, если знаем эндоморфизмы A-подквази-автомата $\mathbf{A}'$ квази-автомата $\mathbf{A}$. ($\mathbf{A}'$ можно называться *ядром* квази-автомата $\mathbf{A}$.) Если характеристическая полугруппа $\bar{F}=F/\delta_A$ обладает единицей, тогда $\mathbf{A}'$ является порожденным. Теорема 3 доставляем главный результат этой работы, где даваем эндоморфизмы (автоморфизмы) порожденных группа-типных квази-автоматов и структуру полугруппы эндоморфизмов (группы автоморфизмы): Обозначаем множество отображений (7). H и множество отображений (8) T. H является подгруппой группы автоморфизмов $G(A)$. T является подполугруппой полугруппы эндоморфизмов $E(A)$, $E(A)=TH$ и $T\cap H=\{\iota\}$, где ι есть единица полугрупы $E(A)$. Можно найти такую подгруппу P полугруппы T, что $PH=HP=G(A)$ $(P\cap H=\{\iota\})$. В следствии 4 показаем, что если $O(\bar{F})>1$, тогда P изоморфно вкладывается в группу автоморфизмов группы H, и если $O(\bar{F})=1$, тогда $H=\{\iota\}$. В теореме 4 даваем число эндоморфизмов и автоморфизмов A-конечных порожденных группа-типных квази-автоматов: $O(E(A))=n^k$ и $O(G(A))=$ $=k!\cdot\left(\dfrac{n}{k}\right)^k$, где $|A|=n$ и $|G|=k$ (G неприводимая система образующих в квази-автомате $\mathbf{A}$). В следствии 1 показываем, что $O(\bar{F})\,|\,|A|$. Лемма 1 является обобшением теоремы Ю. И. Соркина [7]: Все неприводимые системы образующих квази-автомата являются равномошными. Доказаем, что всякий порожденный группа — типный квази-автомат есть прямая сумма изоморфных полусоверщенных квази-автоматов (лемма 4).

ENTZBRUDER VOCATIONAL SECONDARY SCHOOL
H—9700 SZOMBATHELY, HUNGARY

References

[1] Babcsányi, I., A félperfekt kváziautomatákról (On quasi-perfect quasi-automata), *Mat. Lapok*, v. 21, 1970, pp. 95—102.
[2] Babcsányi, I., Ciklikus állapot-független kváziautomaták (Cyclic state-independent quasi-automata), *Mat. Lapok*, v. 22, 1971, pp. 289—301.
[3] Fleck, A. C., Preservation of structure by certain classes of functions on automata and related group theoretic properties, Computer Laboratory, Michigan State University, 1961, (preprint).
[4] Gécseg, F. & I. Peák, Az automaták algebrai elmélete (Algebraic theory of automata), *Mat. Lapok*, v. 17, 1966, pp. 77—134.
[5] Gécseg, F. & I. Peák, *Algebraic theory of automata*, Budapest, 1972.
[6] Trauth, Ch. A., Group-type automata, *J. Assoc. Comput. Mach.*, v. 13, 1966, pp. 170—175.
[7] Sorkin, I. Yu. (Ю. U. Соркин), Теория определяющих соотношений для автоматов, *Problemy Kibernet.*, v. 9, 1963, pp. 45—69.

(Received June 15, 1974)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

Short-circuited k-trees

By I. PÁVÓ

Abstract

In this paper the author introduces the idea of the short-circuited k-tree arising from k-tree of a graph in each component of which a vertex is given in advance. Necessary and sufficient condition is given for short-circuited k-trees to be circuitless and a procedure to generate is shown, too. During this procedure the generation of k-trees published by the author in an earlyer paper [6] is used as well. The authors methods can be applied in designing of general linear electrical networks by topological formulas.

Introduction

During topological design of linear active networks one has to discover k-trees of the graph of electrical network representing common trees in the network reducated in two different ways [4]. More exactly the task is to discover such k-trees of the graph of network which represent connected circuitless subgraphs either short-circuiting all the nullators [7] and deleting the norators or short-circuiting all the norators [7] and deleting the nullators.

The application of topological formulas is very difficult in this manner and it can be considered as a solved problem only in principle [3].

But there is another possibility to design linear active networks by topological formulas which is more congenial also to computer science. Namely the suitable k-trees can be selected from a set of k-trees which was generated by method [6] based on theorem of Ore [5]. Each component of these k-trees has the common property to contain exactly one of the selected vertices from the graph [6]. k-trees with such property may be advantageously used in design of passive networks as well. The suitable k-trees can be produced from the above set by selecting the graphs being circuitless after fusion of certain pairs of vertices in each k-tree of the set.

In the present paper we are going to deal with "short-circuited" in one or more pairs of vertices k-trees being generated by method [6] from the graph of network in the form $\nu(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$, where G is the graph of network, $\mu(G)$ is its adjacency matrix, μ^{-1} is the inverse of μ, $\mathbf{M}_{i_1,\ldots,i_k}$ is an $(i_1, \ldots, i_k)$-reduction of $\mu(G)$, and ν is the operator removing the direction.

Short-circuited k-tree and its properties

Consider a simple graph G (i.e. an undirected graph without loops and multiple edges) with vertices $P_1, \ldots, P_n$. Let us select arbitrarily k different vertices ($k \leq n$) and construct an $F^k_{i_1, \ldots, i_k}$ k-tree of G (we can use the method [6]), where $i_1, \ldots, i_k$ are indices of the selected vertices. Then choose N pairs of vertices of G pair-wise different at least in one vertex.

Definition. A short-circuited in N pairs of vertices k-tree is a graph arising from $F^k_{i_1, \ldots, i_k}$ by considering each chosen pair of vertices one (or in other words each chosen pairs of vertices is separately short-circuited).

Thinking of the directed graph $\mu^{-1}(\mathbf{M}_{i_1, \ldots, i_k})$ we can supply the short-circuited k-tree with unambiguous direction if we direct the edges of the corresponding short-circuited k-tree in the same way. So we get the directed short-circuited in N pairs of vertices k-tree. Further on we shall speak shortly about directed or undirected short-circuited k-trees.

We mention some properties of a short-circuited k-tree following from its definition:

(i) The number of its vertices is $n-N$, the number of its edges is equal to the number of the edges of the suitable k-tree, and the maximal number of its components is k.

(ii) The components of a short-circuited k-tree may generally contain circuits or loops.

(iii) The circuits of an undirected short-circuited k-tree will not be always directed circuits of the corresponding directed one.

(iv) The vertices of the short-circuited k-tree either correspond to the original ones of the k-tree or they have been arisen by fusion. Vertices arisen by fusion are called multiple vertices, vertices with indices $i_1, \ldots i_k$ are the selected vertices and the remaining ones are the usuall vertices.

(v) To a vertex of the directed short-circuited k-tree more than one edge directed outwards can incident. But such a vertex can be only a multiple one. Outwards directed edges are incident to a multiple vertex if and only if they were arisen by fusion of usuall vertices.

(vi) If there is a circuit (or a loop) in a short-circuited k-tree, then it has at least one multiple vertex. Otherwise passing back to the corresponding k-tree, it also contains circuit (or loop) contradicting the definition of k-tree.

Now we are taking an interest in the necessary and sufficient condition for an undirected short-circuit k-tree to contain circuit (or loop).

(vii) A short-circuited k-tree contains a circuit (or a loop) if and only if different vertices $P_1, \ldots, P_l$ can be choosen from its multiple vertices so that between arbitrary two adjoining vertices of the sequence $P_1, \ldots, P_l, P_1$ there exists either an arc progression or a non-multiple vertex connected by a path progression with its adjoining multiple vertices in the corresponding directed short-circuited k-tree.

To verify (vii) we remark that the vertices $P_1, \ldots, P_l$ in question are in order all the multiple ones of a circuit (or a loop) of the short-circuited k-tree (see vi); it is obvious that two arbitrary adjoining vertices of the sequence $P_1, \ldots, P_l, P_1$ are connected by chain. Vertices of this chain are generally different, not more than the first and the last vertices may be equal. This is the situation when $l=1$, otherwise the chain between two multiple adjoining vertices is a path. Because of the construction of the directed k-tree this is an arc progression in the corresponding short-circuited k-tree, if no selected vertex occurs between the middle vertices. Otherwise only one selected vertex may exist between the middle vertices (see v) which is really connected by path progression with its adjoining multiple vertices.

Cycle check on short-circuited k-trees

In what has gone before we examined the necessary and sufficient condition for a short-circuited k-tree to have circuit. Now we are going to study the following problem:

Construct all the $F^k_{i_1, \ldots, i_k}$ k-trees of a simple graph. Fuse one by one the same N pairs of vertices in each of the k-trees. Among the short-circuited k-trees may be graphs which contain circuit (or loop) but circuitless graphs can also occur. The latters are obviously $(k-N)$-trees. Now the problem is how can we choose the $(k-N)$-trees from the short-circuited k-trees, or, what is an equivalent problem, how to recognize the graphs containing circuit (or loop).

We have met a similar problem in the discussion of the generation of k-trees (see [6]). There we could choose k-trees from the set of the generalized trees by complete cycle check. To solve the present problem we will apply the earlyer procedure. First of all we characterize a short-circuited k-tree by row vector representation.

Definition. Row vector representation of a short-circuited in N pairs of vertices k-tree is called the sequence $(s_1, \ldots, s_n)$ with n members which arises from the row vector representation of the original k-tree after the indices of the fused vertices are marked by a common symbol in order of succession of the N pairs of vertices.

It is obvious that the introduction of common symbols may be performed by choosing of a common symbol to the natural number indices of each pair of vertices one by one. Therefore the common symbol can be considered as an index of the corresponding multiple vertex.

Completed row vector representation of a short-circuited k-tree is called a matrix of size $2 \times n$ the first row of which consists of the natural numbers $1, 2, \ldots, n$, taking the introduced common symbols into account, and its second row is the row vector representation of the short-circuited k-tree.

Observe that from a completed row vector representation we can easily pass to the actual directed short-circuited k-tree. Disregarding the columns of matrix containing zero in their second row, the remaining columns indicate the pair of vertices that are connected in the directed graph in question. Therefore the completed row vector representation is called as the representation of the short-circuited k-tree as well.

We remark that the first row in the representation of a short-circuited k-trees can contain of the natural number $1, 2, \ldots, n$ item symbol-elements, while among the element of the second row can occur the number 0. In the first row the number $1, 2, \ldots, n$ can occur no more that one time, while the symbol elements can do it several times, too. Finally notice that the number of the same symbol elements shows the number of the edges being incident with the corresponding multiple vertex and that are directed outwards.

Definition. A function $\varphi(x)$ is called the function associated with a completed row vector representation the domain of which is the set of the elements of the first row, its range is the set of the elements of the second row and the correspondence is defined by the column of the representation in question. Generally $\varphi(x)$ is a function of multivalue.

Remember that there was a "function $\varphi(x)$" in the previous paper [6] as well. The idea of the cycle check was just constructed by application of that function. Presently we introduce the function $\varphi(x)$ with similar design.

Definition. Let be $\varphi(x)$ the associated function with the completed row vector representation

$$\mathbf{R} = \begin{pmatrix} r_1 \dots r_n \\ s_1 \dots s_n \end{pmatrix}.$$

By a cycle check performed on **R** starting with the element r_i we mean the construction of the sequence

$$r_i,\ \varphi(r_i),\ \varphi(\varphi(r_i)),\ \dots\ (1 \leqq i \leqq n).$$

We say that the outcome of the cycle check is finite if we can construct only a finite sequence, i.e. if somewhere in the sequence a zero turns up, which does not belong to the domain φ; otherwise we say that the outcome of the cycle check is infinite.

It the outcome of the cycle check is infinite then, as it can be easily seen, from a certain point the same segment of the above sequence will occur repeatedly.

A fundamental difference appears between "the present cycle check" and the earlier one published in the paper [6]. The cycle check performed on **R** starting at r_i can not be unambiguous therefore the outcome can be several. Such a cycle check can arise if among the members of the sequence performed to the cycle check a symbol element does occur for the function φ is generally of multivalue on symbol elements.

Moreover the construction of the sequence $r_i, \varphi(r_i), \varphi(\varphi(r_i)), \dots$ can also be regarded as walking through a part of the graph, starting at a vertex P_{r_i} of the directed short-circuited k-tree and always proceeding in conformity with the direction of the edges passed along. It goes without saying that if we start cycle checks with each r_i than we walk the whole graph (generally several times) and in case of infinite outcome we get into a directed circuit (or loop) during the walk. But to find out the existence of a circuit (or a loop) it is sufficient if we start cycle checks only with symbol elements namely because of (vi) a circuit (or a loop) always has multiple vertex and symbol elements are just the indices of the multiple vertices.

Definition. By a complete cycle check performed on a completed row vector representation we mean a bunch of all possible cycle checks starting with all symbol elements of the first row of the representation in question. The outcome of a complete cycle check is said to be finite if all checks constituting it have finite outcomes; otherwise, the outcome is said to be infinite.

Because of "the graph theory backgroung" of the cycle check later we shall say that the cycle check is performed on the short-circuited k-tree, in particular starting at its P_{r_i} vertex, by which we mean that the cycle check is performed on the representation starting at the symbol r_i. The idea of the complete cycle check performed on a short-circuited k-tree will be used in similar meaning as well.

