Tomus 4. Fasciculus 3.

ACTA CYBERNETICA

FORUM CENTRALE PUBLICATIONUM
CYBERNETICARUM HUNGARICUM

Szeged, 1979
Curat: Universitas Szegediensis de Attila József nominata

4. kötet 3. füzet

ACTA CYBERNETICA

A HAZAI KIBERNETIKAI KUTATÁSOK
KÖZPONTI PUBLIKÁCIÓS FÓRUMA

Szeged, 1979. december
A Szegedi József Attila Tudományegyetem gondozásában

Estimation of average length of search on random zero-one matrices

By A. BÉKÉSSY

The real content of this short paper is simply a theorem about zero-one matrices. In order to enlighten the background however, reference is made to a certain method of data retrieval.

Let there be given a zero-one matrix of size $m \times n$ such that all of its rows are different from each other. Let us suppose that the rows of this matrix constitute a primary key to a certain file of records stored in a computer. Therefore, the rows of the matrix will be called "names". Our problem is to find the location of any particular name (and the record associated with it) quickly, whenever wanted. The most rapid search-algorithms performing this job, e.g. "binary search" [1] are based on comparisons of the names by their magnitudes and if one complete comparison is counted one decision step then the average number of decision steps to be done for finding any name comes close to the lowest theoretically possible information limit, this latter being $\log_2 m$ if all names are looked for with equal frequencies. A complete comparison of two names, however, requires a considerable amount of time on some computers, so other procedures, though less effective in terms of decision steps, might come into consideration, too, if an elementary decision step is less time consuming.

The simplest looking search strategy would consist of decision steps to be performed column-by-column: given the name to be found the first column of the name-matrix is inspected first. If it consisted of zeros (or ones) only then we pass over to the second column immediately. If not then one decision is counted and the subset of those names is selected whose first column bit was identical to that of the name to be looked for. The second column is then inspected in the same way but restricted to the subset of names selected before, and so on, until the name is completely identified. For finding each name the steps to be made are completely determined by the structure of the name matrix and can be represented by a "search-tree" (Table 1, Fig. 1). The numbers in the nodes show the column no. of the bit the decision should be made on.

The strategy described above would not come into consideration at all should it be done in "run-time" i.e. when the names are looked for repeatedly and be found as quickly as possible. But assumed the file does not change often there might have

been ample time for constructing the corresponding search-tree or, more precisely, an equivalent "search-table" [2] when the file was generated.

The search-table (Table 2) is a list of two pointers. The first column indicated shows the relative location address of that line only; it does not belong to its content. The real first column field is the serial number of the bit to look at, and according to whether it proves to be zero or one the first, resp. the second pointer should be followed by the search-algorithm working in run-time. Zero in the first column would indicate that the search has its end there and the fields belonging to this line would contain the record or a single pointer to that record, for instance.

Names	Column no. 1	2	3	4	5
1.	1	0	1	0	0
2.	1	0	0	0	1
3.	0	1	0	1	0
4.	1	0	0	1	1
5.	1	0	1	1	1
6.	1	0	0	1	0

Table 1. Name-matrix

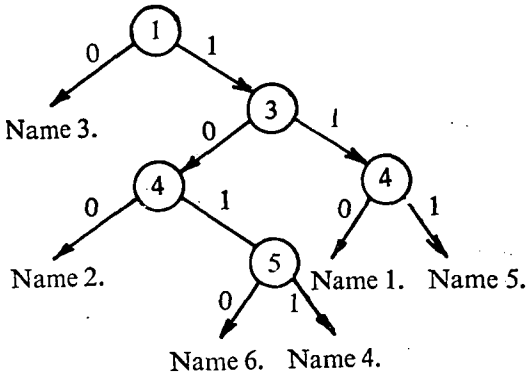


Figure 1
Search-tree to the matrix of Table 1.

A serious objection against the simple strategy described above is that it might, in some cases, result in a highly unbalanced search-tree. For the worst matrices the average number of the necessary decisions is as high as $(m+1)/2$ about. But matrices of ill behaviour, i.e. *matrices with highly unbalanced search-trees are rare.* This is the meaning of the theorem shown below.

Remark. It is possible, in practice, to make the algorithm a bit more flexible: let the decision in turn to be performed on the column in which the zeros and ones

Location	Col. no.	Pointer 1.	Pointer 2.
L+1	1	L+2	L+3
L+2	0	Pointer to record 3.	
L+3	3	L+4	L+9
L+4	4	L+5	L+6
L+5	0	Pointer to record 2.	
L+6	5	L+7	L+8
L+7	0	Pointer to record 6.	
L+8	0	Pointer to record 4.	
L+9	4	L+10	L+11
L+10	0	Pointer to record 1.	
L+11	0	Pointer to record 5.	

Table 2. Search-table to the matrix of Table 1.

Name	Number of Decisions
1.	3
2.	3
3.	1
4.	4
5.	3
6.	4
	18

Ave. number of steps: 18:6=3
Table 3.
Number of decisions to be done according to Table 2.

are distributed most evenly for that particular subset of names that was selected in the previous step. This will help in a lot of cases where the first approach would result in a highly unbalanced tree.

Now we prove the following

Theorem. Let all zero-one matrices of size $m \times n$ with all rows different be considered and supposed to be equiprobable. Let $E(M_{m,n})$ be the arithmetic mean of decisions to be made in order to find each row of matrix $M_{m,n}$ according to the simple strategy described above. Let $\mathscr{E}_{mn}$ be the expectation of the averages $E(M_{m,n})$. Then for all $m \leqq 2^{n-1}+1$

$$\mathscr{E}_{mn} < 1+\left(1+\frac{1}{2}+\ldots+\frac{1}{m-1}\right)/\ln 2 \tag{1}$$

or, because of $1+\frac{1}{2}+\ldots+\frac{1}{m-1}=\ln m+\gamma+0\left(\frac{1}{m}\right)$ (where $\gamma=0.577\ldots$ Euler's constant),

$$\mathscr{E}_{mn} < 1.833\ldots+\log_2 m+O\left(\frac{1}{m}\right), \quad (m, n \to \infty, \ m < 2^{n-1}+1)^* \tag{2}$$

Proof. Let $N(m, n, d_1, d_2, \ldots, d_m)$ be the number of matrices $M_{m,n}$ such that d_i decisions have to be made for finding the i-th row ($i=1, 2, \ldots, n$). Then $E(M_{m,n})=\sum_i d_i/m$ and

$$\mathscr{E}_{mn} = \frac{1}{N} \sum_{d_1, d_2, \ldots, d_m} E(M_{m,n}) N(m, n, d_1, d_2, \ldots, d_m)$$

where $N=\binom{2^n}{m} m!$ is the number of all matrices $M_{m,n}$. The latter expression can be simplified, because of symmetry in the variables d_i, to

$$\mathscr{E}_{mn} = \frac{1}{N} \sum_d d \cdot N(m, n, d) \tag{3}$$

where $N(m, n, d)$ is the number of matrices $M_{m,n}$ such that there are d decisions needed for selecting the *first* row. For this number $N(m, n, d)$ the recursion

$$N(m, n+1, d) = 2N(m, n, d)+2\sum_{j=1}^{m-1}\binom{m-1}{j-1} N(j, n, d-1)\cdot\binom{2^n}{m-j}(m-j)! \tag{4}$$

holds. The first term gives account on the matrices the first column of which consists of zeros (or ones) only. The j-th term under the summation is the number of matrices

* It is thought that the condition $m \leqq 2^{n-1}+1$ is, in fact, not necessary. Also the constant might perhaps be improved to $-0.5+\gamma/\ln 2=0.33\ldots$.

with j zeros in the first column while the first-row-first-column bit is zero, as well. Matrices of j ones in the first column when the first-row-first-column bit is one, are of the same number; therefore the sum should be multiplied by two. The boundary conditions

$$N(m,1,d)=\begin{cases}2 & \text{if} \quad m=1,\ d=0,\\ 2 & \text{if} \quad m=2,\ d=1,\\ 0 & \text{otherwise};\end{cases}$$
$$N(m,n,0)=\begin{cases}2^n & \text{if} \quad m=1,\\ 0 & \text{otherwise}\end{cases} \tag{5}$$

complete the recursion (4).

Introducing the function

$$H(x,n,y)=\sum_{d=0}^{\infty}\sum_{m=1}^{\infty} y^d \frac{x^{m-1}}{(m-1)!} N(m,n,d) \tag{6}$$

we obtain

$$H(x,1,y)=1+xy,$$

$$H(x,n+1,y)=2\cdot\{1+y[(1+x)^{2^n}-1]\}\cdot H(x,n,y)$$

from (4) and (5), with solution

$$H(x,n,y)\equiv 2^n\cdot\prod_{p=0}^{n-1}\{1+y[(1+x)^{2^p}-1]\}. \tag{7}$$

Since by (3) and the definition (6) of H

$$\left.\frac{\partial H}{\partial y}\right|_{y=1}=\sum_m\binom{2^n}{m}mx^{m-1}\mathscr{E}_{mn}$$

it follows easily

$$\mathscr{E}_{mn}=n-\sum_{p=0}^{n-1}\binom{2^n-2^p}{m-1}\Big/\binom{2^n-1}{m-1} \tag{8}$$

or, under the restriction $m\leqq 2^{n-1}+1$

$$\mathscr{E}_{mn}=\sum_{p=0}^{n}\left[1-\prod_{j=1}^{m-1}\left(1-\frac{2^{-p}}{1-j2^{-n}}\right)\right]. \tag{9}$$

Again, for $m\leqq 2^{n-1}+1$

$$\mathscr{E}_{mn}<1+\sum_{p=1}^{n}\left[1-\prod_{j=1}^{m-1}\left(1-\frac{2^{-p}}{1-j2^{-n}}\right)\right]<$$
$$<1+\int_1^{\infty}\left[1-\left(1-\frac{2^{-x}}{1-(m-1)2^{-n}}\right)^{m-1}\right]dx \tag{10}$$

giving the end-result.

Remark 1. For $n \to \infty$, $m = \text{const.}$

$$\mathscr{E}_{mn} \to \mathscr{E}_{m\infty} = \sum_{p=1}^{\infty} \left(1-(1-2^{-p})^{m-1}\right).$$

Could it be proved $\mathscr{E}_{mn} \leqq \mathscr{E}_{m\infty}$ for all n, a better constant would be achieved in the inequality (2).

Remark 2. The problem dealt with here resembles strongly that of calculating the average hight of random trees [3]. Instead of looking, however, for an appropriate link between the two problems the straightforward method presented here seemed to be simpler.

Abstract

The average efficiency of a simple search algorithm defined on random zero-one matrices is estimated.

COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES
P. O. BOX 63
BUDAPEST, HUNGARY
H—1502

References

[1] MARTIN, J., *Data-base organization,* Prentice-Hall, 1975, pp. 254—406.
[2] KNUTH, D. E., *The art of computer programming,* Vol. I, Addison-Wesley, 1968, pp. 315— 16.
[3] RÉNYI, A., G. SZEKERES, On the height of trees, *J. Austral Math. Soc.*, v. 7, 1969, pp. 497—507.

