Tomus 1. Fasciculus 2.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM CYBERNETICARUM HUNGARICUM

Szeged, 1971

Curat: Universitas Szegediensis de Attila József nominata

1. kötet 2. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1971. szeptember
A szegedi József Attila Tudományegyetem gondozásában

Linear regular languages. Part II
The problem of synthesis

By G. T. HERMAN

1. Introduction

In Part I of this paper [3] we discussed the *problem of analysis* for linear sequential circuits, i.e. we gave an algorithm which for every language accepted by a linear sequential circuit (described by the circuit and the function mapping the symbols of the language into inputs to the circuit) produced a regular expression which described that language. We have also shown that the converse cannot be done, there is a regular expression such that the language described by it is not linear regular, i.e. its symbols cannot be mapped into inputs of a linear sequential circuit in such a way that the circuit will accept exactly those words which belong to the language. We shall assume that the reader is familiar with the terminology of [3].

The algorithm of [3] for the analysis of linear sequential circuits had the advantage over similar algorithms by its being a practical algorithm which can be implemented on a digital computer. Such implementation has been reported on in [4].

The problem of synthesis for linear sequential circuits is to give an algorithm which for any given regular expression decides whether or not it describes a linear regular language and, if that is indeed the case, the algorithm must provide us with a linear sequential circuit and a mapping of the symbols of the language into inputs of the circuit, such that the circuit will accept exactly those words which belong to the language. In this paper we shall describe an algorithm which will do this job. Unfortunately, from the practical point of view the algorithm will do little more than show that such algorithm exists, if implemented on a digital computer its operation would be so inefficient that it could not be applied even to very simple cases. This is a usual state of affairs with algorithms in automata theory, but it is our contention that this particular problem should have an implementable solution, similar in simplicity to the one for the problem of analysis. The reader should compare comments in § 5 of [3].

Similarly to [3], this paper will be introductory in the sense that it will make no assumption of knowledge on the part of the reader. Hence, some known definitions and results will be given and proved without reference to original sources.

2. Definitions

All definitions in [3] will be assumed to be known to the reader.

Definition 1. A *finite automaton* is a 5-tuple $M=\langle Q, \Sigma, q, \delta, F\rangle$, where
(i) Q is a finite non-empty set of *states*,
(ii) Σ is a finite non-empty set of *input symbols*,
(iii) $q\in Q$, the *initial state*,
(iv) $\delta: Q\times\Sigma \to Q$, the *direct transition function*,
(v) $F\subset Q$, the set of *accepting states*.

We extend the transition function to a mapping $\bar{\delta}$ from $Q\times I_\Sigma$ into Q as follows:

$$\bar{\delta}(q, e)=q,$$

$$\bar{\delta}(q, xa)=\delta(\bar{\delta}(q, x), a)$$

for all $q\in Q$, $a\in\Sigma$ and $x\in I_\Sigma$.

Since, for all $q\in Q$ and $a\in\Sigma$, $\bar{\delta}(q, a)=\delta(q, a)$, we denote $\bar{\delta}$ by δ as well.

Example 1.

where $\langle\{q_0, q_1, q_2, q_3\}, \{a, b\}, q_0, \delta, \{q_2\}\rangle$

$$\delta(q_i, a)=q_i, \quad \text{for} \quad 0\leqq i\leqq 3,$$

$$\delta(q_i, b)=q_{i+1}, \quad \text{for} \quad 0\leqq i\leqq 2,$$

$$\delta(q_3, b)=q_3,$$

is a finite automaton.

Example 2.

$$\langle\{q_0, q_1\}, \{a, b\}, q_0, \delta, \{q_0\}\rangle$$

where $\delta(q_i, a)=q_i, \quad \text{for} \quad 0\leqq i\leqq 1,$

$$\delta(q_i, b)=q_{1-i}, \quad \text{for} \quad 0\leqq i\leqq 1,$$

is a finite automaton.

Definition 2. Let $M=\langle Q, \Sigma, q, \delta, F\rangle$ be a finite automaton and $x\in I_\Sigma$. We say that M *accepts* x if and only if $\delta(q, x)\in F$. Let $W\in L_\Sigma$. We say that M *accepts* W if and only if the set of those x in I_Σ which are accepted by M is exactly W.

Example 3. The finite automaton given in Example 1 accepts the language described by

$$(((a^*b)(a^*b))a^*).$$

Example 4. The finite automaton given in Example 2 accepts the language described by

$$(a^*((ba^*)(ba^*))^*).$$

Definition 3. A finite automaton $M=\langle Q, \Sigma, q, \delta, F\rangle$ is said to be *linearly realizable* if and only if there exists a linear sequential circuit C and functions α and φ with the following properties. (We assume that the circuit C has k external input

wires and n delays and can be described by matrices **A**, **B** and **C** as in Theorem 1 of [3].)

(i) α maps Σ into k-tuples of 0's and 1's.

(ii) φ maps Q into n-tuples of 0's and 1's.

(iii) For each $p \in Q$ and $a \in \Sigma$,

$$\varphi(\delta(p, a)) = \varphi(p)\mathbf{A} \oplus \alpha(a)\mathbf{B}.$$

(iv) For each $p \in Q$,

$$p \in F \text{ if and only if } \varphi(p)\mathbf{C} = 1.$$

In such a case C is said to be a *linear realization* of M.

Example 5. The finite automaton of Example 2 is linearly realizable. Its linear realization is given in Figure 3 of [3]. α is defined by

$$\alpha(a) = 0, \quad \alpha(b) = 1.$$

φ is defined by

$$\varphi(q_0) = [1, 0], \quad \varphi(q_1) = [1, 1].$$

We shall see later on that the finite automaton of Example 1 is not linearly realizable. We note in passing that the definition of realization that is given here is somewhat restricted, but for the purpose of checking the linearity of regular languages it is as general as needed. For a discussion of various definitions of realization, see for instance [5].

Definition 4. An *initial subautomaton* of a finite automaton $M = \langle Q, \Sigma, q, \delta, F \rangle$ is the finite automaton $\langle Q', \Sigma, q, \delta', F' \rangle$, where

$$Q' = \{p | p \in Q \text{ and } p = \delta(q, x) \text{ for some } x \in I_\Sigma\},$$

$$\delta'(p, a) = \delta(p, a) \text{ for all } p \in Q' \text{ and } a \in \Sigma,$$

$$F' = F \cap Q'.$$

Intuitively, the initial subautomaton is that part of the automaton which consists of all the states which are accessible from the initial state.

Example 6. For the automata of Examples 1 and 2, the initial subautomaton is the automaton itself, since all states are accessible from the initial state.

The following basic result is easy to prove and we shall assume it in the rest of the paper without further reference to it.

Proposition. The language which is accepted by a finite automaton is the same as the language accepted by its initial subautomaton.

Definition 5. Let C be a linear sequential circuit with n delays. With each state $y = [y_1, \ldots, y_n]$ of C we associate a mapping λ_y^C from strings of inputs $x_1 \ldots x_t$ into $\{0, 1\}$ defined as follows:

$$\lambda_y^C(x_1 \ldots x_t) = y\mathbf{A}^t\mathbf{C} \oplus \sum_{i=1}^{t} x_i \mathbf{B}\mathbf{A}^{t-i}\mathbf{C}.$$

I.e. the value of $\lambda_y^C(x_1 \ldots x_t)$ is the same as the output would be at time $t+1$ if the machine was started in state y and received the external input x_i at time i.

Example 7. For the linear sequential machine C of Figure 3 in [3] we have the following.

$$\lambda^C_{[1,0]}(x_1 \dots x_t) = \begin{cases} 1 \text{ if an even number of } x_i \text{ is } 1, \\ 0 \text{ otherwise,} \end{cases}$$

$$\lambda^C_{[1,1]}(x_1 \dots x_t) = \begin{cases} 0 \text{ if an even number of } x_i \text{ is } 1, \\ 1 \text{ otherwise,} \end{cases}$$

$$\lambda^C_{[0,1]}(x_1 \dots x_t) = \lambda^C_{[1,0]}(x_1 \dots x_t),$$

$$\lambda^C_{[0,0]}(x_1 \dots x_t) = \lambda^C_{[1,1]}(x_1 \dots x_t).$$

Definition 6. A linear sequential circuit C is said to be *minimal* if and only if it does not have two different states y_1 and y_2 such that $\lambda^C_{y_1} = \lambda^C_{y_2}$.

Example 8. The linear sequential circuit of Figure 3 in [3] is not minimal.

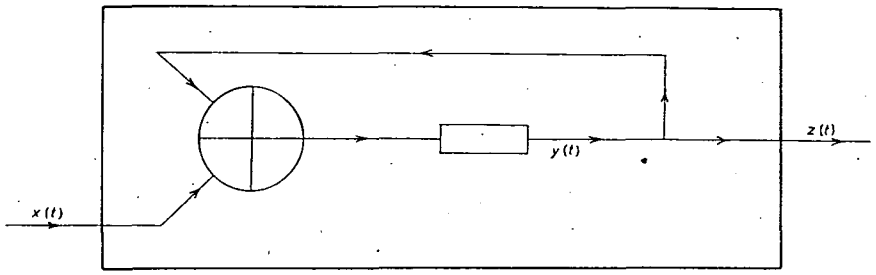


Fig. 1

Example 9. Let C be the linear sequential circuit of Figure 1.

$$\lambda^C_1(x_1 \dots x_t) = \begin{cases} 1 \text{ if an even number of } x_i \text{ is } 1, \\ 0 \text{ otherwise,} \end{cases}$$

$$\lambda^C_0(x_1 \dots x_t) = \begin{cases} 0 \text{ if an even number of } x_i \text{ is } 1, \\ 1 \text{ otherwise.} \end{cases}$$

Hence C is minimal.

Definition 7. Let C and C' be two linear sequential circuits. C and C' are said to be *equivalent* if and only if

$$\{\lambda^C_y | y \text{ is a state of } C\} = \{\lambda^{C'}_{y'} | y' \text{ is a state of } C'\}.$$

Intuitively speaking two linear sequential circuits are equivalent if and only if for every state y of C there is a state y' of C' such that C started in state y behaves the same way as C' started in y', and vice versa.

Example 10. The circuits in Examples 8 and 9 are equivalent.

3. Outline of the argument

Our aim is to provide an algorithm for the synthesis of linear sequential circuits. We are going to do this in the following way.

(a) We give an algorithm which for every regular expression produces a finite automaton such that the language described by the regular expression is the language accepted by the finite automaton.

(b) We prove that a language described by a regular expression is a linear regular language if and only if the initial subautomaton of the automaton produced by the algorithm in (a) is linearly realizable. In particular, we show that from the linear circuit which is the linear realization (if there is one) of the initial subautomaton, we can effectively produce a linear sequential circuit which accepts the language described by the regular expression.

(c) We give an algorithm which for any finite automaton decides whether or not its initial subautomaton is linearly realizable, and if it is, then the algorithm gives a linear sequential circuit which is a linear realization of it.

It is clear that in view of (b), the algorithms in (a) and (c) combine to give the algorithm required for the problem of synthesis for linear sequential circuits.

4. Synthesis for finite automata

Theorem 1. Let Σ be any fixed finite, non-empty set. There is an algorithm which for any given regular expression R over the alphabet Σ will produce a finite automaton $M=\langle Q, \Sigma, q, \delta, F\rangle$ such that M accepts the language $|R|$.

Proof. The algorithm builds up the finite automaton M from finite automata which it has already produced for parts of R. We shall describe what it does for regular expressions of length 1, and then show how it produces the finite automaton for a regular expression of length greater than 1 from finite automata for regular expressions of shorter length.

If $R=o$, then $M=\langle\{q\}, \Sigma, q, \delta, \varnothing\rangle$,

where $\delta(q, a)=q$ for all $a\in\Sigma$.

If $R=e$, then $M=\langle\{q_1, q_2\}, \Sigma, q_1, \delta, \{q_1\}\rangle$,

where $\delta(q, a)=q_2$ for $q\in\{q_1, q_2\}$ and $a\in\Sigma$.

If $R=a$ where $a\in\Sigma$, then $M=\langle\{q_0, q_1, q_2\}, \Sigma, q_0, \delta, \{q_1\}\rangle$, where $\delta(q_0, a)=q_1$ and $\delta(q, b)=q_2$ otherwise.

We now show how to produce from a finite automaton $D=\langle\bar{Q}, \Sigma, \bar{q}, \bar{\delta}, \bar{F}\rangle$ which accepts a regular language $|P|$ and a finite automaton $E=\langle\bar{\bar{Q}}, \Sigma, \bar{\bar{q}}, \bar{\bar{\delta}}, \bar{\bar{F}}\rangle$ which accepts a regular language $|S|$ (P and S are regular expressions), the finite automata which accept the regular languages $|(P+S)|$ and $|(PS)|$, respectively.

Suppose R is of the form $(P+S)$ and D and E are finite automata as defined above. The finite automaton $M=\langle Q, \Sigma, q, \delta, F\rangle$ which accepts $|R|$ is constructed as follows. $Q=\bar{Q}\times\bar{\bar{Q}}$, i.e. ordered pairs of elements from $\bar{Q}$ and $\bar{\bar{Q}}$. $q=\langle\bar{q}, \bar{\bar{q}}\rangle$.

For any $a\in\Sigma$, $p\in\bar{Q}$ and $r\in\bar{\bar{Q}}$,

$$\delta(\langle p, r\rangle, a) = \langle \bar{\delta}(p, a), \bar{\bar{\delta}}(r, a)\rangle,$$

$$F = \{\langle p, r\rangle | p\in\bar{F} \quad \text{or} \quad r\in\bar{\bar{F}}\}.$$

We leave it to the reader to show that M accepts $|(P+S)|$.

Suppose R is of the form (PS) and $D=\langle \bar{Q}, \Sigma, \bar{q}, \bar{\delta}, \bar{F}\rangle$ and $E=\langle \bar{\bar{Q}}, \Sigma, \bar{\bar{q}}, \bar{\bar{\delta}}, \bar{\bar{F}}\rangle$ are finite automata which accept $|P|$ and $|S|$, respectively. The finite automaton $M=\langle Q, \Sigma, q, \delta, F\rangle$ which accepts $|R|$ is constructed as follows.

Let Q' denote the set of all subsets of $\bar{\bar{Q}}$. $Q = \bar{Q}\times Q'$, i.e. a set of ordered pairs, where the first element is a state of D and the second element is a subset of the states of E. $q=\langle \bar{q}, \varnothing\rangle$ if $\bar{q}\notin\bar{F}$ and $q=\langle \bar{q}, \{\bar{\bar{q}}\}\rangle$ if $\bar{q}\in\bar{F}$. δ is defined as follows. For any $p\in\bar{Q}$, $Q_1\subset\bar{\bar{Q}}$ and $a\in\Sigma$,

$$\delta(\langle p, Q_1\rangle, a) = \langle \bar{\delta}(p, a), Q_2\rangle,$$

where

$$Q_2 = \{r | r = \bar{\bar{\delta}}(s, a) \quad \text{for some} \quad s\in Q_1\}\cup Q_3,$$

where

$$Q_3 = \varnothing \quad \text{if} \quad \bar{\delta}(p, a)\in\bar{F} \quad \text{and} \quad Q_3 = \{\bar{\bar{q}}\} \quad \text{if} \quad \bar{\delta}(p, a)\in\bar{F}.$$

Finally,

$$F = \{\langle p, Q_1\rangle | p\in\bar{Q}, Q_1\subset\bar{\bar{Q}} \quad \text{and} \quad Q_1\cap\bar{\bar{F}}\neq\varnothing\}.$$

We now have to show that the finite automaton $F=\langle Q, \Sigma, q, \delta, F\rangle$ does indeed accept $|(PS)|$. The essence of the proof is the following. Given a word w of I_Σ, there may be many ways of breaking up w into two subwords. $w\in|(PS)|$ if and only if one of these ways is such that $w=w_1w_2$ and $w_1\in|P|$ and $w_2\in|S|$. M keeps track simultaneously of all the possible states which E might be in depending on the way w has been broken up.

In order to complete this part of the proof, it is sufficient to prove the following claim.

For any word $w\in I_\Sigma$,

$$\delta(q, w) = \langle \bar{\bar{\delta}}(\bar{q}, w), Q_1\rangle,$$

where $r\in Q_1$ if and only if there exist w_1 and w_2 in I_Σ such that $w=w_1w_2$, $w_1\in|P|$ and $\bar{\bar{\delta}}(\bar{\bar{q}}, w_2)=r$.

We leave the proof, which can best be done by induction on the length of w, to the reader.

If R is of the form P^* and $E=\langle \bar{Q}, \Sigma, \bar{q}, \bar{\delta}, \bar{F}\rangle$ is a finite automaton which accepts $|P|$, $M=\langle Q, \Sigma, q, \delta, F\rangle$ is constructed as follows.

Let $\bar{\bar{q}}$ be such that $\bar{\bar{q}}\notin\bar{Q}$. Let $\bar{\bar{Q}} = \bar{Q}\cup\{\bar{\bar{q}}\}$. Define $\bar{\bar{\delta}}: \bar{\bar{Q}}\times\Sigma \to \bar{Q}$ by

$$\bar{\bar{\delta}}(q, a) = \begin{cases} \bar{\delta}(q, a) & \text{if} \quad q\in\bar{Q} \quad \text{and} \quad a\in\Sigma, \\ \bar{\delta}(\bar{q}, a) & \text{if} \quad q=\bar{\bar{q}} \quad \text{and} \quad a\in\Sigma. \end{cases}$$

Q is the set of all subsets of $\bar{\bar{Q}}$. $q=\{\bar{\bar{q}}\}$. For any $Q_1\subset\bar{\bar{Q}}$ and $a\in\Sigma$,

$$\delta(Q_1, a) = \{r | r = \bar{\bar{\delta}}(s, a) \quad \text{for some} \quad s\in Q_1\}\cup Q_2$$

where $Q_2=\{\bar{q}\}$ if $\bar{\bar{\delta}}(s, a)\in\bar{F}$ for some $s\in Q_1$ and $Q_2=\varnothing$ otherwise. Finally,

$$F=\{Q_1 | Q_1\subset\bar{\bar{Q}} \quad \text{and} \quad \bar{\bar{q}}\in Q_1\}.$$

The proof that M accepts $|P^*|$ is very similar to the proof outlined above and we leave it to the reader.

Even though this theorem proves that the problem of synthesis is solved for finite automata, the algorithm described in it is such that the finite automata produced will in general be far from the simplest of the ones which do the job required. In fact, it will generally be so large that the implementation of the algorithm on an actual computing device is beyond the realm of practical possibility in all but the simplest cases. For instance, for the regular expression in Example 3 the finite automaton produced by the algorithm will have more than 4×10^{12} states, and the finite automaton for the regular expression in Example 4 will have over $10^{10^{10^{10}}}$ states. At the same time, both of the expressions denote languages that can be accepted by fairly simple machines (see Examples 1 and 2).

Although one could give algorithms which work faster than the one described above, there is no existing algorithm which works so fast as to be implementable for non-trivial regular expressions.

5. Some facts about linear sequential circuits and linear realizations

Theorem 2. Let $M=\langle Q, \Sigma, q, \delta, F\rangle$ be a finite automaton and C be a linear sequential circuit which is a linear realization of M with functions α and φ (see Definition 3). Let C be described by the matrices **A**, **B** and **C**. Then, for all $p\in Q$ and $x_1x_2\dots x_t$ in I_Σ,

$$\varphi\big(\delta(p, x_1x_2\dots x_t)\big)=\varphi(p)\mathbf{A}^t\oplus\sum_{i=1}^{t}\alpha(x_i)\mathbf{B}\mathbf{A}^{t-i}.$$

Furthermore,

$$\lambda^C_{\varphi(p)}\big(\alpha(x_1)\alpha(x_2)\dots\alpha(x_t)\big)=1 \text{ if and only if } \delta(p, x_1x_2\dots x_t)\in F.$$

Proof by induction on t.

If $t=0$,

$$\varphi\big(\delta(p, e)\big)=\varphi(p)=\varphi(p)\mathbf{A}^0.$$

Assume that the theorem is true for all words of length t. Then

$$\varphi\big(\delta(p, x_1x_2\dots x_tx_{t+1})=\varphi\big(\delta(\delta(p, x_1x_2\dots x_t), x_{t+1})\big)=$$

$$=\varphi\big(\delta(p, x_1x_2\dots x_t)\big)\mathbf{A}\oplus\alpha(x_{t+1})\mathbf{B}=\left(\varphi(p)\mathbf{A}^t\oplus\sum_{i=1}^{t}\alpha(x_i)\mathbf{B}\mathbf{A}^{t-i}\right)\mathbf{A}\oplus\alpha(x_{t+1})\mathbf{B}=$$

$$=\varphi(p)\mathbf{A}^{t+1}\oplus\sum_{i=1}^{t+1}\alpha(x_i)\mathbf{B}\mathbf{A}^{t+1-i}.$$

The second part of the theorem follows directly from Definitions 3 and 5.

Theorem 3. Let C be a linear sequential circuit with n delays. Let y_1 and y_2 be any two states of C. If there exists a sequence $x_1, x_2, \ldots, x_{n-1}$ of external input conditions such that

$$\lambda^C_{y_1}(x_1 x_2 \ldots x_t) = \lambda^C_{y_2}(x_1 x_2 \ldots x_t).$$

for all t, $0 \leqq t \leqq n-1$, then

$$\lambda^C_{y_1} = \lambda^C_{y_2}.$$

(This is sometimes expressed by saying that C satisfies the n-diagnosability condition.)

Proof. Assume that C can be described by matrices **A**, **B** and **C**. If $\lambda^C_{y_1}(x_1 x_2 \ldots x_t) = \lambda^C_{y_2}(x_1 x_2 \ldots x_t)$, then

$$y_1 \mathbf{A}^t \mathbf{C} \oplus \sum_{i=1}^{t} x_i \mathbf{B} \mathbf{A}^{t-i} \mathbf{C} = y_2 \mathbf{A}^t \mathbf{C} \oplus \sum_{i=1}^{t} x_i \mathbf{B} \mathbf{A}^{t-i} \mathbf{C}$$

and so

$$y_1 \mathbf{A}^t \mathbf{C} = y_2 \mathbf{A}^t \mathbf{C}. \tag{1}$$

Since this is true for $0 \leqq t \leqq n-1$, it is also true for $t \geqq n$. This follows from the fact (Cayley—Hamilton theorem) that any power of **A** can be expressed as a linear combination of powers of **A** less than n. Thus we have that (1) holds for every t. But then for any sequence $x_1 x_2 \ldots x_t$,

$$\lambda^C_{y_1}(x_1 x_2 \ldots x_t) = \lambda^C_{y_2}(x_1 x_2 \ldots x_t)$$

and so $\lambda^C_{y_1} = \lambda^C_{y_2}$.

Theorem 4. For every linear sequential circuit C there exists a minimal linear sequential circuit C' which is equivalent to C.

Proof. Assume that C is described by the matrices **A**, **B** and **C**. (**A** is $n \times n$, **B** is $k \times n$ and **C** is $n \times 1$.) Let **K** be the $n \times n$ matrix

$$[\mathbf{C}, \mathbf{AC}, \ldots, \mathbf{A}^{n-1}\mathbf{C}].$$

First we prove that C is minimal if and only if **K** is a non-singular matrix.

C is not minimal

if and only if

there exist y_1 and y_2 such that $\lambda^C_{y_1} = \lambda^C_{y_2}$ and $y_1 \neq y_2$,

if and only if

there exist y_1 and y_2 such that $y_1 \neq y_2$ and

$$y_1 \mathbf{A}^t \mathbf{C} = y_2 \mathbf{A}^t \mathbf{C} \quad \text{for} \quad 0 \leqq t \leqq n-1,$$

if and only if

there exist y_1 and y_2 such that $y_1 \neq y_2$ and

$$y_1 \mathbf{K} = y_2 \mathbf{K}$$

if and only if

K is singular.

So if **K** is non-singular, then C is already minimal and there is nothing to prove. Let us therefore assume that **K** is singular.

Now we pick a row of **K** which is the linear combination of previous rows. Since the underlying field is the field of two elements this means that the particular row of **K**, say the j'th, is the modulo 2 sum of some of the earlier rows. What is the physical significance of the j'th row of K being the sum of previous rows? It is that the j'th delay in the circuit is superfluous, since its behaviour can be obtained from the output of the other delays.

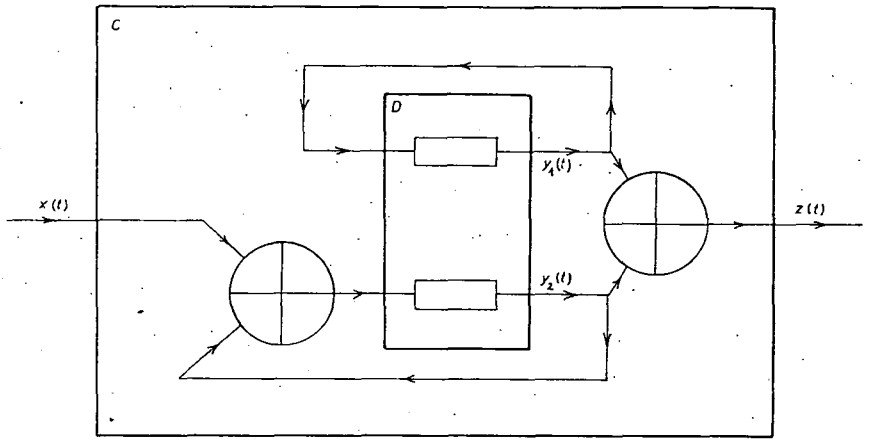


Fig. 2

To see this consider the following argument. We separate the delays of a linear sequential circuit C by considering them to form a separate circuit D with n external input wires and n external output wires. For instance for the circuit of Figure 3 in [3], this would give us Figure 2. For $1 \leqq j \leqq n$, the j'th external output of D at time $t+1$ is the same as the j'th external input of D at time t. Another way of looking at this, is that D takes a state of C as an input and returns the same state of C as an output one unit of time later.

Now suppose that we have a linear sequential circuit C' which is the same as C, except that D is replaced by a circuit D' with n external inputs and n external outputs, such that if D' is given a state y_1 of C as an input, then it returns one unit of time later state y_2 of C such that

$$\lambda^C_{y_1} = \lambda^C_{y_2}.$$

Clearly, such a C' is equivalent to C.

We are now going to show that if **K** is singular, we can always find such a D' with only $n-1$ delays. One of the many possible ways of constructing such a D' is the following.

Suppose the j'th row is the first row of **K** which is the sum of some of the previous rows. Let us denote the rows of **K** by $\mathbf{K}_1, \mathbf{K}_2, \ldots, \mathbf{K}_n$ and let S be that subset of $\{1, \ldots, j-1\}$ such that

$$\mathbf{K}_j = \sum_{i \in S} \mathbf{K}_i.$$

If $j \leqq i \leqq n-1$, the input wire of the i'th delay of D' is connected to the $(i+1)$'st external input wire to D', and the output wire of the i'th delay of D' is connected to the $(i+1)$'st external output wire of D'. If $1 \leqq i \leqq j-1$ the output wire of the i'th delay of D' is connected to the i'th external output wire of D'. If $i \notin S$, the input wire of the i'th delay of D' is connected to the i'th external input wire of D'. If $i \in S$, the input wire of the i'th delay of D' is connected to the output wire of an exclusive or gate whose input wires are the i'th and j'th external input wires to D'. Note that the j'th external output wire of D' is not connected to anything and so it never carries a pulse (it is earthed).

If the input to D' is the state $y_1 = [a_1, a_2, \ldots, a_n]$, then the output to D' one unit of time later will be

$$y_2 = [a_1 + b_1, a_2 + b_2, \ldots, a_{j-1} + b_{j-1}, 0, a_{j+1}, \ldots, a_n],$$

where

$$b_i = \begin{cases} 0 & \text{if } i \notin S, \\ a_j & \text{if } i \in S. \end{cases}$$

In order to show that $\lambda^C_{y_1} = \lambda^C_{y_2}$, it is sufficient to prove that $y_1\mathbf{K} = y_2\mathbf{K}$ (see beginning of this proof).

$$y_2\mathbf{K} = [a_1 + b_1, a_2 + b_2, \ldots, a_{j-1} + b_{j-1}, 0, a_{j+1}, \ldots, a_n] \begin{bmatrix} \mathbf{K}_1 \\ \mathbf{K}_2 \\ \vdots \\ \mathbf{K}_n \end{bmatrix} =$$

$$= \sum_{i=1}^{j-1} a_i\mathbf{K}_i + \sum_{i=1}^{j-1} b_i\mathbf{K}_i + \sum_{i=j+1}^{n} a_i\mathbf{K}_i =$$

$$= \sum_{i=1}^{j-1} a_i\mathbf{K}_i + \sum_{i \in S} a_j\mathbf{K}_i + \sum_{i=j+1}^{n} a_i\mathbf{K}_i =$$

$$= \sum_{i=1}^{j-1} a_i\mathbf{K}_i + a_j\mathbf{K}_j + \sum_{i=j+1}^{n} a_i\mathbf{K}_i =$$

$$= \sum_{i=1}^{n} a_i\mathbf{K}_i =$$

$$= y_1\mathbf{K}.$$

We have now obtained a circuit C' which is equivalent to C and has fewer delays than C. If this circuit is not minimal, then we can repeat this process. Since C has only a finite number of delays, sooner or later we must find a minimal C' which is equivalent to C.

Example 11. Consider the circuit C of Figure 2. For this circuit (see Example 5 in [3]).

$$\mathbf{A} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{C} = \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

$$\mathbf{K} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}, \quad \mathbf{K}_1 = \mathbf{K}_2 = [1, 1].$$

So in the method described above $j=2$ and $S=\{1\}$. This gives us the D' and C' shown in Figure 3.