Short-circuited k-trees without circuit (or loop)

Consider a short-circuited in N pairs of vertices k-tree with $P_{a_1}, \ldots, P_{a_N}$ multiple vertices. Let be $i_1, \ldots, i_k$ the indices of the selected vertices in the corresponding k-tree. Suppose that the complete cycle check performed on the short-circuited k-tree has a finite outcome. It means that all possible cycle checks starting at all multiple vertices come to an end somewhere. Tabulate about the finish of each possible cycle check, that is put down in order those indices of the columns of the completed row vector representation in which the cycle checks have finished together with indicating the vertices where checks were started from. It is obviously enough if the table contains only the indices of the vertices in question. So we get the table:

a_1	$\ldots$	a_j	$\ldots$	a_N
$i_1^1, \ldots, i_{m_1}^1$	$\ldots$	$i_1^j, \ldots, i_{m_j}^j$	$\ldots$	$i_1^N, \ldots, i_{m_N}^N$

The meaning of the above table is the following: Cycle checks starting at P_{a_j} can be performed exactly of number m_j wich finish at vertices $P^j_{i_1}, \ldots, P^j_{i_{m_j}}$ in order, where $j=1, \ldots, N$. Naturally among the numbers $i_1^j, \ldots, i_{m_j}^j$ equal elements may occur as well. Notice that the obtained elements $i_1^j, \ldots, i_{m_j}^j$ where the cycle checks finished can be considered as indices of selected vertices. Namely if the superior element of the column of the representation where the cycle check was finished is not a symbol element then during the cycle check the last touched vertex is just a selected one. If the superior element was a symbol one then the corresponding vertex had arisen with short-circuiting of some selected vertex of the original k-tree.

Definition. A graph with vertices $P_{a_j}, P^j_{i_1}, \ldots, P^j_{i_{m_j}}$, and with edges $(P_{a_j}, P^j_{i_1}), \ldots$ $\ldots, (P_{a_j}, P^j_{i_{m_j}})$ $(j=1, \ldots, N)$ is called the reduced graph of the short-circuited k-tree. It is always undirected.

Notice that the idea of the reduced graph is defined only in that case if the outcome of the complete cycle check is finite. Otherwise the reduced graph is generally more simple as the original one which was reduced and it is a bipartite graph [1].

Theorem. A short-circuited k-tree is without a circuit (or a loop) if and only if the complete cycle check performed on it has a finite outcome and its reduced graph is circuitless.

Proof. To verify the sufficient condition assume that the complete cycle check performed on the short-circuited k-tree has a finite outcome, the reduced graph is circuitless nevertheless the short-circuited k-tree contains circuit (or loop). Then the corresponding subgraph of this circuit (or loop) cannot be a directed one in the directed short-circuited k-tree. Let be all the multiple vertices $P_{a_1}, \ldots, P_{a_l}$ in order that are incident with the circuit (or loop). According to (vii) cycle checks starting at adjoining multiple vertices finish at a place of common column index in the corresponding representation where P_{a_l} is adjoining P_{a_1}, too. Let be P_{i_j} the selected vertex defined by the common column index belonging to the cycle checks starting at P_{a_j} and $P_{a_{j+1}}$ $(j=1, \ldots, l;\ l+1=1)$. So edges $(P_{a_1}, P_{i_1}), \ldots, (P_{a_l}, P_{i_l})$ determine a circuit in the reduced graph contradicting to the starting assumption.

The condition is necessary, too. Namely if the complete cycle check performed on a short-circuited k-tree has an infinite outcome then the corresponding directed short-circuited k-tree contains a circuit (or a loop) and for the same reason so does the

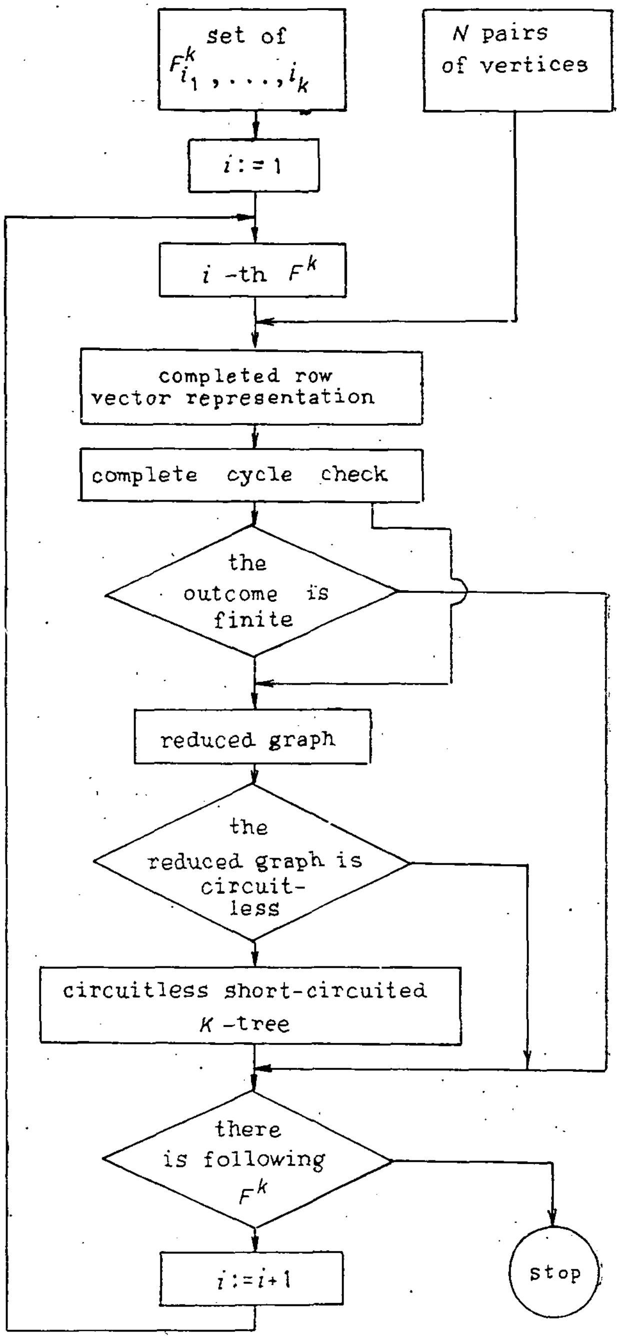

set of
$F^k_{i_1,\ldots,i_k}$
N pairs of vertices
i:=1
i -th F^k
completed row vector representation
complete cycle check
the outcome is finite
reduced graph
the reduced graph is circuit-less
circuitless short-circuited K -tree
there is following F^k
i:=i+1
stop

Fig. 1

undirected one, too. Hereupon let us assume that the complete cycle check has a finite outcome and the reduced graph contains a circuit with multiple vertices $P_{a_1}, \ldots$ $\ldots, P_{a_l}$. For the reduced graph is bipartite between all adjoining multiple vertices mentioned above there is exactly one selected vertex (P_{a_l} is adjoining P_{a_1}!). Let be P_{i_j} between P_{a_j} and $P_{a_{j+1}}$, where $j=1, \ldots, l$ and $l+1=1$. But then a directed path leads either from P_{a_j} to $P_{a_{j+1}}$, or from P_{a_j} and $P_{a_{j+1}}$ to the selected vertex P_{i_j}. Therefore all adjoining multiple vertices are connected in the short-circuited k-tree so it contains a circuit (or a loop). The proof is complete.

In the sequel we are going to construct a procedure that selects short-circuited in N pairs of vertices k-trees which have no circuit (or loop) from among graphs $F^k_{i_1, \ldots, i_k}$:

Step 1. Consider the set of all k-trees $F^k_{i_1, \ldots, i_k}$ where $i_1, \ldots, i_k$ are the indices of the selected vertices given in advance. From the row vector representation of such k-trees and from the pairs of vertices given by conditions of short-circuiting the complete row vector representation of short-circuited k-trees can be constructed. This means the introduction of symbol elements which are the indices of multiple vertices.

Step 2. A complete cycle check is performed on each representation constructed at the above step. If it has a finite outcome so there exists the reduced graph of the corresponding short-circuited k-tree.

Step 3. In the end we control whether the reduced graph is circuitless. In the circuitless case the corresponding short-circuited k-tree will not have a circuit (or a loop) according to the theorem.

One can look over the whole procedure by studying the block diagram on figure 1. We notice that there is a task in the 3-rd part of the procedure to find out whether a graph is without circuit. It may happen in several different ways. In a simple case it is possible by drawing the reduced graph. Further on it can be found out from the incidence matrix of the graph [1]. As for the reduced graph is bipartite its circuit contains only edges of even numbers and we can search for all possible circuits from the table defining the graph after all. To construct such a discussion can easily be made because it consists of steps of finite numbers.

Application

Example 1. Let be given the row vector representation of a 2-tree generated by method [6]:

$$(202527808).$$

It is obvious from the method [6] that this 2-tree has 9 vertices and its selected vertices are P_2 and P_8. We can easily draw the directed 2-tree and it is illustrated on the figure 2.

First short-circuit the pairs of vertices P_2, P_6 and P_4, P_8. Let be defined the symbol elements by equations $2=6=a$ and $4=8=b$. So we get the following completed row vektor representation of the arisen short-circuited 2-tree:

$$\mathbf{R} = \begin{pmatrix} 1 & a & 3 & b & 5 & a & 7 & b & 9 \\ a & 0 & a & 5 & a & 7 & b & 0 & b \end{pmatrix}.$$

The corresponding short-circuited 2-tree has circuit because of performing a cycle check on **R** starting at the element "a" being in the 6-th column of **R** we get the sequence

$$a, 7, b, 5, a, \dots$$

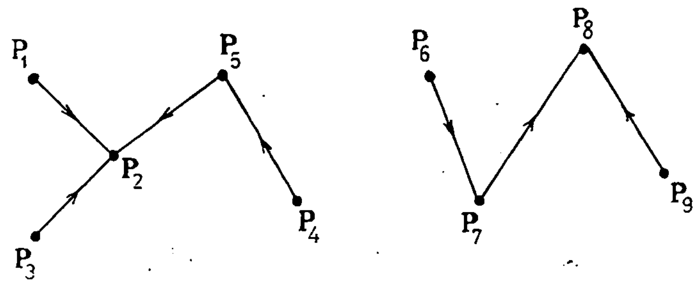


Fig. 2

that is the outcome of the cycle check is infinite. Now the reduced graph does not exist according to its definition. The short-circuited 2-tree can be seen in the figure 3. Notice that the cycle check of infinite outcome means "walking alone the only directed circuit of the graph".

For the second short-circuit the pairs of vertices P_1, P_4 and P_4, P_7 of the 2-tree. The common symbol element is defined by equation $1=4=7=a$, and the completed row vektor representation of the short-circuited 2-tree is:

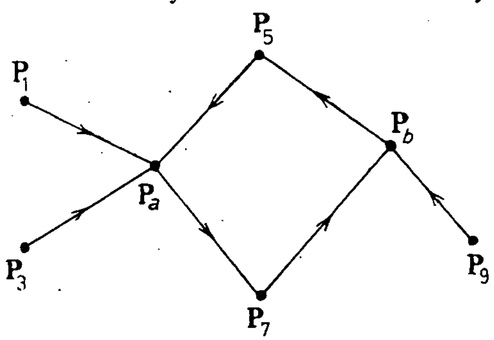


Fig. 3

$$\mathbf{R} = \begin{pmatrix} a & 2 & 3 & a & 5 & 6 & a & 8 & 9 \\ 2 & 0 & 2 & 5 & 2 & a & 8 & 0 & 8 \end{pmatrix}.$$

The complete cycle check performing on **R** has finite outcome because of the following possible cycle checks:

$$a,\ 2,\ 0,$$
$$a,\ 5,\ 2,\ 0,$$
$$\text{and}\quad a,\ 8,\ 0.$$

The table of the reduced graph found out from the sequences of cycle checks is the following:

$$\frac{a}{2,\ 2,\ 8}.$$

The edges of the reduced graph are (P_a, P_2); (P_a, P_2) and (P_a, P_8), therefore this graph contains a circuit consisting of two edges. Either the short-circuited graph or its reduced graph are drawn on the figure 4. Notice that there is not a (directed) circuit in the directed graph but it exists in the undirected one. Otherwise this fact turned out from the complete cycle check of finite outcome as well. The present example shows that the complete cycle check of finite outcome is not a sufficient condition for the short-circuited k-tree to be circuitless.