(Received Jan. 24, 1979)

On the equivalence of candidate keys with Sperner systems

By J. DEMETROVICS

1. Introduction

The use of the relational data model proposed by E. F. CODD [1—3] is to make many problems mathematically describable. In this model all data are represented by two-dimensional tables with rows representing records, and with coloumns representing attributes. Rows are identified by the values of a subset of attributes, if these are not identical for two different rows. These subsets of attributes are called keys and those keys which contain no further keys as subsets are called candidate keys.

Functional dependencies were introduced in 1970 by Codd, but were investigated mathematically only later [4, 5, 8]. In this paper we prove, that for any Sperner system we can construct a relation the set of candidate keys of which is the same as the Sperner system. It is clear, that apart from trivial cases the set of candidate keys of any relation is a Sperner system. At most $\binom{n}{\left[\frac{n}{2}\right]}$ candidate keys may exist in a relation of n attributes and we prove that this limit can be reached by relations with linear dependencies.

2. Definitions

Definition 1. Given the not necessarily different sets $D_1, D_2, \ldots, D_n$, the relation R of n variables denoted by $R(n)$ is a subset of the Cartesian product $D_1 \times D_2 \times \ldots \times D_n$. We shall call the sets D_i domains.

Definition 2. Indices of the domains of the relation $R(n)$ will be called attributes. Values associated to attributes will be called attribute values.

Remark 1. Though the domains of a relation are not necessarily distinct, their attributes are distinct.

In the present paper all domains are sets of natural numbers and the set of their indices in $R(n)$ are denoted by

$$N \quad (N = \{1, 2, \ldots, n\}).$$

Definition 3. The subset of indices $A \subseteq N$ will be said to generate the index k, in notation $A \to k$, if in any row of the relation $R(n)$ the values d_j $(j \in A)$ determine the value d_k uniquely. If in addition, for all rows in $R(n)$, d_k is a linear combination of d_j's $(j \in A)$, than A generates the index k linearly. The subset of indices $B \subseteq N$ will be said to be generated by A if every index in B is generated by A, denoted by $A \to B$. The link $A \to B$ is called a functional dependency in the relation $R(n)$. If A generates every index in B linearly, we say the functional dependency is linear. The set of all functional dependencies in $R(n)$ is denoted by $\{A_i \to B_i\}$ $(i = 1, 2, \ldots, v)$.

Definition 4. Let $A \subseteq N$, $A \neq \emptyset$ and $A \to N$. A is called a candidate key in the relation $R(n)$ if $B \to N$ does not hold for any of its nontrivial subsets B.

Definition 5. The functional dependency $A \to B$ is trivial if $B \subseteq A$. The sets of trivial and nontrivial functional dependencies in the relation $R(n)$ will be denoted by $\mathscr{H}$ and $\mathscr{G}$, respectively.

Remark 2. It is easy to see that in a relation $R(n)$

$$|\mathscr{H}| = 3^n$$

3. The link between candidate keys and Sperner systems

In the present paragraph we shall demonstrate a one-to-one correspondence between the set of candidate keys in a relation $R(n)$ and a Sperner system $\mathscr{S}(n)$ over N.

Definition 6. Let $\mathscr{S} = \{S_1, S_2, \ldots, S_m\} \subseteq 2^N$. $\mathscr{S}$ will be called a Sperner system if it satisfies the following relations,

$$S_i \subset N \quad \text{for} \quad i = 1, 2, \ldots, m; \tag{1}$$

$$S_i \not\subseteq S_j \quad \text{for} \quad i \neq j, \quad i, j = 1, 2, \ldots, m. \tag{2}$$

Trivially, the set of the candidate keys in every relation is a Sperner system or has only one element N. Conversely consider now the following Sperner system:

$$\mathscr{S} = \begin{pmatrix} S_1 = \{a_{11}, a_{12}, \ldots, a_{1m_1}\} \\ S_2 = \{a_{21}, a_{22}, \ldots, a_{2m_2}\} \\ \ldots\ldots\ldots\ldots\ldots\ldots \\ S_m = \{a_{m1}, a_{m2}, \ldots, a_{mm_m}\} \end{pmatrix}$$

with $a_{ij} \in N$ and $\bigcup_{i,j} a_{ij} = N$. This Sperner system is a covering of N.

Theorem 1. To every Sperner system $\mathscr{S}$ a relation $R_{\mathscr{S}}(n)$ of n variables can be constructed with the set of the candidate keys equivalent to the Sperner system $\mathscr{S}$.

Proof. First we shall construct the class of sets $\mathscr{M} = \{A_1, A_2, \ldots, A_t\}$. Let A_j belong to $\mathscr{M}$ iff the following conditions hold:

$$A_j \subseteq N, \quad j = 1, 2, \ldots, t \tag{3}$$

and

$$A_j \cap S_i \neq \emptyset \quad \text{for} \quad i = 1, 2, \ldots, m. \tag{4}$$

We shall choose $\mathscr{F}$ as the set of the elements minimal in $\mathscr{M}$, i.e.

$$A_j \in \mathscr{F} \Leftrightarrow \bar{\exists} A_l \in \mathscr{M}\colon (A_l \subset A_j). \tag{5}$$

From (3), (4) and (5) we have

$$\max\{m_1, m_2, \ldots, m_m\} \leqq |\mathscr{F}| \leqq m_1 \cdot m_2 \cdot \ldots \cdot m_m \tag{6}$$

and

$$A_j \in \mathscr{F} \quad \text{implies} \quad 1 \leqq |A_j| \leqq m. \tag{7}$$

Let us consider the following subsets

$$\mathscr{F}_k (k = 1, 2, \ldots, n) \quad \text{of} \quad \mathscr{F}:$$

$$A_j \in \mathscr{F}_k \Leftrightarrow k \in A_j \in \mathscr{F}. \tag{8}$$

We state that if the k'th index of the relation $R_{\mathscr{S}}(n)$ is determined by the function f_k, the latter identical with the class of sets $\mathscr{F}_k$, then the relation obtained satisfies the conditions of Theorem 1, i.e. the class of the candidate keys in $R_{\mathscr{S}}(n)$ is identical with the given system $\mathscr{S}$. Obviously, this last statement is implied by the following three statements:

a) all the sets S_i in the class $\mathscr{S}$ are keys;
b) no proper subset of S_i is key;
c) there is no candidate key beyond $\mathscr{S}$.

To verify these first we consider

a) Each S_i containing a key K_i $(i=1, 2, \ldots, m)$ is a consequence of

$$\bigcup_{k \in S_i} \mathscr{F}_k = \mathscr{F}. \tag{9}$$

This latter is obvious, as every $A_j \in \mathscr{F}$ is constructed so as to contain at least one element of S_i.

Next we show that the key K_i in S_i equals S_i. To do this

$$\forall a (a \in S_i)\colon \bigcup_{k \in \{A_i \setminus \{a\}\}} \mathscr{F}_k \subseteq \mathscr{F} \setminus \{A\} \quad \text{with} \quad A \in \mathscr{F} \tag{10}$$

is sufficient.

This follows from the existence of an $A \in \mathscr{F}$ with $A \cap S_i = \{a\}$. Indeed, for $j=1, 2 \ldots m$, every S_j contains either $\{a\}$ or some $\{a'\}$ with $\{a'\} \cap S_i = \emptyset$.

So we have proved that every $S_i \in \mathscr{S}$ is identical with a minimal key in the relation $R_{\mathscr{S}}(n)$. Now all we have left to prove is that $R_{\mathscr{S}}(n)$ has no minimal key K beyond those in $\mathscr{S}$.

For an indirect proof let us suppose the existence of such a minimal key. From Remark 1 we have $S_i \cap \bar{K} \neq \emptyset$ for $i=1, 2, \ldots, m$. Let the set A be determined by the sets c_i so that $A \in \alpha$ and $A \cap c_i \neq \emptyset$. It is easy to see, that at least one such set A exists and it is not contained in any of the columns determined by the candidate key K, i.e.

$$\bigcup_{k \in K} \mathscr{F}_k \subseteq \mathscr{F} \setminus \{A\}. \tag{11}$$

This completes the proof of the theorem.

Remark 3. Let us observe that the proof can be carried out the same way if such a class $\mathscr{L}$ of subsets in $\mathscr{M}$ is taken that $\mathscr{M} \supset \mathscr{L} \supset \mathscr{F}$ is fulfilled instead of $\mathscr{F}$. Out of theset he one of minimal cardinality was taken for our proof. If another have been taken, the set of functional dependencies of a form different from $x \to N$ would be changed and the set of candidate keys $\mathscr{S}$ would be unchanged.

The preceding statements can be interpreted as follows: let different prime numbers correspond to each set in the class $\mathscr{F}$, i.e. let $\mathscr{F} = \{p_1, p_2, \ldots, p_h\}$ be in ascending order for simplicity. So the sets in the classes $\mathscr{F}_k$ have their correspondants as well. Let then the function f_k of $|\mathscr{F}_k|$ variables equal the product of the corresponding primes to the sets in $\mathscr{F}_k$.

For example, let $n=5$ and

$$\mathscr{S} = \{\{1, 2, 3\} = s_1, \{3, 4, 5\} = s_2, \{1, 3, 4\} = s_3\}.$$

Then $\mathscr{F} = \{\{3\}=p_1,\ \{1, 4\}=p_2,\ \{1, 5\}=p_3,\ \{2, 4\}=p_4\}$ and $f_1=p_2 \cdot p_3$, $f_2=p_4$, $f_3=p_1$, $f_4=p_2 \cdot p_4$, $f_5=p_3$. Some rows of the relation $R_{\mathscr{S}}(n)$ corresponding to $\mathscr{S}$ are represented in Fig. 1 for

$\mathscr{F}^1 = \{2, 3, 5, 7\}$
$\mathscr{F}^2 = \{2, 5, 7, 11\}$
$\mathscr{F}^3 = \{3, 5, 7, 11\}$
$\mathscr{F}^4 = \{2, 5, 7, 13\}$

1	2	3	4	5
15	7	2	21	5
35	11	2	55	7
35	11	3	55	7
35	13	2	65	7

Fig. 1

i.e. $R_{\mathscr{S}}(5) \in \{(15, 7, 2, 21, 5), (35, 11, 2, 55, 7), (35, 11, 3, 55, 7), (35, 13, 2, 65, 7)\}$.

4. On the maximal number of candidate keys and on linear relations

Definition 7. We shall call the relation $R(n)$ linear provided all the functional dependencies in it are linear.

First we recall here Lemmas 1 and 2 and a Theorem from [8] in stronger forms. Namely, the result of the construction in the proof of Lemma 2 is a linear relation, therefore we can formulate both of them and the Theorem (as a consequence of the two Lemmas) as follows.

Lemma 1. A relation $R(n)$ may have at most $\binom{n}{\left[\frac{n}{2}\right]}$ candidate keys.

Lemma 2. There exists a linear relation $R(n)$ with $\binom{n}{\left[\frac{n}{2}\right]}$ candidate keys.

Theorem 2. There are linear relations $R(n)$ with as many candidate keys as $\binom{n}{\left[\frac{n}{2}\right]}$ and there is no relation $R(n)$ with more candidate keys.