It is easy to see that this circuit is equivalent to the one in Figure 1 ($y_2(t)$ never carries a pulse).

We point out by the way, that the proof of Theorem 4 is constructive. Given a linear sequential circuit, we can actually construct a minimal linear sequential circuit equivalent to it using the proof of Theorem 4.

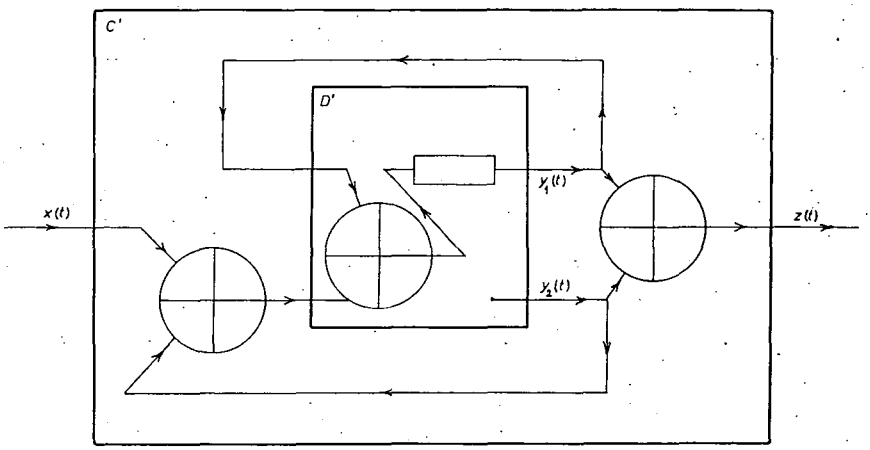


Fig. 3

6. The equivalence of linear realizability of finite automata and linearity of regular expressions

Theorem 5. Let R be a regular expression and M be a finite automaton which accepts $|R|$. Then $|R|$ is a linear regular language if and only if the initial subautomaton of M is linearly realizable. Furthermore, if the initial subautomaton of M is linearly realizable by a linear sequential circuit C, then from C and the mappings φ and α we can effectively produce a linear sequential circuit C' and a function f such that C' accepts the language $|R|$ using f.

Proof. First suppose that the initial subautomaton of $M=\langle Q, \Sigma, q, \delta, F\rangle$ is linearly realizable by a linear sequential circuit C. Let α and φ denote the mappings described in Definition 3.

We now construct a circuit C' which accepts the language $|R|$ using α as the f of Definition 9 in [3].

Consider $\varphi(q)=[y_1, \ldots, y_n]$. We consider each delay of C in turn. If $y_i=0$, we make no alteration to the i'th delay of C. If $y_i=1$, then we connect the output wire of the i'th delay of C to one of the input wires of an exclusive or gate (newly introduced for this purpose) and we connect the output wire of the exclusive or gate to all the wires to which the output wire of the delay used to be connected (all

the old connections being removed). The other input wire of the exclusive or gate is not yet connected up. When we changed all delays in this way, we introduce one more delay (to be considered the first delay of C') whose input wire is not connected to anything (i.e., it is earthed) and whose output wire is connected to the so far free input wires of the newly introduced exclusive or gates. No other alteration is done to C to obtain C'.

For example if C is as in Figure 3 of [3], then C' is given in Figure 4.

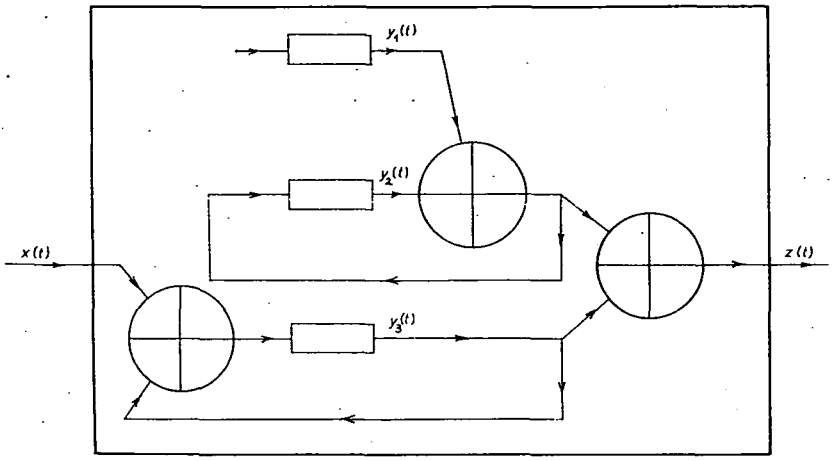


Fig. 4

After time $t=1$, the first delay of C' makes no contribution to the behaviour of C', hence we have that

$$\lambda^{C'}_{[1,0,\ldots,0]} = \lambda^{C}_{\varphi(q)}.$$

This together with Theorem 2 shows that if we let the function f of Definition 9 in [3] to be α, then a word w in I_Σ is accepted by M if and only if it is accepted by C'. In particular, we have shown that $|R|$ is a linear regular language.

Conversely, let us assume that $|R|$ is a linear regular language and let C be a linear sequential circuit which accepts $|R|$. Let C' be a minimal linear sequential circuit which is equivalent to C. We shall prove that C' is a linear realization of the initial subautomaton of M.

Since the number of external input wires for C and C' are the same, we can take α of Definition 3 to be the f of Definition 9 in [3]. φ is defined as follows.

For every state y of C there is one state y' of C' such that

$$\lambda^{C}_{y} = \lambda^{C'}_{y'}$$

(otherwise C and C' would not be equivalent). Furthermore, there is only one such y', for otherwise C' would not be minimal. Let $\mu(y)$ denote this unique y'.

Given a state p of the initial subautomaton of M, there exists an $x = x_1 x_2 \ldots x_t$ in I_Σ such that

$$p = \delta(q, x_1 x_2 \ldots x_t).$$

We define (using the matrices $\mathbf{A}'$, $\mathbf{B}'$ and $\mathbf{C}'$ which describe C')

$$\varphi(p) = \mu[1, 0, \ldots, 0]\mathbf{A}'^{t} \oplus \sum_{i=1}^{t} \alpha(x_i)\mathbf{B}'\mathbf{A}'^{t-i}.$$

The difficulty with this definition is that it is not necessarily unique. It is possible that for some $\bar{x} = \bar{x}_1\bar{x}_2 \ldots \bar{x}_{\bar{t}}$ in I_Σ, $x \neq \bar{x}$ and yet $\delta(q, x) = \delta(q, \bar{x})$. We must show that in such cases

$$(2) \qquad \mu[1, 0, \ldots, 0]\mathbf{A}'^{t} \oplus \sum_{i=1}^{t} \alpha(x_i)\mathbf{B}'\mathbf{A}'^{t-i} = \mu[1, 0, \ldots, 0]\mathbf{A}'^{\bar{t}} \oplus \sum_{i=1}^{\bar{t}} \alpha(\bar{x}_i)\mathbf{B}'\mathbf{A}'^{\bar{t}-i}.$$

Let us denote the left hand side of (2), which is a state of C', by y_1 and the right hand side of (2) by y_2. Since C' is a minimal linear sequential circuit, to show the equality, it is sufficient to prove that

$$\lambda^{C'}_{y_1} = \lambda^{C'}_{y_2}.$$

We are now going to show that for any sequence $s_1 s_2 \ldots s_j \in I_\Sigma$

$$\lambda^{C'}_{y_1}\big(\alpha(s_1)\alpha(s_2) \ldots \alpha(s_j)\big) = \lambda^{C'}_{y_2}\big(\alpha(s_1)\alpha(s_2) \ldots \alpha(s_j)\big).$$

In view of Theorem 3, this is sufficient.

$$\lambda^{C'}_{y_1}\big(\alpha(s_1) \ldots \alpha(s_j)\big) =$$

$$= y_1\mathbf{A}'^{j}\mathbf{C}' \oplus \sum_{i=1}^{j} \alpha(s_i)\mathbf{B}'\mathbf{A}'^{j-i}\mathbf{C}' =$$

$$= \mu[1, 0, \ldots, 0]\mathbf{A}'^{t+j}\mathbf{C}' \oplus \sum_{i=1}^{t} \alpha(x_i)\mathbf{B}'\mathbf{A}^{t+j-i}\mathbf{C}' \oplus \sum_{i=1}^{j} \alpha(s_i)\mathbf{B}'\mathbf{A}'^{j-1}\mathbf{C}' =$$

$$= \lambda^{C'}_{\mu[1,0,\ldots,0]}\big(\alpha(x_1) \ldots \alpha(x_t)\alpha(s_1) \ldots \alpha(s_j)\big) =$$

$$= \lambda^{C}_{[1,0,\ldots,0]}\big(\alpha(x_1) \ldots \alpha(x_t)\alpha(s_1) \ldots \alpha(s_j)\big) =$$

$$= \begin{cases} 1 \text{ if } x_1 \ldots x_t s_1 \ldots s_j \in |R|, \\ 0 \text{ otherwise} \end{cases}$$

$$= \begin{cases} 1 \text{ if } \delta(q, x_1 \ldots x_t s_1 \ldots s_j) \in F, \\ 0 \text{ otherwise} \end{cases}$$

$$= \begin{cases} 1 \text{ if } \delta\big(\delta(q, x_1 \ldots x_t), s_1 \ldots s_j\big) \in F, \\ 0 \text{ otherwise} \end{cases}$$

$$= \begin{cases} 1 \text{ if } \delta(p, s_1 \ldots s_j) \in F, \\ 0 \text{ otherwise} \end{cases}$$

$$= \begin{cases} 1 \text{ if } \delta\big(\delta(q, \bar{x}_1 \ldots \bar{x}_{\bar{t}}), s_1 \ldots s_j\big) \in F, \\ 0 \text{ otherwise} \end{cases}$$

$$= \lambda^{C'}_{y_2}\big(\alpha(s_1) \ldots \alpha(s_j)\big).$$

All that is left to show is that conditions (iii) and (iv) in Definition 3 are satisfied.

For any p in the initial subautomaton of M and for any a in Σ, $p=\delta(q, x)$ for some $x=x_1 \ldots x_t$ in I_Σ and

$$\begin{aligned}\varphi(\delta(p, a)) &= \varphi(\delta(\delta(q, x), a)) = \\ &= \varphi(\delta(q, xa)) = \\ &= \mu[1, 0, \ldots, 0]\mathbf{A}'^{t+1} \oplus \sum_{i=1}^{t} \alpha(x_i)\mathbf{B}'\mathbf{A}'^{t+1-i} \oplus \alpha(a)\mathbf{B}' = \\ &= \left(\mu[1, 0, \ldots, 0]\mathbf{A}'^{t} \oplus \sum_{i=1}^{t} \alpha(x_i)\mathbf{B}'\mathbf{A}'^{t-i}\right)\mathbf{A}' \oplus \alpha(a)\mathbf{B}' = \\ &= \varphi(\delta(q, x))\mathbf{A}' \oplus \alpha(a)\mathbf{B}' = \\ &= \varphi(p)\mathbf{A}' \oplus \alpha(a)\mathbf{B}'.\end{aligned}$$

$$\varphi(p)\mathbf{C}' = 1$$

if and only if

$$\left(\overset{\circ}{\mu}[1, 0, \ldots, 0]\mathbf{A}'^{t} \oplus \sum_{i=1}^{t} \alpha(x_i)\mathbf{B}'\mathbf{A}'^{t-1}\right)\mathbf{C}' = 1$$

if and only if

$$C \text{ accepts } x$$

if and only if

$$x \in |R|$$

if and only if

$$M \text{ accepts } x$$

if and only if

$$\delta(q, x) \in F$$

if and only if

$$p \in F.$$

Corollary. There are finite automata which are not linearly realizable.

Proof. The automaton of Example 1 is such. This is because this automaton accepts the language described in Example 3, and this language is not a linear regular language. (See Theorem 3 of [3].)

7. The linear realizability of finite automata

Theorem 6. If a linearly realizable finite automaton $M=\langle Q, \Sigma, q, \delta, F\rangle$ has n' states and k' symbols in Σ, then it is linearly realizable by a linear sequential circuit with at most k' external input wires and at most n' delays.

Proof. Suppose M has a linear realization C with k external input wires and n delays, where $k>k'$ or $n>n'$ or both. Suppose α and φ are the mappings as in Definition 3.

First we deal with the case when $n>n'$.

Let $q_1, \ldots, q_{n'}$ be the n' states of M. Consider the $n' \times n$ matrix **F** whose i'th row is $\varphi(q_i)$. At least $n-n'$ columns of this matrix **F** will be linearly dependent on the other n' columns, and by relabeling the delays we may assume that it is the last $n-n'$ columns.

Now we proceed as in the proof of Theorem 4. We consider the subcircuit D with n external inputs and n external outputs which contains all the delays and nothing else. We replace this by a circuit D' with n' delays. For $1 \leqq i \leqq n'$, the input wire to the i'th delay is connected to the i'th external input wire to D', and the output wire of the i'th delay is connected to the i'th external output wire of D'.

The other external input wires are not connected to anything. For $n' < i \leqq n$, the i'th external output wire of the D' is connected to the delays through exclusive or gates in such a way that its output is the modulo 2 sum of the outputs of those delays which correspond to those of the first n' columns of **F** which added together give the i'th column.

The circuit C' which we get by replacing D in C by D' is clearly equivalent to C and is also a linear realization of M with functions $\alpha' = \alpha$ and φ' such that $\varphi'(p)$ is the vector consisting of the first n' elements of $\varphi(p)$ (after relabeling of the delays of C).

The case when $k > k'$ can be similarly taken care of. C is altered by attaching in front of it a circuit consisting of exclusive or gates only (no delays) which has k' external input wires and k external output wires, the latter being attached to the external input wires of C. The exact nature of this additional circuit is determined by the linear dependencies between the columns of the matrix whose i'th row is $\alpha(s_i)$, where s_i is the i'th element of Σ.

Theorem 7. There is an algorithm which for any finite automaton decides whether or not its initial subautomaton is linearly realizable, and, in case it is, the algorithm gives a linear sequential circuit which is a linear realization of it.

Proof. First of all, it should be obvious that there is an algorithm which from a given automaton produces its initial subautomaton. Let us assume that this initial subautomaton has n states and k symbols in its alphabet. If it is linearly realizable, then it has a linear realization with n delays and k external input wires. (At most n or k, by Theorem 6, if less, then additional delays and external input wires can be introduced and earthed.) This linear realization can be described by an $n \times n$ matrix **A**, a $k \times n$ matrix **B** and a $n \times 1$ matrix **C** of 0's and 1's. Furthermore given three such matrices we can easily produce a linear sequential circuit which is described by them, and any two circuits described by them will be equivalent and be linear realizations of the same finite automata. (Only the matrices enter the definition of equivalence and linear realization.) Also, given matrices **A**, **B** and **C** it is easy to check (using Definition 3) whether or not the circuit described by them is a linear realization of a given automaton.

So our algorithm will look like this. Try all possible $n \times n$ matrices **A**, $k \times k$ matrices **B** and $n \times 1$ matrices **C** (there will be $2^{n(n+k+1)}$ possibilities). Check one by one whether the circuits described by them are linear realizations of the initial subautomaton of the given automaton. If we find such **A**, **B** and **C**, then our work is done, if we exhaust all possibilities without finding them, then the initial subautomaton is not linearly realizable.

There are more efficient algorithms than the one described above to do what is required in Theorem 7. (One such could be based on the method of Cohn & Even [1].) However, in view of the comments after Theorem 1, it is clear that our total algorithm for the synthesis of linear sequential circuits cannot be made practicable and so we sacrificed efficiency for ease of proof in Theorem 7.

8. Conclusions

The algorithms described in Theorems 1, 5 and 7 together provide us with an algorithm for the synthesis of linear sequential circuits. However, this is a very roundabout as well as inefficient way of doing things, and the possibility of a direct synthesis from regular expressions to circuits remains an intriguing open problem.

In this direction, the reader may find useful two books related to linear sequential circuits which appeared since the writing of Part I. These are [2] and [6].

DEPT. OF COMPUTER SCIENCE
STATE UNIVERSITY OF NEW YORK
AT BUFFALO

9. References

[1] COHN, M. & S. EVEN, Identification and minimization of linear machines, *IEEE Trans.*, v. EC-14, 1965, pp. 367—376.
[2] HARRISON, M. A., *Lectures on Linear Sequential Machines*, Academic Press, New York, 1970.
[3] HERMAN, G. T., Linear regular languages, Part I, *Acta Cybernetica*, v. 1, 1969, pp. 3—11.
[4] HERMAN, G. T., Analysis of linear sequential circuits, *Proceedings of the Fourth Annual Princeton Conference on Information Sciences and Systems*, 1970, pp. 392—396.
[5] HERMAN, G. T., When is a sequential machine the realization of another? *Mathematical Systems Theory*, v. 5, 1971, pp. 115—127.
[6] REUSCH, B., *Lineare Automaten*, Bibliographisches Institut, Mannheim, 1969.

(Received June 24, 1970)

Generation of the k-trees of a graph

By I. PÁVÓ

Abstract

We present a new procedure that generates all k-trees of a graph in each component of which a vertex is given in advance. Our method makes use of a theorem of Ore [7] concerning finite directed graphs, thus providing an application of this theorem that beyond its theoretical interest can be used in practical analysis of electrical networks.

Introduction

Topological formulas are nowadays playing an ever increasing role in the analysis of electrical networks (see [8]). In the applications of such formulas, however simple they are, the question immediately arises how to generate all the trees, and also the 2-trees satisfying certain requirements, of a given graph.

To overcome this problem several methods were proposed in the last decade. In principle, the most simple way to produce all the trees of a graph G with n vertices would be to scan all the sets containing $n-1$ edges and dispose of those not eligible. Naturally for practical purposes such a procedure would be too lengthy and intricate. A procedure usable also in practice was devised by Hakimi and Green [1] and solves the problem by splitting the graph in two parts the trees of which are assumed to be known. From the trees of these subgraphs the trees of the starting one can be composed and also k-trees satisfying certain requirements can be pinpointed. This procedure is, however, also lengthy and cumbersome; indeed, to carry out the splitting and composition of trees is in itself a complicated algorithm, and it must be repeated also for the subgraphs obtained. A similar procedure was devised by Mayeda [3].

Other procedures were designed by Talbot (a new set of topological formulas) and by Mayeda and Seshu [4]. A common feature of these methods is that they choose an arbitrary tree as a starting one and generate the others from this by edge transformations. (After deleting an edge of the starting tree another edge of the graph is substituted, and then the same procedure is applied to the obtained tree.) These methods seem to have certain advantages but we must note that a recursive formula is used that is, from the viewpoint of computational technique, difficult to handle.

The most feasible method for practical use that has been developed up to now

is perhaps that due to Maxwell and Cline [5]. This method is of algebraic character and, to its advantage, is simple, easy to understand, and adaptable to digital computers relatively easily. A snag is, however, that it can be used only to generate k-trees with $k=1$ or 2 and it uses up a relatively large internal storage capacity.

In the present paper we are going to present a new procedure that generates the k-trees, satisfying certain requirements, of a graph; the procedure in the case $k=1$ generates all the trees. Our method is about as simple as the algebraic method mentioned above is, but it can be applied under more general circumstances; namely, it can be used to handle the case $k \geqq 2$ too. It can also be ascribed to its advantage that, fed into computer, it needs considerably less internal storage room than the algebraic method mentioned. The procedure, as presented here, concerns only the generation of k-trees of graphs without multiple edges, but there is, in principle, no difficulty in extending it so as to apply to graphs with multiple edges. Our considerations are based on a well-known theorem of Ore [7] on finite directed graphs.

1. Basic concepts and definitions

Consider a graph with n vertices $\mathbf{P}_1, \ldots, \mathbf{P}_n$ and select arbitrarily a number k $(1 \leqq k \leqq n)$ from among them, these being denoted by $\mathbf{P}_{i_1}, \ldots, \mathbf{P}_{i_k}$ $(1 \leqq i_1 < \cdots < i_k \leqq n)$.

Definition. A *k-tree* $F^k_{i_1, \ldots, i_k}$ of the graph G is an arbitrary graph satisfying the following stipulations:

1. it is a subgraph of G,
2. it contains all the vertices of G,
3. it consists of exactly k connected components,
4. each component contains exactly one of the selected vertices $\mathbf{P}_{i_1}, \ldots, \mathbf{P}_{i_k}$,
5. each of its components are a tree.

In particular, as seen from the definition, $F^1_{i_1}$ denotes a tree of the graph G; for the sake of simplicity we may sometimes drop the lower index and write only F^1. The purpose of this note is to study the generation of k-trees conforming to the above definition.

Let $\mathbf{M}$ be a matrix of size $n \times n$ with the following properties:

(I) Every element of $\mathbf{M}$ equals either 0 or 1,

(II) each row of $\mathbf{M}$ contains at least one element equal to 1.

Consider also a matrix $\mathbf{M}_{i_1, \ldots, i_k}$, where $1 \leqq i_1 < \cdots < i_k \leqq n$, of size $n \times n$ with the following properties:

(I′) the i_j-th row of $\mathbf{M}_{i_1, \ldots, i_k}$ consists purely of zeros $(j=1, \ldots, k)$,

(II′) in other rows of $\mathbf{M}_{i_1, \ldots, i_k}$ there is exactly one element equal to 1 and otherwise these rows too consist purely of zeros, and finally

(III′) at every place where $\mathbf{M}_{i_1, \ldots, i_k}$ contains a 1 the matrix $\mathbf{M}$ too does so.

Definition. If $\mathbf{M}_{i_1, \ldots, i_k}$ satisfies conditions (I′), (II′) and (III′) then we call it an *$(i_1, \ldots, i_k)$-reduction of* $\mathbf{M}$.

Suppose $\mathbf{M}_{i_1, \ldots, i_k}$ satisfies conditions (I′) and (II′), and write $\mathbf{M}_{i_1, \ldots, i_k} = (a_{ij})$, a_{ij} being the element in the intersection of the ith row and jth column of the matrix in question.

Definition. The function

$$\varphi(x) = \begin{cases} j \text{ if } x \neq i_1, \ldots, i_k \text{ and } a_{xj} = 1, \\ 0 \text{ if } x \text{ coincides with one of} \\ \text{the numbers } i_1, \ldots, i_k \end{cases}$$

is called *the function associated with the matrix* $\mathbf{M}_{i_1, \ldots, i_k}$.

As seen, the domain of $\varphi(x)$ is the set $\{1, \ldots, n\}$ and its range is a subset of $\{0, 1, \ldots, n\}$. Shortly, this function makes correspond to each row index an integer equalling zero unless the row with the considered index contains an element equal to 1, in which latter case this integer coincides with the column index of this unique element. It is clear that the correspondence between the matrices $\mathbf{M}_{i_1, \ldots, i_k}$ and the functions associated with them is one-to-one.

This observation is important since it shows that it is possible to characterize the matrix in question with the aid of the row vector $(\varphi(1), \ldots, \varphi(n))$. This characterization will be called *the row vector representation of the matrix* $\mathbf{M}_{i_1, \ldots, i_k}$.

In what follows, both directed and undirected graphs may turn up. Unless otherwise stated, loops are not, in general, admitted. In addition, undirected graphs are assumed to have no multiple edges. Undirected edges will be viewed as pairs of edges directed in opposite directions that connect the same pair of vertices.

To a graph containing the vertices $\mathbf{P}_1, \ldots, \mathbf{P}_n$ make correspond a matrix $\mu(G)$ of size $n \times n$ that in the intersection of the ith row and the jth column contains a 1 if and only if the vertices $\mathbf{P}_i$ and $\mathbf{P}_j$ in this order are connected by a directed edge, the remaining elements of the matrix being equal to zero.

Definition. The matrix $\mu(G)$ is said to be *the adjacency matrix* of the graph G.

It is easily checked that the correspondance $G \to \mu(G)$ between graphs, containing the vertices $\mathbf{P}_1, \ldots, \mathbf{P}_n$, that have only single edges (loops being allowed too) and matrices of type $n \times n$ containing only 0 or 1 as elements is one-to-one. For graphs without loops the adjacency matrix contains purely zeros in its diagonal. Undirected graphs have symmetrical adjacency matrices.

By a well-known theorem of Ore ([7], [6], [2]), a directed graph, possibly with loops, has the property that at each of its vertices exactly one of the edges is directed outwards if and only if it satisfies the following three conditions:

(a) each component of the graph contains exactly one circuit (this may possibly be a loop),

(b) in this unique circuit the edges are directed cyclically,

(c) in each component the edges that do not belong to its circuit are directed towards this circuit.

Definition. An undirected graph without loops is called *a generalized tree* if each of its connected components contains at most one circuit.

In particular a k-tree is a generalized tree.

To introduce a useful notation, for a directed graph G we shall denote by $v(G)$ the undirected graph obtained by retaining undirected edges between those pairs of vertices that are connected in G in at least one direction.

2. Some properties of the graphs $F^k = v(\mu^{-1}(M_{i_1,\dots,i_k}))$

Consider a simple graph, i.e. an undirected graph without loops and multiple edges, that contains the vertices $\mathbf{P}_1, \dots, \mathbf{P}_n$ ($n \geqq 1$), none of which is isolated, and write $\mathbf{M} = \mu(G)$. Fix the integers $i_1, \dots, i_k$ ($1 \leqq i_1 < \dots < i_k \leqq n$) and let the matrix $\mathbf{M}_{i_1,\dots,i_k}$ run over all the $(i_1, \dots, i_k)$-reductions of $\mathbf{M}$.

Theorem 1. For every graph F^k of form $v(\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k}))$ each of the following five assertions hold:

(A) F^k is a subgraph of G,

(B) F^k is a generalized tree,

(C) the vertices $\mathbf{P}_{i_j}$ belong to mutually distinct components of F^k ($j = 1, \dots, k$),

(D) if a component of F^k contains any of the P_{i_j}'s then this component is a tree (which may possibly degenerate to a single vertex),

(E) those components of F^k not containing any of the vertices $\mathbf{P}_{i_j}$ contain at most one circuit and at least one edge each.

Conversely, if a graph F^k satisfies conditions (A), (B), (C), (D), and (E) then it can be represented in at least one way in the form $F^k = v(\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k}))$.

Proof. Choose an arbitrary $(i_1, \dots, i_k)$-reduction $\mathbf{M}_{i_1,\dots,i_k}$ of $\mathbf{M}$. It is obvius that $F^k = v(\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k}))$ is a subgraph of G.

Construct a matrix $\mathbf{M}'_{i_1,\dots,i_k}$ from $\mathbf{M}_{i_1,\dots,i_k}$ by writing 1 in the intersection of the i_j-th row and the i_j-th column instead of 0 for every $j = 1, \dots, k$. Then $\mu^{-1}(\mathbf{M}'_{i_1,\dots,i_k})$ is a directed graph such that to each of its vertices there is exactly one edge incident that is directed outwards. Applying Ore's theorem we obtain that every component of $\mu^{-1}(\mathbf{M}'_{i_1,\dots,i_k})$ contains exactly one circuit or loop. By transition from $\mu^{-1}(\mathbf{M}'_{i_1,\dots,i_k})$ to $\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k})$ we must delete exactly k loops; thus the graph $\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k})$ turns indeed out to be a generalized tree. Since the loops were deleted precisely at the vertices $\mathbf{P}_{i_j}$, we obtain (D) too ($j = 1, \dots, k$).

Furthermore, since to each vertex $\mathbf{P}_{i_j}$ in $\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k})$ there is a loop incident, Ore's cited theorem also implies that each component of F^k may contain at most one of the vertices $\mathbf{P}_{i_j}$; therefore (C) is also established.

Finally, consider those components of F^k containing none of the points $\mathbf{P}_{i_j}$, and consider simultaneously also the corresponding componentes in $\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k})$. These latter contain exactly one directed circuit each. Passing back to $\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k})$, it is clear that the component in question of $\mu^{-1}(\mathbf{M}_{i_1,\dots,i_k})$ contains this circuit; moreover, since a directed circuit contains at least two distinct vertices, it is guaranteed that this component of F^k contains at least one edge. This completes the proof of (E).*

For the proof of the converse of the theorem assume in what follows that F^k satisfies conditions (A), (B), (C), (D), and (E). To accomplish the proof we

* It may happen that a component of F^k is, despite the fact that it contains none of the vertices $\mathbf{P}_{i_j}$, a tree. This is the situation if the subgraph in $\mu^{-1}(\mathbf{M}'_{i_1,\dots,i_k})$ corresponding to this component contains a directed circuit consisting only of two vertices; by transition to F^k, the two edges, directed in opposite directions, that are incident to both of these edges reduce to one single edge, and the component of F^k in question does not contain any circuit.

associate with F^k a directed graph $F^{k'}$ for which the one hand $F^{k'} = v(F^{k'})$ holds, and the other hand for which $\mu(F^{k'})$ coincides with one of the graphs $\mathbf{M}_{i_1,\ldots,i_k}$.

Introduce a directing of the edges in each component of F^k by abiding by the following rules:

1. If the component in question contains a vertex $\mathbf{P}_{i_j}$ and this $\mathbf{P}_{i_j}$ is not an isolated point then direct the edges of this component towards $\mathbf{P}_{i_j}$. (C) and (D) provides for the unique possibility of this.

2. If the component in question contains a circuit then directed the edges of this latter cyclically and the other edges towards the circuit. Such a directing is made possible by (B).

3. In the remaining cases, i.e. when the considered component contains neither a vertex $\mathbf{P}_{i_j}$ nor a circuit then it contains at least one edge. Let us choose an edge and replace it by two edges directed in opposite directions and, if there exist any, direct the other edges towards the circuit constructed just now.