For the third short-circuit the pairs of vertices P_5, P_6 in the considered 2-tree. Let be defined the symbol element "a" by $5=6=a$. So the representation of the arosen short-circuited 2-tree is:

$$\mathbf{R} = \begin{pmatrix} 1 & 2 & 3 & 4 & a & a & 7 & 8 & 9 \\ 2 & 0 & 2 & a & 2 & 7 & 8 & 0 & 8 \end{pmatrix}.$$

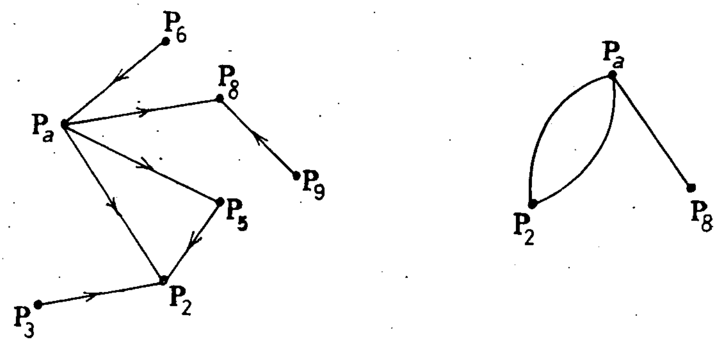


Fig. 4

The steps of the complete cycle check performed on **R** are:

$$a,\ 2,\ 0,$$
$$\text{and}\quad a,\ 7,\ 8,\ 0,$$

so the outcome of the complete cycle check is finite. Because of the reduced graph defined by the table

$$\frac{a}{2,\ 8}$$

is circuitless so the short-circuited 2-tree is too. We can show either the short-circuited 2-tree or its reduced graph on the figure 5.

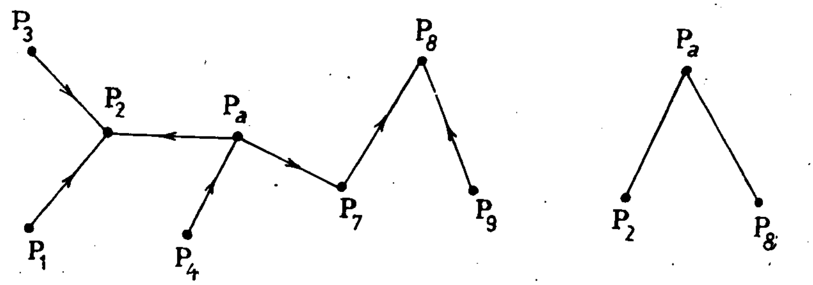


Fig. 5

Example 2. Consider the graph with 6 vertices marked by natural numbers on the figure 6. This is a network graph of a feedback operational amplifier with multiple loops which plays central part in the theory of the linear active electrical networks [2]. The network graph in question is a so called "nullator-norator equiv-

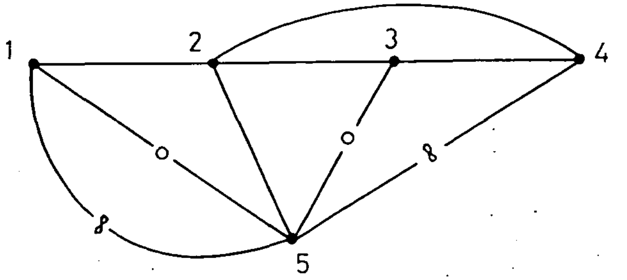


Fig. 6

alent network" [3]. The edges drawn in usual way mean the present of passive element while the symbol "—o—" on the edges refers to a nullator, and the "— 8 —" to a norator. Search all the common 3-trees that become connected circuitless subgraphs after short-circuiting either all the norators or all the nullators while passing over at first all nullators for the second all norators from the graph of network.

First we produce those trees of the graph which contain all the norators and do not contain any of the nullators. Such trees of the graph can be easily produced by method [6]. If we pass over all the norators from the subgraph mentioned above we really get all the 3-trees of the graph which become connected circuitless subgraphs after short-circuiting of endpoints of the norators. After passing over all the norators we get the following result:

(01200), (01400), (03400), (04200), (04400), (05200), (05400).

In the 2-nd step short-circuit the endpoints of all nullators in each of the 3-trees listed above. This means the short-circuiting of pairs of vertices 1,5 and 3,5. From the short-circuited 3-trees arisen by fusion of vertices 1,3 and 5 the circuitless graphs can be selected by method shown in the present paper.

So the only symbol element "a" is defined by equation $1=5=3=a$. The 7 completed row vector representation can be written as following:
the first common row of the representation are

$$a \quad 2 \quad a \quad 4 \quad a$$

while the second rows are in order

$$\begin{matrix} 0 & a & 2 & 0 & 0 \\ 0 & a & 4 & 0 & 0 \\ 0 & a & 4 & 0 & 0 \\ 0 & 4 & 2 & 0 & 0 \\ 0 & 4 & 4 & 0 & 0 \\ 0 & a & 2 & 0 & 0 \end{matrix}$$

and $0 \quad a \quad 4 \quad 0 \quad 0.$

Complete cycle check performed on each of representation in question we get an infinite outcome only in the 1-st and in the 6-th cases. In the remaining cases the reduced graph is a common one defined by table

$$\frac{a}{1,\ 4,\ 5},$$

and it is obviously circuitless.

We obtain the following 3-trees fulfilling the conditions of the present example:
(01400), (03400), (04200), (04400), (05400).

References

RESEARCH GROUP ON MATHEMATICAL LOGIC
AND THEORY OF AUTOMATA OF THE
HUNGARIAN ACADEMY OF SCIENCES
H—6720 SZEGED, HUNGARY
SOMOGYI U. 7.

References

[1] BUSACKER, R. G. & T. L. SAATY, *Finite graphs and networks*, McGraw—Hill Book Comp., New York, 1967.
[2] CHIRLIAN, P., *Integrated and active network analysis*, McGraw—Hill Book Comp., New York, 1969.
[3] DAVIES, A. C., Norator-nullator equivalent networks for controlled sources, *Proc. IEEE*, v. 55, 1967. p. 772.
[4] DAVIES, A. C., Matrix analysis of network containing nullors, *Electronics Letters*, 1966, No. 2, p. 48, No. 3. p. 91.
[5] ORE, O., Theory of graphs, *Amer. Math. Soc., Transl.* Providence, 1962, pp. 68—74.
[6] PÁVÓ, I., Generation of the k-trees of a graph, *Acta Cybernet.*, v. 1, 1971, pp. 57—68.
[7] TELLEGEN, B. D. H., On nullators and norators, *IEEE Trans. Circuit Theory*, v. CT—13, 1966, pp. 466.

(Received Sept. 5, 1974)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

On superpositions of automata

By P. Dömösi

We say that an automaton **A** *realises* an automaton **B** if **B** can be given as an A-homomorphic image of an A-subautomaton of **A**. If there exists a one-to-one homomorphism having the above property then it is said that **B** *can be embetted A-isomorphically* into **A**.

Let **A** be a finite automaton and denote by $C(\mathbf{A})$ the class of all finite superpositions of automata having fewer states than **A**. For any natural number l, let $C_l(\mathbf{A})$ be the class of all automata from $C(\mathbf{A})$ whose factors have not more states than l.

For any finite automaton **A** and natural number l one can raise the following questions:

(a) Whether there exists an $\mathbf{A}_1 \in C_l(\mathbf{A})$ such that $\mathbf{A}_1$ is A-isomorphic to **A**.

(b) Whether **A** can be embetted A-isomorphically into a superposition from $C_l(\mathbf{A})$.

(c) Whether **A** can be realized by an automaton in $C_l(\mathbf{A})$.

Using results published by M. Yoeli [6], we can solve (a). Moreover, by specializing Theorem 4.3.2. stated by F. Gécseg [2], problem (b) can also be solved. In both cases we can give an effective procedure.

In this paper, using a result mentioned by F. Gécseg and some results achieved by R. J. Nelson [5] and H. P. Zeiger [8], we present an algorithm to decide for any automaton **A** whether it can be realized by an automaton **B** from $C(\mathbf{A})$. Moreover, if such **B** exists then it can be given by a procedure presented in this paper.

Before studying these questions, we introduce some notions and notations. In the sequel by an automaton we always mean a finite automaton.

Take two automata $\mathbf{A}_1 = \mathbf{A}_1(X_1, A_1, Y_1, \delta_1, \lambda_1)$ and $\mathbf{A}_2 = \mathbf{A}_2(X_2, A_2, Y_2, \delta_2, \lambda_2)$ with $Y_1 \subseteq X_2$. It is said that the automaton $\mathbf{A} = \mathbf{A}(X, A, Y, \delta, \lambda)$ with $X = X_1$, $A = A_1 \times A_2$ and $Y = Y_2$ is the *superposition* of $\mathbf{A}_1$ by $\mathbf{A}_2$ (in notation: $\mathbf{A} = \mathbf{A}_1 * \mathbf{A}_2$) if for any $x \in X$ and $(a_1, a_2) \in A$,

$$\delta((a_1, a_2), x) = (\delta_1(a_1, x), \delta_2(a_2, \lambda_1(a_1, x)))$$

and

$$\lambda((a_1, a_2), x) = \lambda_2(a_2, \lambda_1(a_1, x))$$

hold.

The concept of superposition can be generalized in a natural way for any finite system of automata $\mathbf{A}_i = \mathbf{A}_i(X_i, A_i, Y_i, \delta_i, \lambda_i)$ $(i = 1, 2, \ldots, n)$ with $Y_j \subseteq X_{j+1}$ $(j = 1, 2, \ldots, n-1)$.

Let k be a natural number and $\mathbf{A}=\mathbf{A}(X, A, Y, \delta, \lambda)$ be an automaton. Then by $\mathbf{A}^k$ we mean the automaton $\mathbf{B}=\mathbf{B}(X, B, Y', \delta', \lambda')$ with

$$B = \underbrace{A\times A\times \ldots \times A}_{k\text{-times}} \text{ and } Y' = \underbrace{Y\times Y\times \ldots \times Y}_{k-\text{times}}$$

such that for any $x\in X$ and $(a_1, a_2, \ldots, a_k)\in B$, we have

$$\delta'\big((a_1, \ldots, a_k), x\big) = \big(\delta(a_1, x), \ldots, \delta(a_k, x)\big)$$

and

$$\lambda'\big((a_1, \ldots, a_k), x\big) = \big(\lambda(a_1, x), \ldots, \lambda(a_k, x)\big).$$

Let $\mathbf{A}_i=\mathbf{A}_i(X, A_i, Y, \delta_i, \lambda_i)$ $(i=1, \ldots, n)$ be a system of automata such that for any $i, j\in\langle 1, \ldots, n\rangle$, $A_i\cap A_j=\emptyset$ if $i\neq j$. Then the automaton $\mathbf{A}=\mathbf{A}(X, A, Y, \delta, \lambda)$ is called the *direct sum* of $\mathbf{A}_i (i=1, \ldots, n)$ if $A=\bigcup_{i=1}^{k} A_i$ and for any $x\in X$ and $a\in A$,

$$\delta(a, x) = \delta_i(a, x) \quad (a\in A_i)$$

and

$$\lambda(a, x) = \lambda_i(a, x) \quad (a\in A_i)$$

hold.

Take an arbitrary automaton $\mathbf{A}=\mathbf{A}(X, A, Y, \delta, \lambda)$. An $x\in X$ is called *reset signal* if there exists an $a\in A$ such that $\delta(b, x)=a$ for any $b\in A$. We say that this a belongs to x. An input signal $x\in X$ is said to be *permutation signal* if $\eta_x\colon a\to\delta(a, x)$ $(a\in A)$ is a permutation of A. Generally, for an automaton $\mathbf{A}$ with input set X, X_R denotes the set of all reset signals and X_P is the set of all permutation signals. An automaton $\mathbf{A}=\mathbf{A}(X, A, Y, \delta, \lambda)$ is *reset, permutation* and *permutation-reset* automaton if respectively $X=X_R$, $X=X_P$ and $X=X_R\cup X_P$.

For any set H let $F(H)$ denote the free semigroup freely generated H. Furthermore, let ap be the last letter in the word $\delta(a, p)$ $\big(a\in A, p\in F(x)\big)$. Let $\mathbf{A}$ be an automaton and B a subset of the state set A of $\mathbf{A}$. Then for any input word p, we set $B^p = \langle c|c=bp|b\in B\rangle$. Moreover we say that a system $\Gamma=\langle B_1, \ldots, B_n\rangle$ of subsets of A is *cover* of $\mathbf{A}$ if $\bigcup_{i=1}^{n} B_i=A$, $B_i\neq B_j$ implies $i\neq j$ and for any $B_i\in\Gamma$ and $x\in X$ there exists a $B_j\in\Gamma$ such that $B_i^x\subseteqq B_j$. For any $B_i\in\Gamma$ take a 1—1 mapping Φ_{B_i} of $\langle 1, 2, \ldots, \bar{\bar{B}}_i\rangle$ onto B_i. We say that a pair $(\mathbf{A}_1, \mathbf{A}_2)$ of automata is an *SR*-system of $\mathbf{A}$ belonging to Γ if the following conditions are satisfied:

$$\mathbf{A}_1 = \mathbf{A}_1(X, \Gamma, \Gamma\times X, \delta_1, \lambda_1), \quad \mathbf{A}_2 = \mathbf{A}_2(\Gamma\times X, \langle 1, \ldots, l\rangle, Y, \delta_2, \lambda_2),$$

where $l=\max_{B_i\in\Gamma} \bar{\bar{B}}_i$; furthermore, for any $x\in X$, $B_i\in\Gamma$ and $k\in\langle 1, \ldots, l\rangle$,

$$B_i^x \subseteqq \delta_1(B_i, x),$$

$$\lambda_1(B_i, x) = (B_i, x),$$

$$\delta_2\big(k, (B_i, x)\big) = \begin{cases} \Phi^{-1}_{\delta_1(B_i, x)}\big(\delta(\Phi_{B_i}(k), x)\big) & \text{if } k \leqq \bar{\bar{B}}_i, \\ \text{arbitrary } m\in\langle 1, 2, \ldots, l\rangle\text{-otherwise,} \end{cases}$$

$$\lambda_2\big(k, (B_i, x)\big) = \begin{cases} \lambda\big(\Phi_{B_i}(k), x\big) & \text{if } k \leqq \bar{\bar{B}}_i, \\ \text{arbitrary } y\in Y\text{-otherwise.} \end{cases}$$

It has been proved (see [5]) that for any such pair $\mathbf{A}_1, \mathbf{A}_2$ the superposition $\mathbf{A}_1 * \mathbf{A}_2$ realises $\mathbf{A}$.