Lemma 3. In a linear relation $R(n)$ all candidate keys have the same length.

Proof. Let A_k be a candidate key. As a consequence of the fact, that the functional dependency $A \to N$ is linear, we have a linear equation system

$$\sum_{j \in A_k} a_{ij}^k x_j = x_i \quad (i = 1, 2, \ldots, n),$$

which is satisfied by every row in $R(n)$. This is true for every candidate key A_k $(k=1, 2, \ldots, m)$, so we have the system

$$\sum_{j \in A_k} a_{ij}^k x_j = x_i \quad (i = 1, 2, \ldots, n;\ k = 1, 2, \ldots, m)$$

with the solution $R(n)$ in the preceding sense. Obviously, the set of indices of an independent set of variables $x_{i_1}, x_{i_2}, \ldots, x_{i_t}$ in this system composes a candidate key in $R(n)$ and conversely. Moreover, independent sets of variables have the same cardinality t, which completes the proof of Lemma 3.

As a consequence of this lemma, for linear relations Theorem 1 does not hold. Neither exist linear relations to every Sperner system $\mathscr{S}$ with the set of their candidate keys equivalent to it, as e.g. for $n=4$ and the Sperner system

$$\mathscr{S} = \{\{1, 2\}, \{1, 3\}, \{3, 4\}\}.$$

Considering a linear equation system as in the proof of Lemma 3 which has all subsets of the variables with the cardinality t independent, we have proved:

Theorem 3. There exists a linear relation $R(n)$ with $\binom{n}{t}$ candidate keys with t being their length.

In [5] it was proved that provided the number of dependencies $k \leqq \sqrt{n}$, a relation $R(n)$ exists with as many candidate keys as $\sqrt{n}!$.

S. OSBORNE and F. TOMPA have recently proved (draft paper) that at most $k!$ candidate keys can be deduced from k dependencies and for each k a relation R_k exists with exactly $k!$ candidate keys.

Each of the papers uses a system of derivation axioms which were introduced in [7] and [4], respectively. The first of them consists of 7 and the second of 4 axioms. Next we shall give a system of 3 axioms which is equivalent to the ones mentioned above.

Definition 8. The functional dependency $A \to B$ is deductible from the set of lunctional dependencies $\mathscr{F} = \{A_i \to B_i,\ i=1, 2, \ldots, k\}$ if it can be obtained from the latter using the derivation rules a; b; and c; a finite number of times.

a; $A \to A'$ with $A \supseteq A'$ is deductible from all $\mathscr{F}$,

b; $(A \to B) \in \mathscr{F}$ and $(B \to C) \in \mathscr{F}$ imply $(A \to C) \in \mathscr{F}$,

c; $(A \to B) \in \mathscr{F}$ and $(A \to C) \in \mathscr{F}$ imply $(A \to (B \cup C)) \in \mathscr{F}$.

By Theorem 4 an example is recalled from [8] in which the number of the undeductible functional dependencies is relatively high and this does not essentially diminish the number of candidate keys.

Theorem 4. Let $k=\binom{n}{\left[\frac{n}{2}\right]}$. A relation T of $n+1$ attributes exists with k undeductible functional dependencies and with the same number of candidate keys.

COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES
KENDE U. 13—17.
BUDAPEST, HUNGARY
H—1502

References

[1] CODD, E. F., A relational model of data for large shared data banks, *Comm. ACM*, v. 13, 1970, pp. 377—387.
[2] CODD, E. F., Normalized data base structure: A brief tutorial, *Proc.* ACM-SIGFIDET Workshop on Data Description, Access and Control 1971.
[3] CODD, E. F., Further normalization of the data base relational model, Courant Computer Science Symposia 6 *Data Base System,* Prentice Hall, Englewood Cliffs, N. J., 1971, pp. 33—64.
[4] ARMSTRONG, W. W., Dependency structures of data base relationships, *Information Processing 74,* North-Holland Publ. Co., 1974, pp. 580—583.
[5] YU, C. T., D. T. JOHNSON, On the complexity of finding the set of candidate keys for a given set of functional dependencies, *Inform. Process. Lett.*, v. 5, 1976, No. 4, pp. 100—101.
[6] SPERNER, E., Eine Satz über Untermengen einer endlichen Menge, *Math. Z.* v. 27, 1928, pp. 544—548.
[7] DELOBEL, C., R. C. CASEY, Decomposition of a data base and the theory of boolean switching functions, *IBM J. Res. Develop.*, v. 17, 1973, pp. 374—386.
[8] DEMETROVICS, J., On the number of candidate keys, *Inform. Process. Lett.*, v. 7, 1978, No. 6. pp. 266—269.

(Received April 2, 1979)

A new statistical solution for the deadlock problem in resource management systems

By Z. GIDÓFALVI

1. Introduction

Nowadays the efficient resource management is an important problem of the national economy, mainly in the case of expensive resources. The resources handled by the operating systems of computers are expensive enough, therefore their optimum usage is an important problem.

The solution may be the proper choose of resource management strategies, on the other hand the economical avoidance of deadlock situations.

The deadlock state is a special, undesirable state of the resource management systems, in which some processes executed simultaneously in the computer get deadlocked by their requests, and there is no way to destroy this situation without external interference. An expressive example could be those processes (P_1 and P_2), which want to transmit data between two tape units (R_1 and R_2), P_1 from R_1 to R_2 and P_2 from R_2 to R_1. Assume a state in which P_1 holds R_1 and P_2 holds R_2, and both request the not acquired units. Then the requests will remain unsatisfied forever, because none of them can release the already acquired resources, and so P_1 and P_2 are deadlocked, and the system is in deadlock state.

In simpler systems both the detection and elimination of deadlock were the operator's duty, but in great systems this way seems to be rather unefficient. It is advisable to entrust this work to the computer, which can make a decision on the basis of theoretically established algorithms. Unfortunately the time requirement of procedures developed for detection and prevention is great enough, and thereby the efforts made for rentability cannot achieve the expected results. The new method described in this paper guaranties a smaller time complexity, and has some other advantages, too.

2. The system

The system can be described after [3] with the triple $\mathrm{RMS}=\langle S, P, R\rangle$, where $S=\{S_1, \ldots, S_z\}$ is the set of system states, $P=\{P_1, \ldots, P_n\}$ is the set of processes, and $R=\{R_1, \ldots, R_m\}$ is the set of resources. Process P_i is a partial function

(P_i: $S \to 2^S$), because it can perform request, aquisition and release operations leading the system from state S_l into a set of states. Processes can interact explicitly, for example by exchanging messages, or implicitly, for example by competing for physical objects such as tape drives. Both types of interaction may cause the blocked state of processes, and they can be modelled by means of resources. The resources modelling the implicit interaction are called reusable resources, because after their request, acquisition and release by processes they are available again, and can be used in further cases. The resources modelling the explicit interaction are called consumable resources, because the message receiver process consumes (acquires) the resource produced (released) by the sender process after having requested it, and the consumed unit will never be available for other processes again. Moreover assume, that R_i has r_i units, where r_i is infinitely large for consumable resources.

The system states can be unambigously characterized by the number of resource units requested and acquired by processes, and can be illustrated by a directed bipartite graph [2, 3, 5, 6], where the nodes are from $P \cup R$, and the edges lead either from a P node to an R node or vice versa with the meaning:

1. $P_i \to R_j$ is a request edge indicating that P_i requests one unit from R_j;
2. $R_j \to P_i$ is an acquisition edge indicating that P_i holds one unit of R_j.

3. Deadlock strategies

If we do not influence the resource operations of P_i, and we have no information about the future requirement of P_i, then deadlock can always occur at the next step, which is to be detected and eliminated. The detection methods [1, 2, 3, 5, 6] decide, whether S_l is a deadlock state or not, and they are based on the following consideration: if there is a sequence, in which the processes can terminate their work one after the other from the state S_l — assuming that none of them will require more resources — S_l is not a deadlock state, because the former sequence is a possible state transition order. The time complexity of algorithms is not polinomial in general resource systems, where the resources are of both types. For reusable resources the algorithms require only mn steps, but the important consumable resources cannot be considered. The most complicated system, with mn time complexity of detection algorithm may have consumable resources, too, but with immediate allocation (all grantable requests to such resources are immediately fulfilled) [6]. There are efficient detection algorithms for restricted systems, too, but unfortunately with the same time complexity.

In the case of deadlock prevention we prohibit certain acquisition operations by means of information about the maximum claim of every P_i for all resources. We permit the acquisition in state S_l, when the maximum claim state $S_{l\max}$ (processes request their whole claim) is not a deadlock state, otherwise we prohibit this operation.

The deadlock avoidance is the simplest method for solving the deadlock problem. It means that the fulfillment of the necessary condition of deadlock should never allowed, since one can decide *a priori*, whether the system will reach a deadlock state or not. Often we use maximum claim information, too. The necessary condition of deadlock by [2, 3, 5] is the presence of at least one directed cycle in the graph

representing the state. Later we will show, that there exist more efficient necessary conditions used in the new statistical method.

There are mixed solutions for the deadlock problem, which apply different strategies for different parts of the system [4, 6].

4. Necessary conditions of deadlock

Deadlock can occur due to one resource (reusable or consumable) and due to more resources. We will divide the necessary conditions accordingly.

Let R be a *reusable resource* with r units requested and/or acquired by the processes $P_1, \ldots, P_n$ with $p_1, \ldots, p_n$ units $(p_i \neq 0)$. The necessary condition of deadlock on R is: $\sum_{i=1}^{n} p_i \geqq r+n$. To prove the expression assume, that the inequality is false. Then the number of edges directed to and from R is less than $r+n$. In the worst case all units of R are assigned to processes, and the remaining $n-1$ edges are requests. If we assume, that all the edges are directed from different processes to R (it is the worst case again), then there must be a process, say P_i, holding resources only. So we can find a sequence of processes beginning with P_i, in which all of the processes can terminate (not requiring more resources), and the state is deadlock-free.

Let R be a *consumable resource* produced (held) by $P_1, \ldots, P_n$. The necessary condition of deadlock on R is, that every process requests the units of R. To prove this statement assume, that there is at least one process producing R only. This process can produce an arbitrary number of units, thus all the others can terminate, and so the state is deadlock-free.

After the previous examinations we may consider at most two edges only between two arbitrary nodes, say P_i and R_j: $P_i \rightarrow R_j$ and $R_j \rightarrow P_i$. The necessary condition due to *more resources* is the presence of at least one directed cycle in the graph representing the examined state, including at least four nodes. To prove the former statement first we show, that a directed cycle in the graph is necessary. It is obvious, because without directed cycles in the graph we can make a queue from nodes, in which the edges from the k-th element are directed to the previous ones only. At the same time this order is a proper sequence of processes, in which they can terminate. Cycles appear through odd number of nodes only, but the cycles with two nodes are unimportant for us, because these ones were examined at the deadlock due to one resource. Therefore we can restrict the necessary condition for cycles including more than two nodes.

Contrary to [2, 5] the above necessary conditions permit directed cycles in the graph, when the conditions of deadlock due to one resource are satisfied.