In this way we obtain a directed graph $F^{k'}$ for which $F^k = v(F^{k'})$ obviously holds.

Consider now the directed graph $F^{k''}$ obtained by adding loops at each point $\mathbf{P}_{i_j}$ of $F^{k'}$. To this $F^{k''}$ we can apply Ore's theorem. We derive that each row of the matrix $\mu(F^{k''})$ contains exactly one 1. By deleting the loops of $F^{k''}$ we can pass back to $F^{k'}$; in terms of the adjacency matrices this amounts to replacing the ones by zeros in each of the rows i_j of $\mu(F^{k''})$. The obtained matrix $\mu(F^{k'})$ is then easily seen to be an $(i_1, \ldots, i_k)$-reduction of the matrix $\mathbf{M} = \mu(G)$. The proof is complete.

From the above proof it becomes clear that to a graph F^k there correspond, in general, several $(i_1, \ldots, i_k)$-reductions. The ambiguity of the construction of these reductions lies in steps 2 and 3 above, where for the directing of the edges there are, in general, several possibilities.

We mention two more interesting properties of the graphs F^k featuring in Theorem 1:

F^k contains at least k components and at most $n-k$ edges.

The remark on the minimal number of components is an easy consequence of (D). That on the maximal number of edges follows from the fact that $F^{k'}$ contains exactly $n-k$ directed edges. Therefore the properties of the correspondance v imply that F^k cannot contain more than $n-k$ edges.

To require that the graph G contains no isolated points is necessary for the existence of an $(i_1, \ldots, i_k)$-reduction for any selection of $i_1, \ldots, i_k$. Still, the assumption on isolated points can be eased if we extend the notion $(i_1, \ldots, i_k)$-reduction for matrices $\mathbf{M}$ that possibly contain rows consisting purely of zeros. In any case, the maximal number of such rows must be limited to k and all such rows must be covered by those of incides $i_1, \ldots, i_k$.

As we pointed out above, the generalized tree F^k in Theorem 1 cannot be represented unambiguously in form $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$. Nevertheless, in case this generalized tree is a k-tree of G, this representation is unambiguous. More precisely, we have the following.

Theorem 2. Assume G is a simple graph without isolated points and with vertices $\mathbf{P}_1, \ldots, \mathbf{P}_n$, and select fixed vertices $\mathbf{P}_{i_1}, \ldots, \mathbf{P}_{i_k}$, where $1 \leqq i_1 < \cdots < i_k \leqq n$. Then the k-tree $F^k_{i_1,\ldots,i_k}$ can be represented unambiguously in form $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$, where $\mathbf{M}_{i_1,\ldots,i_k}$ is a suitable $(i_1, \ldots, i_k)$-reduction of $\mu(G)$.

Proof. That there exists at least one representation of the desired kind follows from the converse Theorem 1 since $F^k_{i_1,\dots,i_k}$ obviously satisfies conditions (A), (B), (C), (D) and (E) of Theorem 1.

In the proof that there exists no more than one representation we may assume $k<n$. Contrary to what we want to prove, suppose that $F^k_{i_1,\dots,i_k}=v(\mu^{-1}(\mathbf{M}^1_{i_1,\dots,i_k}))=v(\mu^{-1}(\mathbf{M}^2_{i_1,\dots,i_k}))$, where $\mathbf{M}^1_{i_1,\dots,i_k}$ and $\mathbf{M}^2_{i_1,\dots,i_k}$ are two different $(i_1,\dots,i_k)$-reductions of $\mu(G)$. We are going exhibit a contradiction.

On account of the assumption that the two matrices are distinct, there exists a $j\neq i_1,\dots,i_k$ $(1\leqq j\leqq n)$ such that in the jth row of $\mathbf{M}^1_{i_1,\dots,i_k}$ the l_1th element is one whereas in the same row of $\mathbf{M}^2_{i_1,\dots,i_k}$ the l_2th element is one, l_1 and l_2 being unequal and both being different from j $(l_1, l_2=1,\dots,n)$. This means that $\mathbf{P}_j$, $\mathbf{P}_{l_1}$, and $\mathbf{P}_{l_2}$ all belong to the same component of $F^k_{i_1,\dots,i_k}$. Assume that $\mathbf{P}_{i_m}$ is the unique one of the selected vertices that also belongs this component $(m=1,\dots,k)$. Then in $\mu^{-1}(\mathbf{M}^1_{i_1,\dots,i_k})$ there exists a directed path leading from $\mathbf{P}_j$ to $\mathbf{P}_{i_m}$ through $\mathbf{P}_{l_1}$, and in $\mu^{-1}(\mathbf{M}^2_{i_1,\dots,i_k})$ there is one leading from $\mathbf{P}_j$ to $\mathbf{P}_{i_m}$ through $\mathbf{P}_{l_2}$. This means that there exist two different paths in some component of $F^k_{i_1,\dots,i_k}$ that connect $\mathbf{P}_j$ with $\mathbf{P}_{i_m}$, contradicting the assumption that $F^k_{i_1,\dots,i_k}$ is a k-tree of G. The proof is complete.

Similarly to what was said after the proof of Theorem 1 the stipulations imposed on G can be eased also here: it is enough to assume that G contains at most k isolated points and these are among the selected ones.

In case $k=1$ we obtain interesting particular cases of Theorem 1 and 2:

Theorem 3. Assume G is a simple graph with vertices $\mathbf{P}_1,\dots,\mathbf{P}_n$, none of which is isolated. Write $\mathbf{M}=\mu(G)$. Fix an integer i, $1\leqq i\leqq n$, and let the matrix $\mathbf{M}_i$ run over all (i)-reductions of $\mathbf{M}$. Then for each graph F of form $v(\mu^{-1}(\mathbf{M}_i))$ the following four assertions hold:

(A) F is a subgraph of G,

(B) F is a generalized tree,

(C) the component containing $\mathbf{P}_i$ of F is a tree, which may possibly degenerate to an isolated point; and finally,

(D) those components of F not containing $\mathbf{P}_i$ contain at least one edge and at most one circuit each.

Theorem 4. Assume $\mathbf{P}_i$ is an arbitrary but fixed point of the connected simple graph G, where $1\leqq i\leqq n$, and F^1 is a tree in G. Then $\mathbf{M}=\mu(G)$ has precisely one (i)-reduction $\mathbf{M}_i$ such that $F^1=v(\mu^{-1}(\mathbf{M}_i))$ holds.

It is necessary to stipulate in this theorem that G is connected since otherwise it would not contain any tree at all.

3. An algorithm to generate the k-trees

Keeping an eye on Theorem 2, we want to generate all the k-trees $F^k_{i_1,\dots,i_k}$ of G by forming all $(i_1,\dots,i_k)$-reductions of $\mu(G)$. Among these there will turn up those representing the k-trees exactly once. Apart from the k-trees these reduced matrices will represent other generalized trees F^k satisfying conditions (A), (B),

(C), (D), and (E). In the sequel we are going to construct a procedure that selects those producing k-trees $F^k_{i_1,\ldots,i_k}$ from among all the $(i_1, \ldots, i_k)$-reductions of $\mu(G)$.

So our procedure will enable us to sift out from among the mentioned generalized trees F^k the k-trees $F^k_{i_1,\ldots,i_k}$, i.e. those generalized trees satisfying (A), (B), (C), (D), and (E) in Theorem 1 that contain no circuit in any of their components.

To start with the description of our method, consider the sets $\{F^k_{i_1,\ldots,i_k}\}$ and $\{\mathbf{M}_{i_1,\ldots,i_k}\}$, where the elements of the second set denote the $(i_1, \ldots, i_k)$-reductions of the matrix $\mathbf{M}=\mu(G)$. As seen from Theorem 2, all the k-trees $F^k_{i_1,\ldots,i_k}$ occur exactly once among the graphs $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$. Also, we observed earlier in Section 1 that the elements of the second set can be given in row vector representation. So in the way described in Theorem 2, the set $\{F^k_{i_1,\ldots,i_k}\}$ can be mapped in a one-to-one way into the set $\{(\varphi(1), \ldots, \varphi(n))\}$ of row vectors, φ running over the functions associated with any of the $(i_1, \ldots, i_k)$-reductions $\mathbf{M}_{i_1,\ldots,i_k}$. We shall describe a procedure that selects those vectors being in the range of the mapping just described. The selected row vectors $(\varphi(1), \ldots, \varphi(n))$ will be those representing the k-trees $F^k_{i_1,\ldots,i_k}$ of the graph G.

Consider a matrix $\mathbf{M}_{i_1,\ldots,i_k}$ satisfying properties (I'), (II'), and (III') described in Section 1 and let φ be the function associated with this matrix.

Definition. By *a cycle check* performed on the matrix $\mathbf{M}_{i_1,\ldots,i_k}$ starting with the integer x we mean the construction of the sequence

$$x, \varphi(x), \varphi(\varphi(x)), \ldots \qquad (1 \leqq x \leqq n).$$

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 of φ; otherwise we say that the outcome of the cycle check is infinite.

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

Definition. By *a complete cycle check* performed on the matrix $\mathbf{M}_{i_1,\ldots,i_k}$ we mean a bunch of cycle checks starting with the integers $1, \ldots, n$ respectively. 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 be infinite.

Now we are going to study the cycle checks from an aspect that will have some importance for our later purposes. To this end, consider a matrix $\mathbf{M}_{i_1,\ldots,i_k}$ and a cycle check performed on it that starts with the integer x. The construction of the sequence $x, \varphi(x), \varphi(\varphi(x)), \ldots$ can be regarded as starting at a vertex $\mathbf{P}_x$ of the directed graph $\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k})$ and walking through a part of the graph, always proceeding in conformity with the direction of the edges passed along. The sequence obtained by a cycle check coincides with the sequence of vertices passed through during such a walk. In case of a cycle check with finite outcome, after a certain time we arrive at a vertex out of which there does not lead any edge. In case of infinite outcome we get into a directed circuit. during the walk.

To be assured that the outcome of a complete cycle check performed on a given matrix is finite we must perform all the n cycle checks constituting this complete check; we may, however, stop earlier if we happen to find an infinite check among these, because this already implies the infiniteness of the complete check.

Theorem 5. Assume G is a simple graph with vertices $\mathbf{P}_1, \ldots, \mathbf{P}_n$, none of which is isolated, and $\mathbf{M}_{i_1,\ldots,i_k}$ is an $(i_1, \ldots, i_k)$-reduction of $\mu(G)$. Then the complete cycle check performed on the matrix $\mathbf{M}_{i_1,\ldots,i_k}$ is of finite outcome if and only if $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$ is a k-tree $F^k_{i_1,\ldots,i_k}$ of the graph G.

Proof. To verify the "if" part assume that $F^k_{i_1,\ldots,i_k}$ is a k-tree of the graph G. According to Theorem 2 there exists a reduction $\mathbf{M}_{i_1,\ldots,i_k}$ such that $F^k_{i_1,\ldots,i_k} = v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$ holds. Furthermore, in each component of the directed graph $\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k})$ all the edges are directed towards the corresponding vertex $\mathbf{P}_{i_j}$ $(j=1, \ldots, k)$. These imply that the cycle check performed on the matrix $\mathbf{M}_{i_1,\ldots,i_k}$ that starts with an arbitrary $i=1, \ldots, n$ is of finite outcome. Therefore the complete cycle check is also of finite outcome.

To prove the "only if" part assume, that the complete cycle check performed on the matrix $\mathbf{M}_{i_1,\ldots,i_k}$ is of finite outcome. Then by an observation made above on cycle checks we see that none of the components of the directed graph $\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k})$ contains a directed circuit.

This entails that the graph $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$ does not contain any circuit. Indeed, assume the contrary, i.e. that this graph does contain some circuit. Consider the directed subgraph of $\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k})$ corresponding to this circuit. This subgraph cannot be a directed circuit; thus it contains at least one vertex with two edges incident to it that are directed outwards. This contradicts the definition of the matrix $\mathbf{M}_{i_1,\ldots,i_k}$.

Moreover, we can derive that the number of edges of $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$ is $n-k$. In fact, this is obvious for $\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k})$. This latter graph, as was pointed out above, does not contain any directed circuit; so, in particular, it does not contain a directed circuit with two edges. Therefore, the correspondance v does not reduce the number of edges.

We obtained that $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$ is a circuit-free graph with $n-k$ edges, and this means that it is indeed a k-tree. Taking Theorem 2 into account, we see that this k-tree can be represented in form $F^k_{i_1,\ldots,i_k}$. The proof is complete.

From Theorem 2 in Section 1 and Theorem 5 in Section 2 we obtain the following algorithm for the generation k-trees $F^k_{i_1,\ldots,i_k}$ of graph G:

I. Construct the adjacency matrix $\mathbf{M}=\mu(G)$ of the graph G.

II. Form all $(i_1, \ldots, i_k)$-reduction $\mathbf{M}_{i_1,\ldots,i_k}$ of $\mathbf{M}$.

III. Perform a complete cycle on the matrices $\mathbf{M}_{i_1,\ldots,i_k}$. Those leading to finite outcomes give the desired k-trees in form $v(\mu^{-1}(\mathbf{M}_{i_1,\ldots,i_k}))$.

(If the graph G has no k-trees at all this will turn out by performing the algorithm since in this case all the cycles have infinite outcomes.)

The algorithm described can be extended so as to apply to the search of the k-trees of a graph G with multiple edges by keeping in mind the following:

Whenever two points of the graph G is connected with more than one edges we replace these with one single edge and call the number of edges replaced the multiplicity of this single one. By performing the above algorithm on the obtained graph G' we get its k-trees. Each of the k-trees of G' that contains no substituted edges is a k-tree of G too.

Now consider the k-trees of G' that contain also substituted edges. By taking

into consideration the substituted edges with their multiplicities we obtain the k-trees of G. From a k-tree in G' we can derive as many k-trees in G as the product of the multiplicity of its edges.

This method provides a way for generating the k-trees of a graph G with multiple edges.

In case $k=1$ our algorithm gives all the trees of a graph G. On account of Theorem 4 we see that the generation of the trees can proceed in n different ways depending on which of the n vertices of G we choose as $\mathbf{P}_i$. We obviously obtain all the trees independently of this choice. This enables us to deliberate as to which of the n ways is the simplest from the angle of computational technique. This problem will be touched upon in the next section.

4. Remarks on adaptation to computers and model examples

We should like to add some computation-technical observations concerning the algorithm outlined in the previous section on the generation of the trees $F^k_{i_1,\ldots,i_k}$ of a graph.

To start with, it seems practical to perform the first two steps of the algorithm immediately on the row vector representation of the considered reduction $\mathbf{M}_{i_1,\ldots,i_k}$. To make this possible we introduce the notion of generating matrix.

Let $\mathbf{M}=\mu(G)=(a_{ij})$ be the adjacency matrix of the graph G.

Definition. The generating matrix $\mathbf{M}_G=(b_{ij})$ associated with the graph G is determined by the formula

$$b_{ij}=\begin{cases}0 & \text{if} \quad a_{ij}=0,\\ j & \text{if} \quad a_{ij}=1.\end{cases}$$

Now the row vector representation of a $(i_1, \ldots, i_k)$-reduction $\mathbf{M}_{i_1,\ldots,i_k}$ of the adjacency matrix $\mathbf{M}=\mu(G)$ of the graph G can be obtained from $\mathbf{M}$ by choosing an element from each row of $\mathbf{M}$ in the following manner:

If the index j of the row considered is different from all i_l $(l=1, \ldots, k)$ then let the choosen element be different from zero, and if $j=i_l$ from some l then choose a 0 (e.g. the element in the diagonal).

Now the cycle check can be performed on the row vector obtained accordingly.

In several cases computational short-cuts can be made in cycle checking. For example, it is easy to see that it is not necessary to start a cycle check at elements which were arrived at in earlier cycle checks. Moreover, if the complete cycle check has infinite outcome we may stop when we stumble upon the first cycle check with infinite outcome.

In principle, it is irrelevant which vertex $\mathbf{P}_i$ we fix when generating the trees of a graph G. In practice, the most clever choice seems to be the one for which the ith row of the generating matrix $\mathbf{M}_G$ contains the largest number of elements different from zero. It can furthermore be observed that the cycle check may have a finite outcome only if the number i occurs in the row vector representation of the matrix $\mathbf{M}_i$.

The remarks made here enable us to sift out those matrices $\mathbf{M}_i$ that are to be

used for cycle checks. Many of the above remarks (e.g. that given in the last sentence of the previous paragraph) can be modified in an obvious way so as to apply also to the case $k \geqq 2$.

Example. Consider the graph G in Fig. 1. We are going to generate all its trees. The generating matrix $\mathbf{M}_G$ of G takes the form

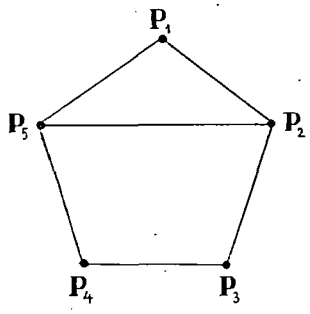


Fig. 1. A simple graph

$$\mathbf{M}_G = \begin{pmatrix} 0 & 2 & 0 & 0 & 5 \\ 1 & 0 & 3 & 0 & 5 \\ 0 & 2 & 0 & 4 & 0 \\ 0 & 0 & 3 & 0 & 5 \\ 1 & 2 & 0 & 4 & 0 \end{pmatrix}.$$

Looking at this matrix we see that it is perhaps the best to choose the vertex $\mathbf{P}_5$ as $\mathbf{P}_i$. If we construct the row vector representations of the $\mathbf{M}_5$ reductions then, according to the remarks made above, we obtain 20 row vectors. Performing cycle checks on these we obtain that 9 of these vectors do not represent any trees. The final result is 11 trees, which are in turn in row vector representation:

(23540), (25230), (25250), (25450),
(51230), (51250), (51430), (51450),
(53450), (55230), (55250).

Observe that from a row vector representation we can easily pass to the actual tree. To do this imagine another row, consisting of the elements $1, 2, \ldots, n$, placed above the row vector representation of $\mathbf{M}_i$; disregarding the one column containing zero, the remaining colums indicate the pairs of vertices that are connected in the tree in question. For example, the tree represented by the row vector (55230) can be seen in Fig. 2.

row vector representation:
(55230)

"completed" row vector:

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 5 & 5 & 2 & 3 & 0 \end{bmatrix}$$

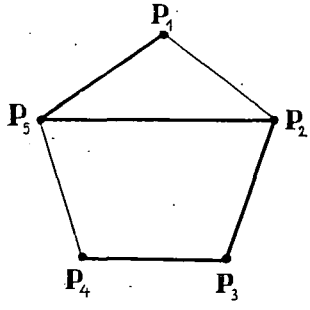


Fig. 2. A tree of the graph of Fig. 1.

As a further example we shall now generate the trees $F^3_{1,2,4}$ of the graph G given in Fig. 1.

Again, we construct the row vector representations of all the eligibe ones of the matrices $\mathbf{M}_{1,2,4}$:

(00201), (00202), (00204), (00401),

(00402), (00404).

Now performing cycle checks on the matrices corresponding to the rows enumerated we obtain finite outcomes in all six cases.

This means that in this case the number of the 3-trees is 6. Fig. 3 illustrates the 3-tree corresponding to the vector (00404).

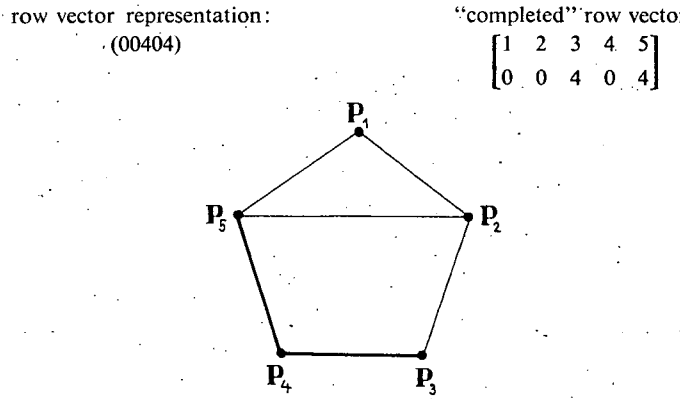


Fig. 3. A 3-tree of the graph of Fig. 1.

The advantages of this method in comparison to others for the generation of *k*-trees of a graph G seems clear if the method is adapted to computer. Namely, the method described in Section 2 is perhaps the most easily fed into computers among the known tree-generation methods. It is also clear that the storage capacity occupied by a programme based on this method is considerably smaller than that needed for the performing of a programme using e.g. the algebraic method [5]. The reason for this is that it is not necessary to store the data representing the tree for further operations, the cycle check decides immediately whether the obtained data (row vector) in fact represent a tree. This is a considerable advantage if we take into account that in pratice the number of the trees can be of magnitude order of several millions [4]. If not programmed clumsily, the complete cycle check does not increase the computing time unfavorably in comparison with time used up by the algebraic method.

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

References

[1] HAKIMI, S. L. & D. G. GREEN, Generation and realization of trees and k-trees, *IEEE. Trans.*, v. CT—11, 1964, pp. 247—255.
[2] HARARY, F., The number of functional digraphs, *Math. Annales*, v. 138, 1959, pp. 203—210.
[3] MAYEDA, W., Reducing computation time in the analysis of networks by digital computer, *IRE Trans.*, v. CT—6, 1959, pp. 136—137.
[4] MAYEDA, W. & S. SESHU, Generation of trees without duplications, *IEEE Trans.*, v. CT—12, 1965, pp. 181—185.
[5] MAXWELL, L. M. & JR. J. M. CLINE, Topological network analysis by algebraic methods, *Proc. IEEE*, v. 113, 1966, pp. 1344—1347.
[6] ORE, O., *Graphs and Correspondences*, Festschrift f. d. 60. Geburtstag von A. Speiser, 1945, pp. 184—191.
[7] ORE, O., *Theory of Graphs*, Am. Math. Soc., Providence, 1962. pp. 68—74.
[8] SESHU, S. & M. B. REED, *Linear graphs and electrical networks*, Addison-Wesley Publ. Co., Massachusette, 1961.

RESUMÉ

Dans cette petite Note nous allons présenter une procedure nouvelle engendrant tous les k-arbres d'une graphe avec un point donné dans chacun de ses composantes connexes. Notre méthode exploite un théorème d'Ore [7] concernant les graphes finites et directées, ainsi rendant une application de ce théorème qui est, hors de son intérêt théorique, aussi utile dans l'analyse des secteurs électriquies pour des buts pratiques.

(Received March 9, 1970)

On the behaviour of some cyclically symmetric networks

By A. ÁDÁM and U. KLING

Zusammenfassung. In diesem Artikel beschäftigen wir uns mit dem folgenden speziellen Typ von Netzwerken: die Punkte des Graphen werden durch $P_1, P_2, \ldots, P_n$ bezeichnet; es existiert ein Zahl k $(1 \leqq k < n)$ so daß von jedem Punkt P_i die Kanten zu den Punkten

$$P_{i-1}, P_{i-2}, \ldots, P_{i-k}$$

und nur zu diesen führen (wobei die Subtraktion modulo n gemeint wird). Wir setzen dasjenige kontinuierliche Modell fort, das im Abschnitt 3 der Arbeit [2] eingeführt wurde. Der Zustand $\mathfrak{A}$ eines derartigen Graphen heißt zyklisch, wenn es eine positive Zahl p gibt, so daß nach einem Zeit-Intervall der Länge p der aus $\mathfrak{A}$ entstehende Zustand mit $\mathfrak{A}$ übereinstimmt. Wir unterscheiden im § 1 reguläre und nicht-reguläre Zustände. In den §§ 2—3 wird das Funktionieren eines Graphen mit einem regulären Anfangszustand diskutiert; wir stellen fest, daß jeder reguläre Zustand zyklisch ist. Im § 4 beschäftigen wir uns mit dem Funktionieren eines Netzwerkes mit einem nicht-regulären Anfangszustand; unser Hauptergebnis besagt, daß kein nicht-regulärer Zustand zyklisch sein kann.

§ 1. Introduction

In this paper we deal with the function of a special graph-theoretical class of networks. (We speak of a *network* if numerical values or numerical functions are assigned to the vertices of a graph.) We shall point out that the behaviour of networks in question can be described more explicitly in comparation to the general model elaborated in Sect. 3 of [2]. It is throughout supposed that the reader is familiar with Sections 1—3 of the former article [2].

Now we delimit the graph-theoretical structure of the networks to be investigated. Let $G(n;\ m_1, m_2, \ldots, m_k)$ (where $1 \leqq m_1 < m_2 < \cdots < m_k < n$) denote the graph consisting of n vertices labelled as $P_1, P_2, \ldots, P_n$, so that the directed edge $\overrightarrow{P_i P_j}$ exists if and only if there is an integer h $(1 \leqq h \leqq k)$ for which the congruence

$$i - j \equiv m_h \pmod{n}$$

holds.[1] *We shall regard the graphs* $G(n;\ 1, 2, \ldots, k)$ (where $1 \leqq k < n$) in the whole

[1] For the isomorphism problem of these graphs see [1] and the most recent papers [3], [4].

paper. We note that the subscripts of the vertices of such a graph (and consequently, also the subscripts of the functions α_i assigned to them) are mostly understood modulo n.[2]

Let a state

$$\mathfrak{A} = \langle \alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t) \rangle$$

(at the instant[3] t) of a graph G (containing n vertices) be considered. Let us denote by $\mathfrak{A}[+p]$ the state of G at the instant $t+p$ where p is an arbitrary non-negative real number. (More precisely: let us apply the continuous model defined in Sect. 3 of [2] for G, starting with $\mathfrak{A}$ at t; let $\mathfrak{A}[+p]$ be the vector

$$\langle \alpha_1(t+p), \alpha_2(t+p), \ldots, \alpha_n(t+p) \rangle.)$$

We say that $\mathfrak{A}$ is a *cyclic* state (and p is its *period*) if there exists a positive p such that $\mathfrak{A} = \mathfrak{A}[+p]$. In the contrary case, $\mathfrak{A}$ is an *acyclic* state.

We use for $\alpha_i(0)$ the shorter notation β_i, too.

Let us consider a network $G(n;\ 1, 2, \ldots, k)$. Assume that there exists at least one vertex P_j with $\alpha_j(t) = 1$. (If this holds for P_j, then each of $\alpha_{j-1}(t)$, $\alpha_{j-2}(t)$, $\alpha_{j-3}(t)$, ..., $\alpha_{j-k}(t)$ is 0.) We say that the vertices

$$P_{i+1}, P_{i+2}, \ldots, P_{j-2}, P_{j-1}, P_j \tag{1}$$

form an *arc* (at the instant t) if

$$1 = \alpha_i(t) > \alpha_{i+1}(t) > \alpha_{i+2}(t) > \cdots > \alpha_{j-k-1}(t) \geqq$$

$$\geqq \alpha_{j-k}(t) = \alpha_{j-k+1}(t) = \alpha_{j-k+2}(t) = \cdots = \alpha_{j-1}(t) = 0$$

(and, of course, $\alpha_j(t) = 1$) hold. Evidently, the number of vertices of an arc is necessarily at least $k+1$. (We emphasize that P_i does *not* belong to the arc (1).) A state of a graph $G(n;\ 1, 2, \ldots, k)$ is called *regular* (at t) if each vertex is contained in an arc (obviously, it may be contained in only one). In a regular state, we denote by $\varphi(P_i, t)$ the first vertex P_j in the sequence

$$P_{i+1}, P_{i+2}, P_{i+3}, \ldots$$

which satisfies $\alpha_j(t) = 1$; in other words, $\varphi(P_i, t)$ is that vertex P_j in the arc containing P_{i+1} which fulfils $\alpha_j(t) = 1$. (P_i and P_{i+1} are in the same arc unless $\alpha_i(t) = 1$.)

In what follows, we shall obtain that *a state of a network* $G(n;\ 1, 2, \ldots, k)$ *is cyclic if and only if it is regular* (Propositions 2, 8).

§ 2. Discussion of the behaviour of a network starting with a regular state

Let us consider a regular state of a network $G(n;\ 1, 2, \ldots, k)$ at the instant 0. Our next aim is to give a detailed discussion of the function α_i associated to a vertex P_i (chosen arbitrarily) of G during the time interval $[0, \tau]$. Our treatment is based

[2] For example, we write simply "the vertex P_{i+l}" instead of "the vertex P_j whose subscript is determined by $j \equiv i+l \pmod{n}$, $1 \leqq j \leqq n$".

[3] In what follows, t will be almost everywhere 0.

upon Sect. 3 of [2]. We shall formulate several consequences of the present discussion in § 3; one of these consequences is anticipated just now:

Proposition 1. If

$$\mathfrak{A} = \langle \alpha_1(0), \alpha_2(0), \ldots, \alpha_n(0) \rangle$$

is a regular state, then we have

$$\alpha_i(\tau) = \alpha_{i+k+1}(0)$$

for each i (*i can be* $1, 2, \ldots, n$).

We are going to perform the discussion. We distinguish three cases according to the possibilities $0<\beta_i<1$, $\beta_i=0$, $\beta_i=1$. Any case is subdivided to some subcases with respect to the smallest integer h satisfying $P_{i+h}=\varphi(P_i, 0)$. In every discussed case, the following statement will be always true: whenever $\alpha_j(t)=0$ and there exists a positive number ε such that $\alpha_j(t')>0$ holds for every t' fulfilling $t-\varepsilon < t'< t$, then $\alpha_{j+1}(t)=1$. We shall apply this method of inference (in a number of steps) without being mentioned explicitly.

Case 1: $0<\beta_i<1$. We distinguish three subcases.