A system $(\mathbf{A}_1, \dots, \mathbf{A}_n)$ of automata is called an *SR-system of* **A** *with rank* k if $A_1 * \dots * \mathbf{A}_n$ realizes $\mathbf{A}$, at least one $\mathbf{A}_i$ $(1 \leqq i \leqq n)$ has k states and none of $\mathbf{A}_1, \dots, \mathbf{A}_n$ has more than k states.

Finally, it is said that **A** *can be mapped MA-homomorphically (MA-isomorphically) onto* **B** if the automaton without output belonging to **A** can be mapped A-homomorphically (A-isomorphically) onto the automaton without output belonging to **B**.

Now we are ready to present our algorithm.

Let $\mathbf{A} = \mathbf{A}(X, A, Y, \delta, \lambda)$ be an arbitrary automaton. We shall investigate whether **A** has an SR-system of rank less than $\bar{\bar{A}}$.

We distinguish the following cases:

(I) If $\bar{\bar{A}} \leqq 2$ then **A** has no SR-system of rank less than $\bar{\bar{A}}$.

(II) Let $X = X_R$ and $\bar{\bar{A}} > 2$. Then every system $\Gamma^{(2)} = \langle B_1^{(2)}, B_2^{(2)} \rangle$ with $B_1^{(2)} \cup B_2^{(2)} = A$ and $1 \leqq \bar{\bar{B}}_1^{(2)}, \bar{\bar{B}}_2^{(2)} < \bar{\bar{A}}$ is a cover of **A**. Giving an SR-system $(\mathbf{A}_1^{(2)}, \mathbf{A}_2^{(2)})$ of **A** belonging to Γ, we get the desired construction.

(III) Let $X = X_p$, $\bar{\bar{A}} > 2$ and assume that **A** can be given as a direct sum of two automata **B** with state set $B = \langle b_1, \dots, b_n \rangle$ and **C** with state set C such that $\bar{\bar{B}} \leqq \bar{\bar{C}}$. In this case $\Gamma^{(3)} = \langle \langle b_1 \rangle, \langle b_2 \rangle, \dots, \langle b_n \rangle, C \rangle$ is a cover of **A**. Therefore, since $\bar{\bar{B}} \leqq \bar{\bar{C}}$ and $\bar{\bar{A}} > 2$ thus every SR-system $(\mathbf{A}_1^{(3)}, \mathbf{A}_2^{(3)})$ of **A** belonging to $\Gamma^{(3)}$ is suitable for our purpose.

(IV) Assume that $X = X_p$, $\bar{\bar{A}} > 2$ and **A** cannot be given as a direct sum of any two automata. Consider all proper subsets C_j of A having at least two elements and for any C_j give a cover $\Gamma_j = \langle C_j^p \mid p \in F(X) \rangle$. For any such Γ_j, let us consider an SR-system $(\mathbf{B}_j, \mathbf{A}_j)$ of **A** belonging to Γ_j. If one of these SR-systems has rank less than $\bar{\bar{A}}$ then it is a suitable SR-system of **A**. If none of them has rank less than $\bar{\bar{A}}$ then take all pairs $(\mathbf{B}_j, \mathbf{A}_j)$ such that the number of states of $\mathbf{B}_j * \mathbf{A}_j$ is less than $\bar{\bar{A}}!$. (In this case this is only a formal requirement since the number of states of any $\mathbf{B}_j * \mathbf{A}_j$ is less than $\bar{\bar{A}}!$). For any subset C_{ij} of the state set of such $\mathbf{B}_j$ having at least two elements, let us construct a cover $\Gamma_{ij} = \langle C_{ij}^p \mid p \in F(X) \rangle$ of $\mathbf{B}_j$ and an SR-system $(\mathbf{B}_{ij}, \mathbf{A}_{ij})$ belonging to this cover. If one of these triples $(\mathbf{B}_{ij}, \mathbf{A}_{ij}, \mathbf{A}_j)$ is of rank less than $\bar{\bar{A}}$ then we get a desired SR-system of **A**. If there exists no such system let us consider all systems $(\mathbf{B}_{ij}, \mathbf{A}_{ij}, \mathbf{A}_j)$ for which the number of states of $\mathbf{B}_{ij} * \mathbf{A}_{ij} * \mathbf{A}_j$ is less than $\bar{\bar{A}}!$. Now repeating the above process, we get the following cases:

(IV. A) We get an SR-system $(\mathbf{A}_1^{(4)}, \dots, \mathbf{A}_n^{(4)})$ of **A** with rank less than $\bar{\bar{A}}$.

(IV. B) For all sequences $(\mathbf{B}, \mathbf{A}_1, \dots, \mathbf{A}_n)$, $\bar{\bar{B}} \geqq \bar{\bar{A}}$ and the number of states of $\mathbf{B} * \mathbf{A}_1 * \dots * \mathbf{A}_n$ is not less than $\bar{\bar{A}}!$. In this case **A** cannot be realized by a superposition of automata having fewer states than **A**.

(V) Assume that $X = X_R \cup X_P$, $X_R \neq \emptyset$, $X_P \neq \emptyset$ and $\bar{\bar{A}} > 2$. If the X-subautomaton of **A** having input set X_P can be given as a direct sum then let us apply to this X-subautomaton the procedure presented in (III); in the opposite case let us apply to it the procedure given in (IV). In case (IV. B) the automaton **A** cannot be realized by a superposition of automata having fewer states than **A**. If we get (IV. A) then one can apply (III) or, using (VII), we get a desired SR-system $(\mathbf{A}_1^{(5)}, \dots, \mathbf{A}_n^{(5)})$ of **A**.

(VI) Let $X \setminus (X_R \cup X_P) \neq \emptyset$, $\bar{\bar{A}} > 2$ and consider the construction given by H. P. Zeiger in [8]: For any $x \in X \setminus X_P$, let $a(x)$ denote the state of **A** such that $\delta(a', x) \neq$

$\neq a(x)$ where $a' \in A$ is arbitrary. Consider the cover $\Gamma^{(6)} = \langle B | B = A \setminus \langle a \rangle, a \in A \rangle$ and take the automaton $\mathbf{A}_1^{(6)} = \mathbf{A}_1^{(6)}(X, \Gamma^{(6)}, \Gamma^{(6)} \times X, \delta_1, \lambda_1)$ such that for any $x \in X$ and $B \in \Gamma^{(6)}$,

$$\delta_1(B, x) = \begin{cases} B^x & \text{if} \quad x \in X_P, \\ A \setminus \langle a(x) \rangle\text{-otherwise}, \end{cases}$$

and

$$\lambda_1(B, x) = (B, x).$$

Now choosing a suitable automaton $\mathbf{A}_2^{(6)}$, we get an SR-system $(\mathbf{A}_1^{(6)}, \mathbf{A}_2^{(6)})$ of $\mathbf{A}$ such that the number of states of $\mathbf{A}_2^{(6)}$ is less than $\mathbf{A}$, $\mathbf{A}_1^{(6)}$ is permutation-reset; moreover, if $X_P \neq \emptyset$ then the X-subautomata of $\mathbf{A}$ and $\mathbf{A}_1^{(6)}$ having input set X_P are A-isomorphic (see [5]).

Thus we get the following subprocedures.

(VI. A) If $A_1^{(6)}$ is a reset automaton then apply (II) to it. In this case $(\mathbf{A}_1^{(2)}, \mathbf{A}_2^{(2)}, \mathbf{A}_2^{(6)})$ is a required system.

(VI. B) If $\mathbf{A}_1^{(6)}$ has a permutation signal then apply (V) to it. If $\mathbf{A}_1^{(6)}$ has no SR-system with rank less than $\bar{A}$ then neither has $\mathbf{A}$. In the opposite case $(\mathbf{A}_1^{(5)}, \mathbf{A}_2^{(5)}, \ldots, \mathbf{A}_n^{(5)}, \mathbf{A}_2^{(6)})$ is an SR-system of $\mathbf{A}$ with rank less than $\bar{A}$.

(VII) Assume that $X \setminus X_R \neq \emptyset$, $X_R \neq \emptyset$ and the superposition $\mathbf{A}_1 * \mathbf{A}_2 * \ldots * \mathbf{A}_n$ of the automata $\mathbf{A}_i = A_i(X_i, A_i, Y_i, \delta_i, \lambda_i)$ $(\bar{A}_i < \bar{A};\, i = 1, \ldots, n)$ realises the X-subautomaton $\mathbf{B}$ with input set $X \setminus X_R$ of the automaton $\mathbf{A}$. Let ψ be an A-homomorphism of an A-subautomaton of the superposition $\mathbf{A}_1 * \mathbf{A}_2 * \ldots * \mathbf{A}_n$ onto $\mathbf{B}$. For any $x \in X_R$ take an element $(a_1(x), \ldots, a_n(x))$ of $A_1 \times A_2 \times \ldots \times A_n$ such that $\psi((a_1(x), \ldots, a_n(x)))$ is an element of A belonging to x. Construct the automaton $\mathbf{A}_i^{(7)} = \mathbf{A}_i^{(7)}(X_i', A_i, Y_i', \delta_i', \lambda_i')$ $(i = 1, \ldots, n)$ with $X_1' = X$ and $Y_n' = Y$ such that for any $j (= 2, \ldots, n)$ and $k (= 1, \ldots, n-1)$, $X_j' = A_1 \times A_2 \times \ldots \times A_{j-1} \times X$ and $Y_k' = A_1 \times \ldots \times A_k \times X$; furthermore, for any $i (= 1, \ldots, n)$, $x_i \in X_i'$, and $a_i \in A_i$,

$$\delta_i'(a_i, x_i) =$$

$$= \begin{cases} \delta_i(a_i, x_i) & \text{if} \quad i = 1 \text{ and } x_i \notin X_R, \\ a_i(x_i) & \text{if} \quad i = 1 \text{ and } x_i \in X_R, \\ \delta_i(a_i, \lambda_{i-1}(a_{i-1}, \ldots, \lambda_1(a_1, x), \ldots)) & \text{if} \quad i > 1,\ x_i = (a_1, a_2, \ldots, a_{i-1}, x),\ x \notin X_R, \\ a_i(x) & \text{if} \quad i > 1,\ x_i = (a_1, a_2, \ldots, a_{i-1}, x),\ x \in X_R, \end{cases}$$

$$\lambda_i'(a_i, x_i) =$$

$$= \begin{cases} (a_i, x_i) & \text{if} \quad i = 1, \\ (a_1, a_2, \ldots, a_i, x) & \text{if} \quad 1 < i < n \text{ and } x_i = (a_1, \ldots, a_{i-1}, x), \\ \lambda(\psi(a_1, a_2, \ldots, a_n), x) & \text{if} \quad i = n,\ x_i = (a_1, \ldots, a_{n-1}, x) \text{ and } \psi((a_1, a_2, \ldots, a_n)) \text{ is} \\ & \qquad \text{defined, arbitrary } y \in Y\text{-otherwise.} \end{cases}$$

The system $(\mathbf{A}_1^{(7)}, \ldots, \mathbf{A}_n^{(7)})$ given above is an SR-system of $\mathbf{A}$ with rank less than $\bar{A}$.

We now show that the process given above is right. Superpositions of automata with one-element state sets have one-element state sets, too. Moreover, the state set is never void. Therefore (I) is obviously valid.

It can be seen directly from the definition that (II) and (III) are valid.

After proving (IV) and (VII), the validity of (V) follows obviously, and (VI) is valid by the results published in [8].

In order to deal with the construction given in (VII) take the partial mapping $\psi': A_1 \times A_2 \times \ldots \times A_n \to A$ given as follows: For any $(a_1, a_2, \ldots, a_n) \in A_1 \times A_2 \times \ldots \times A_n$, let

$$\psi'((a_1, \ldots, a_n)) = \begin{cases} \psi((a_1, \ldots, a_n)) & \text{if} \quad \psi((a_1, \ldots, a_n)) \text{ is defined,} \\ \text{undefined-otherwise.} \end{cases}$$

It can be proved easily that ψ' is an A-homomorphism of a suitable A-subautomaton of $\mathbf{A}_1^{(7)} * \ldots * \mathbf{A}_n^{(7)}$ onto $\mathbf{A}$, i.e., the superposition $\mathbf{A}_1^{(7)} * \mathbf{A}_2^{(7)} * \ldots * \mathbf{A}_n^{(7)}$ realizes $\mathbf{A}$. This shows the applicability of (VII).

It remains to show that (IV) is valid. To do this consider the following two results.