5. The statistical method

Opposite to the former methods this strategy considers the system being determined by statistical information about its antecedents up to the moment of examination, and gives an approximate estimation about the existence of deadlock. We can assign a number v_i $(0 \leqq v_i \leqq 1)$ to every process and resource — the protec-

tion degree of P_i or R_i —, and w_i as the counterpart of v_i ($w_i=1-v_i$). The protection degree of a process or resource depends on its importance and value. Really w_i means the maximum allowed probability of coming the i-th node into a deadlock. The strategy examines the occurrence probability of deadlock on the basis of statistical information until this probability is smaller than the smallest w_i of all processes and resources concerned. If it is successful, then the deadlock occurrence probability is surely smaller than permitted, because it is surely smaller than the occurrence probability of necessary condition.

This strategy can be applied for detection and prevention, too, combined with detection algorithms, because deadlock can occur even if the deadlock occurrence probability is smaller than permitted, and naturally it is to be detected and recovered. At the application for detectional purposes we decide whether the prescribed protections can be satisfied or not — after the execution of a request or acquisition operation — in the future, too, and the exact detection algorithm is used accordingly. At the application for preventional purposes we examine the possible effects of acquisition before every acquisition operation, and we permit it, if the prescibed protections can be satisfied after the change, too. Naturally deadlock can occur in this case, too (with a small probability), thus the use of detection algorithms is necessary from time to time.

First of all let us see the statistical data needed for the decision. To gather them it is advisable to join the sampling of states with the resource operations, since in such cases there is a need for other administration activities, too. Let M be the operation counter with the initial value of zero, which is incremented by every operation. Moreover let $\mathbf{A}$ a hipermatrix of range $m \times n$, where a vector of five elements ($\mathbf{A}_{ij}$) belongs to R_i and P_j with the following meaning of coordinates:

1. k_{ji} — which shows the number of operations in which the graph had a request edge between P_j and R_i (the number of edges are irrelevant);
2. k_{jiu} — which shows, whether P_j had a request edge to R_i at the last operation between them ($k_{jiu}=1$ if it had, otherwise 0);
3. b_{ji} — which shows the number of operations in which the graph had an acquisition edge between P_j and R_i (the number of edges are irrelevant);
4. b_{jiu} — which shows, whether R_i had an acquisition edge to P_j at the last operation between them ($b_{jiu}=1$ if it had, otherwise 0);
5. M_{jiu} — which shows the value of M at the last rewriting of k_{ji} and b_{ji}.

This appearingly great amount of data makes possible to avoid updating all matrix elements at every operation (it would require $5mn$ steps). Rewriting of $\mathbf{A}_{ij}$ is needed only if P_j executes some kinds of operation with R_i or if we make use of the occurrence probability of the $R_i \rightarrow P_j$ and the $P_j \rightarrow R_i$ edges at the examination of the occurrence probability of necessary condition.

Thus k_{ji} gives the frequency of request occurrence, and b_{ji} gives the frequency of acquisition occurrence between P_j and R_i. The probability of these requests and acquisitions can be measured with their relative frequency:

the probability of $P_j \rightarrow R_i$ edge is k_{ji}/M, and

the probability of $R_i \rightarrow P_j$ edge is b_{ji}/M.

So we get a graph like the usual one representing the state, which characterizes

the antecedents of the system up to the moment of examination, and whose edges are existing with the former probabilities.

We will divide the examinations about occurrence probability of necessary condition according to the previous chapter. So we introduce the numbers:

s_i — for the value $\sum_{j=1}^{n} p_j$ at the reusable resource R_i, and for the number of requesting processes at the consumable resource R_i,

q_i — for the number of occurrence frequency of necessary condition on the resource R_i (both types),

q_{iu} — which shows, whether the necessary condition was satisfied after the last request or release operation with R_i ($q_{iu}=1$ if it was, otherwise 0),

M_{iu} — which shows the value of M at the last change of q_i, and

$w_{i\min}$ — which is the smallest w_k of R_i or the P_j's connected to it (the connected processes have at least one edge to the reusable resource R_i, or they are producers of the consumable resource R_i).

With the above numbers we can formulate the feasibility of the prescribed protections. The protection degrees can be satisfied in the case of deadlock due to one resource, if the occurrence probability of necessary condition (q_i/M) is smaller than the smallest w_k of R_i and the processes concerned $(w_{i\min})$, thus $q_i/M < w_{i\min}$.

To the determination of deadlock probability due to more resources assume, that P_j is executing an operation with the resources of $R^* \subseteq R$. Then we try to build the graph, beginning with P_j through the resources of R^* in the case of request, and beginning with the resources of R^* through P_j in the case of acquisition, i.e., we try to put the nodes into levels from the root (s) through the directly, secondarily etc. accessible nodes, where the k-th level consists of such resources or processes, to which there is an edge directed from at least one node of the $(k-1)$-th level. Thereby every node on the k-th level is accessible from the root (s) (P_j or R^*). Executing the request or acquisition operation the edges between the first and second level are existing surely, and our purpose is to examine the probability of a directed cycle in the graph, due to the newly introduced edges. Assume that after a request operation of P_j, P_j appears once more say on the k-th level. This means a directed cycle, which includes the nodes $P_j, a_1, \ldots, a_{k-2}, P_j$. Mentioned previously the probability of $P_j \to a_1$ edge is considered to be 1, and the probability of a certain, say $a_n \to a_{n+1}$, edge in the cycle is $k_{n,n+1}/M$ or $b_{n,n+1}/M$ depending on the type of the edge (request or acquisition). Since the processes perform their request and acquisition operations in a nondeterministic manner, the existence of any two edges are independent events, and so the probability of any cycle in the graph can be obtained by multiplying the occurrence probabilities of the edges in the cycle. The correlation between the cycles are ignored and so the access probability of a node accessible through more than one paths is substituted with the highest probability of paths.

But the complete tracing of the cycles is unnecessary, because we are interested in the feasibility of the prescribed protections, and not in the exact probability of a cycle. If there is an a_i in the former example, for which the probability of getting from P_j to a_i is smaller than the smallest w_k of nodes on the mentioned path, then the further examination of this cycle can be ignored, since the prescribed protec-

tion degrees are already satisfied on this path. For the sake of simplicity the existence of low probability parallel paths is neglected in this algorithm.

Summarizing we must continue the cycle detection from P_j until:

1. for all paths from P_j the condition mentioned above is fulfilled, and so we can satisfy the protection degrees, or until

2. P_j occurs once more in the graph, and the condition has not been fulfilled yet, so in this case we cannot satisfy the protection degrees for the nodes in the cycle.

In the case of an acquisition operation executed by P_j the graph building and the examinations can be made similarly.

The time complexity of the algorithm ([6]) performing the above examinations is less than mn, since the examination of feasibility on one resource requires one step only (altogether m steps), and the number of steps needed at the cycle detection is surely less than mn, because not all the $2mn$ edges are considered. The other essential advantage of the strategy is the option of giving different protection degrees to processes and resources. Thus it is possible to rise the protection degree of a process as a function of the performed work.

The strategy can be mixed with others, too. Thus it is easy to apply prevention or avoidance methods instead of statistical ones on singular resources. This is the case on every R_i, where $w_{i\min}=0$.

DEPT. OF MATHEMATICS
ELECTRICAL ENGINEERING FACULTY
TECHNICAL UNIVERSITY BUDAPEST
STOCZEK U. 2./H.
BUDAPEST, HUNGARY
H—1111

References

[1] COLDEWEY, H. D., J. GONSCHOREK, H. HOECHNE, E. SCHÖNBERGER, F. STETTER, Ein Modell zur Deadlockerkennung, *Ang. Informatik*, v. 17, 1975, pp. 65—69.

[2] HOLT, R. C., On deadlock in computer systems, Ph. D. thesis, Cornell Univ. Ithaca N. Y., 1971.

[3] HOLT, R. C., Some deadlock properties of computer systems, *Comput. Surveys*, v. 4, 1972, pp. 179—196.

[4] HOWARD, J. H., Mixed solution for the deadlock problem, *Comm. ACM*, v. 16, 1973, pp. 427—430.

[5] SHAW, A. C., *The logical design of operating systems*, Prentice Hall, 1974.

[6] GIDÓFALVI, Z., *Erőforrásgazdálkodó rendszerek pattállapotai és feloldásuk*, dissertatio, BME, 1978.

(Received Sept. 2, 1978)

On the Garden-of-Eden problem for one-dimensional cellular automata

By L. K. BRUCKNER

Definitions. 1. A *cellular automaton* (further on shortly CA) is a structure

$$U = (A, Z^d, x, f)$$

where

A — finite, nonempty set. It represents the set of states that can be taken by any cell.

Z^d — set of d-tuples of integers.

x — $(\xi_1, \ldots, \xi_n)$, neighborhood function $\xi_i \in Z^d$, $(i=1, \ldots, n)$. The neighborhood of a cell is the n-tuple

$$N(s) = s+\xi_1, s+\xi_2, \ldots, s+\xi_n$$

where $s+\xi_i$, $(1 \leqq i \leqq n)$ is the componentwise sum of the d-tuples. We say, that the neighborhood size of the CA is n.

f — local transformation function, $f: A^n \to A$. As it is a discrete function, it can be defined by a table. This table will be called the defining table of the local transformation function.

2. The local transformation function (its defining table) will be called balanced, if each $a \in A$ occurs as a value of the function just as many times.

3. A configuration is a mapping $c: Z^d \to A$. The set of all configurations will be denoted by C $(C=(c|c: Z^d \to A))$.

4. A mapping $F: C \to C$ defined from a local transformation f as follows will be called global transformation function

$$F(c) = c^1 \quad \text{if} \quad c^1(s) = f\left[c(N(s))\right] \quad s \in Z^d$$

where $c(N(s))$ is the restriction of mapping c to $N(s)$.

5. $a_0 \in A$ is a quiescent state if $f(a_0, \ldots, a_0) = a_0$. A quiescent state is always supposed to exist.

REMARK. Instead of any neighborhood an arbitrary d-dimensional array can be considered that contains the given neighborhood. The transition depends on the joined cells trivially, i.e., it is independent of them.

Further we assume such a neighborhood.

6. A configuration $c \in C$ is of finite support if only a finite number of values of $c: Z^d \to A$ differs from a_0. This finite set of cells in c will be called the support of the configuration. Denote by C_F the set of all configurations of finite support.

7. A configuration $c \in C$ will be called Garden-of-Eden or briefly Eden if there is no $c_1 \in C$ such that $F(c_1)=c$. This concept can be defined considering C_F instead of C as well.

It would be important to know whether a given CA contains Eden-configuration or not, for it is closely related to function F. More exactly, if there exists an Eden-configuration in a CA, the mapping F is not surjective.

Amoroso and Patt created an algorithm [1] by means of which it can be decided for an arbitrary one-dimensional CA whether it contains Eden or not.