Case 1/a: $h > 2k+1$, in other words, each of $P_{i+1}, P_{i+2}, \ldots, P_{i+2k+1}$ differs from $\varphi(P_i, 0)$. This assumption implies (by the definition of the regular state)

$$\beta_i > \beta_{i+1} > \cdots > \beta_{i+k} > \beta_{i+k+1} \geqq \beta_{i+k+2} \geqq \cdots \geqq \beta_{i+2k+1}.$$

The behaviour of α_i in $[0, \tau]$ can be described as follows:

(i) in the interval $[0, \tau(1-\beta_i)]$ the value of α_i grows linearly from β_i to 1,

(ii) in the interval $[\tau(1-\beta_i), \tau(1-\beta_{i+1}))$ α_i is constantly 1,

(iii) in the interval $[\tau(1-\beta_{i+1}), \tau(1-\beta_{i+k+1})]$ α_i is constantly 0,

(iv) in the interval $[\tau(1-\beta_{i+k+1}), \tau]$ (of length $\tau\beta_{i+k+1}$) the value of α_i grows linearly from 0 to $\tau\cdot\beta_{i+k+1}/\tau = \beta_{i+k+1}$.

Indeed, P_i gets edges exactly from the vertices $P_{i+1}, P_{i+2}, \ldots, P_{i+k}$. None of $\alpha_{i+1}, \ldots, \alpha_{i+k}$ can be 1 in the interval $[0, \tau(1-\beta_{i+1}))$. However, at every instant t of the interval $[\tau(1-\beta_{i+1}), \tau(1-\beta_{i+k+1}))$, (exactly) one of $\alpha_{i+1}(t), \ldots, \alpha_{i+k}(t)$ is 1. In the interval $[\tau(1-\beta_{i+k+1}), \tau)$ α_{i+k+1} is constantly 1, thus each of $\alpha_{i+1}, \ldots, \alpha_{i+k}$ is constantly 0. We have also $\alpha_{i+1}(\tau)=\cdots=\alpha_{i+k}(\tau)=0$, hence α_i may grow in $[\tau(1-\beta_{i+k+1}), \tau]$.

Case 1/b: $k+2 \leqq h \leqq 2k+1$. Then

$$\beta_i > \beta_{i+1} > \cdots > \beta_{i+h-k-1} \geqq \beta_{i+h-k} = \beta_{i+h-k+1} = \cdots = \beta_{i+h-1} = 0,$$

$$1 = \beta_{i+h} > \beta_{i+h+1} \geqq \beta_{i+h+2} \geqq \cdots \geqq \beta_{i+h+k}.$$

The condition of the case implies the inequalities

$$i+2 \leqq i+h-k \leqq i+k+1 \leqq i+h-1 \leqq i+2k,$$

thus $\beta_{i+k+1}=0$. The behaviour of α_i satisfies the assertions (i), (ii) of Case 1/a, moreover,

(iii) in the interval $[\tau(1-\beta_{i+1}), \tau]$ α_i is constantly 0. Indeed, since $\alpha_{i+k+1}(t')<1$ at each instant t' of the interval $[0, \tau)$, the behaviour of $\alpha_{i+1}, \ldots, \alpha_{i+k}$ is similar to Case 1/a (with τ instead of $\tau(1-\beta_{i+k+1})$).

Case 1/c: $h = k+1$. Then

$$\beta_i > \beta_{i+1} = \beta_{i+2} = \ldots = \beta_{i+k} = 0,$$

$$1 = \beta_{i+k+1} > \beta_{i+k+2} \geqq \beta_{i+k+3} \geqq \ldots \geqq \beta_{i+2k+2}.$$

The behaviour of α_i can be described as follows:

(i) in the interval $[0, \tau(1-\beta_i)]$ the value of α_i grows linearly from β_i to 1,

(ii) in the interval $[\tau(1-\beta_i), \tau]$ α_i is constantly 1.

Indeed, none of $\alpha_{i+1}, \alpha_{i+2}, \ldots, \alpha_{i+k}$ can reach 1 in the interval $[0, \tau(2-\beta_{i+k+2}))$, furthermore $\tau < \tau(2-\beta_{i+k+2})$.

Case 2: $\beta_i = 0$. We distinguish four subcases:

Case 2/a: $h = k+1$. We can prove by ideas similar to Case 1/c that α_i grows linearly from 0 to 1 in the whole interval $[0, \tau]$.

Case 2/b: $h = k$. Then

$$\beta_i = \beta_{i+1} = \ldots = \beta_{i+k-1} = 0,$$

$$1 = \beta_{i+k} > \beta_{i+k+1} \geqq \beta_{i+k+2} \geqq \ldots \geqq \beta_{i+2k+1}.$$

The behaviour of α_i is as follows:

(i) in the interval $[0, \tau(1-\beta_{i+k+1})]$ α_i is constantly 0,

(ii) in the interval $[\tau(1-\beta_{i+k+1}), \tau]$ α_i grows linearly from 0 to

$$(\tau - \tau(1-\beta_{i+k+1}))/\tau = \beta_{i+k+1}.$$

Case 2/c: $1 \leqq h \leqq k-1$ and $\beta_{i+k+1} = 0$. Then

$$\beta_i = \beta_{i+1} = \ldots = \beta_{i+h-1} = 0,\ 1 = \beta_{i+h} > \beta_{i+h+1} > \ldots$$

$$\ldots > \beta_{i+k+1} > \beta_{i+k+2} \geqq \beta_{i+k+3} \geqq \ldots \geqq \beta_{i+2k+2}.$$

The same conclusions (i), (ii) are true as in Case 2/b.

Case 2/d: $1 \leqq h \leqq k-1$ and $\beta_{i+k+1} = 0$. Then

$$\beta_i = \beta_{i+1} = \cdots = \beta_{i+h-1} = 0,$$

$$1 = \beta_{i+h} > \beta_{i+h+1} \geqq \beta_{i+h+2} \geqq \cdots \geqq \beta_{i+k+1} = 0.$$

In this case α_i is constantly 0 in the whole interval $[0, \tau]$.

Case 3: $\beta_i = 1$. This case can be discussed similarly to Case 1. The single modification is that $\tau(1-\beta_i) = 0$, thus the conclusions (i) do not occur in the subcases.

§ 3. Propositions on the behaviour of a network starting with a regular state

We are going to expose some statements which summarize the discussion performed in the preceding paragraph. Let g be the least common multiple of $k+1$ and n.

Proposition 2. Any regular state is cyclic; $g\tau/(k+1)$ *is a suitable period.*

Proof. If we apply Proposition 1 $g/(k+1)$ times, then we get

$$\alpha_i(0) = \alpha_{i+(k+1)}(\tau) = \alpha_{i+2(k+1)}(2\tau) = \cdots = \alpha_{i+g}(g\tau/(k+1)) = \alpha_i(g\tau/(k+1))$$

for every i.

Proposition 3. If $\mathfrak{A}$ is a regular state, then the state $\mathfrak{A}[+t]$ is regular for each non-negative t.

Proof. Assume that the instant of $\mathfrak{A}$ is denoted by 0. Let d be the greatest integer so that $d\tau \leqq t$. We get by successive application of Proposition 1 that the conclusion of the present proposition is true for $d\tau$. By analyzing § 2, we obtain that it holds for t too (because $t - d\tau < \tau$). The proof is completed.

An easy consequence of our former investigations is

Proposition 4. If $\mathfrak{A}$ is a regular state and t is a non-negative number, then the number of arcs of $\mathfrak{A}$ equals to the number of arcs of $\mathfrak{A}[+t]$.

Let us fix a vertex P_i, let us consider the sequence

$$P_i, P_{i+(k+1)}, P_{i+2(k+1)}, P_{i+3(k+1)}, \ldots, P_{i-(k+1)} \tag{2}$$

consisting of $g/(k+1)$ (distinct) vertices and the sequence

$$P_{i+1}, P_{i+(k+1)+1}, P_{i+2(k+1)+1}, P_{i+3(k+1)+1}, \ldots, P_{i-(k+1)+1} \tag{3}$$

which consists likewise of $g/(k+1)$ vertices. Either $n, k+1$ are relatively prime to each other (thus $g = n(k+1)$ and both of (2), (3) contain all the vertices) or (2), (3) are disjoint.[4] Let us define the instants v_h and w_h by

$$v_h = \tau(h - \beta_{i+(h-1)(k+1)}) \quad \text{and} \quad w_h = \tau(h - \beta_{i+(h-1)(k+1)+1})$$

(where h can be $1, 2, \ldots, g/(k+1)$). This definition implies immediately

Lemma 1. For any h,

$$\tau(h-1) \leqq v_h \leqq \tau h \quad \textit{and} \quad \tau(h-1) \leqq w_h \leqq \tau h.$$

Lemma 2. For any h we have one of the three possibilities

(a_1) $v_h < w_h$

(a_2) $v_h = w_h = \tau h$

(a_3) $w_h = \tau(h-1)$ *and* $v_h = \tau h$

(*according as*

(b_1) $\beta_{i+(h-1)(k+1)} > \beta_{i+(h-1)(k+1)+1}$

(b_2) $\beta_{i+(h-1)(k+1)} = \beta_{i+(h-1)(k+1)+1} = 0$

(b_3) $\beta_{i+(h-1)(k+1)} = 0, \beta_{i+(h-1)(k+1)+1} = 1$).

[4] For, if (2), (3) contain a vertex in common, then some multiple of $k+1$ is congruent to 1 modulo n, hence n and $k+1$ are relatively primes.

Proof. The equivalence of (a_i) and (b_i) can be shown easily (for all the three values of i), the proof is completed by the remark either (b_1) or (b_2) or (b_3) is true since the state is regular.

Lemma 3. If $v_{h-1} < w_{h-1}$ *and* $v_h < w_h$ *for some* $h(\geqq 2)$, *then either* $w_{h-1} = v_h = \tau(h-1)$ *or* $w_{h-1} < v_h - \tau$.

Proof. The supposition implies

$$\beta_{i+(h-2)(k+1)} > \beta_{i+(h-2)(k+1)+1},$$

$$\beta_{i+(h-1)(k+1)} > \beta_{i+(h-1)(k+1)+1}.$$

The sequence (consisting of $k+1$ numbers)

$$(4) \qquad \beta_{i+(h-2)(k+1)+1}, \beta_{i+(h-2)(k+1)+2}, \beta_{i+(h-2)(k+1)+3}, \cdots, \beta_{i+(h-1)(k+1)}$$

is monotonically decreasing unless $\beta_{i+(h-1)(k+1)} = 1$ (by the regularity of the state), thus we can distinguish two cases.

Case 1: (4) is monotonically decreasing. Then the number

$$\beta_{i+(h-2)(k+1)+1} - \beta_{i+(h-1)(k+1)} \left(= (v_h - \tau - w_{h-1})/\tau\right)$$

is positive, hence $w_{h-1} < v_h - \tau$.

Case 2: $\beta_{i+(h-1)(k+1)} = 1$. Then, on the one hand, $v_h = \tau(h-1)$; on the other hand, $\beta_{i+(h-2)(k+1)+1} = 0$, this implies $w_{h-1} = \tau(h-1)$.

By use of the numbers v_h, w_h we can explicitly characterize the behaviour of α_i in the interval $[0, g\tau/(k+1))$:

Proposition 5. Let us consider a regular state at the instant 0. *The function* α_i, *assigned to a vertex* P_i, *satisfies the following four assertions:*

(A) *If* $(1 \leqq h \leqq g/(k+1)$ *and*$)$ $v_h < w_h$, *then* α_i *is constantly* 1 *in the interval* $[v_h, w_h)$.[5]

(B) *If* $(2 \leqq h \leqq g/(k+1)$ *and*$)$ $w_{h-1} < v_h < w_h$, *then* α_i *grows linearly in the interval* $[v_h - \tau, v_h]$ *from* 0 *to* 1.

(C) *If* $v_1 < w_1$, *then* α_i *grows linearly in the interval* $[0, v_1]$ *from* $1 - v_h/\tau$ *to* 1.

(D) *The value of* α_i *is* 0 *at all the instants of the interval* $[0, g\tau/(k+1))$ *which are not referred to in* (A), (B) *and* (C).

Proof. Let an instant t lying in $[0, g\tau/(k+1))$ be considered. There exists a number h such that $\tau(h-1) \leqq t < \tau h$ (where $1 \leqq h \leqq g/(k+1)$). By using Proposition 1 successively $h-1$ times (with $t-\tau, t-2\tau, t-3\tau, \ldots, t-\tau(h-1)$ instead of 0), we get

$$\alpha_i(t) = \alpha_{i+(k+1)}(t-\tau) = \alpha_{i+2(k+1)}(t-2\tau) = \cdots$$

$$\ldots = \alpha_{i+(h-2)(k+1)}\big(t-\tau(h-2)\big) = \alpha_{i+(h-1)(k+1)}\big(t-\tau(h-1)\big),$$

i.e. the behaviour of α_i in the interval $[\tau(h-1), \tau h)$ is the same as the behaviour of $\alpha_{i+(h-1)(k+1)}$ in $[0, \tau)$ (with the appropriate translation).

[5] Since $w_h = v_{h+1}$ may occur, two or more intervals of this character can be joined.

First we show (A). The function $\alpha_{i+(h-1)(k+1)}$ takes the value 1 exactly in the sub-interval

$$[\tau(1-\beta_{i+(h-1)(k+1)}), \tau(1-\beta_{i+(h-1)(k+1)+1}))$$

of $[0, \tau)$ by Cases 1/a, 1/b, 1/c, 3/a, 3/b, 3/c of the discussion in § 2 (even if at least one of

$$\beta_{i+(h-1)(k+1)} = 1, \quad \beta_{i+(h-1)(k+1)+1} = 0$$

is true).

In order to verify (B), let $t(\geqq\tau)$ be such an instant that $\alpha_i(t)=1$ but, for every positive ε, there exists a t^* fulfilling $\alpha_i(t^*)<1$ and $t-\varepsilon<t^*<t$. Then $\alpha_{i+(h-2)(k+1)}$ has the analogous property at the instant $t-(h-2)$, and $\tau \leqq t-\tau(h-2) < 2\tau$. By analyzing the discussion and by Proposition 1, we get that $\alpha_{i+(h-2)(k+1)}$ grows linearly in $[t-\tau(h-1), t-\tau(h-2)]$ from 0 to 1, consequently α_i behaves in $[t-\tau, t]$ analogously.

(C) follows from the discussion immediately.

(D) is equivalent to the subsequent statement: any function α_i is 0 at t unless t is contained in an interval $(t', t'+\tau]$ such that $\alpha_i(t'+\tau)=1$. This statement follows easily from the discussion and Proposition 1 in the interval $[0, 2\tau]$, it can be extended for any non-negative t by Proposition 1.

The last assertion we state relying upon § 2 is the evident

Proposition 6. The following three statements are equivalent for a regular state:
(A) *The state is steady.*
(B) *Every arc of the state consists of exactly $k+1$ vertices.*
(C) *$k+1$ is a divisor of n and the number of arcs in the state is $n/(k+1)$.*

§ 4. Study of non-regular states

The purpose of this paragraph is to show that only the regular states are cyclic. First we define the *irregularity indices* of an arbitrary permitted state[6] $\mathfrak{A}$ by the following three rules:

(i) if $\beta_{i-1}<\beta_i<1$, then i is an irregularity index,
(ii) if $\beta_{i-1}=\beta_i>0$, then i is an irregularity index,
(iii) if $\beta_{i-1}=\beta_i=0$ and each of $\beta_{i+1}, \beta_{i+2}, \ldots, \beta_{i+k}$ is <1, then i is an irregularity index.

(The conditions in (i), (ii), (iii) exclude each other.) We agree that no remaining number (out of the set $\{1, 2, \ldots, n\}$) is an irregularity index. The *irregularity number* of the state $\mathfrak{A}$ is the number of its irregularity indices.

If (i) or (iii) holds for i, then i is called a *strong irregularity index*, the number of strong irregularity indices is the *strong irregularity number* of $\mathfrak{A}$. If (ii) holds for i then we call i a *weak irregularity index*.

Lemma 4. The irregularity number of $\mathfrak{A}$ is 0 if and only if $\mathfrak{A}$ is a regular state.

Proof. It is obvious that the definition of the regular state does not admit any of the possibilities (i), (ii), (iii). — Conversely, assume that no vertex fulfilling

[6] A state is permitted if $\alpha_i=1$, $P_j \in \chi(P_i)$ imply $\alpha_j=0$.

the condition of either (i) or (ii) or (iii) occurs in $\mathfrak{A}$; let P_i be an arbitrary vertex. If $\beta_i = 1$, then

$$\beta_{i-k} = \beta_{i-k+1} = \beta_{i-k+2} = \cdots = \beta_{i-1} = 0$$

(since the state is permitted). If $0 < \beta_i < 1$, then $\beta_{i-1} > \beta_i$ (since otherwise (i) or (ii) would be violated). If $\beta_i = 0$, then either one of $\beta_{i+1}, \beta_{i+2}, \ldots, \beta_{i+k}$ is 1 or $\beta_{i-1} > \beta_i$ (in consequence of (iii)). Thus $\mathfrak{A}$ is a regular state.

Lemma 5. Let $\mathfrak{A}$ be a state at the instant 0 and t be a positive instant such that the functioning of the network is defined (at least) in the interval $[0, t]$. If i is not a strong irregularity index at 0, then i is a strong irregularity index nor at t.

Proof. Let t^* be the (possibly non-existing) least real number such that $0 \leqq t^* \leqq t$ and none of $\alpha_{i+1}, \alpha_{i+2}, \ldots, \alpha_{i+k}$ takes the value 1 in the interval $[t^*, t]$. Either $t^* = 0$ or there exists a number q such that $1 \leqq q \leqq k$ and to every positive ε there exists a t' satisfying both $t^* - \varepsilon < t' < t^*$ and $\alpha_{i+q}(t') = 1$.

Case 1: $t^* > 0$ and $q < k$. We have

$$\alpha_{i-1}(t^*) = \alpha_i(t^*) = 0,$$

the functions α_{i-1}, α_i are equal and increase linearly in the whole interval $[t^*, t]$ from 0 to $(t - t^*)/\tau$. (Necessarily $t - t^* < \tau$; if the contrary were true, we should get a contradiction to the hypothesis that the functioning is defined in $[0, \tau]$.)

Case 2: $t^* > 0$ and $q = k$. We have

$$\alpha_{i-1}(t^*) \geqq \alpha_i(t^*) = 0.$$

Three subcases are possible:

Case 2/a: $\alpha_{i-1}(t^*) = 0$. This subcase can be treated similarly to Case 1.

Case 2/b: $\alpha_{i-1}(t^*) > 0$ and $t - t^* < \tau$. Then α_i increases linearly in the whole interval $[t^*, t]$ from 0 to $(t - t^*)/\tau$. α_{i-1} increases linearly from

$$\alpha_{i-1}(t^*) \text{ to } \begin{cases} \alpha_{i-1}(t^*) + (t - t^*)/\tau & \text{in } [t^*, t] \text{ if } \alpha_{i-1}(t^*) + (t - t^*)/\tau \leqq 1, \\ 1 \text{ in } [t^*, t^* + \tau(1 - \alpha_{i-1}(t^*))] & \text{if } \alpha_{i-1}(t^*) + (t - t^*)/\tau > 1. \end{cases}$$

In the second of these cases α_{i-1} is constantly 1 in $[t^* + \tau(1 - \alpha_{i-1}(t^*)), \tau]$.

Case 3: $t^* = 0$ and $\beta_{i-1} > \beta_i$. Let us assume that t is so large that all the intervals to be discussed are in $[0, t]$. (If this assumption is not fulfilled, then the subsequent discussion is altered so that it breaks off at the instant t.) In the interval $[0, \tau(1 - \beta_{i-1})]$ both α_{i-1} and α_i increase linearly. In $[\tau(1 - \beta_{i-1}), \tau(1 - \beta_i))$ α_{i-1} is constantly 1 and α_i increases linearly. In $[\tau(1 - \beta_i), t]$ α_i is constantly 1 and α_{i-1} is constantly 0.

Case 4: $t^* = 0$ and $\beta_{i-1} = \beta_i$. Then $t < \tau$, furthermore α_{i-1}, α_i are equal and increase from 0 to t/τ similarly as in Case 1.

Case 5: t^* does not exist. Then there is at least one number q such that $1 \leqq q \leqq k$ and $\alpha_{i+q}(t) = 1$, thus $\alpha_i(t) = 0$. i fulfils the conditions of neither (i) nor (iii) at t.

Lemma 6. If the strong irregularity number of a state $\mathfrak{A}$ *at the instant* 0 *is positive and the functioning of the network in the interval* $[0, \tau]$ *is defined, then the strong irregularity number of the state* $\mathfrak{A}[+\tau]$ *is* 0.

Proof. Let i be an arbitrary index. If i is not a strong irregularity index, then we can apply Lemma 5. Otherwise, let us define t^* and q as in the proof of Lemma 5. If $t^* > 0$ then Cases 1, 2 of the preceding proof remain valid; if t^* does not exist, then the inference of Case 5 can be applied. We have still to study the cases when $t^* = 0$ and i fulfils (i) or (iii).

If (i) is true, then

$$\alpha_i(\tau(1-\beta_i)) = 1 \quad \text{and} \quad \alpha_{i-1}(\tau(1-\beta_i)) = 0.$$

i is not a strong irregularity index at $\tau(1-\beta_i)$ consequently nor at τ (by Lemma 5).

If (iii) holds, then it is easy to see that the functioning of the graph is defined at most in the interval $[0, \tau)$; this contradicts the supposition of Lemma 6.

Lemma 7. Let $\mathfrak{A}$ *be a state at the instant* 0 *such that the strong irregularity number of* $\mathfrak{A}$ *is* 0. *If the functioning of the network in the interval* $[0, \tau]$ *is defined, then the irregularity number of* $\mathfrak{A}[+\tau]$ *is* 0.

Proof. Whenever j is an arbitrary index and t' is an instant such that $0 \leqq t' \leqq \tau$, then j cannot be a strong irregularity index at t' (by Lemma 5). We shall study a function α_i in $[0, \tau]$. Let us define t^* and q in the same manner as at beginning of the proof of Lemma 5.

Case 1: $t^* > 0$. Necessarily $q = k$ (since now the value 1 "steps" from j to $j+1$, similarly to the case of a regular state, discussed in § 2). Hence $\alpha_{i-1}(t^*) > \alpha_i(t^*) = 0$. In the interval

$$[t^*, t^* + \tau(1-\alpha_{i-1}(t^*))]$$

α_{i-1}, α_i increase parallel (i.e. $\alpha_{i-1} - \alpha_i$ remains constant). In the interval

$$[t^* + \tau(1-\alpha_{i-1}(t^*)), \tau]$$

(provided that it exists) α_{i-1} is constantly 1 and α_i continues its growth.

Case 2: $t^* = 0$. We distinguish two subcases.

Case 2/a: $\beta_{i-1} = \beta_i$. This assumption implies that the functioning of the network is defined only in $[0, \tau(1-\beta_i))$, i.e. it contradicts the supposition of Lemma 7.

Case 2/b: $\beta_{i-1} > \beta_i$. In the interval $[0, \tau(1-\beta_{i-1})]$, α_{i-1} and α_i increase parallel. In

$$[\tau(1-\beta_{i-1}), \tau(1-\beta_i))$$

α_{i-1} is constantly 1 and α_i continues its growth. In $[\tau(1-\beta_i), \tau]$ α_i is constantly 1 and α_{i-1} is constantly 0.

Case 3: t^* does not exist. We get $\alpha_i(\tau) = 0$ similarly to Case 5 of the proof of Lemma 5, hence i does not fulfil the condition of (ii).

Proposition 7. If the state $\mathfrak{A}$ *(at the instant* 0*) is non-regular, then either* $T'_{\max}$ *is defined for* $\mathfrak{A}$ *and* $0 < T'_{\max} < 2\tau$ *or* $\mathfrak{A}[+2\tau]$ *is regular.*[7]

[7] $T'_{\max}$ was introduced in [2].

Proof. Assume that the states $\mathfrak{A}[+t]$ are definable whenever $0 \leqq t \leqq 2\tau$. The state $\mathfrak{A}[+\tau]$ cannot have a strong irregularity index (by Lemma 6), hence the state $\mathfrak{A}[+2\tau]$ is regular (by Lemmas 7 and 4).

Proposition 8. Any non-regular state is acyclic.

Proof. Let $\mathfrak{A}$ be a non-regular state (at the instant 0). If the state $\mathfrak{A}[+t]$ is not definable for every positive t (i.e. if $T'_{\max}$ does exist), then $\mathfrak{A}$ is obviously acyclic. Assume that $\mathfrak{A}[+t]$ is defined for every t. Let $\mathfrak{A}$ be cyclic and p be a period of it, we shall get a contradiction. Let d be the least integer such that $dp \geqq 2\tau$ holds. On the one hand,

$$\mathfrak{A} = \mathfrak{A}[+p] = \mathfrak{A}[+2p] = \cdots = \mathfrak{A}[+dp],$$

thus $\mathfrak{A}[+dp]$ is non-regular. On the other hand, $\mathfrak{A}[+2\tau]$ is regular by Proposition 7, hence also $\mathfrak{A}[+dp]$ is regular by Proposition 3.

§ 5. On some possibilities for future researches

Let us consider a graph. Denote by A the set of its permitted states (i.e. all the mappings of the vertex set into the interval $[0, 1]$ such that the restriction mentioned in Footnote 6 is satisfied), by $A_r (\subset A)$ the set of its regular states. We define two partitions π_1, π_2 of A and a further partition π_3 of A_r in the following manner:

$\mathfrak{A}(\in A)$, $\mathfrak{A}'(\in A)$ are in a common class mod π_1 if there exists an integer s such that $0 \leqq s \leqq n-1$ and

$$\alpha_1 = \alpha'_{1+s},\ \alpha_2 = \alpha'_{2+s},\ \ldots,\ \alpha_{n-1} = \alpha'_{s-1},\ \alpha_n = \alpha'_s$$

where $\mathfrak{A} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$, $\mathfrak{A}' = \langle \alpha'_1, \alpha'_2, \ldots, \alpha'_n \rangle$.

$\mathfrak{A}(\in A)$, $\mathfrak{A}'(\in A)$ are in a common class mod π_2 if the inequalities $\alpha_i < \alpha_j$ and $\alpha'_i < \alpha'_j$ are equivalent to each other for every index pair i, j.

$\mathfrak{A}(\in A_r)$, $\mathfrak{A}'(\in A_r)$ are in a common class mod π_3 if there exists a non-negative real number t such that $\mathfrak{A}[+t] = \mathfrak{A}'$.

The partitions π_1 and π_2 generate a sublattice of the lattice of all partitions of A; similarly, π_1, π_2 and π_3 generate a sublattice in the partition lattice of A_r. Various questions (concerning both the lattice-theoretical properties and numerical problems) can be raised on the lattices generated in this manner.

Finally, we mention a problem of this character. Let A_h be the set of the states $\mathfrak{A} = \langle \alpha_1, \alpha_2, \ldots, \alpha_n \rangle$ fulfilling the three requirements:

(i) $\alpha_i = 1$ holds for exactly one index i,

(ii) the state is permitted,

(iii) whenever l and l' are two indices such that $1 \leqq l < l' \leqq n$, $P_l \notin \{P_i\} \cup \chi(P_i)$, $P_{l'} \notin \{P_i\} \cup \chi(P_i)$, then the inequalities $0 < \alpha_l < 1$, $0 < \alpha'_l < 1$, $\alpha_l \neq \alpha'_l$ hold.

It is easy to see that a randomly chosen element $\mathfrak{A}' = \langle \alpha'_1, \alpha'_2, \ldots, \alpha'_n \rangle$ of A satisfies $\mathfrak{A}'[+t] \in A_h$ with probability 1 where $t = \tau(1 - \max(\alpha'_1, \alpha'_2, \ldots, \alpha'_n))$.

Let us consider the graphs $G(3; 2)$, $G(4; 3)$, $G(5; 4)$, ..., $G(n; n-1)$, ... Starting with the general member $G(n; n-1)$ of this sequence, we denote by Ω_n the factor set $A_h^{(n)}/\pi_2$ where $A_h^{(n)}$ denotes the set A_h with respect to the graph $G(n; n-1)$. Ω_n is a finite set. On the other hand, let us define the subsets $A_h^{(n,x)}$ of $A_h^{(n)}$ so that $\mathfrak{A} \in A_h^{(n,x)}$ if and only if the regular state $\mathfrak{A}[+t]$ (with the least possible $t (\geqq 0)$)

(exists and) consists of x arcs ($x \leqq n/2$). The sets $A_h^{(n,x)}$ are pairwise disjoint (for varying x), moreover, $\mathfrak{A} \in A_h^{(n,x)}$, $\mathfrak{A}' \in A_h^{(n,x')}$, $\mathfrak{A} \equiv \mathfrak{A}' \pmod{\pi_2}$ imply $x = x'$. Let $\Omega_n^{(x)}$ be the subset of Ω_n which consists of the classes whose elements are in $A_h^{(n,x)}$. It is interesting to examine the asymptotical behaviour of the numerical function

$$f(n, x) = \frac{|\Omega_n^{(x)}|}{|\Omega_n|}.$$

(Evidently, $\sum_{x=1}^{[n/2]} f(n, x) \leqq 1$.) A discussion shows that the first values of $f(n, x)$ are:

x \ n	2	3	4	5	6
1	1	1	1/2	1/6	1/24
2			1/2	5/6	17/24
3					1/4

We conjecture that $f(n, [(n-1)/2])$ converges to 1 if n tends to the infinity.