Theorem 1. Let $\mathbf{A}$ be an automaton with n states. Then for any natural number k, every connected A-subautomaton of the A-direct power $\mathbf{A}^k$ of $\mathbf{A}$ is MA-isomprphic to a suitable A-subautomaton of the A-direct power $\mathbf{A}^n$.

Theorem 2. (R. J. Nelson [5]). Every permutation automaton is strongly connected or can be given as a direct sum of strongly connected permutation automata.

We now prove two lemmas. Applying them, we get Theorem 3 which shows the validity of (IV).

Lemma 1. Let n and l be arbitrary natural numbers such that $1 < l < n$. Furthermore, let $\mathbf{A}$ be a connected permutation automaton with n states having an SR-system $(\mathbf{A}_1, \ldots, \mathbf{A}_m)$ of rank less than or equal to l.

Assume that an SR-system $(\mathbf{B}, \mathbf{C})$ of $\mathbf{A}$ has the following properties.

(a) $\mathbf{B} * \mathbf{C}$ is an MA-homomorphic image of a connected A-subautomaton of $\mathbf{A}^n$,

(b) $(\mathbf{A}_1, \ldots, \mathbf{A}_i)$ $(1 < i \leq m)$ is an SR-system of $\mathbf{B}$.

Then, using (IV), one can find an SR-system $(\mathbf{B}_1, \mathbf{C}_1)$ of $\mathbf{B}$ and a natural number t such that

(c) $\mathbf{B}_1 * \mathbf{C}_1 * \mathbf{C}$ is MA-homomorphic image of an A-subautomaton of $\mathbf{A}^{n \cdot t}$,

(d) $\mathbf{A}_1 * \ldots * \mathbf{A}_{i-1}$ realises $\mathbf{B}_1$,

(e) $\mathbf{C}_1$ has a number of states not exceeding l.

Proof. Using Theorem 2, it can be proved easily that every connected A-subautomaton of $\mathbf{A}^n$ is strongly connected permutation automaton. Therefore, the same is true for $\mathbf{B} * \mathbf{C}$, too. Thus $\mathbf{B}$ (as the first component of $\mathbf{B} * \mathbf{C}$) should be strongly connected permutation automaton. From this it follows, by an easy computation, that $\mathbf{A}_1 * \ldots * \mathbf{A}_{i-1}$ has a strongly connected A-subautomaton $\mathbf{D}$ such that $\mathbf{D} * \mathbf{A}_i$ realizes $\mathbf{B}$.

Let us denote by $F(X)$ the input semigroups of $\mathbf{A}$ and $\mathbf{D}$. Moreover, let D and A_i be the state sets of $\mathbf{D}$ and $\mathbf{A}_i$, respectively. Take an A-homomorphism ψ of a suitable A-subautomaton of $\mathbf{D} * \mathbf{A}_i$ onto $\mathbf{B}$. For any $d \in D$, define the set

$$\Delta(d) = \langle \psi((d, a_i)) \mid a_i \in A_i \rangle. \tag{1}$$

Since $\mathbf{B}$ is strongly connected thus $\Gamma = \langle (\Delta(d))^p \mid p \in F(x) \rangle$ is a cover of $\mathbf{B}$ for any $d \in D$.

Accomplishing a step of (IV), we get an SR-system $(\mathbf{B}_1, \mathbf{C}_1)$ of $\mathbf{B}$ belonging to Γ.

On the other hand, since the number of states of $\mathbf{A}_i$ does not exceed l and, by definition (1), $\overline{\overline{\Delta(d)}} \leqq l (d \in D)$ thus $\mathbf{C}_1$ has not more states than l. Therefore, (e) is valid.

Define a partitition Π on D as follows: $d_1 \equiv d_2(\Pi)$ if and only if $\Delta(d_1) = \Delta(d_2)(d_1, d_2 \in D)$. Then, by (1), Π is congruent. Therefore, $\mathbf{B}_1$ is an MA-homomorphic image of $\mathbf{D}$, i.e., (d) is valid.

Now in order to prove our Lemma it is enough to show that, choosing a suitable natural number t, (c) is also true. Since $\mathbf{B}$ is a permutation automaton thus $\overline{\overline{(\Delta(d))p}} = \overline{\overline{\Delta(d)}}$ holds for arbitrary $d \in D$ and $p \in F(X)$. Therefore, it is easy to prove that for any $d \in D$ and $p \in F(X)$,

$$(\Delta(d))^p = \Delta(dp). \tag{2}$$

By this equality (2), we can use the notation $\Delta(d)(d \in D)$ for the elements of Γ.

For any $\Delta(d) \in \Gamma$, let $\Phi_{\Delta(d)}$ be the one-to-one mapping of $\langle 1, 2, \ldots, \overline{\overline{\Delta(d)}} \rangle$ onto $\Delta(d)$ determined *by* $\mathbf{C}_1$. Moreover, let ψ' be the MA-homomorphism of a suitable connected A-subautomaton of $\mathbf{A}^n$ onto $\mathbf{B} * \mathbf{C}$. Since this subautomaton is strongly connected permutation automaton (see Theorem 2) thus the number of elements of arbitrary class of the partitition induced by ψ' is the same natural number t_1. Denote by C the state set of $\mathbf{C}$ and let $t = t_1 \cdot \overline{\overline{\Delta(d)}} \cdot \overline{\overline{C}} (d \in D)$.

For arbitrary state $(\Delta(d), c_1, c)$ of $\mathbf{B}_1 * \mathbf{C}_1 * \mathbf{C}$, let

$$\Omega(\Delta(d), c_1, c) = \Big\langle (a_1, a_2, \ldots a_{n \cdot t}) \Big| \bigcup_{i=0}^{t-1} \langle \psi'((a_{i \cdot n+1}, \ldots, a_{(i+1) \cdot n})) \rangle =$$

$$= \Delta(d) \times C, \psi'((a_1, \ldots, a_n)) = (\Phi_{\Delta(d)}(c_1), c) \Big\rangle. \tag{3}$$

We show that for any pair $(\Delta(d), c_1, c), (\Delta(d'), c_1', c')$,

$$(\Delta(d), c_1, c) \neq (\Delta(d'), c_1', c') \Rightarrow \Omega(\Delta(d), c_1, c) \cap \Omega(\Delta(d'), c_1', c') = \emptyset. \tag{4}$$

Assume that $\Delta(d) \neq \Delta(d')$. Then it can also be assumed that there exists a $b \in \Delta(d)$ with $b \notin \Delta(d')$. Take a state $(a_1'', \ldots, a_n'')$ from $\mathbf{A}^n$ such that $\psi'((a_1'', \ldots, a_n'')) \in \in \langle b \rangle \times C$. Then, by (3), every element $(a_1, \ldots, a_{n \cdot t})$ of $\Omega(\Delta(d), c_1, c)$ has a part $(a_{i \cdot n+1}, \ldots, a_{(i+1) \cdot n})$ $(0 \leqq i \leqq t-1)$ which is equal to $(a_1'', \ldots, a_n'')$, and for any element $(a_1', \ldots, a_{n \cdot t}')$ of $\Omega(\Delta(d'), c_1', c')$ we have $(a_{j \cdot n+1}', \ldots, a_{(j+1) \cdot n}') \neq (a_1'', \ldots, a_n'')$ $(j = 0, 1, \ldots, t-1)$. Therefore (4) is true.

Let $\Delta(d) = \Delta(d')$ and assume that $(c_1, c) \neq (c_1', c')$. Then by (3) for any pair $(a_1, a_2, \ldots, a_{n \cdot t}) \in \Omega(\Delta(d), c_1, c)$, $(a_1', a_2', \ldots, a_{n \cdot t}') \in \Omega(\Delta(d'), c_1', c')$ we have that $(a_1, \ldots, a_n) \neq (a_1', \ldots, a_n')$. This completes the proof of (4).

Let us show that for any state $(a_1, a_2, \ldots, a_{n \cdot t})$ of the A-direct power $\mathbf{A}^{n \cdot t}$ defined by (3) and for any input word $p \in F(X)$

$$(a_1, a_2, \ldots, a_{n \cdot t}) \in \Omega(\Delta(d), c_1, c) \Rightarrow (a_1, \ldots, a_{n \cdot t}) \cdot p \in \Omega((\Delta(d), c_1, c) \cdot p). \tag{5}$$

Since $\mathbf{B}$ and $\mathbf{B} * \mathbf{C}$ are permutation automata thus

$$(\forall (d, p))(d \in D, p \in F(X))(\overline{\overline{(\Delta(d))^p}} = \overline{\overline{\Delta(d)}}, \overline{\overline{(\Delta(d) \times C)^p}} = \overline{\overline{\Delta(d) \times C}}),$$

i.e. $(\Delta(d)\times C)^p = (\Delta(d))^p\times C$. Thus for arbitrary element $(a_1, a_2, \ldots, a_{n\cdot t})$ of $\Omega(\Delta(d), c_1, c)$ we have

$$\bigcup_{i=0}^{t-1} \langle \psi'((a_{i\cdot n+1}, \ldots, a_{(i+1)\cdot n})\cdot p)\rangle = (\Delta(d))^p\times C. \tag{6}$$

From $(a_1, \ldots, a_{n\cdot t})\in\Omega(\Delta(d), c_1, c)$ and (3)

$$\psi'((a_1, a_2, \ldots, a_n)\cdot p) = (\Phi_{\Delta(d)\cdot p}(c_1'), c') \tag{7}$$

where c_1' and c' are the second and third components of $(\Delta(d), c_1, c)\cdot p$. Hence used the definition (3) implies the (6) and (7) the (5) is valid, too.

From (4) and (5) we have that an A-subautomaton of A-direct power $\mathbf{A}^{n\cdot t}$ can be mapped MA-homomorphically onto $\mathbf{B}_1 * \mathbf{C}_1 * \mathbf{C}$. The classes of this homomorphism are represented by definition (3). This completes the proof of Lemma 1.

The following holds.

Lemma 2. Let $(\mathbf{B}, \mathbf{C})$ be an SR-system of a connected permutation automaton $\mathbf{A}$ and assume that $\mathbf{C}$ has fewer states than $\mathbf{A}$. Then it can be found an SR-system $(\mathbf{B}'\ \mathbf{C}')$ of $\mathbf{A}$ such that

(a) $\mathbf{B}'$ is MA-isomorphic to a strongly connected A-subautomaton of $\mathbf{B}$,

(b) using (IV) $\mathbf{C}'$ can be constructed as the second component of an SR-system of $\mathbf{A}$,

(c) $\mathbf{C}'$ has not more states than $\mathbf{C}$,

(d) $\mathbf{B}' * \mathbf{C}'$ is strongly connected,

(e) $\mathbf{B} * \mathbf{C}$ realises $\mathbf{B}' * \mathbf{C}'$.

Proof. Let ψ be an A-homomorphism of an A-subautomaton $\mathbf{M}$ of $\mathbf{B} * \mathbf{C}$ onto $\mathbf{A}$ and take a fixed state (b_0, c_0) of $\mathbf{M}$. Since $\mathbf{A}$ is strongly connected thus it can be assumed that $\mathbf{M}$ is also strongly connected.

Let $\mathbf{B}=\mathbf{B}(X, B, Y, \delta_B, \lambda_B)$ and take

$$\Delta(b_0) = \langle \psi(b_0, c) | c\in C\rangle, \quad\text{and}\quad \Delta(b) = (\Delta(b_0))^p \tag{8}$$

where $b=b_0p$ $(b\in B, p\in F(X)$ and C is the state set of $\mathbf{C})$.

Since $\mathbf{M}$ is strongly connected thus $\Delta(b_0)$ is non-empty. Therefore, $\Gamma = \langle(\Delta(b_0))^p | p\in F(X)\rangle$ is a cover of $\mathbf{A}$.

Denote by $(\mathbf{B}_1, \mathbf{C}')$ an SR-system of $\mathbf{A}$ belonging to Γ. By (8) and the construction of Γ, it can be seen that $\mathbf{C}'$ satisfies conditions (b) and (c) of Lemma 2.

Now let us define the automaton $\mathbf{B}'=\mathbf{B}'(X, B', \Gamma\times X, \delta_{B'}, \lambda_{B'})$ in the following way: $B'=\langle b | b=b_0p, p\in F(X)\rangle$ and for any $x\in X$ and $b\in B$, $\delta_{B'}(b, x)=\delta_B(b, x)$ and $\lambda_{B'}(b, x)=(\Delta(b), x)$.

By our construction, it is clear that $(\mathbf{B}', \mathbf{C}')$ is an SR-system of $\mathbf{A}$; furthermore, conditions (a) and (d) of Lemma 2 is satisfied.

Again, since $\mathbf{A}$ is a permutation automaton thus

$$(\Delta(b))^p = \Delta(bp) \tag{9}$$

for any $b\in B'$ and $p\in F(X)$.