Algorithm of Amoroso and Patt:

Construct a finite graph (a tree) each node of which will be a subset of all n-tuples in A^n with equal images of f. Select an element $b \in A$, let the (unique) node N at level 0 be the set of all n-tuples $(a_1, \ldots, a_n)$ such that $f(a_1, \ldots, a_n)=b$. For each node N at level i (>0) construct for each $a \in A$ a node N_a at level $i+1$ as it follows. If $(a_1, \ldots, a_n)$ is an element in the set corresponding to N, the set corresponding to N_a will consist of exactly those n-tuples $(a_2, \ldots, a_n, d)$, $(d \in A)$ for which $f(a_2, \ldots, a_n, d)=a$. A directed arc labeled by a is then drawn from node N to node N_a.

If for each $(a_1, \ldots, a_n)$ at N there are no such elements d, then this node N_a is not included in the graph and we say that N is terminal node in the graph.

If during the construction process a number of nodes appear at the same or different levels, all nodes associated with the same subset of A^n then each one will be a distinct node in the graph, but only one, arbitrarily chosen, will be extended. This process must terminate since there is a bound on the number of possible subsets of A^n.

Theorem 1. In a CA (A, Z, x, f) if there is no Eden of finite support then there is no terminal node in the graph constructed above.

Proof. If we get the terminal node across the arcs $bb_1 \ldots b_k$, it is easy to show that the configuration having this support has no preimage under F.

Theorem 2. In a CA (A, Z, x, f) if the defining table of the local transformation function is not balanced then there exists an Eden of finite support in the CA.

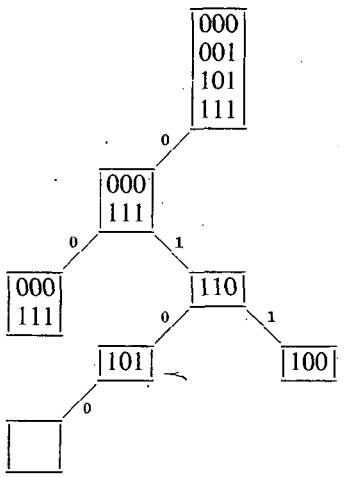
Proof. Will be given by means of the algorithm of Amoroso and Patt. Let n be the neighborhood size of the CA. Denote by $|A|$ the number of elements in the set A. We shall see if there is a node in the graph which contains k n-tuples, then continuing the process we get a node containing not more than $k-1$ n-tuples.

Suppose the contrary. Construct n levels of the graph, in this case the equal nodes too. On level n there are $|A|^n$ nodes. The number of n-tuples in each node is k because of our assumption. It should be clear from the construction process that for each $a_i \in A$ there are $|A|^{n-1}$ nodes on level n, which contains n-tuples having image a_i under f. (If it were not so, we should have a terminal node.) We show that all of such n-tuples are contained exactly k-times on level n. On the

first level there are k n-tuples containing a_i as a last element, (for each $a_i \in A$) because we join all $a_i \in A$ to each n-tuple of level 0. These elements in the tailes of the n-tuples are shifting forward on each level, and to each tail every $a_i \in A$ joins. So we get all of the n-tuples of A k-times on level n. For each $a_i \neq a_j$ if the sets of n-tuples are disjoint and the multiplicity is always k then for each $a_i \in A$ there are exactly $|A|^{n-1}$ n-tuples having image a_i. So the table of the local transformation function is balanced which is contradiction.

So the statement is valid for each k, and for $k=1$ we shall get to a terminal node. Then from Theorem 1 it follows the existence of an Eden configuration.

REMARK. The converse of the theorem above is not true, there exists a transformation function which has balanced defining table, but the CA contains Eden configuration. For example: $A=2$, $n=3$, denote the two states of the cells 0 and 1. Let us construct the graph beginning with the n-tuples having image 0. (This defines the function f as well.)



We have got to a terminal node, so the CA has Eden configuration.

We have the following sufficient condition:

Theorem 3. If the defining table of the local transformation function is balanced and it holds for each $a_i \in A$ that taking away the first elements from the n-tuples having image a_i under f, there are no equal ones among the left $n-1$-tuples: or this is valid for the last elements, then the CA has no Eden configuration.

Proof. Construct the graph beginning with the n-tuples having image an arbitrary $a_i \in A$.

1. If the last $n-1$-tuples are different, on the following level all of the n-tuples appear. This is valid for all nodes so further we shall have the same nodes.

2. If the first $n-1$-tuples are different then from each n-tuple we get $|A|$ n-tuples which are in different nodes. Therefore, we shall never get to a terminal node.

Abstract

This article gives a necessary and a sufficient condition for the existence of an Eden configuration in one-dimensional cellular automata.

References

[1] AMOROSO, S. and J. N. PATT, Decision procedure for surjectivity and injectivity of parallel maps for tessellation structures, *J. Comput. System Sci.*, v. 6, 1972, pp. 448—464.
[2] AMOROSO, S. and G. COOPER, Tessellation structures for reproduction of arbitrary patterns, *J. Comput. System Sci.*, v. 5, 1971, pp. 455—464.
[3] AMOROSO, S. and G. COOPER, The Garden-of-Eden theorem for finite configurations, *Proc. Amer. Math. Soc.*, v. 44, 1970, pp. 189—197.

(Received April 5, 1979)

Linear parallel maps of tessellation automata

By E. KATONA

The s-state cellular automaton, where the local map computes the modulo-s sum of all neighbour-states, has very useful algebraic properties (see [1]). The principle of this cellular automaton will be generalized in this paper, as far as it is possible, applying inhomogeneous tessellation automata (defined in [2], [6] and below), so we get the concept of *linear parallel maps.* A wide class of parallel maps is obtained in such a way, keeping the good algebraic properties of the structure in [1].

The present work discusses the fundamental characteristics of linear parallel maps, giving some examples and applications to demonstrate the theoretical results. Finally, we mention an open problem showing that many further investigations are possible in the area.

1. Definitions

In [6] a general conception is developed for practical construction and application of cellular automata. On the basis of it we have given a strong generalization for the concept of cellular automaton in [2], those definitions are repeated below.

DEFINITION. An *inhomogeneous cellular automaton* (in short CA) is a four-tuple (C, A, N, Φ), where

$C=\{c_1, \ldots, c_m\}$ is the finite *set of cells,*

$A=\{0, 1, \ldots, s-1\}$ is the finite *set of cell-states,*

$N: c_i \mapsto (c_{i_1}, \ldots, c_{i_{n_i}})$ is the *neighbourhood function,*

which assigns to each cell its neighbours. The specification of neighbours may be different cell by cell, i.e. the cellular automaton has a totally arbitrary topology.

$\Phi: c_i \mapsto f_i$ is the *function-system,* which assigns to each cell an $f_i: A^{n_i} \to A$ local transition function (*local map* in short). The local maps also may be different cell by cell.

Note that a nearly homogeneous topology with only a few different local maps is sufficient in practice, yet the theoretical studies need no such a restriction.

The CA works as usual: at time $t=0$ each cell has an initial state. From t to $t+1$ each cell changes its state synchromously so that the new state of a cell depends only on its neighbours, according to the local map.

DEFINITION. An *inhomogeneous tessellation automaton* (in short TA) is a triple (C, A, N), where the components correspond to the above-mentioned ones. In this case the function-system is time-varying: at time t the TA executes a function-system Φ_t.

FURTHER USUAL DEFINITIONS. A *configuration* is a possible global state of the TA, formally a mapping $\alpha: C \to A$. It will be denoted always by Greek letters, and the notation $\alpha=(a_1, \ldots, a_m)$, where $a_i=\alpha(c_i)$, will be used too. We denote by $\mathscr{A}$ the set of all configurations.

The *parallel map* induced by a given function-system is a mapping $F: \mathscr{A} \to \mathscr{A}$, where $F(\alpha)=\beta$ if for all i $f_i(a_{i_1}, \ldots, a_{i_{n_i}})=\beta(c_i)$ (the n_i-tuple $(a_{i_1}, \ldots, a_{i_{n_i}})$ denotes the neighbourhood of c_i in α). A mapping $F: \mathscr{A} \to \mathscr{A}$ will be called a parallel map, if and only if there exists a function-system inducing F.

2. Linear parallel maps

In [1] an s-state homogeneous CA was investigated, with arbitrary dimension and neighbourhood-index, where the local map of any cell computes the sum (in the sense modulo s) of all neighbour-states. The parallel map of this CA is *linear* in the following sense: for any configurations α, β: $F(\alpha+\beta)=F(\alpha)+F(\beta)$, where $\alpha+\beta$ denotes the configuration satisfying $(\alpha+\beta)(c_i)=\alpha(c_i)+\beta(c_i)$ for any i. This property is indispensable to prove the most important characteristic of this CA: it reproduces an arbitrary pattern in s^q steps if q is great enough (see [1]).

In the following we define the concept of linear parallel maps in general.

Let (C, A, N) be a TA such that the set A forms a *finite commutative ring* with identity element, whose operations are denoted by $+$ and $\cdot$. In the most simple case A is a residue-class-ring, namely $A=\{0, 1, \ldots, s-1\}$ and the two operations are the modulo-s addition and multiplication.

Further a configuration α will be considered as a vector $(a_1, \ldots, a_m)$ over the ring A. If $\alpha=(a_1, \ldots, a_m)$ and $\beta=(b_1, \ldots, b_m)$ then the configurations $\alpha+\beta$ and $k \cdot \alpha$ are defined in the usual way: $\alpha+\beta:=(a_1+b_1, \ldots, a_m+b_m)$ and $k \cdot \alpha:=(ka_1, \ldots, ka_m)$.

It is clear that the following holds:

Theorem 1. The set $\mathscr{A}$ of all configurations forms an m-dimensional vector-space over the ring A.

DEFINITION. A parallel map F is called a *linear parallel map,* if it is a linear transformation of the vector-space $\mathscr{A}$ onto itself: $F(\alpha+\beta)=F(\alpha)+F(\beta)$ and $F(k \cdot \alpha)=k \cdot F(\alpha)$ holds for any $k \in A$ and $\alpha, \beta \in \mathscr{A}$.

It is well-known from algebra (see [3]), that a map $F: \mathscr{A} \to \mathscr{A}$ is linear iff there exists a matrix K of type $m \cdot m$ over A such that for any $\alpha (\in \mathscr{A})$, $F(\alpha)=K \cdot \alpha$ holds (here α is considered a matrix of type $m \cdot 1$). Using this fact the following result can be proved.

Theorem 2. A parallel map F is linear iff each local map has the form $f_i(a_{i_1}, \ldots, a_{i_{n_i}})=k_{ii_1}a_{i_1}+\ldots+k_{ii_{n_i}}a_{i_{n_i}}$, where $k_{ii_1}, \ldots, k_{ii_{n_i}} \in A$.

Proof. If F is a linear parallel map then $\exists K: \forall \alpha: F(\alpha)=K\cdot\alpha$. So we get: $f_{i_1}(a_{i_1}, \ldots, a_{i_{n_i}})=k_{i1}a_1+\ldots+k_{im}a_m$ for any i. However if c_j is not a neighbour of c_i, then k_{ij} must be equal to 0.