MATHEMATICAL INSTITUTE OF THE
HUNGARIAN ACADEMY OF SCIENCES
BUDAPEST, HUNGARY

INSTITUT FÜR NACHRICHTENTECHNIK DER
TECHNISCHEN HOCHSCHULE
MÜNCHEN, BR DEUTSCHLAND

References

[1] ÁDÁM, A., Research problem 2—10 (Isomorphism problem for a special class of graphs), *Journal of Combinatorial Theory*, v. 2, 1967, p. 393.
[2] ÁDÁM, A., Simulation of rhythmic nervous activities, II. (Mathematical models for the function of networks with cyclic inhibition), *Kybernetik*, v. 5, 1968, pp. 103—109.
[3] DJOKOVIĆ, D. Ž., Isomorphism problem for a special class of graphs, *Acta Math. Acad. Sci. Hung.*, v. 21, 1970, pp. 267—270.
[4] ELSPAS, B. & J. TURNER, Graphs with circulant adjacency matrices, *Journal of Combinatorial Theory*, v. 9, 1970, pp. 297—307.
[5] KLING, U. & G. SZÉKELY, Simulation of rhythmic nervous activities, I. (Function of networks with cyclic inhibitions), *Kybernetik*, v. 5, 1968, pp. 98—103.

(Received April 17, 1970)

On measure-theoretic problems involving retrospective sequential functions

By A. MÁTÉ and J. SZŰCS

1. Introduction

The present paper can be regarded as self-contained inasmuch as it does not rely on outside repositories of references to an extent we would think underisable, yet, we think it should, in a proper setting, be considered as a continuation of, or an addendum to, L. Klukovits's paper [6] in the first issue of these Acta.[1] Otherwise it might be questionable whether the present paper, investigating pure measure-theoretic properties of certain types of functions, should appear in a periodical of *cybernetics*. Though these researches might have some potential applications to cybernetics and to the theory of automata, this aspect of the problem will not be elaborated here in detail. Perhaps some additional research in this area may be useful.

Yet, from a cybernetical angle, our study can be viewed as an investigation, on a theoretical level, of the relation between the behaviours of an automaton, firstly, if an arbitrarily large, but only a finite, number of input signs is successively fed into it and, secondly, if the feeding of input signs is repeated infinitely many times.

The approach to the characterization of the behaviours of automata is achieved through studying measure-theoretic properties of retrospective sequential functions, the precise definition of which, along with other definitions, may be found below. We shall point out that under certain natural conditions such functions are measurable, or, in more specific circumstances, they are even continuous. They map Borel sets onto sets which, in a natural sense, can be called Lebesgue-measurable; we shall give an example which illustrates that the image of a Borel set may be a non-Borel set, even in a very simple case.

2. Preliminary notions

Since the sections that follow depend to a considerable extent on different sets of notions we think it undesirable to accumulate here all the necessary definitions,

[1] The cited paper contains some inaccuracies and a considerable number of proofs in it are presented in an unnecessarily complicated way. Our observations concerning this matter are presented on p. 89.

and we collect here only the concepts that play a rôle throughout the whole of these notes.

The very concept around which all that follows centres is that of the *retrospective sequential function,* shortly RS function. The domain of such a function is the Cartesian product

$$X = \bigtimes_{n=1}^{\infty} X_n, \tag{2.1}$$

where X_n is intuitively interpreted as the set of *input signs* that can be fed into a given automaton at the nth stage. The range is a subset of the Cartesian product

$$Y = \bigtimes_{n=1}^{\infty} Y_n, \tag{2.2}$$

where Y_n is, intuitively, the set of *output signs* that can be emitted immediately after the digestion of the input sign absorbed at the nth stage. The automaton in question is to be imagined as having a fixed initial state that completely determines its reactions to sequences of input signs. The RS function assotiated with this automaton makes correspond to an infinite sequence of input signs the sequence of output signs the automaton emits while receiving the former.

This intuitive description of RS functions may easily be put in the form of a precise definition: a function f mapping the set X into Y is called an RS function if, under f, the first n signs of the image sequence are uniquely determined by the first n signs of the argument sequence for every positive integer n. This specific property of an arbitrary RS function f enables us to consider its restrictions to finite sequences. In notations, for every positive integer n put

$$X|n = \bigtimes_{k=1}^{n} X_k, \quad Y|n = \bigtimes_{k=1}^{n} Y_k; \tag{2.3}$$

the function $f|n$ sends, by definition, all sequences in $X|n$ to sequences in $Y|n$ in the same way as f handles these sequences as finite segments of infinite sequences. The notations

$$n|X = \bigtimes_{k=n+1}^{\infty} X_k, \quad n|Y = \bigtimes_{k=n+1}^{\infty} Y_k \qquad (n \geqq 0) \tag{2.4}$$

will sometimes prove useful, too.

In all our considerations, each of the sets X_n and Y_n will be vested with a *measurability structure*, by which we mean an ordered pair consisting of a set, the underlying space, and a σ-ring defined on this set, this latter being usually suppressed in the notational framework. The spaces $X|n$, $Y|n$, X and Y will be endowed with the measurability structures that are the products of their respective factors. The σ-rings determining these structures are the *minimal* ones generated by the sets of all rectangular sets or, in case of an infinite number of factors, by the sets of all *cylindrical* sets; here a subset of e.g. X is said to be cylindrical if, for some n, it is the Cartesian product of a set measurable in $X|n$ with the whole set $n|X$. (As seen, the notion of cylindrical sets already depends on the concept of measurability in products of finite numbers of spaces.)

A part of our study depends only on this measurability structure, without the need of actually considering measures. In other parts we have also to assume that certain measures are given on the described σ-rings. Sometimes we shall also consider the completions of these measures; these are, in general, defined on larger σ-rings, and this fact should carefully be kept in mind since, unless specifically mentioned our results may not hold for these extensions of the measures involved.

Another point to be stressed is that, up to Section 5, when measures are considered the measures on the product measurability structure of X and Y are never assumed to be the products of the measures on the respective factors; on the other hand, all our counter-examples are so constructed that, when measures on X and Y are at all considered, these are the products of the measures defined on the respective factors.

3. Measurability of RS functions

A very simple necessary and sufficient condition in order that an RS function be measurable in the sense that the whole inverse image of any measurable set is measurable is provided by

Theorem 3.1. An RS function f is measurable if and only if the functions $f|n$ are measurable for all positive integers n.

Proof. The "only if" part of the assertion is quite obvious and needs no comment whatsoever. Not much more complicated is the reverse implication, either. Indeed, observing that the inverse of a function does not spoil set-theoretical operations such as union and difference, the desired result immediately follows from the minimality restrictions, as imposed in Section 2, on product spaces.

Here, of course, the question might be raised how far these minimality restrictions are indispensable. The situation is, perhaps, illuminated by

Counter-example 3.1. The tacit assumption that in Theorem 3.1 measurability on Y means belonging to the minimal σ-ring generated by cylindrical sets cannot be omitted even in the simplest case.

This assertion is intended to be a vague intuitive description of the situation rather than a precise mathematical statement.

To consider a σ-ring, larger than the minimal one, of measurable sets in Y is senseless unless motivated in some suggestive way. Thus, what we are going to do will be to introduce measures on X_n and Y_n and consider the σ-rings that are the domains of the completions of the product measures on X and Y.

Now we actually set to describe the counter-example in question. Choose X_n and Y_n as coinciding with the two-element discrete space, containing the integers 1 and 2, such that the measure of each of its one-element subsets is 1/2. Let μ be the product measure on $X=Y$, and $\bar{\mu}$ its completion.

Define the RS function f mapping X into itself by the stipulation that for an arbitrary sequence $x=\{x_n\}_{n=1}^{\infty}$ the image $f(x)=y=\{y_n\}_{n=1}^{\infty}$ be such that $y_{2n-1}=1$ and $y_{2n}=x_n$. In compliance with the clause in Theorem 3.1, $f|n$ is clearly measurable with respect to the (minimal) measurability structure on $X|n=Y|n$, this being the discrete structure.

Now the function f being one-to-one, for any set $Z \subseteq X$ we have $Z = f^{-1}(f(Z))$. Here obviously $\mu^*(f(Z)) = 0$,[2] so $f(Z)$ is always measurable with respect to $\bar{\mu}$; thus, providing Z is chosen nonmeasurable with respect to $\bar{\mu}$, this set is an example for a measurable set whose full inverse image under f is not measurable. To make our considerations complete, we only have to point out that X has a subset which is not measurable with respect to $\bar{\mu}$; this, however, follows from the fact that X endowed with the measure $\bar{\mu}$ is essentially identical as a measure space to the interval (0,1) with the usual Lebesgue measure on it.

Finally, we remark that if the measurability structures of the spaces occurring here are coupled with certain topological ones then some simple conditions ensure the completion measurability of an RS function. These conditions and the proofs are analogous as in the cases of Lemma 4.2 and Theorem 4.3; the proofs in this case are even slightly simpler. We do not formulate these results here since they do not seem as natural as well as have no such a consequence as their counterparts in the next section (see Theorem 5.1 below).

4. Questions concerning the transportation of measurability

The question studied here, a much more difficult one than that envisaged in the previous section, concerns the transportation of measurability. More exactly, the problem to which we try to find an answer here is under what circumstances it is guaranteed that the image of a measurable set under an RS function is measurable again. This problem seems to depend much more on the topological structures of the spaces involved and on measures rather than on measurabilities than we experienced it in connection with the question studied in the previous section. Thereby we are forced to impose further restrictions on the spaces X_n and Y_n, and it will be convenient to do this along with a short description of the related concepts.

Throughout the rest of the paper we assume that, for each positive integer n, the spaces X_n and Y_n are endowed with topologies induced by metrics under which these spaces are complete and separable metric spaces. The topologies on $X|n$, $Y|n$, X, and Y are defined as the products of the topologies on their respective factors. As is well known, it is possible also on these spaces to introduce metrics with respect to which they are complete and separable metric spaces. For example, if we denote the distance function on X_n by ϱ_n then the function

$$\varrho(x, x') = \sum_{n=1}^{\infty} 2^{-n} \frac{\varrho_n(x_n, x_n')}{1 + \varrho_n(x_n, x_n')} \tag{4.1}$$

serves as such a metric on X. Since our main concern is the possibility of the introduction of such metrics rather than the particular distance functions chosen, we shall suppress these latter in the notational framework; nevertheless, we might refer to the spaces involved as metric when it were enough to say metrizable in a certain way.

Measures on these spaces will also be considered. μ and ν will denote two Borel measures on X and Y, respectively; here a Borel measure, by definition, is a

[2] The asterisk * in superscript indicates outer measure.

σ-finite measure explained on the σ-ring of all Borel sets, this being the smallest σ-ring generated by e.g. all closed sets. $\bar{\mu}$ and $\bar{\nu}$, called Lebesgue measures, will denote the completions of μ and ν. In an obvious way we can also define the restrictions μ_n, ν_n, $\mu|n$ and $\nu|n$, of the measures μ and ν, to the spaces X_n, Y_n, $X|n$ and $Y|n$, respectively; e.g. for a Borel set $H \subseteq X|n$ put $\mu|n(H) = \mu(H \times n|X)$. It is usually not assumed that μ and ν are the products of the measures μ_n and ν_n.

A simple condition in order that an RS function under the circumstances specified above be in a sense measurability transporting is

Theorem 4.1. If the RS function f is such that $f|n$ is Borel-measurable for any positive integer n then f maps all Borel sets onto Lebesgue-measurable ones.

Here the Borel measurability of a function means that the whole inverse image under it of a Borel set is again a Borel set.

Proof. It follows from Theorem 3.1 that, under the given assumption, f is itself Borel-measurable; so it maps Borel sets onto analytic or, by another name, Suslin sets (see e.g. [2, 2.2.14 on p. 70]). As is well known, every analytic set is Lebesgue-measurable (see [2, 2.2.1.2. Theorem on p. 68]), which completes the proof.

To illustrate how far the assumption in this theorem is necessary and whether the conclusion goes far enough we give several counter-examples. The assumption that $f|n$ is Borel-*measurable* when we want to prove that f is *measurability transporting* may seem artificial; Counter-example 4.1, however, shows that it is not enough to suppose that $f|n$ is measurability transporting. Counter-example 4.2 shows that the given assumption does not ensure that f maps every Lebesgue-measurable set onto a Lebesgue-measurable set. It is not certain, either, that, under this assumption, the image of every Borel set is a Borel set; this will be shown later, in Counter-example 5.1.

Counter-example 4.1. The assumption that, for any positive integer n, the function $f|n$ maps every subset of $X|n$ onto a Borel set of $Y|n$ does not imply the conclusion of Theorem 4.1.

In the example we are going to give, the validity of the assumption that $f|n$ maps every set onto a Borel set will be ensured by choosing as $Y|n$ a finite discrete space, every subset of which is, of course, a Borel set. To elaborate, choose the spaces $X_2, X_3, \ldots$ and $Y_1, Y_2, \ldots$ as identical to a two-element discrete space, with points 1 and 2, such that either of its one-element subsets is of measure 1/2. Explain the Borel measure on Y as the product of those defined on the spaces Y_n; define X_1 as identical to Y, with the same topology and measure defined on it. Finally, choose the Borel measure on X as identical to the product of the measures explained on the spaces X_n.

Now choose as f_1 an arbitrary function from X_1 into Y whose range is not Lebesgue-measurable. Then the function f that makes correspond to every $x = \{x_n\}_{n=1}^{\infty}$ the sequence $f_1(x_1)$, independently of x_n for $n \geqq 2$, is an RS function that satisfies our requirements, yet it does not map the whole set X onto a Lebesgue-measurable set.

Counter-example 4.2. The assumption of Theorem 4.1 does not assure that the image under f of a Lebesgue-measurable set is Lebesgue-measurable.

Using the same spaces X and Y as in the counter-example just before, define the RS function f from X into Y so that it send a sequence $x=\{x_n\}_{n=1}^{\infty}$ to a sequence y that is identical to $x_1 \in X_1 = Y$, independently of the values of x_n for $n \geqq 2$. It is obvious that the function $f|n$ is Borel-measurable for each positive integer n. If we select an arbitrary set $X_1' \subseteq X_1 = Y$ that is not Lebesgue-measurable, then the image under f of the set $X' = X_1' \times \{1\} \times \{1\} \times \cdots \subseteq X$ is X_1'; now the set X' is Lebesgue-measurable, since its outer Borel measure is zero; yet its image is not so.

It seems to be a rather difficult problem to give conditions that subtly differentiate between cases when Lebesgue-measurable sets are mapped onto this same kind of sets and when they are, possibly, not. Nevertheless, the following two results, however rough they are, point in this direction.

Lemma 4. 2. Assume that the space X is locally compact and that the measure of every compact set in X is finite.[3] *Suppose, furthermore, that the RS function f is such that, n running over all positive integers, the function $f|n$ is Borel-measurable, and moreover, with some positive constant C,*

$$(\nu|n)^*(f|n(G_n)) \leqq C\mu|n(G_n) \tag{4.2}$$

holds for any open set G_n in $X|n$.[2,4] *Then f maps all Lebesgue-measurable sets onto Lebesgue-measurable ones.*

We remind that the local compactness of X is an additional assumption and, as said at the beginning of this section, all the spaces considered here are assumed to be complete and separable metric spaces. We also recall that in order for the product of topological spaces to be locally compact it is necessary and sufficient that all factors, with the possible exception of a finite number of them, be compact and the non-compact factors be locally compact (see [1, Proposition 11 on p. 65]). Taking this into account, we can reformulate the lemma accordingly.

The point in adopting (4. 2) as an assumption of the lemma is that it ensures that the mapping f does not increase the outer measure of any set more than C times; thus, in particular, it maps sets of zero outer measure onto sets also of zero outer measure, and this implies the assertion of the lemma.

Proof. Since every Lebesgue-measurable set can be represented as the union of two sets of which one is Borel-measurable and the other is of zero outer Borel measure, the assertion will follow from the previous Theorem if we show that f maps every set of zero outer Borel measure onto a set also of zero outer Borel measure. To accomplish this, let Z be an arbitrary subset of zero outer Borel measure of X. Since in a locally compact and separable metric space every Borel set is a Baire set, and a Baire measure on a locally compact space is always regular, pro-

[3] Usually, Borel measures are considered on locally compact spaces and it is traditionally included in their definition that they are finite on compact sets. Here we cannot conform to this tradition since it would involve some unnecessary restrictions on the measures considered.

[4] It is enough to require the conditions depending on n in this lemma and in the next theorem only for large enough integers, though the statements so obtained are not real generalizations since they easily follow from the assertions, analogous to the given ones, arrived at by grouping the factors of X as $(X_1 \times \cdots \times X_k) \times X_{k+1} \times \ldots$, and those of Y similarly. Moreover, it does not represent any real change to require only for large n's that $f|n$ is Borel-measurable since then the same follows for every positive integer n.

vided it is finite on compact sets (see [3, Theorem E on p. 218 and Theorem G on p. 228]), for an arbitrarily small positive ε there exists an open subset G of X with $\mu(G)<\varepsilon$ such that $Z\subseteqq G$.

n being an arbitrary positive integer, let U run over all the open subsets of $X|n$, and write

$$G_n = \bigcup\{U\colon U\times n|X \subseteqq G\}. \tag{4.3}$$

Then G_n is an open subset of $X|n$, and, obviously, we have

$$G_n\times n|X \subseteqq G_{n+1}\times(n+1)|X, \tag{4.4}$$

and

$$G = \bigcup_{n=1}^{\infty} G_n\times n|X; \tag{4.5}$$

moreover, on account of (4.2), we obtain

$$\begin{aligned} &\nu^*\big(f(G_n\times n|X)\big) \leqq \nu^*\big(f|n(G_n)\times n|Y\big) \leqq {}^{5} \\ &\leqq (\nu|n)^*\big(f|n(G_n)\big) \leqq C\mu|n(G_n) = C\mu(G_n\times n|X). \end{aligned} \tag{4.6}$$

Here all the sets are actually Lebesgue-measurable; for the first one this is stated by the previous theorem. For the second and the third one this fact will not be used, so we do not go into details and only note that in the proofs similar arguments involving analytic sets may be used. So, writing $\bar{\nu}$ instead of ν^*, the last four centred lines imply

$$\bar{\nu}\big(f(G)\big) = \bar{\nu}\left(\bigcup_{n=1}^{\infty} f(G_n\times n|X)\right) \leqq C\mu(G). \tag{4.7}$$

Since G, by its choice, includes Z, we obtain

$$\nu^*\big(f(Z)\big) \leqq \bar{\nu}\big(f(G)\big) \leqq C\mu(G)<C\varepsilon. \tag{4.8}$$

Since ε can be selected arbitrarily small we have $\nu^*\big(f(Z)\big)=0$, which completes the proof of the lemma.

Though in the proof of this lemma we made a relevant use of the local compactness of X, this assumption can actually be dispensed with if we stipulate that μ is totally finite, and we can derive

Theorem 4.3. Assume that the measure μ is totally finite and the RS function f mapping X into Y is such that, n running over all positive integers, $f|n$ is Borel-measurable, and, moreover, with some positive constant C,

$$(\nu|n)^*\big(f|n(B_n)\big) \leqq C(\mu|n)(B_n) \tag{4.9}$$

holds for any Borel set B_n in $X|n$.[6] Then f maps all Lebesgue-measurable sets onto Lebesgue-measurable ones.

[5] An easy argument invoking analytic sets shows that here actually equality holds, and would continue to hold even if the set $f|n(G_n)$, which could easily be shown to be Lebesgue-measurable, were replaced by any subset of $Y|n$. This is, however, irrelevant for our purposes.

[6] See footnote [4].

Proof. Any complete and separable metric space is either countable or of continuum cardinality (see [4, IV on p. 320]), therefore to each X_n there is a compact and separable metric space X_n' of the same cardinality. Since a compact metric space is necessarily complete, on account of a well-known result (see [7, 2° on p. 358]), there exists a one-to-one function g_n mapping X_n' onto X_n that is Borel-measurable in both ways.[7]

Now define the function g from X', this being the product space $X_1' \times X_2' \times \cdots$, onto X componentwise, i.e. put

$$g(x_1' x_2' \ldots) = g_1(x_1') g_2(x_2') \ldots \tag{4.10}$$

Then g is obviously one-to-one and Borel-measurable in both ways. Determine the Borel measure μ' on X' so that g be also measure-preserving. Instead of the RS function $f(x)$ we may consider the RS function $f(g(x))$ mapping X' into Y and the assertion of the theorem directly follows from the previous lemma on account of the compactness of X'.

5. The problem of transportation of measurability in a special case[8]

In this concluding section we shall be concerned only with the following special case: $X_1, X_2, \ldots$ and $Y_1, Y_2, \ldots$ are all identical discrete spaces, with a finite number $N \geqq 2$ of points, and the Borel measures μ_n and ν_n on X_n and Y_n, respectively, are such that the measure of a one-point set is $1/N$; finally we determine μ and ν as the products of the measures μ_n and ν_n, respectively. It is easy to see that in this case all RS functions are continuous. Moreover, the assumptions of Theorem 4.3 are satisfied for any RS function f. Indeed, $f|n$ is Borel-measurable for many reasons, e.g. since it is defined on a discrete space. The assumption (4.9) is also satisfied with $C=1$. The argument showing this is simply that the measure of a set in $X|n$ is a constant multiple of the number of the (finite) sequences contained in it; this measure may only decrease by performing the mapping $f|n$, as a consequence of the phenomenon that two different sequences may have a common image. So in this case we have

Theorem 5.1. Every RS function maps all Lebesgue-measurable sets onto Lebesgue-measurable sets.

The proof of this theorem does not, in fact, need such sophisticated tools as have been used to accomplish it. Namely, cylindrical sets being compact, their image is also compact, and the considerations based on (4.9) that establish the full strength of the theorem are largely simplified by the fact that the measure μ is the product of the measures μ_n.

[7] Actually, the phrase "in both ways" need not be added; namely, it is easy to show that if a one-to-one function which maps a complete and separable metric space onto another is Borel-measurable then its inverse is so, too.

[8] The more ambitious reader is advised also to consult L. Kalmár's paper [5], where a case with generality lying between that of the cases dealt with in this and the previous sections is studied from a somewhat different angle.

The paper of Klukovits [6], which spurred us to investigate measure-theoretic problems involving RS functions, considered only the particular case studied in this section. We take now a closer look at the relationship between some of his results and some of our considerations here. It will turn out, in particular, that many of the proofs in his paper can be radically shortened by using some simple devices of topology.

Theorem 1 in the paper in question says that two RS functions differing only on a set of Lebesgue measure zero coincide. The functions in question being continuous, this is naturally true, since in this case the measure of any non-empty open set is positive, and thus the set of coalescence is dense.

Lemma 1 claims that if the range of an RS function is Lebesgue-measurable then the image under it of any Lebesgue-measurable set is so, too. This is a consequence of Theorem 5. 1 of ours, though the assumption on the range is superfluous. Independently of our result just referred to, the fact that the range of f is measurable is obvious since, being a continuous image of a compact set, it is in fact compact. In the proof of the cited lemma, the author leaves to the reader the verification of the assertion that, under the assumption of the measurability of the range, the image of every cylindrical set is Lebesgue-measurable. Cylindrical sets being compact, the task of the reader in proving this is indeed not difficult. He may, however, be annoyed by not finding a way to weave the measurability of the range into his considerations.

Theorem 2 states that an RS function f is measure-preserving if and only if it is an onto mapping. Here the proof of the necessity can be contraced into a few lines as follows: the range of an RS function f being compact, its complement is open. The stipulation that f is measure-preserving implies that the measure of this open set is zero; so it is empty, which means that f is indeed onto.

Lemma 2 asserts that the range of a "finite-state RS function without one-to-one state" is Lebesgue-measurable. (The phrase is not an exact quotation; the author writes fsrsf for what we called a finite-state RS function.) In whatever way the above attributes may specify the notion of RS function, the range is a continuous image of a compact set, therefore it is compact, and so measurable.

Theorem 4 announces that the range of any "fsrsf" is Lebesgue-measurable. Actually, the range is again compact.

The concluding result of these notes is

Counter-example 5. 1. There exists an RS function under which the image of a certain Borel set is not a Borel set.

In order to give such an example, for every positive integer n, identify the spaces X_n and Y_n with the discrete space consisting of the points 1 and 2 and choose N_n as the discrete space containing exactly the positive integers; let N be the topological product of the spaces N_n.

Decompose the space $X = Y$ as

$$X = Z_1 \times Z_2 \times Z_3, \tag{5. 1}$$

where

$$Z_j = \bigtimes_{k=0}^{\infty} X_{3k+j} \qquad (j=1,2,3). \tag{5.2}$$

Let Z_j' be the subspace of Z_j which consists precisely of the sequences that contain an infinite number of ones. It is easy to see that Z_j' is homeomorphic to N. Indeed, a homeomorphism between these spaces can be described as follows: for an arbitrary element z of Z_j' form groups of consecutive elements constituting z so that each group consist purely of 2's except that it end with a 1. The numbers of elements in each group, in turn, form a sequence of positive integers which, if considered as the image of z, determines a homeomorphism between Z_j' and N. Denote this homeomorphism from Z_3' onto N by h; for a sequence $z \in Z_3'$ denote by $h_n(z)$ the nth integer forming the sequence $h(z)$.

Now, following closely the lines of the example for an analytic set that is not a Borel set given in [2, 2.2.11 on p. 68], our example can be described as follows:

Choose a countable open base $U(n)$ of $Z_1' \times Z_2'$ and define a closed subset of $Z_1' \times Z_2' \times Z_3'$ by

$$C = \left\{(z_1, z_2, z_3):\ (z_1, z_2) \notin \bigcup_{n=1}^{\infty} U\big(h_n(z_3)\big)\right\}. \tag{5.3}$$

It is obvious that all the closed subsets of $Z_1' \times Z_2'$ occur among the slices

$$C_{z_3} = \{(z_1, z_2):\ (z_1, z_2, z_3) \in C\}. \tag{5.4}$$

Now, on the one hand,

$$S = \{(z_1, z_3):\ (z_1, z_2, z_3) \in C \quad \text{for some} \quad z_2\} \tag{5.5}$$

is an analytic subset of $Z_1' \times Z_3'$; and, on the other hand, the slices

$$S_{z_3} = \{z_1:\ (z_1, z_3) \in S\} = \{z_1:\ (z_1, z_2) \in C_{z_3} \quad \text{for some} \quad z_2\} \tag{5.6}$$

run over all the analytic subsets of Z_1', since Z_3' is homeomorphic to N (see [2, 2. 2. 10 on p. 65]).

Finally, the intersection of S with the diagonal of $Z_1' \times Z_3'$, the latter being a set closed in the relative topology, is an analytic subset of $Z_1' \times Z_3'$. The projection of this set into Z_1',

$$T = \{z_1:\ (z_1, z_3) \in S \quad \text{and} \quad z_1 = z_3\}, \tag{5.7}$$

is therefore analytic; now the complement of T, $Z_1' - T$, is not analytic since it does not occur among the sets S_{z_3}. Indeed, the assumption $Z_1' - T = S_{z_3}$ is equivalent to saying that for any $z_1 \in Z_1' = Z_3'$

$$(z_1, z_1) \notin S \quad \text{holds if and only if} \quad (z_1, z_3) \in S. \tag{5.8}$$

This is, however, certainly not true for $z_1 = z_3$, implying that $Z' - T$ is not analytic, as asserted. Thus, since the complement of a Borel set is again a Borel set, and so a fortiori an analytic set, we may conclude that T is not a Borel set.

To complete our example, we shall determine an RS function that maps a Borel subset of $Z_1' \times Z_2' \times Z_3' \subseteq X$ essentially onto T. To this end, define a diagonal

plane of the set C:

$$D=\{(z_1, z_2, z_3)\colon (z_1, z_2, z_3)\in C \quad \text{and} \quad z_1=z_3\}, \tag{5. 9}$$

and consider the function f from $Z_1\times Z_2\times Z_3$ (without accents ′!) into itself such that

$$f(z_1, z_2, z_3)=(z_1, c, c), \tag{5. 10}$$

where c is an arbitrary but fixed sequence in $Z_2=Z_3$, e.g. $c=111\ldots$. In view of (5. 1), f can be rewritten as an RS function mapping $X=Y$ into itself.

Now the set D, being a set closed in the relative topology on the Borel set $Z_1'\times Z_2'\times Z_3'$, is itself a Borel set, and its image under f, the set $T\times\{c\}\times\{c\}$, was proved to be a non-Borel set just before.

MATHEMATICS DEPARTMENT
UNIVERSITY OF SZEGED, HUNGARY

References

[1] BOURBAKI, N., *Topologie générale*, chapitre I—II, Hermann and Co., Paris, 1940.
[2] FEDERER, H., *Geometric measure theory*, Springer Verlag, Berlin—Heidelberg—New York, 1969.
[3] HALMOS, P. R., *Measure theory*, eleventh printing, Van Nostrand, Princeton — New Yersey—Toronto—London—New York, 1966.
[4] HAUSDORFF, F., *Grundzüge der Mengenlehre*, Chelsea Publishing Co., New York, 1949.
[5] KALMÁR, L., Über arithmetische Funktionen von unendlich vielen Variablen, welche an jeder Stelle bloß von einer endlichen Anzahl von Variablen abhängig sind, *Colloq. Math.*, v. 5, 1957, pp. 1—5.
[6] KLUKOVITS, L., Some measure problems concerning the retrospective sequential functions, *Acta Cybernetica*, Szeged, v. 1, 1969, pp. 27—40.
[7] KURATOWSKI, C., *Topologie I*, second edition, Warszawa—Wrocław, 1948.