For any $(b, k)\in B'\times\langle 1, 2, \ldots, \overline{\overline{\Delta(b)}}\rangle$, take

$$\Omega(b, k) = \langle (b, c) | c\in C, \psi((b, c)) = \Phi_{\Delta(b)}(k)\rangle \tag{10}$$

where $\Phi_{\Delta(b)}$ is the one-to-one mapping of $\langle 1, 2, \ldots, \overline{\overline{\Delta(b)}}\rangle$ onto $\Delta(b)$ determined by $\mathbf{C}'$. By (9), the set $\Omega(b, k)$ given by (10) is defined for any (b, k) from $B' \times \langle 1, 2, \ldots, \ldots, \max_{b \in B'} \overline{\overline{\Delta(b)}}\rangle$. On the other hand, since the mappings $\Phi_{\Delta(b)}: \langle 1, 2, \ldots, \Delta(b)\rangle \rightarrow \Delta(b)$ defined by $\mathbf{C}'$ are 1—1 thus the sets $\Omega(b, k)(b \in B', k \in \langle 1, \ldots, \overline{\overline{\Delta(b)}}\rangle)$ forms a partition of a given subset of $B \times C$. Taking into consideration that ψ is a homomorphism this partition can be induced by a homomorphism ψ' onto $\mathbf{B}' * \mathbf{C}'$ because of (9). Therefore, $\mathbf{B} * \mathbf{C}$ realizes $\mathbf{B}' * \mathbf{C}'$ which ends the proof of Lemma 2.

It can be proved that if **A** is an permutation automaton with n states then none of the strongly connected A-subautomata of $\mathbf{A}^n$ has more states than $n!$ Thus the validity of (IV) follows from.

Theorem 3. Let n and l be natural numbers with $1 < l < n$. Moreover, assume that the connected permutation automaton **A** with n states has an SR-system $(\mathbf{A}_1, \ldots, \ldots, \mathbf{A}_m)$ of rank l. Then, using (IV), we get an SR-system $(\mathbf{B}_1, \ldots, \mathbf{B}_m)$ of **A** with rank not exceeding l such that $\mathbf{A}^n$ has an A-subautomaton which can be mapped MA-homomorphically onto $\mathbf{B}_1 * \ldots * \mathbf{B}_m$.

Proof. Let $\mathbf{B}_{m+1}$ an automaton with one state having the same input set as **A**; moreover, under any input signal x, $\mathbf{B}_{m+1}$ produces the same output signal x.

Let $\mathbf{B}=\mathbf{A}$, $\mathbf{C}=\mathbf{B}_{m+1}$ and $i=m$. It is clear hat for any $(\mathbf{B}, \mathbf{C})$ and natural i, the conditions of Lemma 1 are satisfied. By Lemma 2, it can be assumed that for the pair $(\mathbf{D}_0, \mathbf{B}_m)$ $(\mathbf{D}_0=\mathbf{B}_1, \mathbf{B}_m=\mathbf{C}_1)$ given at the first step of (IV), $\mathbf{D}_0 * \mathbf{B}_m$ is strongly connected, i.e., $\mathbf{A}^{n \cdot t}$ has a strongly connected A-subautomaton which can be mapped MA-homomorphically onto $\mathbf{D}_0 * \mathbf{B}_m$. Since $\mathbf{B}=\mathbf{A}$ thus $(\mathbf{D}_0, \mathbf{B}_m)$ is an SR-system of **A**; i.e., we can disregard $\mathbf{B}_{m+1}$.

Using Theorem 1, there is an A-subautomaton of $\mathbf{A}^n$ which can be mapped MA-homomorphically onto. $\mathbf{D}_0 * \mathbf{B}_m$. Thus the system $\mathbf{B}=\mathbf{D}_0$, $\mathbf{C}=\mathbf{B}_m$, $i=m-1$ satisfies the conditions of Lemma 1.

By Lemma 2, it can be assumed that for any pair $(\mathbf{D}_1, \mathbf{B}_{m-1})$ obtained at the second step of (IV), $\mathbf{D}_1 * \mathbf{B}_{m-1}$ is strongly connected. Again, using Lemma 2, it can also be shown that $\mathbf{D}_1 * \mathbf{B}_{m-1} * \mathbf{B}_m$ is strongly connected. This, by Theorem 1, implies that $\mathbf{A}^n$ has a strongly connected A-subautomaton which can be mapped MA-homomorphically onto $\mathbf{D}_1 * \mathbf{B}_{m-1} * \mathbf{B}_m$. Therefore, the system $\mathbf{B}=\mathbf{D}_1$, $\mathbf{C}=\mathbf{B}_{m-1} * \mathbf{B}_m$, $i=m-2$ satisfies the conditions of Lemma 2. Repeating this process, we get an SR-system $(\mathbf{B}_1, \ldots, \mathbf{B}_m)$ of **A** such that

(a) $\mathbf{A}_1$ realizes $\mathbf{B}_1$ and the number of states of $\mathbf{B}_1$ and $\mathbf{B}_i$ $(i=2, \ldots, m)$ do not exceed l,

(b) $\mathbf{A}^n$ has an A-subautomaton which can be mapped MA-homomorphically onto $\mathbf{B}_1 * \ldots * \mathbf{B}_m$,

(c) the system $(\mathbf{B}_1, \ldots, \mathbf{B}_m)$ (except one-state components) can be given by applications of (IV).

This completes the proof of Theorem 3 and at the same time we proved that our process is right.

We now show the validity of.

Theorem 4 (see [2]). There exists an automaton **A** with four states such that **A** can be realized by a superposition of three automata having fewer states than **A** but no superposition of two automata having fewer states than **A** realizes **A**.

Proof. Let $\mathbf{A}=\mathbf{A}(X, A, A\times X, \delta, \lambda)$ be the automaton with $X=\langle x_1, x_2\rangle$ given by the transition table below

δ	x_1	x_2
a_1	a_2	a_2
a_2	a_3	a_3
a_3	a_4	a_1
a_4	a_1	a_4

The $\lambda: A\times X\rightarrow A\times X$ output function induces the identical mapping.

It can be proved easily that any cover of $\mathbf{A}$ has at least four elements. Therefore, using a result by M. Yoeli [7], $\mathbf{A}$ cannot be realized as a superposition of two automata having fewer states than $\mathbf{A}$.

Now take an SR-system $(\mathbf{B}_1, \mathbf{A}_3)$ belonging to the cover $\Gamma_0=\langle\langle a_1, a_2\rangle^p|p\in F(X)\rangle$ of $\mathbf{A}$. Furthermore, let $(\mathbf{A}_1, \mathbf{A}_2)$ be an SR-system belonging to the cover $\Gamma_1=\langle\langle\langle a_1, a_2\rangle, \langle a_3, a_4\rangle\rangle^p|p\in F(X)\rangle$ of $\mathbf{B}_1$. By the constructions of Γ_0 and Γ_1, it can be proved easily that $\mathbf{A}_1$, $\mathbf{A}_2$ and $\mathbf{A}_3$ have fewer states than four. This ends the proof of Theorem 4.

THE BOULDING SOCIETY OF
SOUTHERN HUNGARY /DÉLÉP/
SZEGED, HUNGARY
BOCSKAI U 10–12.

References

[1] Gécseg, F., О композиции автоматов без петель. *Acta Sci. Math. (Szeged),* v. 26, 1965, pp. 269—272.
[2] Gécseg, F. & I. Peák, *Algebraic theory of automata,* Disquisitiones Matematicae, Akadémiai Kiadó, Budapest, 1971.
[3] Глушков, В. М., Абстрактная теория автоматов, *Uspehi Mat. Nauk.,* v. 16:5 (101), 1961, pp. 3—62.
[4] Hartmanis, J., Loop-free structures of sequential machines, *Information and Control,* v. 5, 1962, pp. 25—44.
[5] Nelson, R. J., *Introduction to automata,* Wiley, New York—London—Sydney, 1968.
[6] Yoeli, M., The cascade decomposition of sequential machines, *IRE Trans. Electronic Comput.,* v. 10, 1961, pp. 587—592.
[7] Yoeli, M., Generalized cascade decomposition of automata, *J. Assoc. Comput. Mach.,* v. 12, 1965, pp. 411—422.
[8] Zeiger, H. P., *Loop-free synthesis of finite-state machines,* Ph. D. Thesis, Dept. of Electrical Engineering, M.I.T., 1964.

(Received Jan. 24, 1974)

Acta Cybernetica, Tom. 2, Fasc. 4, Szeged, 1975.

A three-dimensional cellular space

(A challenge to Codd-ICRA)

By A. DOMÁN

Abstract

A three-dimensional cellular space is suggested. Three-dimensionality enables the designer to form structures of highly consize character. As an advantage to the traditional — generally two dimensional — cellular spaces it yields not only reduction in cells number but also, as a result of the paths, some increase in operation speed. By the way, crossover problem (so central in planar constructions) simply does not arise.

A next relevant feature of our space is the complete interchangeability of the inputs and unlike in Neumann-space the outputs too (full symmetry). As a consequence, full symmetry ensures a considerably simple possibility of hardware implementation by IC technology, MSI certainly suffices. In addition, along design symmetry properties of the cell can be well made use of.

A *signal* several means for signal propagation, gates, gating and storage elements have been defined. An attempt has been made to gain effectivity. Besides the undoubtedly advantageous features like hardware simplicity and software effectivity there are inevitably some disadvantages. A characteristic feature of cellular automata: self-reproduction have to be given up. An other feature, viz. growing automata and growing structures again, have to be sacrificed. These features, however, cannot be considered disadvantages as far as computational applications are concerned. A practical difficulty arises from the spatial characteristics. Along design traditional drawing have to be replaced, at least partly, by a sort of a model making technique.

Introduction

Cellular space is a homogeneous, clocked network of identical mutually connected automata where the next state of each automaton depends only on its own and on its neighbors' last states.

In the recent few years a number of papers have been published dealing with several types of cellular spaces. A representative collection of articles (until around the mid 60's) can be found in Burks [1]. Cellular automata, i.e. automata "planted" in cellular space, or configurations in cellular space have been first studied by von Neumann [12] motivated by brain researches and has its way back to the famous Hixon Symposium in 1948 (see Jeffress ed. 1951). Since Burks [1] a Conference on Biologically Motivated Automata represents the last development [9].

As for the motivations of most of the studies into cellular automata, it seems that cellular automata are considered as organisation *principles* or new types of computation *techniques*, rather than as a matter of hardware reality, or a true medium to be produced so as to act as a host medium for constructing and breeding growing and selfreproducing automata.

One exception to that approach comes from a Hungarian group aiming at the implementation and effectivization of cellular spaces (see Fáy [6], Fáy—Takács [7], Fáy [5], Takács [14] Dettai [3], Szőke [13], Fazekas [8], Kaszás [11], Huszár [10], Dettai [4]).

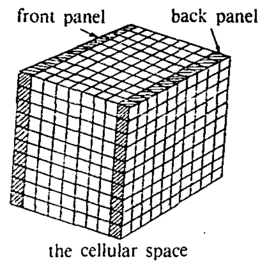


Fig. 1

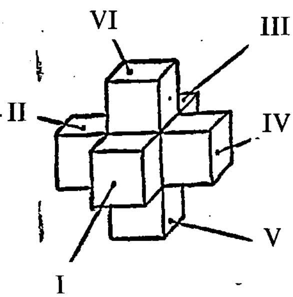


Fig. 2
The neighborhood template of the three-dimensional space with the names of neighbors.
(First (I) or front neighbor; second (II) or left neighbor; third (III) or back neighbor; fourth (IV) or right neighbor; fifth (V) or bottom neighbor; sixth (VI) or top neighbor.)

By the present paper, we would like to join this direction of research as far as the effectivization of cellular automata (or space) is concerned. Yet there is a difference, for all the last practically motivated papers (Dettai [4] excepted) are centering around the effectivization of Codd's cellular space while the one here is not. It turned out that at present a Codd-cell (even in its late version, called Codd-ICRA) is too complex to be implemented economically by integrated circuit technology. It's true, it could be done both theoretically as well as practically; it is only *economy* which is questionable in my opinion at least. At any rate, I feel I have found a version for cellular space which is competitive with respect to ICRA's performance and definitely much simpler than it is in its circuitry realization.

As for the preliminaries an attempt is made to be fairly self-contained. As for further backgrounds, underlying ideas and motivations, however, readers are better adviced to refer to Codd [2], and to the publications referred to above.

Preparations

It is well enough for our present purposes if the cellular space for developing and studying, is visualized as a rectangular block of identical cubes of finite number (see Fig. 1). This number is around one million in practice.

The *neighbors* of a cell in this space are the cells sharing common faces with the *center cell*, i.e. the cell in question. In other words, a next-neighborhood is accepted with six neighbors of each cell. The neighborhood *template* can be seen on Fig. 2.

In our six-neighbor space each cell has eight states. These are denoted by 0, 1, 2, ..., 7. Usually blank is used for 0.

We need a convention for telling the transition rules.

This again is explained through an example:

S cell state	N neighbors state	O next state
3	$\bar{3}$ 5 $\bar{7}$	6

This excerpt of the *transition table* reads as follows: *If* the present state of a cell is 3 *and* it has *no* neighbor in state 3 *and* at least one neighbor in state 5 *and* *no* neighbor in state 7 *then* this cell's next state will be 6.

Blank entry in column N means that the next state is completely independent from the neighbors.

According to this convention the complete *transition function* is contained in Table 1.