Conversely, if each f_i forms a linear combination, then the matrix K is constructable in the following way: if c_j is a neighbour of c_i then let k_{ij} be the coefficient corresponding to c_j in the formula of f_i. If c_j is not a neighbour of c_i then $k_{ij}:=0$. □

3. One-to-one parallel maps

From [3] we recall the following result: A matrix K over the ring A has an inverse K^{-1} iff the determinant of K is not a zero-divisor. (An element $a(\in A)$ is called zero-divisor if there exists a nonzero element $b(\in A)$ such that $a\cdot b=0$.)

Applying this result to parallel maps we get

Theorem 3. A linear parallel map F is one-to-one iff the determinant of its matrix is not a zero-divisor.

Considering a two-state CA we have a more concrete result

Theorem 4. A linear parallel map F, induced by a two-state CA, is one-to-one iff the number of such distinct permutations p: $C\to C$, where for any i, $p(c_i)$ is a "real" neighbour of c_i, is odd. (c_j is called a "real" neighbour of c_i, if the local map f_i depends on the state of c_j.)

Proof. The ring $A=\{0, 1\}$ is zero-divisor-free (it is a field), therefore we must prove: the determinant D of the matrix of F has the value 1 iff the right-hand condition above holds. By the definition, $D=\sum_p(-1)^I\cdot k_{1p_1}\ldots k_{mp_m}$ where p is a permutation $C\to C$ and p_i denotes $p(c_i)$. The factor $(-1)^I$ may be eliminated because in the field $\{0, 1\}$ the elements -1 and 1 are equal. Further we have: $D=1$ iff the number of permutations satisfying $k_{1p_1}\cdot\ldots\cdot k_{mp_m}=1$ is odd, where $k_{ij}=1$ means that c_j is a real neighbour of c_i. □

Example. Let (C, A, N, Φ) be a one-dimensional two-state CA interconnected into a circle (i.e., c_1 and c_m are neighbours), where $N(c_i)=(c_i, c_{i+1})$ and $f_i(a_i, a_{i+1})=a_i+a_{i+1}$ for any i. The parallel map of this CA is *not one-to-one;* because there are two permutations satisfying the condition in Theorem 4: $p_1(c_i)=c_i$ and $p_2(c_i)=c_{i+1}$ for any i. However this parallel map can be made *one-to-one* modifying only one among the local maps on $f_i(a_i, a_{i+1})=a_i$, since in this case we have only the permutation p_1.

In these two examples the bijectivity of F was independent of m. Note that in other cases it depends strongly on the size of the given CA (see [4]).

4. Decision procedure for bijectivity and reversibility

DEFINITION. A parallel map F is called *(locally) reversible*, if it is one-to-one and F^{-1} is also a parallel map. The function-system, which generates F^{-1}, is called the *reverse function-system.*

In this paragraph a simple procedure is presented (realizable in practice by the

simulation of the CA), which decides that whether or not a linear parallel map F induced by a CA (C, A, N, Φ) is one-to-one (i.e. bijectiv). If it is, then we can *decide its reversibility and construct the reverse function-system too.*

Let the CA start from a configuration $\varepsilon_i=(0, \ldots, 0, \overset{i}{1}, 0, \ldots, 0)$ until it reaches a cycle (i.e. until a time t_2, for which $\exists t_1(<t_2)$: $F^{t_1}(\varepsilon_i)=F^{t_2}(\varepsilon_i)$). If this procedure was executed for any ε_i $(i=1, \ldots, m)$, then we have one of the following two cases:

(i) There exists an ε_i such that in its cycle (from $F^{t_1}(\varepsilon_i)$ to $F^{t_2}(\varepsilon_i)$) the configuration ε_i is not occuring. This implies that there exist α, β such that $\alpha \neq \beta$ and $F(\alpha)=F(\beta)$, i.e., F is *not one-to-one.*

(ii) Each ε_i generates a cycle returning into itself. Consequently each ε_i has a predecessor $F^{-1}(\varepsilon_i)$. Since $\varepsilon_1, \ldots, \varepsilon_m$ is a basis of the vector-space $\mathscr{A}$, therefore F is one-to-one, F^{-1} is linear too and its matrix consists of the columns $F^{-1}(\varepsilon_1)$, ..., $F^{-1}(\varepsilon_m)$. Moreover, F^{-1} is a *parallel map* iff its matrix has zero elements for each (i, j) where c_j is not a neighbour of c_i.

Summarizing our-results, we get

Theorem 5. A linear parallel map F is *one-to-one* iff for any i the configuration $\varepsilon_i=(0, \ldots, 0, \overset{i}{1}, 0, \ldots, 0)$ has a predecessor $F^{-1}(\varepsilon_i)$. Furthermore F is *reversible* iff every $F^{-1}(\varepsilon_i)$ has only the neighbouring cells of c_i in nonzero state. □

Note that a *homogeneous CA* shows the same behaviour for any ε_i, consequently it is sufficient to investigate it starting from ε_1. Our procedure is powerful practically only in this case.

Example. A homogeneous CA (C, A, N, Φ) is considered (a so-called Lindenmayer-CA, investigated in [4]) where

$C = \{c_{1,1}, \ldots, c_{8,8}\}$ is a two dimensional matrix,

$A = \{0, 1\}$,

$N: c_{ij} \mapsto (c_{ij}, c_{i-1,j}, c_{i+1,j}, c_{i,j-1}, c_{i,j+1})$

for any i, j. The indexes are interpreted cyclically (for example $N(c_{1,1})= =(c_{1,1}, c_{8,1}, c_{2,1}, c_{1,8}, c_{1,2})$), hereby the edges are interconnected into a torus (homogeneous topology).

$\Phi: c_{ij} \mapsto f$ for any i, j, where f computes the modulo-2 sum of all neighbour-states. So Φ induces a linear parallel map.

Starting this CA from the configuration $\varepsilon_{4,4}$ a four-step cycle is obtained (Figure 1). Since the CA is homogeneous, so each ε_{ij} shows an analogous behaviour.* This implies that *our CA generates a one-to-one parallel map* F. But, at the same time, in the configuration $F^{-1}(\varepsilon_{4,4})$ (see step 3) there are 16 cells, not neighbouring with $c_{4,4}$, in nonzero state. Consequently *the CA is not locally reversible.*

* If the torus-connection is rejected and dummy cells are used on the edges of CA (inhomogeneous topology!), then we need a test for the configurations $\varepsilon_{1,1}$, $\varepsilon_{2,1}$, $\varepsilon_{2,2}$, $\varepsilon_{3,1}$, $\varepsilon_{3,2}$, $\varepsilon_{3,3}$, $\varepsilon_{4,1}$, $\varepsilon_{4,2}$, $\varepsilon_{4,3}$, too.

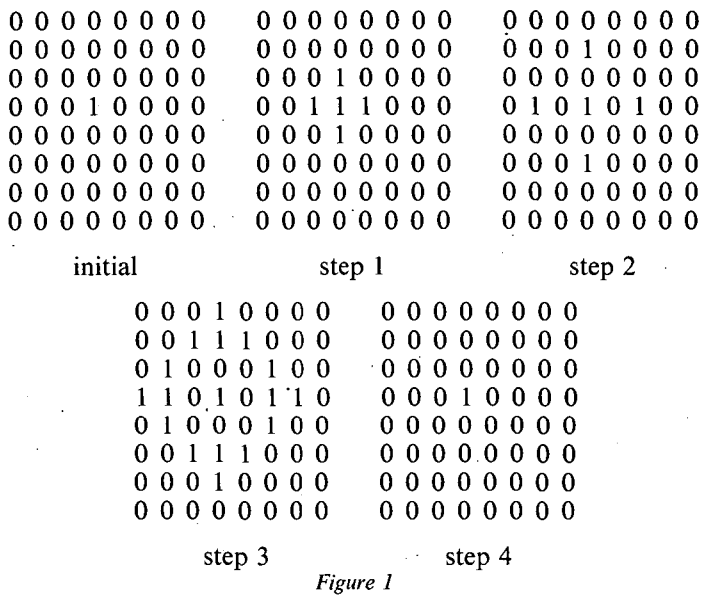


Figure 1

Note that we can construct a *reverse* CA to the above-mentioned one, as follows: let N be a homogeneous torus-topology on the set $C=\{c_{1,1}, \ldots, c_{8,8}\}$ such that e.g. $N(c_{4,4})$ contains the cells having the state 1 in $F^{-1}(\varepsilon_{4,4})$ (see step 3 in Figure 1), and let Φ be such that each local map computes the modulo-2 sum of all neighbour-states. This CA generates the parallel map F^{-1}.

5. The problem of the synthesis

In this point we mention an important open problem in the theory of TA (it was investigated in [5], in case of one-dimension) which, if only linear parallel maps are considered, simplifies into a matrix-theoretical problem.

It is obvious that for some linear parallel maps F_1 and F_2 with matrixes K_1 and K_2, respectively, the mapping $F_2 \circ F_1$ is linear too, and has the matrix $K_2 \cdot K_1$. We have the following problem:

If (C, A, N) is a TA and $F: \mathscr{A} \to \mathscr{A}$ is an arbitrary linear mapping, then whether or not there exist linear parallel maps $F_1, \ldots, F_n$ such that $F_n \circ \ldots \circ F_1 = F$. If we have no such a row of parallel maps, then what extension of the neighbourhood function N is needed to that?

Because in the matrix of any parallel map each element $k_{ij}=0$, for which c_j is not a neighbour of c_i, so the problem described above alters into an algebraic question: which index-sets $I=\{(i_1, j_1), \ldots, (i_r, j_r)\}$ do satisfy, that an arbitrary matrix M over A is decomposable into a product $M_1 \cdot \ldots \cdot M_n$, where each component M_i contains nonzero elements only with indexes in I?

Note that in [5] it was proved that for a usual (i.e. infinite, homogeneous) one-dimensional TA such finite index set (i.e. a finite neighbourhood) does not exist.

6. Concluding remarks

In this work a special class of cellular automata was studied having linear parallel maps and — consequently — linear algebraic properties. In Theorem 2 were characterized the local maps inducing linear parallel maps; further it was shown that many interesting problems in the theory of cellular automata (e.g. the Garden-of-Eden problem, reversibility, synthesis-problem) can be investigated easily in this linear case, applying the results of linear algebra.

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

References

[1] HAMILTON, W. L. & J. R. MERTENS, Reproduction in tessellation structures, *J. Comput. System Sci.*, v. 10, 1975, pp. 248—252.
[2] KATONA, E., Local and global reversibility of finite inhomogeneous cellular automaton, *Acta Cybernet.*, v. 3, 1978, pp. 287—292.
[3] RÉDEI, L., *Algebra I.*, chapter III., Akadémiai Kiadó, Budapest, 1967.
[4] MARTONI, V., The cellular automaton of Lindenmayer, *Sejtautomaták*, Gondolat Kiadó, Budapest, 1978, (in Hungarian).
[5] AMOROSO, S. & I. J. EPSTEIN, Indecomposable parallel maps in tessellation structures, *J. Comput. System Sci.*, v. 13, 1976, pp. 136—142.
[6] LEGENDI, T., Cellprocessors in computer architecture, *Computational Linguistics and Computer Languages*, v. 11, 1977, pp. 147—167.