(Received March 19, 1970)

Versuch einer automatentheoretischen Beschreibung von Selektionsprozessen

Von K. BELLMANN und M. GÖSSEL

Wir betrachten eine Population P von Individuen mit einer beliebigen meßbaren Eigenschaft x. Die Phänotypenwerte seien etwa normalverteilt. Mit $\bar{x}$ bezeichnen wir das Populationsmittel von x.

Zur genaueren phänomenologischen Beschreibung von Selektionsprozessen werden zweckmäßig die Begriffe "Selektionsdifferenz" $S(t_i)$ und "Response" $R(t_i)$ benutzt, die durch

$$S(t_i) = \bar{x}_E(t_i) - \bar{x}(t_i)$$
$$R(t_i) = \bar{x}(t_{i+1}) - \bar{x}(t_i) \tag{1}$$

definiert sind. Dabei sind $\bar{x}(t_i)$ der Mittelwert von x in der i-ten Generation der Population, $\bar{x}_E(t_i)$ der Mittelwert von x der aus der i-ten Generation selektierten Eltern, und t charakterisiert die diskrete Zeit, die durch die Generationsfolge gegeben ist.

Wir benötigen noch die Größe $\tilde{R}(t_i)$, die durch

$$\tilde{R}(t_i) = \bar{x}(t_{i+1}) - \bar{x}(t_0) \tag{2}$$

definiert sei. $\tilde{R}(t_i)$ stellt den Gesamtresponse von der 0-ten bis zur i-ten Generation dar.

Aus (1) und (2) folgt unmittelbar

$$\tilde{R}(t_k) = \sum_{i=0}^{k} R(t_i) \tag{3}$$

und damit auch $\tilde{R}(t_0) = R(t_0)$. (t_k gibt den Endpunkt des Selektionsprozesses an.)

Wir betrachten im folgenden den Selektionsprozeß als Ganzes und schließen dabei auch Intervalle ohne Selektion ein, für die $S(t_j) = 0$ gilt. Relaxation der Selektion ist also ausdrücklich zugelassen.

Wenn durch den Züchtungsprozeß ein möglichst großer Mittelwert $\bar{x}$ angestrebt wurde, vermindert sich $\bar{x}$ im allgemeinen ohne künstliche Selektion im Verlaufe der Zeit. Dieser Prozeß wird im folgenden als Selbstreduktion von P bezeichnet.

Die Änderung des Mittelwertes $\bar{x}$ von Generation zu Generation wird durch zwei sich überlagernde Prozesse bestimmt:

1. Veränderung von $\bar{x}$ durch künstliche Selektion (d.h. $S \neq 0$).
2. Veränderung von $\bar{x}$ ohne (künstliche) Selektrion, d. h. durch Selbstreduktion (d.h. $S = 0$).

Wir setzen voraus, daß sich beide Prozesse linear überlagern. Nach diesen Vorbereitungen soll der Selektionsprozeß von einem abstrakteren Standpunkt aus untersucht werden, was uns zu einem automatentheoretischen Modell dieses Prozesses führen soll. Eine Folge von Werten $S(t_i)$, die auf das System Population als Input einwirkt, verursacht eine Folge von Werten $R(t_i)$, die die man als Output des Systems ansehen kann, wobei wir, wie oben erläutert, t_i als diskrete Zeit des Systems auffassen. (Das Vorgehen in der praktischen Züchtung besteht darin, daß eine bestimmte Anzahl Generationen lang eine Selektion bestimmter Intensitäten durchgeführt wird. Danach erfolgt die Nutzung ohne Selektion. Ein solches Vorgehen wird durch eine Inputfolge $C_1 C_2 \ldots C_K 0\,0\,0 \ldots$ beschrieben. Dabei sind die C_i $(i=1, 2, \ldots, K)$ die entsprechenden Selektionsintensitäten.) Eine derartige Input-Output-Beziehung wird mathematisch adäquat durch einen abstrakten Automaten beschrieben.

Wir betrachten hier wegen der vorausgesetzten linearen Überlagerung der unter 1. und 2. aufgeführten Prozesse einen Spezialfall des abstrakten Automaten, den linearen Automaten [2, 3, 5]. Außerdem können wir uns auf eindimensionalen Input und Output beschränken, da Selektionsdifferenz und Response skalaren Charakter haben.

Ein linearer Automat wird durch die Überführungs- und Ergebnisfunktion

$$\begin{aligned} \mathbf{z}(t_{i+1}) &= \mathbf{A}\mathbf{z}(t_i) + \mathbf{B}x(t_i) \\ y(t_i) &= \mathbf{C}\mathbf{z}(t_i) + \mathbf{D}x(t_i) \end{aligned} \tag{4}$$

beschrieben. $\mathbf{z}$ ist ein n-dimensionaler Zustandsvektor, x ein eindimensionaler Input (vektor), y ein eindimensionaler Output (vektor). $\mathbf{A}$, $\mathbf{B}$, $\mathbf{C}$, $\mathbf{D}$ sind Matrizen entsprechender Dimension mit konstanten Matrixelementen.

Die Überführungsfunktion bestimmt aus dem Zustand $\mathbf{z}(t_i)$ und dem Input $x(t_i)$ den Folgezustand $\mathbf{z}(t_{i+1})$. Die Ergebnisfunktion bestimmt aus dem Zustand $\mathbf{z}(t_i)$ und dem Input $x(t_i)$ den zugehörigen Output $y(t_i)$. Die Dimension n des Zustandsvektors heißt auch die Dimension des linearen Automaten.

Der Zustand $\mathbf{z}(t_j)$ ist durch den Initialzustand $\mathbf{z}(t_0)$ und die auf den Automaten wirkende Inputfolge $x(t_0), x(t_1), x(t_2), \ldots$ durch

$$\mathbf{z}(t_j) = \mathbf{A}^j \mathbf{z}(t_0) + \sum_{i=0}^{j-1} \mathbf{A}^{j-i-1} \mathbf{B} x(t_i) \tag{5}$$

bestimmt.

Für den Output gilt entsprechend

$$y(t_j) = \mathbf{C}A^j \mathbf{z}(t_0) + \sum_{i=0}^{j} \mathbf{M}(t_{j-i}) x(t_i) \tag{6}$$

mit

$$\mathbf{M}(t_k) = \begin{cases} \mathbf{D} & \text{für} \quad k=0 \\ \mathbf{C}\mathbf{A}^{k-1}\mathbf{B} & \text{für} \quad k>0 \end{cases}$$

Ist die Dimension n des Automaten endlich, dann gibt es eine endliche Zahl $r \leqq n$, so

daß für alle m

$$\mathbf{M}(t_{m+r+1}) = \alpha_1\mathbf{M}(t_{m+1}) + \alpha_2\mathbf{M}(t_{m+2}) + \cdots + \alpha_r\mathbf{M}(t_{m+r}) \tag{7}$$

gültig ist.

Der Wert von r kann aus der Folge der Matrizen

$$[\mathbf{M}(t_1)] \quad \begin{bmatrix} \mathbf{M}(t_1) & \mathbf{M}(t_2) \\ \mathbf{M}(t_2) & \mathbf{M}(t_3) \end{bmatrix} \quad \begin{bmatrix} \mathbf{M}(t_1) & \mathbf{M}(t_2) & \mathbf{M}(t_3) \\ \mathbf{M}(t_2) & \mathbf{M}(t_3) & \mathbf{M}(t_4) \\ \mathbf{M}(t_3) & \mathbf{M}(t_4) & \mathbf{M}(t_5) \end{bmatrix} \ldots \tag{8}$$

als der größte Rang entnommen werden, der einmal erreicht, bei Fortführung der Folge erhalten bleibt.

Die Kenntnis der Beziehung (7) erlaubt, explizit eine Realisierung des linearen Automaten anzugeben. (Minimierungsprobleme bei linearen Automaten, auf die wir hier nicht eingehen, sind ausführlich z.B. in [2, 4] untersucht.)

Eine mögliche Realisierung ist durch

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ 0 & 0 & 0 & \ldots & 0 \\ . & . & . & \ldots & . \\ 0 & 0 & 0 & & 1 \\ \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_r \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} \mathbf{M}(t_1) \\ \mathbf{M}(t_2) \\ \mathbf{M}(t_3) \\ . \\ . \\ \mathbf{M}(t_r) \end{bmatrix} \tag{9}$$

$$\mathbf{C} = [1 \quad 0 \quad 0 \quad \ldots 0] \qquad \mathbf{D} = \mathbf{M}(t_0)$$

gegeben.

Befindet sich das zu realisierende System im Initialzustand $\mathbf{z}(t_0) = 0$, so ist $\mathbf{M}(t_k)$ nach (7) als (Impuls-) Antwort auf die Inputfolge 1 0 0 0 ... bestimmt. Auf die Folge C_1 0 0 0 ... antwortet das System mit $C_1 \cdot \mathbf{M}(t_k)$ (C_1 gibt wieder die Selektionsintensität an).

Auf die Inputfolge 0 0 0 0 ... reagiert das System vom Initialzustand $\mathbf{z}(t_0) = 0$ mit dem Output 0 0 0 0 ..., wie man ebenfalls unmittelbar aus (7) abliest.

Wir nehmen an, daß sich das System Population zunächst im genetischen Gleichgewicht befindet. Der dem System zuzuordnende Initialzustand ist dann $\mathbf{z}(t_0) = 0$, da die Population in diesem Falle auf die Inputfolge der Selektionsdifferenzen 0 0 0 0 ... mit der Response-Outputfolge 0 0 0 0 ... reagiert. Um in einfacher Weise das Modell für das genetische System bestimmen zu können, ist das Verhalten der Population auf die Inputfolge C_1 0 0 0 ... zu untersuchen. Ist dann das Modell bestimmt, so läßt sich eine Reaktion auf eine beliebige, etwa praktisch vorliegende Inputfolge vorhersagen. Die Response-Impuls-Antwort $R_I(t_k)$ erhält man, wenn man auf die Population die Selektions-Inputfolge 1 0 0 0 ... einwirken läßt. In der Praxis sind verschiedenartige Response-Impuls-Antworten möglich.

Wenn wir von zufälligen Mutationen absehen können, ist $R_I(t_k)$ eine monotone nicht wachsende Funktion mit

$$\underset{k \to \infty}{R_I(t_k)} \to 0.$$

D.h. die Änderungen des Populationsmittel von Generation zu Generation ohne

weitere Selektion werden immer geringer. Es scheint sinnvoll, $R_I(t_k)$ durch

$$R_I(t_k) = \begin{cases} b_k & \text{für} \quad k < j \\ b_j e^{-a(k-j)} & \text{für} \quad k \geqq j \end{cases} \tag{10}$$

zu approximieren.

Die Beziehung (8) nimmt dann für alle $m \geqq 0$ die Form

$$R_I(t_{m+j+1}) = 0 \cdot R_I(t_{m+1}) + 0 \cdot R_I(t_{m+2}) + \cdots + 0 \cdot R_I(t_{m+j-1}) + e^{-a} \cdot R_I(t_{m+j}) \tag{11}$$

an, und aus (10) und (11) erhalten wir

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 \ldots 0 \\ 0 & 0 & 1 \ldots 0 \\ \cdot & \cdot & \cdot \ldots \cdot \\ \cdot & \cdot & \cdot \ldots \cdot \\ 0 & 0 & 0 \ldots 1 \\ 0 & 0 & 0 \ldots e^{-a} \end{bmatrix} \quad \mathbf{B} = \begin{bmatrix} b_1 \\ b_2 \\ \cdot \\ \cdot \\ b_j \end{bmatrix} \tag{12}$$

$$\mathbf{C} = [1 \quad 0 \quad 0 \ldots 0] \qquad \mathbf{D} = b_0.$$

Da $R_I(t_k)$ eine monotone nicht wachsende Funktion ist, gilt für $k \neq 0$

$$b_k \leqq 0,$$

wenn $b_0 > 0$ ist.

Ein Beispiel ist in [1] betrachtet.

Herrn Prof. Dr. W. Kämmerer möchten wir für Diskussionen herzlich danken.

ZENTRALINSTITUT FÜR
KYBERNETIK UND INFORMATIONSPROZESSE
DDR – 1199 BERLIN
RUDOWER CHAUSSEE 5

Literatur

[1] BELLMANN, K. & M. GÖSSEL, Zur automatentheoretischen Beschreibung von Selektionsprozessen, *Biometrische Zeitschrift*, J. 13, 1971, S. 79—86.
[2] GILL, A., *Linear Sequential Circuits*, Mac-Graw-Hill, New York, 1966.
[3] HO, B. L. & R. E. KALMAN, Effective construction of linear state-variable modells from input/output functions, *Regelungstechnik*, J. 14, 1966, S. 545—548.
[4] KALMAN, R. E., Irreducible realizations and the degree of a rational matrix, SIAM, J. 13, 1965, S. 520—544.
[5] KÄMMERER, W., *Einführung in mathematische Methoden der Kybernetik*, Akademie-Verlag, Berlin, (in Vorbereitung).

Zusammenfassung. Es wird ein Selektionsprozeß als linearer Automat beschrieben.
Abstract. The process of selection is considered as a linear automaton.
Résumé. Рассматривается селекционный процесс как линейный автомат.

(Eingegangen am 4. Mai 1970)

On a minimization algorithm for Boolean functions

By F. MÓRICZ

1. Logical design of circuits with a single output, using solid-state integrated circuits, as primitive elements, leads to several non-traditional optimization problems which require to find, for any given Boolean function, a formula (or all formulas), composed from fixed Boolean functions as primitive elements, representing the given function and minimal with respect to a given objective function.

In this note our purpose is to present an algorithm which (provided the objective function satisfies a simple restriction) obviously leads to the exact solution of the problem and, moreover, a limited number of its steps which can be implemented on a digital computer delivers a fairly good approximative solution. (Of course, the approximation will be the better the more steps are performed and thus a larger computer may provide a better approximation.)

In view of the general nature of the problem, the algorithm will be formulated here in a very general and comprehensive way which, for each practical application, must be specified in accordance with the given particular primitive elements and objective function.

2. Assume $\Theta = \{\vartheta_1, \vartheta_2, \ldots\}$ is a functionally complete system[1] of a finite number of Boolean functions, or in other words, of logical operations in a general sense where the number of operands of each operation, i.e. the number of arguments of each function ϑ_i, can be arbitrary. Let $X = \{x_1, x_2, \ldots\}$ be the (countable) set of the available Boolean variables. (In an actual realization of the algorithm to be formulated we have, of course, to limit ourselves to a finite set of variables.)

The *formulas* considered here are all those composed of the constants 0 (falsity) and 1 (truth) and of the given Boolean variables by means of operations belonging to Θ.[2] We say that two formulas, F and G, are *identically equal* or that the equality $F = G$ is an *identity* if for every valuation their values coincide; here by *valuation* we mean a mapping which makes correspond to each of the variables belonging to X one of the constans 0 and 1.

A *substitution instance* of a formula F is, by definition, any formula that can be obtained by replacing all occurrences in F of some variables, say $x_{i_1}, \ldots, x_{i_r}$ ($i_1, \ldots, i_r$ are different positive integers), by an equal number of formulas, say

[1] See, e.g., A. Ádám [1], Chapter 4.

[2] In other words, the following symbol strings are called formulas: (i) 0 and 1; (ii) any element of X; (iii) $\vartheta(F_1, \ldots F_r)$ where $\vartheta \in \Theta$ is a Boolean function with r variables and $F_1, \ldots, F_r$ are formulas; (iv) nothing else.

$H_1, \ldots, H_r$, respectively; for the formula thus obtained we introduce the notation:

$$F(x_{i_1} := H_1; \ldots; x_{i_r} := H_r).$$

Here the variables $x_{i_1}, \ldots, x_{i_r}$ and formulas $H_1, \ldots, H_r$ are called *substituends* and *substituents*, respectively. Analogously, by a substitution instance of an identity $F=G$ we mean any equality of the form

$$F(x_{i_1} := H_1; \ldots; x_{i_r} := H_r) = G(x_{i_1} := H_1; \ldots; x_{i_r} := H_r),$$

which is easily shown to be also an identity.

Consider now a formula F and an identity $G=H$ the left-hand side G of which is a (proper or non-proper) subformula of F. For each occurrence of G in F decide independently whether it is to be left unchanged or replaced by H and proceed accordingly. Any of the formulas that can be obtained in this way are said to arise from F by a *direct application* of the identity $G=H$; the number of such formulas is 2^s, where s denotes the number of occurrences of G in F. If the formula F is left untouched we speak of a *trivial* direct application. If G is not a subformula of F then the only possible direct application is the trivial one.

The formula F' is said to be obtained from F by an *application* of the identity $G=H$ if F' arises from F by a direct application of some substitution instance of the identity $G=H$. Analogously as above, calling an application *trivial* if it leaves F untouched, in case G has no substitution instance that is a subformula of F the only possible application of the identity $G=H$ is the trivial one.

We note that the minimization problem mentioned above has been studied in detail so far mainly in the case of the classical propositional calculus, i.e. when $\Theta=\{\vartheta_1, \vartheta_2, \vartheta_3\}$, where $\vartheta_1(x_1, x_2) = x_1 \wedge x_2$, $\vartheta_2(x_1, x_2) = x_1 \vee x_2$ and $\vartheta_3(x_1) = \bar{x}_1$ (negation). For practical applications also important is the case when consists of Sheffer's alternative or Peirce's joint denial only[3], or of some of their generalizations for several variables, known as NAND and NOR elements.

3. Let $c(F)$ be a mapping from formulas to real numbers. We call the number $c(F)$ the *weight* of the formula F. An identity $G=H$ is said to be *weight-reducing* if:

$$c(G) > c(H).$$

We shall assume that $c(F)$ satisfies the following requirement, which in most cases of practical application does indeed hold:

(*) If the formula F' is obtained from F by a non-trivial direct application of a weight-reducing identity then we have

$$c(F') < c(F).$$

This condition ensures that the direct application to a formula of a weight-reducing identity is always efficient in the sense that it reduces the weight of this formula. Some care must be taken, however, in connection with non-direct applica-

[3] See the classical papers of C. S. Peirce [2] and H. M. Sheffer [3].

tions since the analogous assertion is not necessarily true for them even in the most simple cases occurring in practice.[4]

In most cases that are practically important, the meaning of the weight function $c(F)$, playing the role of an objective function to be minimized, is either the length of the formula F,[5] or the cost involved in its technical realization under specified circumstances.

4. After these preliminaries the minimization problem can be formulated precisely as follows: Assume that we are given a functionally complete system $\Theta = \{\vartheta_1, \vartheta_2, \ldots\}$ of a finite number of Boolean functions, a countable set $X = \{x_1, x_2, \ldots\}$ of Boolean variables, and an objective function $c(F)$, satisfying property $(*)$, defined for all formulas that can be composed by means of the given primitive elements. For a given Boolean function, represented by a formula F, consider the set $\mathscr{F} = \mathscr{F}(F) = \{F_1, F_2, \ldots\}$ of all formulas identical to F. Any formula F_{i_0}, belonging to $\mathscr{F}$, such that

$$c(F_{i_0}) \leqq c(F_i)$$

holds for all formulas $F_i \in \mathscr{F}$, is said to be a *minimal representation* of F. (In general, there exist several such formulas F_{i_0}.) An algorithm which, for any given formula F, selects a minimal representation of F is called a *minimization procedure.*

5. Now we have reached the stage where we can outline the ideas on which our minimization procedure is based.

(1) Enter the formula F given as input datum, possibly in a converted form suitable for the computer, on a list called the "*list of formulas to be minimized*".

(2) For any formula G newly entered on the list of formulas to be minimized, form all its subformulas, and then all those formulas H that have at least one substitution instance which is a subformula of G, and, finally, enter on a list called the "*list of the left-hand sides of applicable identities*" all those of these formulas H that do not yet occur there.

(3) For any formula H newly entered on the list of the left-hand sides of applicable identities, generate all formulas K having a weight less than H has. For each of these formulas K check whether it is identically equal to H; if yes then enter the identity $H = K$ on a list called the "*list of applicable identities*".

(4) Apply directly to every formula occurring on the list of formulas to be minimized all (weight-reducing) identities newly recorded on the list of applicable identities and also all those substitution instances of these identities that are weight-

[4] E.g. in case of the classical propositional calculus, taking the total number of occurrences of variables in the formula F as $c(F)$,

$$x_1 \wedge x_1 \wedge x_1 \wedge x_2 = x_1 \wedge x_2 \wedge x_2$$

is obviously a weight-reducing identity, but its substitution instance

$$x_1 \wedge x_1 \wedge x_1 \wedge (x_3 \wedge x_4 \wedge x_5) = x_1 \wedge (x_3 \wedge x_4 \wedge x_5) \wedge (x_3 \wedge x_4 \wedge x_5)$$

is not.

[5] There are many different weight functions called the lenght of a formula, e.g. those defined as the number of occurrences of variables or as the number of occurrences of variables and function symbols, etc. in the formula in question.

reducing.[6] Add those of the resulting formulas which are not yet contained in the list of formulas to be minimized to this list.

If the list of formulas to be minimized is not enlarged in step (4) then the algorithm is concluded by printing out one of the formulas with minimal weight occurring on this list; otherwise it continues at (2) again.

6. It is easy to see that our algorithm finally leads to an exact solution of the minimization problem formulated above. Indeed, if M is a minimal representation of the formula F given as input datum then $F=M$ is an identity.

If M has smaller weight than F has, i.e. the identity $F=M$ is weight-reducing, then it will sooner or later occur on the list of applicable identities, for F, as a subformula of itself, is to be found on the list of the left-hand sides of applicable identities. Then, by a direct application of the identity $F=M$ to F, we obtain M as a formula to be added to the list of formulas to be minimized. Hence, finally, either M or another formula of the same weight will be printed.

If, however, the weight of M equals that of F then F is already itself of minimal weight. Each of the generated weight-reducing identities can be applied to F only trivially, and thus the algorithm concludes after the first performance of step (4), and the only formula on the list of formulas to be minimized will be F, as a minimal representation of itself.

We emphasize that each of the formulas on the list of formulas to be minimized (among others F itself) has to stay on this list even if a formula identically equal to it of smaller weight is added to this list. Otherwise the application of a weight-reducing identity might impede later, possibly more advantageous, application of another such identity.

In practice, storage capacity or available running time limitations might prevent the continuation of the algorithm until its conclusion. If one is forced to interrupt the algorithm, we propose to print out one of the formulas with minimal weight from the list of formulas to be minimized as an approximative solution.

7. It is expedient to give the input formula of the algorithm in the so-called *Łukasiewicz bracket-free notation* (shortly Ł-notation; also known as Polish notation), or to convert it into that form by a supplementary algorithm.[7] The Ł-notation considerably simplifies the performing the algorithm.

Among others, if the formulas are written in Ł-notation, it is relatively easy to construct, by making use of the so-called *push down store*,[8] the sub-algorithms for the following tasks:

[6] In view of condition (*), a non-trivial direct application of a weight-reducing identity always reduces the weight of the formula in question, but in case of a non-direct application, as we already noted, it might happen that some substitution instance of a weight-reducing identity is not weight-reducing (see footnote[4]).

In principle, a direct application of all identities newly recorded on the list of applicable identities to every formula occurring on the list of formulas to be minimized would suffice. However, disregarding the weight-reducing substitution instances of these identities would lenghten our algorithm to such an extent that it were not practically feasible any more.

[7] See J. Łukasiewicz and A. Tarski [4], pp. 30—50. As for a simple proof of the unambiguous character of this notational system see, e.g., L. Kalmár [5], pp. 11—15.

[8] See F. L. Bauer and K. Samelson [6].

(i) Elimination of the Boolean constants from a given formula;

(ii) Production, for a given formula G, of all formulas that have at least one substitution instance which is a subformula of G;

(iii) Determination of the truth-value of a given formula for a given valuation of the variables occurring in it and, by repetitions of this sub-algorithm, the decision of the question whether two formulas are identically equal or not;

(iv) Application of an identity to a given formula.

8. As for the implementation of the algorithm on a computer, the most delicate part is (3), since it requires producing, given a formula H, all formulas K that have a smaller weight than H has. This part can perhaps most easily be realized in practice by splitting it into two steps:

(v) Production of all *formula types* of given weight. A formula type associated with a given formula containing no Boolean constants can be obtained, by definition, by replacing all Boolean variables occurring in this formula by a common one, x (without subscript), say.

(vi) Production of all the formulas of a given type containing variables from a given set only, e.g. that of the variables occurring in the input formula.

Empirical evidence (in cases that are practically most important) shows that weight-reducing identities with a smaller weight on the left-hand side, when applied, are more efficient than those with a left-hand side having a greater weight. Therefore it is advisable to apply the former ones first.

Hence, it is appropriate to generate and store the formulas in order of increasing weight. Thereby, the algorithm, even if it is interrupted, delivers a well approximating solution.

9. In the above version of the algorithm its run is controlled by the formula F to be minimized, at least in the sense that only those weight-reducing identities are produced which are non-trivially directly applicable to F or to another formula, arisen from F, occurring on the list of formulas to be minimized. In this of the way a great deal of computing time and storage room may be spared if we have only one formula to minimize.

If, however, we want to minimize several formulas, the above way might be disadvantageous. Indeed, in this case the algorithm produces, separately for each of the formulas to be minimized, all formulas, built up from the available stock of variables, that have smaller weights than those occurring on the list of the left-hand sides of applicable identities have. This might result in a very redundant repetition in the production of formulas.

An alternative version of the algorithm consists, e.g. in case only positive integers occur as weights, in generating and tabulating a "*complete* system of *independent* weight-reducing identities" up to a given ceiling for their left-hand side. In more detail, this version produces a set $\mathfrak{M}$ of weight-reducing identities such that

(α) Any weight-reducing identity such that the weight of its left-hand side does not exceed the given ceiling can be obtained, and therefore its direct applications can be replaced, by a finite number of direct applications of identities which either belong to $\mathfrak{M}$ or are weight-reducing substitution instances of identities belonging to $\mathfrak{M}$;

(β) $\mathfrak{M}$ is minimal in the sense that no identity belonging to $\mathfrak{M}$ can be obtained

by a finite number of direct applications of either other identities in $\mathfrak{M}$ or such substitution instances of these as are weight-reducing.

This variant of the algorithm is advantageous if we have to minimize a large number of formulas, since it requires to draw up the above chart of weight-reducing identities together with all their appropriate substitution instances only once, and then we have only to attempt to apply directly the identities in this chart to the formulas to be minimized. Nevertheless, we have to take into consideration that generating and tabulating the chart in question might require an enormous storage capacity.

10. Another variant of our algorithm consists in that besides the weight-reducing identities we admit such ones as leave the weights of their left-hand sides unchanged. More precisely, for every formula on the list of the left-hand sides of applicable identities we generate all the identities $H = K$ such that the weight of K does not exceed that of H; and then we apply any such substitution instance of each of these as are not weigh-augmenting to all formulas on the list of formulas to be minimized. (See especially steps (3) and (4) of the algorithm described above.)

For example, in case of classical propositional calculus with the weight of a formula meaning its length (see footnote[5]), this variant of our algorithm enables us to make use of the associative and commutative laws of conjunction and disjunction; these identities obviously do not change the length of a formula, but they may prepare for the application of another, strictly weight-reducing, identity[9].

Using this version of the algorithm we may, possibly, arrive at a minimal representation of the starting formula much quicker, though the price of this may be a much larger storage capacity used up. In yet another possible variant of our algorithm, for which the remarks made just now apply still more strongly, we may allow the application of certain weight-augmenting identities as well; e.g., in case of the classical propositional calculus the use of distributive law in the direction $(x_1 \vee x_2) \wedge x_3 = (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$ may sometimes prove useful by preparing the way for the application of a powerful weight-reducing identity.[10]

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

[9] The situation is illustrated by the following simple example, for which the author is indebted to an oral communication of G. Specker:

$$(\dots((x_1 \vee x_2) \vee x_3) \dots \vee x_n) \vee x_1 = (\dots(((x_1 \vee x_1) \vee x_2) \vee x_3) \dots \vee x_{n-1}) \vee x_n =$$
$$= (\dots((x_1 \vee x_2) \vee x_3) \dots \vee x_{n-1}) \vee x_n.$$

[10] The following example may serve as an illustration:

$$((x_1 \vee x_2) \wedge x_3) \vee x_2 = (x_1 \wedge x_3) \vee (x_2 \wedge x_3) \vee x_2 = (x_1 \wedge x_3) \vee x_2.$$

References

[1] ÁDÁM, A., *Truth functions and the problem of their realization by two-terminal graphs*, Akadémiai Kiadó, Budapest, 1968.
[2] PEIRCE, C. S., A Boolian algebra with one constant (c. 1880), *Collected papers* IV., pp. 13—18.
[3] SHEFFER, H. M., A set of five independent postulates for Boolean algebras with application to logical constants, *Trans. Amer. Math. Soc.*, v. 14, 1913, pp. 481—488.
[4] ŁUKASIEWICZ J. & A. TARSKI, Untersuchungen über den Aussagenkalkül, Sprawozdania zposiedzén Towarzystwa Naukowego Warszawskiego, Wydz. III, 23, 1930.
[5] KALMÁR, L., Another proof of the Markov-Post theorem, *Acta Math. Acad. Sci. Hung.*, v. 3, 1952, pp. 1—27.
[6] BAUER, F. L. & K. SAMELSON, Sequential formula translation, *Comm. ACM*, v. 3, 1960, pp. 76—83; addendum on p. 351.