Table 1.
The transition function

S	N	O
0		0
1	4 6 $\bar{5}$	1
1	4 $\bar{6}$ 5	7
1	4 6 5	4
1	4 $\bar{6}$ $\bar{5}$	4
1	7	4
1	$\bar{4}$ $\bar{7}$	1
2		1
3	4 7	5
3	3 5	5
3	$\bar{3}$ 5 $\bar{7}$	6
3	7 $\bar{4}$	3
3	$\bar{7}$ $\bar{5}$	3
4		2
5	4 7	3
5	3 6	3
5	$\bar{3}$ $\bar{4}$	5
5	3 5	5
5	2 7	5
6	3	3
6	$\bar{3}$	6
7	2	2
7	$\bar{2}$	7

Needless to say, the transition function, just like that of Codd's, is a *partially defined* one. The function is so simple that its implementation by integrated circuit technology is quite a straightforward routine.

This convention of defining the transition function is legitimized by its peculiar logical structure.

A novel feature, compared with the usual cell spaces, is that our cells have *two modes of operation*. The transition function, defined by Table 1, refers to the first mode of operation called *ordinary mode* (OM). The second mode is called *shift mode* (SM). In this mode the whole cellular space acts as a collection of parallel coupled serial shift registers connecting the front panel to the back panel. In this mode any configuration of the front panel shifts forward during the mode. By this, one can transfer any configuration into the inside of the space. Of course, spatial figures (or structures) have to be sliced up and the slices have to put serially (step by step) in touch with the front panel's cells. Fig. 3 shows this "shift-in" procedure.

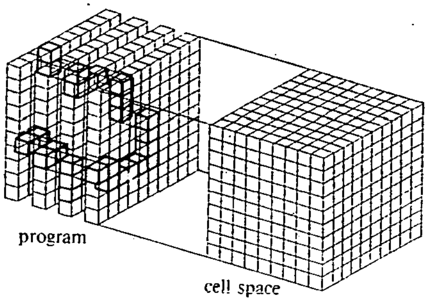


Fig. 3

The concept of the *empty* or *quiescent* state of the space is identical with the usual. In the space with each cell in state 0 and all the edge cells having no input, no change occurs.

The problem of *clearing or erasure* is solved by the shift mode. All we have to do is to switch to SM from OM and shift out everything to the back panel.

Basic structures and functions

Paths. As in Codd [2], a *path* can, in the simplest case, be defined as a linear row of neighboring cells all in state 1. Paths, then can have bends (corners), branches, loops etc. Typical paths are shown on Figures 4-5.

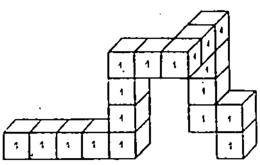


Fig. 4
A typical simple path

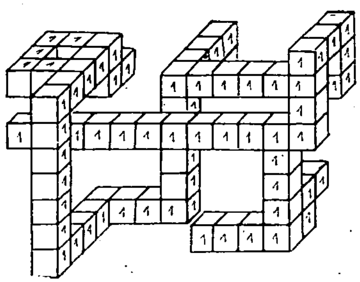


Fig. 5
A typical path

Sometimes it is possible — and preferable too — to resort to planar representation. If so we use the traditional means, for instance: cells are represented by squares, empty cell means cell in state 0, filled square means cell in state one, the other cell's states are denoted by numbers placed in the representing square. Unlike Codd we do *not* sheathe paths.

Signals. We have only one type of *signal* which is a pair of adjacent cells with state 2 and 4 respectively. So our signals are always of the form "24". (By contrast, Codd has four signals: 04, 05, 06 and 07.)

Signals propagate along paths with 4 heading and 2 tailing. So signal propagation is shown on the shot-figures:

1 1 2 4 1 1 1 1 1 1	at $t=1$
1 1 1 2 4 1 1 1 1 1	$t=2$
1 1 1 1 2 4 1 1 1 1	$t=3$
1 1 1 1 1 2 4 1 1 1	$t=4$
1 1 1 1 1 1 2 4 1 1	$t=5$

Propagating in opposite directions along the same path *signal collision* may occur. Signal collision results annihilation independently from the parity of the signal distance

Signal collision with odd parity:

events	shots
1 1 2 4 1 1 1 4 2 1 1 1 1	t
1 1 1 2 4 1 4 2 1 1 1 1 1	$t+1$
1 1 1 1 2 4 2 1 1 1 1 1 1	$t+2$
1 1 1 1 1 2 1 1 1 1 1 1 1	$t+3$
1 1 1 1 1 1 1 1 1 1 1 1 1	$t+4$

Signal collision with even parity:

1 1 2 4 1 1 1 1 4 2 1 1 1	t
1 1 1 2 4 1 1 4 2 1 1 1 1	$t+1$
1 1 1 1 2 4 4 2 1 1 1 1 1	$t+2$
1 1 1 1 1 2 2 1 1 1 1 1 1	$t+3$
1 1 1 1 1 1 1 1 1 1 1 1 1	$t+4$

Path branching:

```
        1                    1                    1
1 2 4 1 1 1 1        1 1 2 4 1 1 1        1 1 1 2 4 1 1
        1                    1                    1
        1                    1                    1
        t                   t+1                  t+2

              1                    4
        1 1 1 1 2 4 1        1 1 1 1 1 2 4
              4                    2
              1                    4
             t+3                  t+4
```

Signal collisions in crosspaths:

```
        1                  1                  1
        1                  1                  1
1 2 4 1 1 1 1      1 1 2 4 1 1 1      1 1 1 2 4 1 1
        1                  4                  2
        4                  2                  1
        t                 t+1                t+2

        1                  4
        4                  2
1 1 1 1 2 4 1      1 1 1 1 1 2 4
        1                  1
        1                  1
       t+3                t+4
```

odd distance

```
      1                1                1
      1                1                4
1 2 4 1 1 1      1 1 2 4 1 1      1 1 1 2 4 1
      1                1                4
1 2 4 1          1 1 2 4          1 1 1 2
      t               t+1              t+2

      4                2
      2                1
1 1 1 1 2 4      1 1 1 1 1 2
      2                1
1 1 1 1          1 1 1 1
     t+3              t+4
```

even distance

Collision in fork:

```
1 1 2 4 1                1 1 1 2 4                1 1 1 1 2                1 1 1 1 1
        1 1 1 1                  1 1 1 1                  4 1 1 1                  2 4 1 1
1 1 2 4 1                1 1 1 2 4                1 1 1 1 2                1 1 1 1 1
```

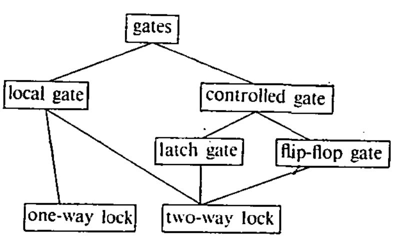


Fig. 6

Gates. In our construction gates play the key roles. Structurally one has four basic types of gates contrasted with Codd who had only one. (Whereas he had four signals while we manage with one.)

The four gate types are as follows (see Fig. 6). Local gates (being not controllable) are always permanent.

Local one-way locks consist of two cells placed beside the path to be gated, i.e. the *subordinate path:*

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1    subordinate path
          5 6
         /   \
slave cell    master cell
```

symbol: 

A signal propagating along the subordinate path will be annihilated by the local one-way lock if the signal first encounters the master cell then the slave cell. So, in contrast with Codd, no matter on which side of the path is the gate placed.

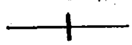
The function of the local one-way lock is:

```
1 1 1 1 1 4 2     1 1 1 1 4 2 1     1 1 1 1 2 1 1     1 1 1 1 1 1 1
    5 6               5 6               5 6               5 6
     t                t+1               t+2               t+3

2 4 1 1 1 1 1     1 2 7 1 1 1 1     1 1 2 4 1 1 1     1 1 1 2 4 1 1
    5 6               5 6               5 6               5 6
     t                t+1               t+2               t+3
```

Local two-way locks differ from the one-way locks that the slave cell is dropped:

```
1 1 1 1 1 1 1 1 1 1 1
          6
  local two-way lock
```

symbol: 

Again, no matter whether it is on the right or left hand side of the path (from an approaching signal). A signal coming from any direction will be annihilated by the local two-way lock:

```
2 4 1 1 1 1 1     1 2 4 1 1 1 1     1 1 2 1 1 1 1     1 1 1 1 1 1 1
    6                 6                 6                 6
    t                t+1               t+2               t+3

1 1 1 1 1 4 2     1 1 1 1 4 2 1     1 1 1 1 2 1 1     1 1 1 1 1 1 1
    6                 6                 6                 6
    t                t+1               t+2               t+3
```

Controlled gates have access by paths either to the master cell or to the slave cell. If the control path leads to the master cell we got the *flip-flop type gate* (see Fig. 7).

```
                     1 1 1 1 1 1 1 1 1 1 1——subordinate path
slave cell (in state 5)——5 6——master cell (in state 6)
support head——————7 5 1
support cell (in state 5)——1 1
                           1
                           1
                           1
             control path  1
```

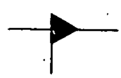
symbol: 

Fig. 7
Flip-flop gate

As it can be seen from Fig. 7 two additional cells are placed. The first is the first neighbor of the slave cell called *support cell,* the second is the second neighbor of the support cell called *support head.* Of course, other arrangements are equally possible as far as they are produced by a transformation permitted by the transition function. These other arrangements are considered to be isomorphic to the one on Fig. 7.

Owing to the support cell and support head control becomes possible through the control path.

Upon receiving a signal first the support cell goes into state 3 then in the next shot both the master and the slave cell change to 3:

```
1 1 1 1 1    1 1 1 1 1    1 1 1 1 1    1 1 1 1 1    1 1 1 1 1
  5 6          5 6          5 6          5 6          3 3
7 5 1        7 5 1        7 5 4        7 3 2        7 3 1
    1            4            2            1            1
    4            2            1            1            1
    2            1            1            1            1
    1            1            1            1            1
    1            1            1            1            1
```

Now the configuration of both the master and slave cells in state 3 are neutral making no effect on the signal passing the gate on the subordinate path:

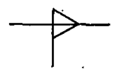
```
2 4 1 1 1 1 1 1    1 2 4 1 1 1 1 1    1 1 2 4 1 1 1 1
      3 3                3 3                3 3
    7 3 1              7 3 1              7 3 1
        1                  1                  1
        1                  1                  1
        1                  1                  1
        1                  1                  1
        1                  1                  1

1 1 1 2 4 1 1 1    1 1 1 1 2 4 1 1    1 1 1 1 1 2 4 1
      3 3                3 3                3 3
    7 3 1              7 3 1              7 3 1
        1                  1                  1
        1                  1                  1
        1                  1                  1
        1                  1                  1
        1                  1                  1
```

On the other hand:

```
1 1 1 1 1 1 4 2    1 1 1 1 1 4 2 1    1 1 1 1 4 2 1 1
    3 3                3 3                3 3
  7 3 1              7 3 1              7 3 1
      1                  1                  1
      1                  1                  1
      1                  1                  1
      1                  1                  1
      1                  1                  1
```

```
1 1 1 4 2 1 1 1    1 1 4 2 1 1 1 1    1 4 2 1 1 1 1 1    4 2 1 1 1 1 1 1
    3 3                3 3                3 3                3 3
  7 3 1              7 3 1              7 3 1              7 3 1
      1                  1                  1                  1
      1                  1                  1                  1
      1                  1                  1                  1
      1                  1                  1                  1
      1                  1                  1                  1
```

symbol: 

Flip-flop gate with master and slave cells in state 3 is called to be in "off"-state while the other (with master cell in state 6 and slave in 5) is "on"-state. As a synonym a gate in "on"-state is also called *closed gate* and an off gate is called *open gate*. The transition from on to off is referred to as *opening* and from off to on as *closing*.

A new signal propagating along the control path is capable of closing an open flip-flop gate:

```
1 1 1 1 1 1       1 1 1 1 1 1       1 1 1 1 1 1
    3 3               3 3               3 3
  7 3 1             7 3 4             7 5 2
      4                 2                 1
      2                 1                 1
      1                 1                 1
      1                 1                 1
- - - 1 - - - - - - - - 1 - - - - - - - - 1 - -
    1 1 1 1 1         1 1 1 1 1         1 1 1 1 1
      5 6               3 6               3 3
    7 3 2             7 3 1             7 3 1
        1                 1                 1
        1                 1                 1
        1                 1                 1
        1                 1                 1
        1                 1                 1
```

Latch gate differes structurally from the flip-flop gate that the control path is leading directly to the support cell rather than to the slave cell and passing the support (see Fig. 8).

```
                  1 1 1 1 1 1 1 1 1 1 1 1   subordinate path
slave cell ———————————————— 5 6 ——— master cell
support head ——————————— 7 5 ——— support cell
                            1
                            1
                            1
                            1
                            1
                            1 control path
```

Fig. 8
Latch gate in ON state

A latch gate, once closed, can never be opened. It has the latch property:

```
1 1 1 1 1 1   1 1 1 1 1 1   1 1 1 1 1 1   1 1 1 1 1 1   1 1 1 1 1 1
    5 6           5 6           5 6           5 6           5 6
  7 5           7 5           7 5           7 5           7 5
    1             7             2             1             1
    4             2             1             1             1
    2             1             1             1             1
    1             1             1             1             1
    1             1             1             1             1
    1             1             1             1             1
```

An open latch gate, however, can be closed:

```
1 1 1 1 1     1 1 1 1 1     1 1 1 1 1     1 1 1 1 1     1 1 1 1 1
    3 3           3 3           3 3           5 3           5 6
  7 3           7 3           7 5           7 5           7 5
    1             4             2             1             1
    4             2             1             1             1
    2             1             1             1             1
    1             1             1             1             1
    1             1             1             1             1
    1             1             1             1             1
```

A direct version of a controlled two-way lock is on Fig. 9.