(Received April 23, 1979)

Schützenberger's monoids*

By K. H. KIM and F. W. ROUSH

In [1], Schützenberger proposed the following problem. "Give an algorithm to construct inductively all finite monoids M which contain a submonoid P satisfying

$$(U_s)\ m, m' \in M \,\&\, mm', m'm \in P \Leftrightarrow m, m' \in P$$

$$(N_d)\ m \in M \Rightarrow P \cap MmM \neq \emptyset$$

and (to limit the problem to its essential) which are such that P is not a union of classes of a nontrivial congruence on M."

Definition 1. *A (U_s, N_d)-submonoid* of a monoid M is a submonoid P satisfying the two conditions (U_s) and (N_d). Such a submonoid is *simple* if P is not a union of classes of a nontrivial congruence on M.

Theorem 1. *Let P be a simple (U_s, N_d)-submonoid of a finite monoid M. Then P contains all invertible elements of M. If $x \in P$ and the $\mathscr{H}$-class of x is a group then P contains the entire $\mathscr{H}$-class of x. Some element of the lowest $\mathscr{D}$-class of M belongs to P. All $\mathscr{H}$-classes of the lowest $\mathscr{D}$-class D_0 of M contain only one element. And P contains the centralizer of any element of $D_0 \cap P$.*

Proof. Let P be a simple (U_s, N_d)-submonoid of M, and let D_0 be the lowest $\mathscr{D}$-class of M. Condition (N_d) is equivalent to stating that P contains some element z of D_0. Suppose x belongs to P and the $\mathscr{H}$-class of x is a group. Let e be the identity element of this $\mathscr{H}$-class and y any other element of the $\mathscr{H}$-class. Then $ex = xe = x$ implies e belongs to P. And $yy^{-1} = y^{-1}y = e$ implies y belongs to P. Therefore P contains the entire $\mathscr{H}$-class of x, and also the $\mathscr{H}$-classes of all elements of P in D_0. Also P contains the $\mathscr{H}$-class of the identity element of M. Therefore it contains all invertible elements of M.

Let α be the equivalence relation $x \mathscr{L} y$ if and only if $x = y$ or $x, y \in D_0$ and $x \mathscr{H} y$. We claim α is a congruence. Let $x, y \in D_0$ and $x \mathscr{H} y$. Let e be the idempotent of this $\mathscr{H}$-class. Let $a \in M$. Then $ax = (ae)x$ and $ay = (ae)y$. The $\mathscr{D}$-class D_0 is a finite simple semigroup, and $ae \in D_0$ and $x \mathscr{H} y$ in D_0. By the structure of

* This work was supported by Alabama State University Faculty Research Grant R-78-6.

finite simple semigroups (Suschkevitch's theorem) this implies $(ae)x \mathscr{H} (ae)y$. Likewise $xa\mathscr{H}ya$. Therefore α is a congruence. If the $\mathscr{H}$-classes of D_0 contain more than one element the congruence α is nontrivial. Since P is a union of classes of α, this would mean P is not simple. Therefore the $\mathscr{H}$-classes of P contain only a single element.

Let c belong to the centralizer of $z \in D_0 \cap P$. Then $cz=zc=zzc=zcz=z(zcz)z$ since z is idempotent. But $z(zcz)z$ lies in the $\mathscr{H}$-class of z. Since this $\mathscr{H}$-class contains only one element $cz=zc=z$. Therefore $c \in P$. This proves the theorem.

NOTATION. (1) Let $|X|$ denote the cardinality of a set X.
(2) Let B_n denote the semigroup of binary relations on an n-element set.
(3) Let T_n denote the semigroup of transformations on an n-element set.

Corollary. *Let M, P be as in the preceding theorem and let $|M|>1$, and let 1 be the monoid identity. Then M cannot be abelian, contain a zero, be an inverse semigroup, B_n, T_n, or $GLS(n, F)$.*

Proof. The preceding theorem implies that D_0 must contain more than one $\mathscr{H}$-class, else D_0 would be a single zero element and $P=M$. For $|M|>1$, P would not be simple. In particular M cannot contain a zero. This rules out all the above types of semigroups except T_n.

Suppose $M=T_n$, $n>1$. Then the symmetric group belongs to P. Therefore all rank 1 transformations belong to P. This implies all transformations belong to P, by condition (U_s). Therefore for $n>1$, M is not simple.

Proposition 2. *Let P be a (U_s, N_d)-submonoid of the finite monoid M. Let α be the relation $x\alpha y$ if and only if for all $u, v \in M$ ($uxv \in P$ if and only if $uyv \in P$). Then α is a congruence on M and P is a union of classes of α. Let M_0, P_0 be the quotients of M, P by α. Then P_0 is a simple (U_s, N_d)-submonoid of M_0.*

Proof. It is immediate that α is an equivalence relation, and a computation shows that α is a congruence. Suppose $x\alpha y$ and $y \in P$. Take $u=v=1$, the identity of the monoid. Then $x \in P$. Therefore P is a union of classes of α. Let M_0, P_0 be the quotients of M, P by α. Suppose P_0 is a union of classes of some congruence β. Let M_1, P_1 be the quotients of M_0, P_0 by β. Let $h_1: M \to M_0$ and $h_2: M_0 \to M_1$ be the quotient homomorphisms. Let γ be the congruence on M such that $x\gamma y$ if and only if $(x)h_1h_2=(y)h_1h_2$. If β is a nontrivial congruence, there exist x, y such that $x\gamma y$ but not $x\alpha y$. By symmetry we may assume that for some $u, v \in M$, $uxv \in P$ and $uyv \notin P$. Therefore $(uxv)h_1h_2 \in P_1$ but $(uyv)h_1h_2 \notin P_1$. But $(x)h_1h_2 = (y)h_1h_2$. Therefore $(uxv)h_1h_2=(uyv)h_1h_2$. This is a contradiction. This proves the proposition.

Definition 2. Let G be a free monoid on generators $x_1, x_2, \ldots, x_k$. If W is a word of G a *segment* of G is a word formed by the i-th through j-th letters of W in order, for some $i<j$. If $i=1$, the segment is called *initial*. If $j=n$ the segment is called *terminal*. Let G_n be the homomorphic image of G in which $W_1=W_2$ if and only if W_1 and W_2 have the same length n initial segment or $W_1=W_2$.

Theorem 3. *Let W_0 be a word of length n in G such that no initial segment of W_0 equals a terminal segment of W_0, other than the segment W_0 itself. Let*

$P=\{1, W_0\}$. Then P is a (U_s, N_d)-submonoid of G_n. Let α be the relation on G_n such that $x\alpha y$ if and only if for all $u, v\in G_n$: $uxv\in P$ if and only if $uyv\in P$. Then P/α is a simple (U_s, N_d)-submonoid of G_n/α. Suppose the last letter of W_0 is not x_1. Let S be the set $\{1, x_1^n,$ all segments of W_0, Wx_1^{n-r} such that W is a terminal segment of length r of W_0 which also equals a nonterminal segment of $W_0\}$. Then S contains exactly one element from each class of α. Products in G_n/α can be described as follows. Take the product Y in G_n and reduce as follows. If $Y=$some element of S, the product is Y. Suppose Y does not equal an element of S. Suppose an initial segment t of Y equals a terminal segment of W_0 of length r, where r is a maximum. Then if t equals a nonterminal segment of W_0 the product in G_n/α is tx_1^{n-r}. If t does not equal a nonterminal segment of W_0 the product is t. If no initial segment of Y equals a terminal segment of W_0, then the product is x_1^n.

Proof. The set $P=\{1, W_0\}$ is a submonoid of G_n since $W_0^2=W_0$. Suppose for some $W_1, W_2\in G_n$, $W_1W_2=W_2W_1=1$. Then $W_1=W_2=1$. Suppose $W_1W_2=$ $=W_2W_1=W_0$. Then if $W_1, W_2\notin P$, some initial segment of W_0 equals a final segment. This is contrary to assumption. Therefore P satisfies condition (U_s). The lowest $\mathcal{D}$-class of G_n consists of all length n words. Therefore W_0 belongs to this lowest $\mathcal{D}$-class. Therefore P satisfies condition (N_d). It follows from Proposition 2 that G_n/α is simple. It remains to describe the relation α. Suppose x has the property that x is not a segment of W_0 and x is not 1 and no initial segment of x equals a final segment of W_0. It follows that uxv equals W_0 if and only if u equals W_0. Since x_1^n also has this property, $x\alpha x_1^n$. Suppose $x\notin S$ and an initial segment t of x equals a terminal segment of W_0 of length r, where r is maximal. Suppose t equals a nonterminal segment of W_0. If $uxv=W_0$ then $ux=W_0$ since x is not a segment of W_0. Therefore $ut=W_0$. Therefore $utx_1^{n-r}=$ $=W_0$. Suppose $utx_1^{n-r}v=W_0$. Then $utx_1^{n-r}=W_0$ since the length of tx_1^{n-r} is n. Therefore $ut=W_0$ since the last letter of W_0 is not x_1. Therefore $uxv=W_0$. This proves $x\alpha tx_1^{n-r}$. Suppose t does not equal a nonterminal segment of W_0. Then we have $x\alpha t$ by a similar argument. This proves that S contains at least one element from every class of α. Suppose $y\alpha z$ where $z=1$. Then $y\in P$. Therefore $y=1$ or W. But $zx_1\notin P$ implies $yx_1\notin P$ which implies $y\neq W$. So $y=1$. Suppose $y\alpha z$ where z is a nonterminal segment of W_0. Let $W_0=z_1zz_2$ in G. Then $z_1yz_2=$ $=W_0$. Suppose y had length greater than z. Then $z_1yz_3=W_0$ in G_n where z_3 is obtained from z_2 by omitting the last letter of z_2. But $z_1zz_3\neq W_0$. This contradicts $y\alpha z$. Therefore the length of y is not more than the length of z. So $z_1yz_2=$ $=z_1zz_2$ in G. So $y=z$. If z is a terminal segment of W_0 which does not equal a nonterminal segment of W_0, we have shown above that $z\alpha zx_1^{n-r}$ where r is the length of z. Suppose $y\alpha zx_1^{n-r}$ where z is any terminal segment of W of length r. Then y does not equal a nonterminal segment of W_0. Let $W_0=z_1z$ in G. Then $z_1y=W_0$ in G_n. Therefore $y=zz_2$ for some z_2. Suppose an initial segment of y of length greater than r equals a terminal segment of W_0. Then $y\alpha z$ will be false. This proves no two elements of S belong to the same class of α. Moreover it completely describes the relation α. The description of multiplication in G_n/α follows. This proves the theorem.

CONCLUDING REMARK. This construction can be generalized in a number of ways. For certain words W_0, P will have more than two elements. More than one

word W_0 of length n can be chosen. A similar construction can be made where G_n is replaced by the free monoid band on $x_1, x_2, \ldots, x_k$ and W_0 is replaced by the word $x_1 x_2 \ldots x_k$. This will give (U_s, N_d)-simple submonoids of semigroups which are bands.