(Received March 31, 1970; Section 10 added May 8, 1970)

On some generalizations of cyclic networks

By A. ÁDÁM

Zusammenfassung. Die ersten Abschnitte der Arbeit geben eine vollständige Beschreibung der endlichen zusammenhängenden gerichteten Graphen, die mindestens zwei Zyklen enthalten und in denen jeder Punkt und jede Kante entweder in einem Zyklus oder in (genau) zwei Zyklen liegt. Bezeichnen wir durch C_1 die Klasse dieser Graphen. Sei G ein Element von C_1 und k eine Zahl, die kleiner als die Längen der Zyklen von G ist; bezeichnen wir durch $\mathfrak{A}_k(G)$ den Graphen, dessen Punktmenge mit der Punktmenge von G übereinstimmt, so daß die Kante $\overrightarrow{AB}$ in $\mathfrak{A}_k(G)$ genau dann existiert, wenn $A \neq B$ und B aus A in G durch höchstens $k-1$ Kanten erreichbar ist. Sei C_2 die Klasse aller Graphen $\mathfrak{A}_k(G)$ wobei G die Elemente von C_1 durchläuft.

In den letzten Abschnitten wird es danach bestrebt, die in der früheren Arbeit [2] ausgearbeiteten Untersuchungen (über das Verhalten der Netzwerke mit einer speziellen graphentheoretischer Struktur) auf die in C_2 enthaltenen Graphen zu verallgemeinern. Es gelang nicht, alle erzielten Aussagen zu beweisen, folglich enthält die Arbeit auch unentschiedene Vermutungen (sowohl über die Struktur wie über das Verhalten).

§ 1. Introduction

In [2] certain cyclically symmetric networks were studied. These networks can be obtained in such a manner that we start with a single cycle and draw some additional edges in it.

Let us alter the mentioned procedure so that we start with a graph G satisfying the following four requirements (instead of being a cycle):

G is a finite connected directed graph,

to any edge e of G there exists at least one cycle containing e,

G contains at least two cycles,

whenever z_1, z_2, z_3 are three different cycles of G, then there exists no vertex lying in all of z_1, z_2, z_3.

The collection of these graphs G will be called the class C_1. We shall define a class of graphs (the class C_2) by adding edges to any graph in C_1 in an appriopriate manner. In §§ 3—4 we study the graph-theoretical structure of the members of the classes C_1, C_2; in §§ 6—8 the behaviour of the networks of type C_2 is analyzed. Since I did not succeed in solving all the arising problems, the paper also contains conjectures besides the propositions verified.

§ 2. Some graph-theoretical definitions

We shall always consider finite graphs having at least one edge. "Graph" will mean a *directed* one unless (rarely) we speak of a non-directed tree explicitly. Self-loops are (in general) permitted. Among the graphs containing parallel edges with the same orientation (especially, at least two self-loops on the same vertex), only the two graphs seen on Fig. 1 are allowed (cf. Remark 2 at the end of § 3).

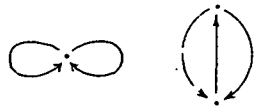


Fig. 1

Let G be a directed graph. A sequence

$$A_0, e_1, A_1, e_2, A_2, \ldots, e_n, A_n \tag{1}$$

consisting of the vertices $A_0, A_1, \ldots, A_n$ and the edges $e_1, e_2, \ldots, e_n$ of G (alternatively) is called a *directed edge sequence* (of length n) if each e_i ($1 \leqq i \leqq n$) goes from A_{i-1} to A_i. If, in addition, $A_0, A_1, \ldots, A_n$ are different vertices, then (1) is a *path*. If $A_0, A_1, \ldots, A_{n-1}$ are different but $A_0 = A_n$, then (1) is a *cycle*. Let $Z(A)$ be the number of cycles of G which contain the vertex A; let $Z(e)$ be the number defined for the edge e analogously. We denote by M_G the minimal cycle length that occurs in the graph G.

In case of undirected graphs (or if the orientation of the edges is disregarded), the concepts analogous to path and cycle are called *chain* and *circuit*, respectively.

Let A be a vertex of the directed graph G, assume that A is incident to exactly k edges oriented *towards* A and to exactly l edges oriented *outwards from* A. Then we say that the *indegree* of A is k, the *outdegree* of A is l, and the *degree* of A the ordered pair (k, l). — If G is undirected, then the *degree* $d(P)$ of the vertex P is the number of edges incident to P.

Let H be a subgraph of G. If H contains all the vertices (but, possibly, not all the edges) of H, then we say that H is an *e-subgraph* of G. The subgraph H of G is called a *p-subgraph* of G if the following condition is satisfied: whenever A and B are contained in H and the edge e of G is incident to A and B, then e is contained in H too.[1] For each subgraph H of G, there exists exactly one graph $\mathfrak{S}_G(H)$ such that $\mathfrak{S}_G(H)$ is a p-subgraph of G and H is an e-subgraph of $\mathfrak{S}_G(H)$.

Let G be a directed graph fulfilling[2] $M_G \geqq 3$ and k be a number such that $2 \leqq k < M_G$. Let us form a graph H conforming to the following two rules:

the vertex set of H equals to the vertex set of G.

the (directed) edge $\overrightarrow{AB}$ ($A \neq B$) exists in H if and only if in G there is a path of the length $< k$ from A to B.

The obtained graph H is denoted by $\mathfrak{A}_k(G)$. Obviously, G is an e-subgraph of $\mathfrak{A}_k(G)$ and $\mathfrak{A}_2(G) = G$ is always true.

Let C be a class consisting of directed graphs. Then we denote by $\mathfrak{A}(C)$ the class of all the graphs $\mathfrak{A}_k(G)$ where G runs through the members of C and, for any G, k runs through the numbers satisfying $2 \leqq k < M_G$.

[1] If e is a self-loop, then the same vertex is considered as A as well as B.

[2] The condition $M_G \geqq 3$ means that G contains neither self-loops nor (oppositely oriented) parallel edge pairs.

Now we introduce two classes of connected directed graphs. Let C_1 consist of all the graphs having at least two cycles and satisfying the inequalities

$$1 \leqq Z(A) \leqq 2 \quad \text{and} \quad 1 \leqq Z(e) \leqq 2$$

identically.[3,4] Let C_2 be[5] $\mathfrak{A}(C_1)$.

§ 3. The structure of the graphs in C_1

Construction I. The construction consists of four steps.

Step 1. Let T be a non-directed tree with at least one edge. For each vertex P of T, we denote by $e_1^{(P)}, e_2^{(P)}, \ldots, e_{d(P)}^{(P)}$ the edges incident to P (in an arbitrary manner). (Evidently, every edge gets *two* notations.)

Step 2. Let us form a directed graph G_1 by what follows: the vertices of G_1 correspond one-to-one with the edges of T; if the vertex A of G_1 corresponds to the edge $e_p^{(P)} = e_q^{(Q)}$ of T, then edges go from A to the vertices corresponding to $e_{p+1}^{(P)}$ and $e_{q+1}^{(Q)}$ and only to these vertices (in case $p = d(P)$, $e_1^{(P)}$ plays the role of $e_{p+1}^{(P)}$).

Step 3. Choose a subset V' of the set of vertices of G_1 arbitrarily. For any element A of V', perform the following procedure:

Replace A by two vertices A' and A'';

if an edge had gone to A, then let it go to A',

if an edge had gone from A, then let it go from A'';

finally, supplement the graph with a new edge leading from A' to A''.

Evidently, this process can be carried out for all the vertices in V' simultaneously. We denote the resulting graph by G_2. (See Fig. 2.)

Step 4. Instead of any edge of G_2, we draw a *path* of arbitrary length ($\geqq 1$). (Of course, the inner vertices of these paths have the degree (1, 1).) We denote the resulting graph by G.

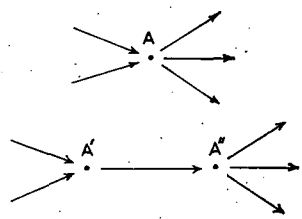


Fig. 2

Theorem 1. Any graph G arising by Construction I belongs to the class C_1.

Proof. First we show that G is connected. It is sufficient to verify that G_1 is connected because Steps 3, 4 cannot spoil the connectedness. Let A, B be two vertices of G_1. If the edges e_A, e_B of T, corresponding to A and B (resp.), are adjacent, then A and B can clearly be joined by a chain. — Let now A, B be arbitrary vertices

[3] The word "identically" means that the conditions are required for each vertex A and for each edge e, respectively.

[4] These four inequalities do not form an independent system: if $Z(A) \leqq 2$ and $Z(e) \geqq 1$ are true, then also $Z(A) \geqq 1$ and $Z(e) \leqq 2$ hold.

[5] Our present notation differs from that of [2]: the graph, denoted by $G(n; 1, 2, \ldots, k)$ in [2], is now denoted by $\mathfrak{A}_{k+1}(z_n)$, where z_n is the cycle of length n.

of G_1. There exists a chain in T with edges

$$e_A = e_1, e_2, \ldots, e_k = e_B.$$

Let the vertices of G_1 corresponding to these edges (respectively) be

$$A = A_1, A_2, \ldots, A_k = B.$$

We have shown that A_i, A_{i+1} can be joined by a chain (for each i, $1 \leqq i < k$); this implies that the same holds for A and B.

Let A be an arbitrary vertex of G_1 and e_A be the corresponding edge of T. There exist two vertices P, Q of T such that $e_A = e_p^{(P)} = e_q^{(Q)}$ (where p, q are suitable numbers). A is of degree (2, 2) by Step 2, and, moreover, A is a cut vertex since any chain going from $e_{p+1}^{(P)}$ to $e_{q+1}^{(Q)}$ passes through A. These considerations imply that a sequence of vertices of G_1 determines a cycle if and only if it corresponds to the edge sequence

$$e_1^{(P)}, e_2^{(P)}, \ldots, e_{d(P)}^{(P)}$$

for some vertex P of T. Hence $Z(A) = 2$, $Z(e) = 1$ are identically satisfied in G_1.

Step 3 of the construction does not alter the number of cycles and the identical validity of $Z(A) = 2$. For an edge e of G_2, either $Z(e) = 2$ or $Z(e) = 1$ holds according as e is a new edge (i.e. going from an A' to an A'') or not.

Step 4 does not modify the number of cycles, either. Denote by e' an edge of G_2, let A be an arbitrary inner vertex and e be an arbitrary edge of the path (in G) replacing e' by virtue of Step 4. We have obviously $Z(A) = Z(e) = Z(e')$. If A is a vertex of G_2, then $Z(A) = 2$ holds in G as well as in G_2. Thus $1 \leqq Z(A) \leqq 2$ and $1 \leqq Z(e) \leqq 2$ are identically satisfied in G.

Lemma 1. Assume that the graph G satisfies $Z(A) = 2$ and $Z(e) = 1$ identically. Then any two cycles of G have at most one vertex in common.

Proof. Let z_1, z_2 be two cycles containing (at least) two common vertices. Let A be a common vertex such that the edges of z_1 and z_2, starting from A, are different. Let us pass from A on z_1 to the first other common vertex $B (\neq A)$, then let us pass from B to A on z_2. Thus we have got a third cycle containing A; this, however, contradicts $Z(A) = 2$.

Theorem 2. Every graph G belonging to the class C_1 may be produced by Construction I.

Proof. Let G be contained in C_1. The condition $Z(A) \geqq 1$ implies that any vertex of G has a positive outdegree and a positive indegree. Neither the outdegree nor the indegree of a vertex A can exceed 2, because if e.g. the indegree were $k (> 2)$, then each of the k edges starting from A could be extended to a cycle, hence $Z(A) \geqq k > 2$ would follow; this is a contradiction.

Thus the degree of any vertex of G is either (1, 1) or (2, 1) or (1, 2) or (2, 2). There is at least one vertex whose degree differs from (1, 1) (otherwise G would be a single cycle).

In what follows, we shall define a decomposition procedure for G that consists of four steps corresponding to Steps 4., 3., 2., 1. of Construction I, respectively.

Step 1. If A is of degree (1, 1), then we delete A and contract the two edges incident to A into one edge. This can be performed for all the vertices with degree

(1, 1) simultaneously (without essential difficulties). Let us denote the resulting graph[6] by G_1'. It is clear that $G_1' \in C_1$, and, furthermore, that only the degrees (2, 1), (1, 2), (2, 2) may occur in G_1'.

Now we establish three lemmas on G_1' (the proof of Theorem 2 will be continued later).

Lemma 2. Let e be an edge of G_1', going from A to B. Then $Z(e)=2$ if and only if $d(A)=(2, 1)$ and $d(B)=(1, 2)$.

Proof. The sufficiency is trivial. Conversely, suppose $Z(e)=2$; if the outdegree of A is 2, then $Z(A)\geqq 3$; if the indegree of B is 2, then $Z(B)\geqq 3$.

Lemma 3. Let e, A, B be as in Lemma 2. Then $Z(e)=1$ if and only if $d(A)$ is either (1, 2) or (2, 2) and $d(B)$ is either (2, 1) or (2, 2).

Proof. First we show that each of the following four statements leads to a contradiction:

(a) $d(A)=(2, 1)$ and $d(B)=(2, 1)$
(b) $d(A)=(1, 2)$ and $d(B)=(1, 2)$
(c) $d(A)=(2, 1)$ and $d(B)=(2, 2)$
(d) $d(A)=(2, 2)$ and $d(B)=(1, 2)$.

Indeed, (a) implies $Z(e')\geqq 3$ for the single edge e' going out from B, (c) implies $Z(B)\geqq 3$; (b) and (d) can be disproved analogously (by interchanging A and B).

Since the possibilities (a)—(d) and the ones of Lemma 2 are excluded, only those allowed in Lemma 3 remain.

Lemma 2 implies immediately.

Lemma 4. If $Z(e)=Z(e')=2$ for two different edges e, e' of G_1', then e and e' are not adjacent.

Proof of Theorem 2 (continued).

Step 2. Consider the graph G_1' (resulting by Step 1), and choose an edge e of G_1' satisfying $Z(e)=2$. Contract the two vertices A, B incident to e into one vertex (i.e. delete e, A and B, and introduce a new vertex C so that any edge ($\neq e$) which has been incident to A or B will now be incident to C).[7] This process can be performed for all the edges fulfilling $Z(e)=2$ simultaneously (by Lemma 4). Let the resulting graph be denoted by G_2'. Obviously, $G_2' \in C_1$, and, moreover, $d(A)=(2, 2)$, $Z(A)=2$, and $Z(e)=1$ are identically valid in G_2'.

Step 3. Consider G_2', and define an undirected graph T' in the following manner: the vertices of T' correspond in a one-to-one way to the cycles of G_2'; two vertices P, Q of T' are joined by an edge if and only if the corresponding cycles of G_2' have a vertex in common.

Next we state two lemmas on T'. The first of them follows from $Z(A)=2$ (holding in G_2') and Lemma 1 at once:

[6] If parallel edges with the same orientation do not occur in G, then either the same holds for G_1' or G_1' is one of the graphs of Fig. 1.

[7] If there has been an edge e' going from B to A (of course, satisfying $Z(e')=1$), then e' will become a self-loop of the new vertex C.

Lemma 5. Let us assign to any edge e of T' the (unique) common vertex of the two cycles in G_2' corresponding to the vertices incident to e. This assignment is a one-to-one correspondence between all the edges of T' and all the vertices of G_2'.

Lemma 6. T' is a tree.

Proof. First we show that T' has no circuit. Assume that t is a circuit of minimal length in T', let t consist of

$$P_1, e_1, P_2, e_2, \ldots, P_k, e_k, P_1 \qquad (k \geqq 3)$$

(i.e. $P_1, P_2, \ldots, P_k$ are the vertices and $e_1, e_2, \ldots, e_k$ are the edges of t, passed through as they follow). Let

$$A_1, A_2, \ldots, A_k$$

be the vertices of G_2' corresponding to

$$e_1, e_2, \ldots, e_k$$

(resp.) and

$$z_1, z_2, \ldots, z_k$$

be the cycles of G_2' corresponding to

$$P_1, P_2, \ldots, P_k$$

(resp.). Let us form a directed edge sequence in G_2' so that we pass

on z_1 from A_k to A_1, afterwards
on z_2 from A_1 to A_2,
on z_3 from A_2 to A_3,
...
finally, on z_k from A_{k-1} to A_k.

This sequence z is a cycle (otherwise t cannot be minimal). Thus $Z(A_i) \geqq 3$ $(1 \leqq i \leqq k)$, which is a contradiction.

We are going to show that T' is connected. Suppose the contrary. The disconnectedness of T' implies (by Step 3) that G_2' is either disconnected or has an edge e fulfilling $Z(e) = 0$. Both alternatives are contradictory (the first one is because the connectedness of G_2' is equivalent to the connectedness of G, by Steps 1, 2).

Proof of Theorem 2 (final part). The proof is completed by noting that the decomposition procedure, described in this proof (together with Lemmas 2—6), is an exact counterpart of Construction I.

Remark 1. To a vertex A of the graph G_1 (produced by Step 2 of Construction I) a self-loop is incident exactly if the edge in T, corresponding to A, is a final edge in T (i.e. it is incident to a vertex of degree 1). A graph G produced by Construction I contains no self-loop (i.e. cycle of length 1) exactly if each self-loop of G_1 is eliminated either in Step 3 or in Step 4.

Remark 2. It is easy to see that if a connected directed graph G satisfying $Z(e) \geqq 1$ identically has two parallel edges with the same orientation, then either G is one of the graphs of Fig. 1 or G has a vertex A such that $Z(A) \geqq 3$. This fact justifies the agreement posed in the fourth sentence of § 2.

Remark 3. A graph G_1 (produced by Step 2 of Construction I) contains a pair of oppositely oriented parallel edges (i.e. a cycle of length 2) exactly if T has a vertex of degree 2. A graph G produced by Construction I does not contain a pair of oppositely oriented parallel edges exactly ife ach pair of this property of edges of G_1 is eliminated either in Step 3 or in Step 4 (of course, the possibility that G_1 contains no such pair is included).

Problem. How to describe all connected directed graphs fulfilling $1 \leqq Z(e) \leqq 2$ identically?[8]

§ 4. Some conjectures on the class C_2

By definition, each graph G contained in the class C_2 has at least one e-subgraph G' such that $G = \mathfrak{A}_k(G')$ where k is a suitable number fulfilling $k < M_{G'}$. It is an open problem whether or not the statement of unicity of this presentation holds. This problem would be solved in the affirmative sense if a method were given for constructing G' from G such that the resulting graph G' is the unique e-subgraph such that $G = \mathfrak{A}_k(G')$. In this § some conjectures related to this question will be exposed. The unicity statement is formulated in Conjecture 3.

In what follows, we shall make use of two further classes of connected directed graphs. Let C_3 contain a graph G if and only if G has an automorphism α such that α permutes the vertices of G cyclically and there exists an edge from A to $\alpha(A)$ for any vertex A.[9] Let G belong to the class C_4 exactly if the following assertion is fulfilled: whenever

$$G = \mathfrak{A}_k(G'), \quad k < M_{G'} \quad \text{and} \quad G' \in C_1$$

are satisfied for G', k, and z is a cycle of G', then[10, 11] $\mathfrak{S}_G(z) = \mathfrak{A}_k(z)$.

[8] This condition implies the identical fulfilment of $Z(A) \geqq 1$. Fig. 3 shows a graph in which $Z(e) = 1$ for each edge and $Z(A) = 3$ for some vertex.

[9] The relation $G \in C_3$ holds exactly if the vertices of G can be labelled by the numbers $1, 2, \ldots, n$ such that

$$G = G(n; 1, m_2, m_3, \ldots, m_k),$$

where n is the number of vertices of G, k is a suitable number and, on the right-hand side, the notation means that an edge $\overrightarrow{A_i A_j}$ exists exactly if i—j is congruent to one of $1, m_2, \ldots, m_k$ modulo n.

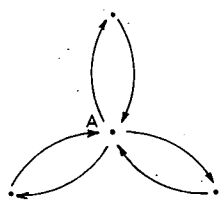


Fig. 3

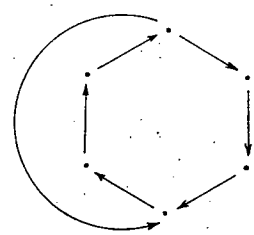


Fig. 4

[10] The inclusion $\mathfrak{S}_k(z) \supseteq \mathfrak{A}_k(z)$ is trivially satisfied; we require now the converse inclusion. It is obvious that $\mathfrak{A}_k(z) \in C_3$.

[11] Let G' be the graph of Fig. 4 and z be the longer cycle of G'. Evidently $\mathfrak{A}_3(G') \notin C_4$. If all the cycles of G' are of the same length, then $\mathfrak{A}_k(G') \in C_4$.

Let G be a graph in C_2. If a p-subgraph G_1 of G belongs to the class C_3, then we say that G_1 is a C_3-subgraph of G. If G_1 is a C_3-subgraph of G and there exists no C_3-subgraph G_2 of G such that $G_1 \subset G_2 \subset G$, then we say that G_1 is a maximal C_3-subgraph of G.

Conjecture 1. Let G be a graph contained in C_2. Let G' be an e-subgraph of G and k be a natural number such that

$$G' \in C_1, \quad k < M_{G'} \quad \text{and} \quad G = \mathfrak{A}_k(G').$$

If G_1 is a C_3-subgraph of G, then there exists a cycle z of G' such that z contains all the vertices of G_1.

Conjecture 2. Let G, G' be two graphs as in Conjecture 1. Assume that G is contained in C_4. A p-subgraph G_1 of G is a maximal C_3-subgraph of G if and only if there exists a cycle z of G' such that $G_1 = \mathfrak{S}_G(z)$.

Proposition 1. If Conjecture 1 holds, then so does Conjecture 2 as well.

Proof. Let z be a cycle of G'. $\mathfrak{S}_G(z)$ is a C_3-subgraph of G in consequence of $G \in C_4$. Let G_2 be a proper C_3-subgraph of G such that $\mathfrak{S}_G(z) \subset G_2$. Conjecture 1 implies the existence of a cycle z' of G' containing all the vertices of G_2. The vertex set of z is a proper subset of the vertex set of z'; this contradiction shows that $\mathfrak{S}_G(z)$ is a maximal C_3-subgraph, thus the sufficiency statement of Conjecture 2 is proved.

Conversely, let G_1 be an arbitrary maximal C_3-subgraph of G. Consider $\mathfrak{S}_G(z)$, where z is the cycle whose existence is stated in Conjecture 1. $\mathfrak{S}_G(z)$ is a C_3-subgraph of G by $G \in C_4$. The maximality of G_1 implies $G_1 = \mathfrak{S}_G(z)$.

Conjecture 3. Suppose $G \in C_2$. Then there exists exactly one pair (G', k) (consisting of an e-subgraph G' of G and of a natural number k) such that $G = \mathfrak{A}_k(G')$.

Proposition 2. If Conjecture 2 holds and $G \in C_2 \cap C_4$, then the conclusion of Conjecture 3 is valid for G.

Proof. Let G', G'' be two e-subgraphs of G and k_1, k_2 be natural numbers such that

$$k_1 < M_{G'}, \quad k_2 < M_{G''}, \quad G' \in C_1, \quad G'' \in C_1, \quad G = \mathfrak{A}_{k_1}(G') = \mathfrak{A}_{k_2}(G'').$$

Conjecture 2 implies the equivalence of the following three assertions (i), (ii), (iii) for a p-subgraph G_1 of G:

(i) the vertices of G_1 coincide with the vertices of a cycle of G',

(ii) G_1 is a maximal C_3-subgraph of G,

(iii) the vertices of G_1 coincide with the vertices of a cycle of G''.

Hence the vertex sets of the cycles of G' coincide with the vertex sets of the cycles of G''. Let z' be a cycle of G' and z'' be a cycle of G'' such that z', z'' contain precisely the same vertices; let A be a vertex of z' (and of z''). We shall label the vertices (in question) as they follow A on z' or on z''. From A, edges (of G) go to the first, second, ..., k_1-th vertices (and only to these) of z'; analogously, from A edges go to the first, second, ..., k_2-th vertices (and only to these) of z''. This implies $z' = z''$ and $k_1 = k_2$, thus also $G' = G''$ (because $Z(e) \cong 1$ is identically satisfied in G' and in G'').

§ 5. Some lemmas

Let A, B be two vertices of a graph G and a be a directed edge sequence from A to B. It is well-known that we can select a path a_1 from a such that a_1 leads from A to B, too; more precisely, a_1 may be constructed by iterating the method that we omit a cycle out of a directed edge sequence (unless it is a path). This fact will be used sometimes in this §.

Lemma 7. Let A, B be two vertices of a connected graph G. If $Z(e) \geqq 1$ is identically satisfied in G, then there exists a path a of G such that the beginning vertex of a is A and the end vertex of a is B.[12]

Proof. First we show that the conclusion is satisfied by some directed edge sequence. Since G is connected, there exists a chain whose vertices are

$$A = A_0, A_1, A_2, \ldots, A_{m-1}; A_m = B$$

where m is the length of the chain. For every subscript i $(0 \leqq i < m)$ there exists either the edge $\overrightarrow{A_i A_{i+1}}$ or the edge $\overrightarrow{A_{i+1} A_i}$.

Suppose that there exists a directed edge sequence b from A to A_i $(0 \leqq i < m)$, we shall prove the analogous statement for A, A_{i+1}. If $\overrightarrow{A_i A_{i+1}}$ does exist, then the existence of the required sequence is obvious. If $e = \overrightarrow{A_{i+1} A_i}$ exists, then let c be the path which originates from a cycle containing e by deleting e. b and c form together a directed edge sequence from A to A_{i+1}. We can select a path from the directed edge sequence constructed above between A and B. This completes the proof.

In the subsequent lemmas, we consider a graph G belonging to C_1 and we denote by d the greatest common divisor of the lengths of all cycles of G. For any pair A, B of vertices of G, the number of cycles containing both A and B is either 0 or 1 or 2.

Lemma 8. Let G be a graph belonging to C_1 and A, B be two vertices of G. Denote by $\pi(A, B)$ the number of paths going from A to B. The following three assertions are true:

(a) *If there is at most one cycle containing both A and B, then*

$$\pi(A, B) = 1.$$

(b) *If there exist two cycles containing both A and B, then*

$$\textit{either} \quad \pi(A, B) = 2 \quad \textit{and} \quad \pi(B, A) = 1,$$

$$\textit{or} \quad \pi(A, B) = 1 \quad \textit{and} \quad \pi(B, A) = 2.$$

(c) *Suppose that the first alternative of* (b) *holds. Let l_1, l_2 be the lengths of the paths leading from A to B and l_3 be the length of the path going from B to A. Then*

$$l_1 \equiv l_2 \equiv -l_3 \pmod{d}.$$

[12] If A and B coincide, then a path of length 0 fulfils the conclusion.

Remark. The assertions (a), (c) hold also in symmetrized form (by interchanging A and B).

Proof. We use induction with respect to the number of cycles of G. If G has two cycles, then the lemma is evidently valid.

Assume that the number of cycles is m and the lemma is true for the graphs having at most $m-1$ cycles. We shall rely upon Construction I without any explicit reference (this is justified by Theorem 2). Let z be a cycle of G such that z corresponds to a vertex of degree 1 of T. G has exactly one cycle z_1 such that z, z_1 have at least one vertex in common. The vertices of z can be denoted (uniquely) by

$$F_1, F_2, \ldots, F_t, \quad D_1, D_2, \ldots, D_w$$

such that z passes through the vertices in this ordering and exactly $F_1, F_2, \ldots, F_t$ are the common vertices with z_1. Also z_1 passes over the F's according to increasing subscripts.(See Fig. 5.) $t=1$ is possible. $t=1$ implies that the degree of F_1 is (2, 2), $t>1$ implies that the degree of F_1 is (2, 1) and the degree of F_t is (1, 2); in both cases, all the remaining vertices of z are of degree (1, 1).

Fig. 5

Denote by G_1 the graph resulting if $D_1, D_2, \ldots, D_w$ (and the edges incident to them) are deleted. Clearly $G_1 \in C_1$.

We distinguish six cases with respect to the situation of A and B. (The cases arising when A, B are interchanged are not treated separately.)

Case 1: neither A nor B occurs in z. Then the connectibility of A and B is the same in G as in G_1.

Case 2: $A=D_i$ and $B=D_j$ (where $1 \leqq i<j \leqq w$). Then (a) is trivially fulfilled.

Case 3: $A=F_i$ and $B=F_j$ $(1 \leqq j<i \leqq t)$. We have

$$\pi_1(A, B)=\pi_1(B, A)=1$$

for the function π_1 defined in G_1, hence

$$\pi(A, B)=2 \quad \text{and} \quad \pi(B, A)=1,$$

i.e. the first alternative of (b) holds. Let l_1, l_2 be the lengths of the paths from A to B along z_1, z, respectively; let l_1^*, l_2^* be the lengths of z_1, z (resp.); let l_3 be the length of the path from B to A. Then

$$l_1+l_3=l_1^*, \quad l_2+l_3=l_2^*,$$

hence, on the one hand, $d \mid l_1^*=l_1+l_3$, thus

$$l_1 \equiv -l_3 \pmod{d};$$

on the other hand, $l_1 - l_2 = l_1^* - l_2^*$. Since both of l_1^*, l_2^* are multiples of d, the same holds for their difference, thus

$$l_1 \equiv l_2 \pmod{d}.$$

Also (c) is verified.

Case 4: $A = D_i$ and $B = F_j$ $(1 \leqq i \leqq w,\ 1 \leqq j \leqq t)$. (a) is trivially fulfilled.

Case 5: A does not occur in z and $B = F_i$ $(1 \leqq i \leqq t)$. (a) follows from the induction hypothesis.

Case 6: A does not occur in z and $B = D_i$ $(1 \leqq i \leqq w)$. Because $\pi(A, F_i) = 1$ by Case 5, it is clear that $\pi(A, B) = 1$. — Analogously, $\pi(F_1, A) = 1$, hence $\pi(B, A) = 1$. The proof is completed.