```
            1 1 1 1 1 1 1 1 1 1
                              1
                              1
                              1 1 1 1 1 1 1 1 1 1   subordinate path
slave cell ---------------- 5 6 — master cell
support head ------------ 7 5 1
                            | 1
support cell ---------------┘ 1
                              1
                              1
                              1   control path
```

Fig. 9
Controlled two-way lock (in ON state)

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   subordinate path
        5 6   6 5
      7 5 1 1 1 5 7
            1
            1
            1
            1
            1   control path
```

Fig. 10
A controlled two-way lock as a combination of two controlled flip-flop gates

By combining gates of the above types one can produce new structures capable of performing new functions. For instance a controlled two-way lock can be seen on Fig. 10.

Also gates are suitable to control more than one path simultaneously. Some immediate versions can be seen on Figures 11—13. These intensively make use of the advantages due to the three-dimensionality of the space.

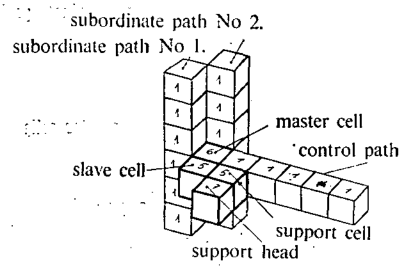


Fig. 11

Multifunctional usage of gates is utterly impossible in Codd's twodimensional space. We feel that this possibility in our space gives ample compensation for the difficulties arousing from the clumsy spatial representation techniques in design and demonstration.

```
                                  subordinate path No 2.
master cell ─────────\x ──────── (orthogonal to the drawing)
slave cell ────────── 5 6 x ──── subordinate path No 1.
support head ─── 7 5 1           (orthogonal to the drawing)
support cell ────────  1
                       1
                       1
                       1
                       1
```

Fig. 12

Top view of a bi-functional controlled two-way lock (cf. Fig. 11)

```
master cells ─────────────\x───── subordinate paths
slave cell                x 6 x   (orthogonal to the drawing)
(serving two masters) ──── 5 6 x
support head ─────────── 7 5 1
support cell ────────────/1
                          1
                          1
                          1
                          1
                          1
                          1  control path
```

Fig. 13

A multifunctional, threedimensional gate-arrangement for controlling four paths simultaneously in two-way lock mode

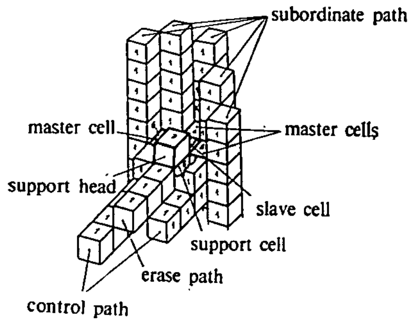


Fig. 14

The combination possibilities for the basic gate types are virtually indefinite. To close this paragraph let us see just one more (a truly spatial) multifunctional gate arrangement having two control paths and five subordinate paths (see Fig. 14).

A multifunctional double control one-way lock system. Support head is the bottom neighbor of the support cell. The system is capable of performing quite a sophisticated control function on five paths occupying only a volume less then ten cells.

Other structures and functions

Gating. Gates do not represent the only possibilities for performing gating functions. An other possibility is offered by signal collisions. OR-functions can be achieved by "forks". Signal 24 entering at point A or B or both simultaneously results a signal 24 at the output point C (see Fig. 15).

```
                  5 6
A   1 1 1 1 1 1 1
                    1 1 1 1 1 1 1   C
B   1 1 1 1 1 1 1
                  5 6
```

in symbols:

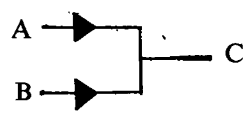


Fig. 15

Logical operation OR performed by a path fork rather than by a gate

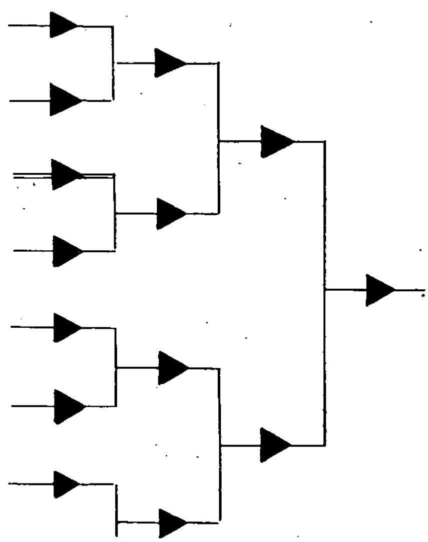


Fig. 16

An extension of this "gating by fork" can be seen on Fig. 16.

A *coincidence-* or AND-*gate* is somewhat more complex. Detailed shots of its operation can be seen on Fig. 17.

Into the control path a gate restoring loop is inserted by which the control signal opens the gate for 4 shots.

During that period the open gate permits the subordinate signal to pass, otherwise not. Thus the arrangement produces a *time slot* for the subordinate signal to pass. The minimal time slot is 4 shots.

In drawing this *interference gate* will be denoted this way: 

The number in the symbol indicates the time slot. Blank means minimal slot i.e. shots.

```
      1 1 1
      1   1
1 2 4 1 1 1 5 7
          6 5
1 2   1 1 1
  4   1   1 1 1
  1 1 1
     t = 1
```

```
      1 1 1
      1   1
1 1 2 7 1 1 5 7
          6 5
1 1   1 1 1
  2   1   1 1 1
  4 1 1
     t = 2
```

```
        1 1 1
        4   1
1 1 1 2 4 1 5 7
          6 5
1 1   1 1 1
  1   1   1 1 1
  2 4 1
     t = 3
```

```
    4 1 1
    2   1
1 1 1 1 2 4 5 7
        6 5
1 1   1 1 1
  1   1   1 1 1 1
   t = 4
```

```
1 2 4
    2 4 1
    1   1
1 1 1 1 1 2 3 7
          6 5
1 1   1 1 1
  1   4   1 1 1
  1 1 2
     t = 5
```

```
        1 2 7
        1   1
1 1 1 1 1 1 3 7
          6 3
1 1   4 1 1
  1   2   1 1 1 1
  1 1 1
     t = 6
```

```
      1 1 2
      1   4
1 1 1 1 1 1 3 7
          3 3
  1   2 4 1
  1   1   1 1 1
  1 1 1
   t = 7
```

```
      1 1 1
      1   2
1 1 1 1 1 4 3 7
          3 3
1 1   1 2 4
  1   1   1 1 1
  1 1 1
     t = 8
```

```
        1 1 1
        1   1
1 1 1 1 1 2 5 7
          3 3
1 1   1 1 2
  1   1   4 1 1
     t = 9
```

```
1 1 1
    1 1 1
    1   1
1 1 1 1 1 1 5 7
          3 5
1 1   1 1 1
  1   1   2 4 1
  1 1 1
     t = 10
```

```
      1 1 1
      1   1
1 1 1 1 1 1 5 7
          6 5
1 1   1 1 1
  1   1   1 2 4
  1 1 1
     t = 11
```

By a suitable delay between signals leaving points A and B respectively one can manage that they meet again right at an Interference Gate I_i $(i=1, 2, \ldots, 7)$. In this case this gate will be turned off (if previously it was ON) since signal can reach gates only through the open time slots.

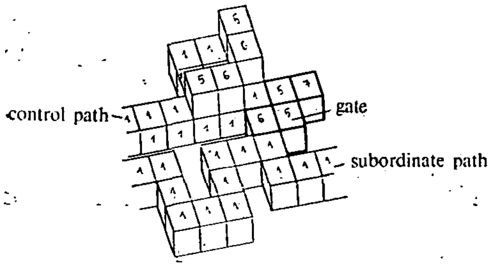


Fig. 17

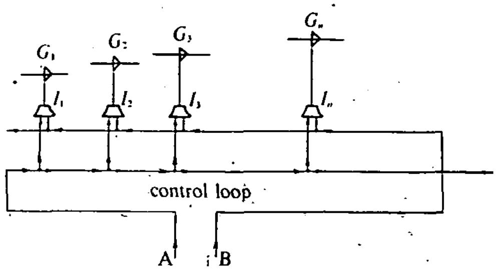


Fig. 18
The interference principle

Storage. Making use of the possibilities given by the spatial structure of gates one can easily form a *storage element* with read-write-erase capabilities (see Figures 19, 20). Fig. 19 shows a version with a one-way lock.

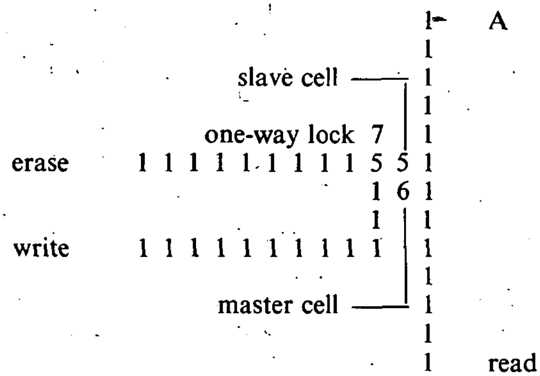


Fig. 19
A storage element using a one-way lock with read-write-erase capabilities

By path "write" one can control the master cell thereby gating the signal propagating along path "read". If the gate is ON then no signal can pass it. This state, say *empty state*, of the storage element is read out by the *lack* of the read signal. A read-signal appearing *beyond* the gate, at point A, indicates the *off state* of the storage element.

As it can be seen from Fig. 19 path "erase" is just a control path of a latch gate while, at the same time, from the "write"-path the same gate is a flip-flop gate. This flip-flop path is used for writing-in purposes (i.e. for writing in both states nought and one). On the other hand, the other path, i.e. the latch path can be eminently used for performing the erase-function of a storage element. On Fig. 20 a two-way lock version of the latter storage element is shown.

```
                              1   read
                              1
                              1
                              1
                        7     1
erase   1 1 1 1 1 1 1 1 5 5 6 1
                        1 6   1
write   1 1 1 1 1 1 1 1       1
                              1
                              1
                              1
                              1
                              1   read
```

Fig. 20
Storage element using two-way lock with two accesses for reading (cf. Fig. 19)

Conclusions

It has been demonstrated that quite effective components and functions can be gained in a cellular space containing unquestionably simple cells to implement. The number of ways of combination possibilities for the main components, such as several types of paths, gates and storage elements is virtually indefinite.

Of course, these elementary components themselves throw meager light on the true design advantages and disadvantages. Further R & D is necessary to be able to make decision in this respect.

COMPUTING INSTITUTE
MINISTRY OF LABOUR
H—1089 BUDAPEST, HUNGARY

References

[1] BURKS, A. W., *Essays on cellular automata,* University of Illinois Press, Urbana Illinois, 1968.
[2] CODD, E. F., *Cellular automata,* Academic Press, New York and London, 1968.
[3] DETTAI, I., A version of von Neumann's cell, *KGM ISzSzI Technical Report* Budapest, 1974.
[4] DETTAI, I., Effectivity problems and suggestions concerning Codd-ICRA cellular space, *KGM ISzSzI Technical Report* Budapest, 1974.
[5] FÁY, G., Circuitry realization of Codd's cellular automaton's cell, *Conference Paper* Székesfehérvár, 1973.
[6] FÁY, G., A lecture on DSL ALPHA and Codd-ICRA, *Conference Lecture* Vienna, 1974.
[7] FÁY, G. & D. V. TAKÁCS, Galois perceptron cell assemblies in cellular space, *Conference paper* Paris, 1974.
[8] FAZEKAS, B., Design for a "RETINA", *KGM ISzSzI Technical Report* Budapest, 1974.
[9] HERMANN, G. T., Conference on Biologically Motivated Automata held at MITRE Virginia June, 1974.
[10] HUSZÁR, A., Testing equipment for automaton cell circuitry, *KGM ISzSzI Technical Report* Budapest, 1974.
[11] KASZÁS, O., Character generation in Codd-ICRA, *KGM ISzSzI Technical Report* Budapest, 1974.

[12] von Neumann, J., *The theory of self-reproducing automata*, ed. A. W. Burks, University of Illinois Press, Urbana Illinois, 1966.
[13] Szőke, M., Codd-ICRA: Regularization of Codd's transition function and its implementability, *KGM ISzSzI Technical Report* Budapest, 1974.
[14] Takács, D. V., A bootstrap for Codd's cellular space, *Conference Paper* Székesfehérvár, 1973.
[15] Takács, D. V., Galois connection and DPL ALPHA system, *Conference Paper* Paris, May. 1974.

(Received Oct. 29, 1974)

INDEX—TARTALOM

Felelős szerkesztő és kiadó: Kalmár László
A kézirat a nyomdába érkezett: 1975. október hó
Megjelenés: 1976. július hó
Példányszám: 1000. Terjedelem: 6,62 (A/5) ív
Készült monószedéssel, íves magasnyomással
az MSZ 5601 és az MSZ 5602-55 szabvány szerint
75-4875 — Szegedi Nyomda — F. v.: Dobó József ig.