Lemma 8 implies immediately

Lemma 9. Let G be a graph belonging to C_1 and A, B be two vertices of G. If a, b, c are three directed edge sequences such that both of a, b lead from A to B and c goes from B to A, then

$$l_1 \equiv l_2 \equiv -l_3 \pmod{d},$$

where l_1, l_2, l_3 are the lengths of a, b, c respectively.

§ 6. Some notions concerning the behaviour of networks

We recall the continuous model of the behaviour of a network,[13] exposed in Section 3 of [1]. The subsequent treatment is — essentially — an extension of that of [2]. The mentioned behaviour may be shortly summarized as follows:

(1) To any vertex A_i a function $\alpha_i(t)$ is assigned. The domain of α_i is either the (real) interval $[0, \infty)$ or an interval $[0, T'_{\max})$ where $T'_{\max}$ is some positive number (common for the vertices). The range of α_i is the interval $[0, 1]$.

(2) For any number t lying in the domain of the functions α_i (where $1 \leqq i \leqq n$, n is the number of vertices), if the edge $\overrightarrow{A_j A_k}$ exists and $\alpha_j(t) = 1$, then $\alpha_k(t) = 0$.

(3) The initial values $\alpha_i(0)$ of the functions are assumed to fulfil the requirement posed in (2) (with 0 as t).

(4) If the value of the function α_i is less than 1, then it increases linearly unless it must be 0 in consequence of (2).

(5) If the value of the function α_i is 1, then it remains constantly 1 unless it must be 0 in consequence of (2).

(6) If $\overrightarrow{A_j A_k}$ exists and the function α_j, α_k reach the value 1 at some instant t_0 simultaneously, then (t_0 is denoted by $T'_{\max}$ and) the functions are not defined for numbers $t \geqq t_0$.

If the functioning of a network is defined at an instant t, then the vector

$$\mathfrak{B} = \langle \alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t) \rangle$$

is called the *state* of the network at t. Let us form the state $\mathfrak{D}$ of the network at

[13] By a network we mean a graph (without self-loops) together with numerical functions, depending on the time, assigned to the vertices in a one-to-one manner.

the instant $t+t'$ (where t' is a non-negative number and this new state $\mathfrak{D}$ is formed from $\mathfrak{B}$ in agreement with the above rules (1)—(6)); $\mathfrak{D}$ is denoted by $\mathfrak{B}[+t']$ too.

The state $\mathfrak{B}$ is called *cyclic* if there exists a positive t such that $\mathfrak{B}=\mathfrak{B}[+t]$; each suitable t is called a *period* of the network with the initial state $\mathfrak{B}$. $\mathfrak{B}$ is called *steady* if $\mathfrak{B}=\mathfrak{B}[+t]$ is true for every positive t. Any steady state is obviously cyclic. By a *proper cyclic state* a non-steady cyclic state is meant. If $\mathfrak{B}$ is a proper cyclic state, then clearly there exists a positive number t_0 such that $\mathfrak{B}=\mathfrak{B}[+t]$ holds exactly if $t=gt_0$ where g can be 0, 1, 2, 3, ...

In the remaining part of this §, the concept of regular state will be introduced. Let us consider a graph G' belonging to the class C_1. Denote by d the greatest common divisor of the lengths of the cycles of G'. We define a partition Π of the vertex set of G' in the following manner: let $A\equiv B \pmod{\Pi}$ be true exactly if there exists a path a (of length $\geqq 0$) such that the beginning vertex of a is A, the end vertex of a is B and the length of a is a multiple of d. We have to show that Π is an equivalence.

Lemma 10. The relation Π is reflexive, symmetric and transitive.

Proof. The reflexivity is evident since paths of length 0 are allowed.

Next we prove the symmetry. Suppose $A\equiv B \pmod{\Pi}$. There exists a path a from A to B and a path b from B to A by Lemma 7. The length of a is a multiple of d by the supposition; either the fact that a, b form together a cycle or Lemma 8 (c) implies that also the length of b is a multiple of d; consequently, $B\equiv A \pmod{\Pi}$.

Finally, we show the transitivity. Assume $A\equiv B \pmod{\Pi}$ and $B\equiv C \pmod{\Pi}$. There exists a path a from A to B and a path b from B to C such that the lengths of a and b are multiples of d. Hence C can be reached from A on a directed edge sequence whose length is $\equiv 0$ modulo d. By Lemmas 7, 8, 9, the same holds for the path(s) leading from A to C (and there exists such a path).

Lemma 11. Let A, B, C, D be four vertices of G'. If $A\equiv B \pmod{\Pi}$, and there exists an edge from A to C, and there exists an edge from B to D, then $C\equiv D \pmod{\Pi}$.

Proof. Lemmas 7, 9 and the definition of Π imply the existence of two paths a, b such that a goes from C to A, the length of a is $\equiv -1 \pmod{d}$, b goes from A to B, the length of b is $\equiv 0 \pmod{d}$. Hence the directed edge sequences, going from C to D, are of length congruent with $-1+0+1 = 0 \pmod{d}$, thus $C\equiv D \pmod{\Pi}$. The proof is completed.

Since Lemma 11 is valid and $Z(A)\geqq 1$ is identically satisfied, it is easy to see that there are d equivalence classes modulo Π and we can label these classes by

$$E_1, E_2, \ldots, E_d \tag{6.1}$$

so that if an edge comes from a vertex in E_i $(1\leqq i\leqq d)$, then it terminates at an element of E_{i-1} (where, of course, E_d plays the role of E_{1-1}). This enumeration of the classes is unique apart from cyclic translation.

Let us consider a network $G(\in C_2)$, having the vertices $A_1, A_2, \ldots, A_n$, and an e-subgraph $G'(\in C_1)$ of G such that $G=\mathfrak{A}_k(G')$ and $2\leqq k<d$.[14] A state

$$\mathfrak{B}=\langle \alpha_1(t), \alpha_2(t), \ldots, \alpha_n(t)\rangle$$

[14] Throughout the following parts of the paper, this terminology will be used extensively.

of G will be called *regular*[15] (at the instant t) if it satisfies the following three conditions:

(a) If $A_i \equiv A_j \pmod{\Pi}$, then $\alpha_i(t) = \alpha_j(t)$.

(b) If $\alpha_i(t) = 1$ for the vertices A_i lying in a class E_j (occurring in (6. 1)), then $\alpha_{i'}(t) = 0$ for every

$$A_{i'} (\in E_{j-1} \cup E_{j-2} \cup \cdots \cup E_{j-k+1})$$

where the expressions $j-1, j-2, \ldots$ are meant modulo d.

(c) If $0 \leqq \alpha_i(t) < 1$, $A_i \in E_j$ and $\alpha_m(t) < 1$ for every

$$A_m (\in E_{j+1} \cup E_{j+2} \cup \cdots \cup E_{j+k-1}),$$

then $\alpha_i(t) < \alpha_{i'}(t)$ for each $A_{i'} (\in E_{j-1})$, where the expressions $j+1, j+2, \ldots$ and $j-1$ are again viewed modulo n.[16]

By comparing the notions of cyclic and regular states with how these concepts had been introduced in [2], one can ascertain that the cyclic states were defined in precisely the same manner and the regularity was introduced in an almost full analogy (the difference is motivated by the modification of the graph-theoretic structure).

§ 7. The cyclicity of regular states

Consider a network $G (\in C_2)$ (as in the definition of the regular state). Suppose that we start with a regular state of G at the instant 0. The behaviour of G may be studied in detail in analogy to the discussion in § 2 of [2]. In studying a function α_i assigned to a vertex A_i, the only modification here is that now the sequence of sets

$$H_1^{(i)}, H_2^{(i)}, H_3^{(i)}, \ldots$$

must be considered, where $H_h^{(i)}$ consists of the vertices from which A_i is reachable by a path of length h (instead of the vertex sequence

$$P_{i+1}, P_{i+2}, P_{i+3}, \ldots$$

in [2]); clearly, any set $H_h^{(i)}$ is a subset of E_{j+k} (where h can be 1, 2, 3, ..., and j is determined by $A_i \in E_j$), thus any two vertices lying in a common $H_h^{(i)}$ have the same initial value (by the requirement (a) in the definition of the regular state). The discussion and inferences, being in analogy with the respective parts of [2], lead to the following statements:

Proposition 3. If we start with a regular state at the instant 0 *and* $A_{i_1} \in E_j$, $A_{i_2} \in E_{j+k}$, *then*

$$\alpha_{i_1}(\tau) = \alpha_{i_2}(0)$$

(*the expression* $j+k$ *is meant* mod d).

Denote by g the least common multiple of d and k.

Proposition 4. Any regular state is cyclic, $g\tau/k$ *is a suitable period.*

[15] The regularity of a state depends on which e-subgraph of G is distinguished as G'. If Conjecture 3 is valid, then this dependence is apperent only.

[16] If $A_i \in E_j$, $\alpha_i(t) = 1$ and $A_{i'} \in E_{j-k}$, then both $\alpha_{i'}(t) = 0$ and $\alpha_{i'}(t) > 0$ are permitted.

Proposition 5. If $\mathfrak{B}$ is a regular state, then the state $\mathfrak{B}[+t]$ is regular for each non-negative t.

Proposition 6. Let $\mathfrak{B}$ be a regular state. $\mathfrak{B}$ is steady if and only if k is a divisor of d and there exists a number j such that $1 \leqq j \leqq d/k$ and the equivalence

$$\alpha_i(0)=1 \Leftrightarrow A_i \in E_j \cup E_{j+k} \cup E_{j+2k} \cup E_{j+3k} \cup \cdots \cup E_{j+d-k}$$

holds.

§ 8. On the regularity of cyclic states

Our Proposition 4 is an exact analogon of Proposition 2 of the article [2]. For the networks of the type investigated in [2], the converse statement is true as well: only the regular states are cyclic ([2], Proposition 8). Now we are going to make some considerations (without any claim for completeness) on the question whether or not a similar assertion concerning the networks lying in C_2 holds.

First we characterize the steady states (without presupposing the regularity). Let A, B be two vertices of a graph G; we say that A is k-reachable from B if there exists a path of length $\leqq k$ from B to A ($A=B$ is permitted).

Proposition 7. Let G, k, G' have the same meaning as in the definition of the regular state. Let $\mathfrak{B}$ be a state of the network G (at the instant 0). Denote by H the set of the vertices A_i satisfying $\alpha_i(0)=1$. The state $\mathfrak{B}$ is steady if and only if the following three conditions are fulfilled:

(i) *$A_i \notin H$ implies $\alpha_i(0)=0$ for all the vertices A_i of G.*

(ii) *If $A \in H$, $B \in H$ and A is $(k-1)$-reachable from B in G', then $A=B$.*

(iii) *To any vertex A of G there exists a vertex $B(\in H)$ such that A is $(k-1)$-reachable from B in G'.*[17]

Proof. If (i), (ii), (iii) are fulfilled, then at each vertex of G outside H at least one edge of G coming from an element of H terminates (by the operation $\mathfrak{A}_k$), hence all the initial values $\alpha_i(0)$ remain unchanged.

Assume that one of (i), (ii), (iii) is not satisfied. If (ii) were not true, then state in question would not be permitted. If either (i) or (iii) were not valid, then a vertex A_i would exist such that $\alpha_i(t)$ would increase in an interval $[0, t')$ with an appropriate positive t'. Thus the state could not be steady. The proof is complete.

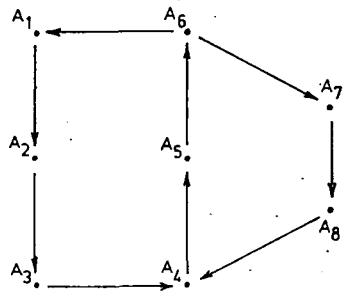


Fig. 6

[17] In case $A \in H$ the statement is satisfied with $B=A$ trivially.

Consider the graph $G=\mathfrak{N}_3(G')$, where G' is the graph on Fig. 6. If we put

$$\beta_1=\beta_2=\beta_4=\beta_5=\beta_7=\beta_8=0, \quad \beta_3=\beta_6=1 \quad (\text{where } \beta_i=\alpha_i(0))$$

then we get a steady state (since (i), (ii), (iii) are fulfilled) that is not regular. Thus the statement "any cyclic state is regular" does not hold. However, it can be expected that all non-regular cyclic states are steady, or (equivalently):

Conjecture 4. Any proper cyclic state of a network of type C_2 is regular.

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

References

[1] ÁDÁM, A., Simulation of rhytmic nervous activities, II. (Mathematical models for the function of networks with cyclic inhibition), *Kybernetik*, v. 5, 1968, pp. 103—109.
[2] ÁDÁM, A., & U. KLING, On the behaviour of some cyclically symmetric networks, *Acta Cybernetica*, v. 1, 1971, pp. 69—79.

(Received July 22, 1970)

VIDEC: A universal visual input for digital electronic computers

By D. MUSZKA and A. SÁRA

In the investigations concerning pattern recognition, the adoption of electronic computers recently became more and more frequent. For a long time it had been impossible because of the absence of suitable technical aids. Though lateron there came some visual readin instruments connectable with digital electronic computers into being, the most of them, however, were special purpose machines which, according to the demand of the commercial data processing, served in general just as reading-in vouchers, for the identifications of alphanumerical signs.

Another, but much smaller group of visual readin instruments has already been useful in scientific research — primarily in medical technics —, as they have proved to be suitable for analysing more complicated visual information sets. These instruments, however, have never been able to come into general use because of their elaborateness and, so, their high prices.

Among these instruments let us mention just for an example the FIDAC instrument, well-known in England, built by Ledley and his collaboraters, in which the film to be processed is resolved by a cathode ray tube (as light source), controlled by a digital sweep circuit, and the greyness degree of its picture-elements is forwarded in an appropriately encoded form to the computer.

In Hungary, on the request of medical research institutes, the Cybernetical Laboratory is continuing researches to develop a suitable technical appliance. The primary goal of this work consists in developing a relatively not expensive visual readin instrument which, connected to any digital electronic computer, is capable of processing both transilluminatable and light-reflecting pictures. In the following, I should like to give a review on our so-scalled VIDEC instrument, which constitutes the produce of the first stage of our research work.

The pictures to be processed (e.g. *X*-ray photographs, diagrams drawn by EEG, ECG registering devices etc.) can be fixed on the superficies of a cylinder made of glass. Taking into account that most of the pictures to be processed are such that the greyness degree of the picture-elements of the visual patterns on the pictures does not carry any essential new information (for instance this is the case of an ECG diagram) we decided to consider the elements, under an appropriate resolution of the picture, as dark and light points.

Along the generating line of the cylinder there are placed 5—8 point-like sources of white light. Near the inner and outer surface of the picture cylinder, according to the position of the sources of light, there are built in 5—8 diodes of appropriate

resolving power. (In case of a transilluminatable picture the inner, in case of a light-reflecting picture, the outer diodes are activated, respectively.) The photodiodes and the sources of light are placed along the generating line proportionally in the sense that if the size of the picture to be processed, taken in the direction of the generating line, is "*b*" and we have "*n*" photodiodes then the distances between the photodiodes equal "*b*/*n*". (This distance can be changed on the instrument, and before starting the visual input one has to take care of its appropriate adjustment.) The number of the photodiodes is determined by how many channels of the computer

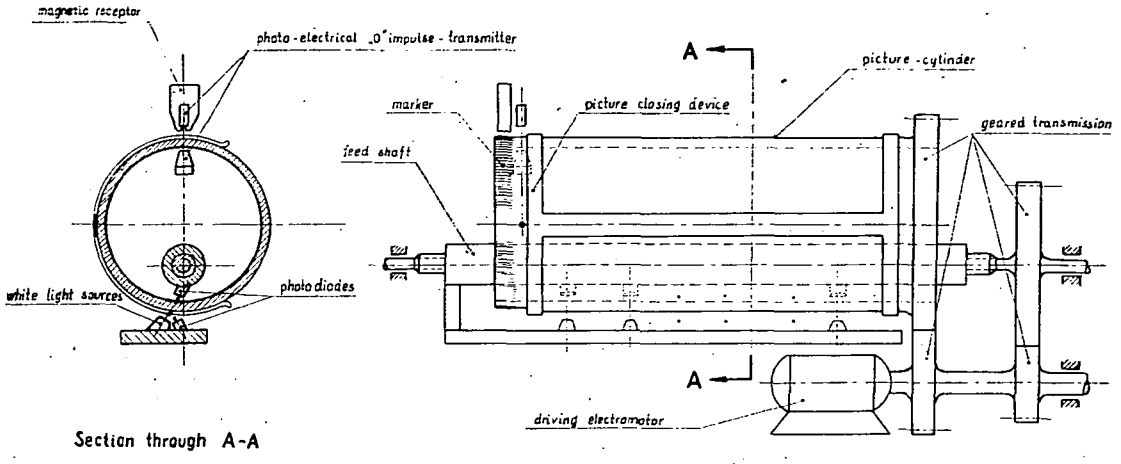


Fig. 1

the instrument is connected to are ready to receive information simultaneously. In case of our instrument, "*n*" cannot exceed 8. The length of generating line of the cylinder is 240 mm, so the size "*b*" of the picture to be processed can be 240 mm at most. The circumference of the picture cylinder is 270 mm, thus taking into account the size of the picture fixing device as well, the size "*a*" of the picture to be processed is 240 mm at most.

On the surface of the picture cylinder, outside the field of picture, there are magnetic Marker signs of density corresponding to the finest resolution applicable. The attainable finest resolution (raster) is 0,5 mm × 0,5 mm, so the number of the Markers is 480. The Markers cannot be found all along the circumference, just on the section of 240 mm. The device fixing the edges of the picture is placed inside the remaining stripe of width 30 mm, on the picture cylinder.

In the following, let "Δe" always denote the resolving fineness in both directions (the possible values of "Δe" are 0,5—1, 0—2,0 mm). The choice of the value "Δe" depends in part on the resolving fineness required by the task in question, in part on the operative memory capacity of the computer we are to work with. In fact, in case of the finest resolution (0,5 mm × 0,5 mm) and of the maximal size of picture (240 mm × 240 mm), 480 × 480 = 230.400 bits as information have to be stored which, in case of an average computer word length (30 bits), means 8*K* word memory capacities.

Here we should point out the fact that the proportional placement of photo-

diodes mentioned above implies that, after the input, the visual information is stored in the memory unit of the computer in a form unfit for direct processing. In fact, the bits corresponding to neighboring picture-elements are no longer "neighboring" in the memory. Therefore — with the aid of the VIDECORD ordering program[1] to be presented at the end of this talk — the visual information set stored in the memory before processing has to be rearranged into a form suitable for processing. This problem of rearrangement could be eliminated if the photodiodes were placed directly beside each other at a distance corresponding to the resolving finess. However, the construction of photo-perceptors of a sufficiently small size gives a grave technical problem as well as an additional expenditure. Therefore, the divided proportional placement of the photodiodes which made it possible to use any photo-perceptor and optics seemed to us more practical. Furthermore, the running time of the VIDECORD ordering program in most of the cases of the pattern recognition programs is much less than that of an effective pattern recognition program.

The operation unit of the instrument receives the starting impulse from the computer. Then the actuating engine of the instrument starts working and rotates the picture cylinder with 750 revolutions per minute. Thus, the time needed for the input of one picture varies between 5 and 8 second. (In case of a five channel and eight channel input system this is 8 and 5 seconds, respectively.)

The maximal speed of the information transport (in case of the finest resolution and 8 channels) is 54.000 bits per second. Just for comparison: the maximal information transport speed of a modern punched tape reader, RC-2000 of GIER Co., is 16.000 bits per second.

The actuating engine of the instrument, rotating the picture cylinder simultaneously, moves the photodiodes and sources of light with the aid of a suitable cog-wheel transmission and pulling spindle in axial direction. The cog-wheel is chosen in such a way that during a full revolution of the picture cylinder the photodiodes and sources of light cover the distance "Δe". This cog-wheel transmission has, of course, to be retooled, according the choice of the resolving fineness.

The operation unit counts the number of revolutions of the picture cylinder (with the aid of the photo-electric "0"-impulses), and after "$b/(n \cdot \Delta e)$" revolutions it automatically changes the direction of rotation of the actuating engine, not allowing any further information flow, and it gives a signal to the electronic computer to begin the VIDECORD-program. Now if the photodiodes and sources of light moving backwards come again into ground position (the first diode ought to be just at the edge of the picture) the operation unit stops the actuating engine and it changes the sense of rotation again.

During the working of the instrument, the signs of the photodiodes, arriving continuously, get on Schmitt-triggers. The output levels of the Schmitt-triggers control gate stages. The signs gating the gate stages are furnished by Marker signs. The Marker signs are emitted by a magnetic perceptive head. At the value "Δe"$=$ $=0{,}5$ mm every Marker sign is effective. At "Δe"$=1{,}0$ mm every second, at "Δe"$=2{,}0$ mm every fourth Marker sign is effectual, respectively.

The output impulses of the gate stages are directly connected to the input unit of the computer. The effectual Marker signs also serve to synchronize the com-

[1] The ordering program is maked by P. Hunya.

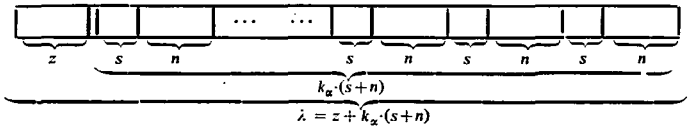
puter. Thus, the picture is transmitted to the computer in a form of parallel-series information. Each photodiode scans just a given zone of the picture, but their joint information gives the entire field of picture. In the further development of the instrument we have set ourselves the task of making the instrument capable of distinguishing the nuances retraceable in the picture. This, on the one hand, raises the requisite concerning the memory capacity of the computer, on the other hand, increases the electronics of the instrument. In fact, in this case, the signs of the photodiodes are conducted onto analogue-digital converters, instead of Schmitt-triggers, and according to the greyness degree of the picture-element, at least 3 bit code per picture element has to be transmitted to the computer.

To carry out pattern recognition experiments (human chromosome analysis) the instrument is connected to a MINSK-22 electronic computer working in our Laboratory.

* * *

The VIDECORD ordering program consists of two parts. The first part of it is suitable for horizontal investigations, and it converts the information into a so-called row-continuous form. If vertical investigations are needed, then the second part of the program, which arranges the information, previously brought into a row-continuous form, is also be used. The entire VIDECORD program leaves the original information set fixed, therefore for the rearranged information as large additional memory capacity is needed as for storing the original information set.

Let the length of word of the computer adapted to the VIDEC instrument be "λ" bits in general. This word plotted against the input visual information is of the following form:



where z = the number of possibly unused bits (e.g. sign-bit);

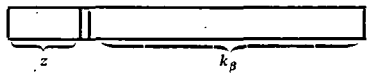
n = the number of elementary informations read-in an effectual Marker sign. It is identical with the number of the input channels (this can be 5, 6, 8);

s = the number of unused bits depending on the input system (ordinarily 0 or 1);

k_α = the number of picture characters of "n" bits stored in one word.

Thus the memory capacity needed to store a picture consisting of $e \cdot e = e^2$ picture-elements is $C = \dfrac{e^2}{k_\alpha \cdot n}$ words. (The possible values of "e" are 120, 240 and 480.) After each k_α-th effectual Marker sign the VIDEC instrument gives the computer a distinct operating sign to write the information, read in up to that time, in the memory and to increase automatically the adress. k_α has to be chosen such that "e/k_α" is an integer, and that the inequality $k_\alpha(s+n) \leqq \lambda < (k_\alpha+1) \cdot (s+n)$ holds. The first condition is necessitated by the requirement that, after a full revolution of the picture cylinder, the computer should start storing into a new word by all means, the second one aims the best exploitation of the memory. The values of k_α on the VIDEC instrument can be adjusted 3, 4, 5, 6 and 8.

The placement of the information brought into a row-continuous or column-continuous form in one word is the following:



$$\gamma = z + k_\beta$$

where k_β = the number of the bits used in one word, whether after a row-continuous or a column-continuous rearrangement.

After the rearrangement the memory capacity needed for storing is $C' = \frac{e^2}{k_\beta}$

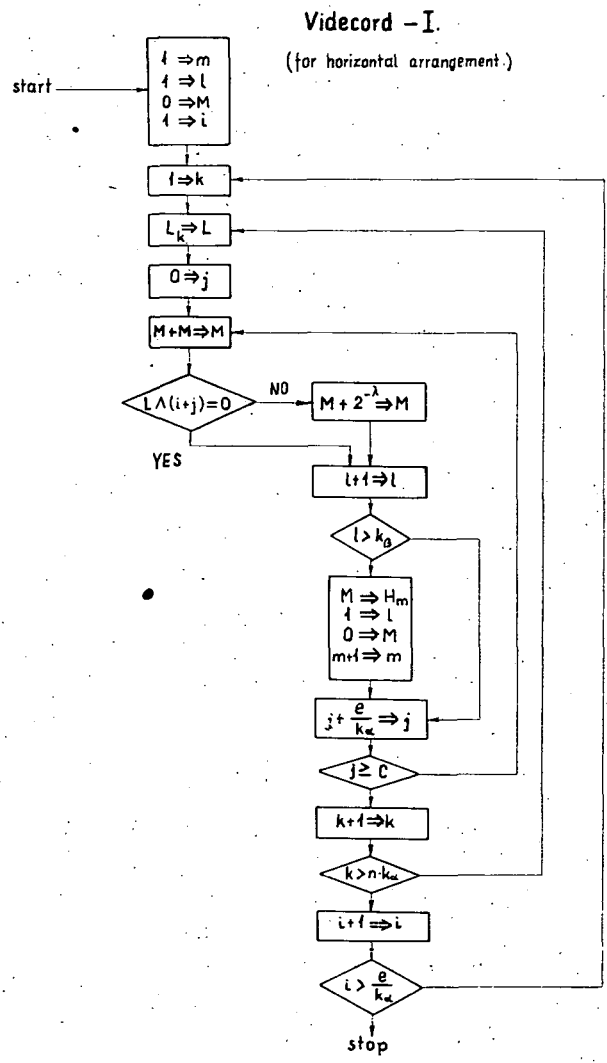


Fig. 2

word. k_β is to be chosen such that "e/k_β" is an integer and that the inequality $k_\beta \leqq \lambda$ holds.

Before the run of the VIDECORD program the following parameters are to be made known to the computer:

e^2 (the number of the picture-elements);
n (the number of the input channels);
k_α (the number of picture characters stored in one word);
k_β (the number of bits used in one word after the rearrangement.)

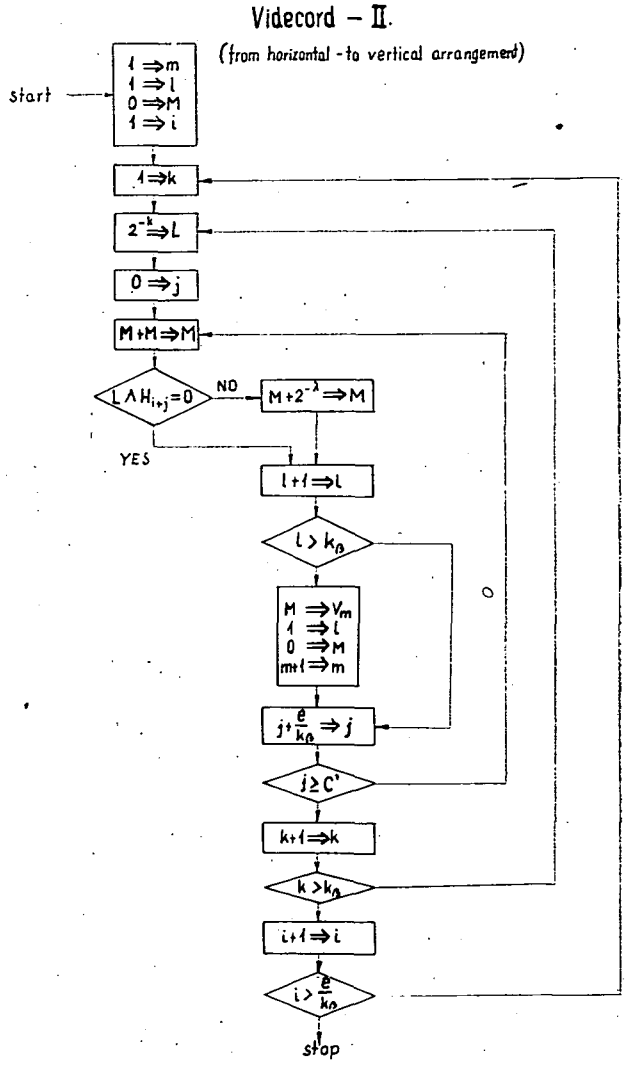


Fig. 3

In the block diagram of the VIDECORD program we use, in addition, the following notation:

H_m = the m-th word of the information brought into a row-continuous form;
V_m = the m-th word of the information brought into a column-continuous form;
i, j, k, l, m = indices;
M = work compartement in the operative memory;
L_k = logical constants depending on the input system separating the corresponding picture-elements ($k = 1, 2, 3, \ldots, n \cdot k_z$).

CYBERNETICAL LABORATORY OF THE
JÓZSEF ATTILA UNIVERSITY,
ARADI VÉRTANÚK TERE 1,
SZEGED, HUNGARY

(Received Dec. 22, 1969)


08 290010 012008 29

INDEX—TARTALOM

Felelős szerkesztő és kiadó: Kalmár László
A kézirat nyomdába érkezett: 1970. november hó
Megjelenés: 1971. szeptember hó
Példányszám: 1000. Terjedelem: 7,68 (A/5) ív
Készült monószedéssel, íves magasnyomással,
az MSz 5601-90 és az MSz 5602-55 szabvány szerint
70-6401 — Szegedi Nyomda