Abstract

We study pairs, P, M, $P \subset M$ of monoids such that P contains an element of the lowest $\mathcal{D}$-class of M and $mm', m'm \in P$ if and only if $m, m' \in P$ for all $m, m' \in M$. Such pairs are called simple if P is not a union of classes of a nontrivial congruence on M. We show that simple finite pairs P, M have certain characteristics which rule out most familiar semigroups. However we do construct an infinite family of simple, finite P, M pairs.

MATHEMATICS RESEARCH GROUP
ALABAMA STATE UNIVERSITY
MONTGOMERY, ALABAMA 36101
U. S. A.

Reference

[1] SCHÜTZENBERGER, M. P., Problem 15, *Algebraic Theory of Semigroups,* Coll. Math. Soc. János Bolyai 20, North-Holland Publishing Company, 1978.

(Received June 5, 1978)

The cardinality of closed sets in pre-complete classes in k-valued logics

By J. DEMETROVICS and L. HANNÁK

Introduction

Let $E_k=\{0, 1, \dots k-1\}$. By a k-valued function we shall mean a function $f: E_k^n \to E_k$, and by P_k we denote the set of all those functions. If A is a subset of P_k, $[A]$ will stand for the set of all superpositions over A. (The definition of a superposition over A is the following:

1. $f \in A$ is a superposition over A.

2. If $g_0(x_1, \dots, x_n)$, $g_1(x_{11}, \dots, x_{1m_1})$, $\dots$ $g_n(x_{n1}, \dots, x_{nm_n})$ are either superpositions over A or $g_i(x_{i1}, \dots, x_{im_i})=x_j$ $(i=1, \dots, n)$ then $g_0(g_1(x_{11} \dots x_{1m_1}), \dots, g_n(x_{n1}, \dots x_{nm_n}))$ is a superposition over A.)

The set $A \subset P_k$ is closed if $A=[A]$. We call A complete if $[A]=P_k$. The closed set $\mathcal{M}$ is precomplete if $\mathcal{M} \subsetneqq A \subsetneqq P_k$ implies $[A]=P_k$. I. ROSENBERG [8] has given a complete description of the precomplete classes in P_k. In order to formulate his theorem we need some definitions. An h-ary relation R is a subset of E_k^h. If g is an n-ary k-valued function and R is an h-ary relation we say that f preserves R if $(f(x_1^1, \dots, x_1^n), \dots, f(x_h^1, \dots, x_h^n)) \in R$ whenever $(x_1^1, \dots, x_h^1) \in R, \dots$ $\dots, (x_1^n, \dots, x_h^n) \in R$ an h-ary relation R is called central if it fulfils the following conditions:

1. $(a_1, \dots, a_h) \in R$ whenever not all of $a_1, \dots, a_h$ are distinct,

2. for each permutation π of $1, 2, \dots, h$, $(a_1, \dots, a_h) \in R$ if and only if $(a_{\pi(1)}, \dots, a_{\pi(h)}) \in R$,

3. $\emptyset \neq \bigcap_{(a_1, \dots, a_{h-1}) \in E_k^{h-1}} \{c | (a_1, \dots, a_{h-1}, c) \in R\} \neq E_k$.

For $a \in E_k$ we denote by $[a]_l$ the l-th digit $(l=0, \dots, m-1)$ in the expansion $a=\sum_{l=0}^{m-1} [a]_l \cdot h^l$ of a in the scale of h.

We may now state the theorem of Rosenberg as follows:

There are 6 types of precomplete classes in P_k and every proper closed subset of P_k is contained in at least one precomplete class.

This 6 types are the following:

1. $\mathcal{M}_\mu$ the set of all functions which preserve a partial order μ of E_k with greatest and least element.

2. S_π, the set of all functions which preserve the graph of a nonidentical permutation π where π is the product of cycles with the same prime length.

3. L_σ the set of all functions which preserve the quaternary relation

$$\sigma = \{(a_1, a_2, a_3, a_4)/a_1+a_2 = a_3+a_4\}$$

where $\langle E_k, +\rangle$ is an elementary Abelian p-group.

4. K_θ, the set of all functions which preserve the non trivial equivalence-relation θ of E_k^2.

5. C_ϱ, the set of all functions which preserve the h-ary central relation ϱ $(1 \leqq h \leqq k)$.

6. H_R, the set of all functions which preserve the relation R, where R is for some h $(3 \leqq h \leqq k)$ and for some surjection $\Phi: E_k \to E_{h^m}$ the h-ary relation

$$|\{[\Phi(x_1)]_l, \ldots, [\Phi(x_n)]_l\}| < h \quad \text{for} \quad l = 0, \ldots, m-1.$$

(Such a relation R is called h-regular.)

If A is a closed subset of P_k, $v(A)$ will denote the cardinality of the set of all closed sets contained in A. Let us denote by $\mathfrak{c}$ the cardinality of the continuum.

Ju. I. Yanov and A. A. Mučnik [5] have proved that $v(P_k)=\mathfrak{c}$ for $k>2$. The general result of E. Post [10] implies that $v(P_2)=\aleph_0$.

It is a natural question to determine $v(A)$ when A is a precomplete class. In this paper we shall prove the following three statements:

I. if $k>2$ and M is a precomplete class of type 1., 4., 5., or 6. then $v(M)=\mathfrak{c}$,

II. if $k>2$ then $v(S_\pi) \geqq \aleph_0$ for all precomplete classes of type 2.,

III. $v(S_\pi)=\mathfrak{c}$ if k is not prime.

The precomplete class L_σ was investigated by many authors. A. Salomaa [8] J. Demetrovics and J. Bagyinszki ([2] and [3]) proved $v(L_\sigma)<\aleph_0$ in the case if k is prime. J. Bagyinszki [1] and A. Szendrei [9] showed that if k is square-free then there are finitely many closed linear classes in P_k. A. Salooma [8] proved, that $v(L_\sigma) \geqq \aleph_0$ if k is not square-free and D. Lau [7] showed that $v(L_\sigma)=\aleph_0$ in this case.

1. §.

The proof of the first statement is based on the construction of Ju. I. Janov and A. A. Mučnik [5]. They have proved, that the set of functions $\{g_i\}$ defined by

$$g_i(x_1, \ldots, x_i) = \begin{cases} b & \text{if } |\{j|x_j = c\}| = i \text{ or} \\ & \quad |\{j|x_j = b\}| = 1 \text{ and} \\ & \quad |\{j|x_j = c\}| = i-1 \\ a & \text{in all other cases} \end{cases}$$

has the property

$$g_i \notin [\bigcup_{j \neq i} g_j]$$

(a, b and c are pairwise distinct fixed elements of E_k, $k>2$).

Let μ be a fixed partial order of E_k, let a be its least element, c its greatest one and $a<b<c$ such that $\{x|b<x<c\}=\emptyset$. In this case every g_i preserves μ, that is $v(\mathcal{M}_\mu)=\mathfrak{c}$. If θ is a non-trivial equivalence, then we can choose $a\neq b$ such that $a\equiv b(\theta)$. Let c be an arbitrary element of E_k $(c\neq a,\ c\neq b)$. Since $g_i(x_1, \ldots, x_n)\in\{a, b\}$ all g_i preserve θ and $v(K_\theta)=\mathfrak{c}$. If ϱ is a central relation of E_k then g_i preserves ϱ whenever a is an element of the centre of ϱ. Hence $v(C_\varrho)=\mathfrak{c}$.

If R is an h-regular relation, then we can chose arbitrary distinct elements a, b, c. Every g_i preserves every h-regular relation of E_k.

Thus we have proved

Theorem 1. If $k>2$ then

$$v(M_\mu) = \mathfrak{c}$$
$$v(K_\theta) = \mathfrak{c}$$
$$v(C_\varrho) = \mathfrak{c}$$
$$v(H_R) = \mathfrak{c}$$

for all μ, θ, ϱ, R defined in I. ROSENBERG's theorem.

A permutation of E_k, π can be written as a product of disjoint cycles. Such a cycle will be denoted by c_i. If

$$\pi = c_1 \ldots c_n \quad \text{and} \quad c_i = (a_{i1}, \ldots, a_{im_i})$$

then $\{c_i\}$ will denote the set $\{a_{i1}, \ldots, a_{im_i}\}$.

Lemma 1. Let $k\geqq 3$, π be a permutation in the form $\pi=c_1 \ldots c_m$. If $m>1$ and there are $i, j\leqq m$ such that $i\neq j$,

$$|\{c_i\}| = k_1, \quad |\{c_j\}| = k_2 \quad \text{and} \quad k_1|k_2 \quad (k_1 \text{ devides } k_2)$$

then a set of closed classes of cardinality $\mathfrak{c}$ preserving π can be constructed.

Proof. We can assume, that

$$c_1 = (0, \ldots, a_n), \quad c_2 = (1, 2, \ldots, a_m) \quad \text{and} \quad |\{c_1\}|\big||\{c_2\}|.$$

May be that $\{c_1\}=\{0\}$ or $\{c_2\}=\{1, 2\}$.

Let $m\geqq 3$ and

$$g_m(a_1, \ldots, a_m) = \begin{cases} b\in c_2, & \text{if } \{a_1, \ldots, a_m\}\subset\{c_2\} \text{ and } \big|\{j|a_j=b\}\big|=1 \text{ and} \\ & \text{all } a_j\neq b \text{ is equal to } \pi^{-1}(b), \\ d\in c_1, & \text{if } \{a_1, \ldots, a_m\}\subset\{c_1\}\cup\{c_2\} \text{ and the previous} \\ & \text{condition does not hold,} \\ a_1 & \text{in all other cases.} \end{cases}$$

One can easily see, that since $|\{c_1\}|\big||\{c_2\}|$, $g_m(x_1, \ldots, x_m)$ preserves π.

We shall prove that $g_m\notin\left[\bigcup_{m\neq j} g_j\right]=G_m$ for all $m\geqq 3$. Let us suppose that $g_k\in G_k$ i.e.

$$g_k(x_1, \ldots, x_k) = \mathfrak{A}(x_1, \ldots, x_k)$$

where $\mathfrak{A}$ is a superposition over G_k. Let $g_s(x_{i_1}, \ldots, x_{i_s})$ be a function in $\mathfrak{A}$. If $s<k$ then we can find an x_l such that $x_l \notin \{x_{i_1}, \ldots, x_{i_s}\}$. If we choose $x_l=1$ and $x_1=x_2, \ldots, x_{l-1}=x_{l+1}, \ldots, x_k=2$ then, by the definition, $g_k(x_1, \ldots, x_k)=1$, and $g_s(x_{i_1}, \ldots, x_{i_s}) \in c_1$ that is $\mathfrak{A} \neq 1$ holds. (All g_m preserve the set $\{c_1\} \cup \{c_2\}$ and $\{a_1, \ldots, a_m\} \cap c_1 \neq \emptyset$, $\{a_1, \ldots, a_m\} \subset \{c_1\} \cup \{c_2\}$ imply $g_m(a_1, \ldots, a_m) \in c_1$. If $s>k$ then we have at least one pair

$$x_{i_k}, x_{i_l} \quad \text{such that} \quad i_k = i_l.$$

Let $x_{i_k}=x_{i_l}=1$ and all $x_j=2$ with $j \neq i_k$. In this case we have also $g_s(x_{i_1}, \ldots, x_{i_s}) \in c_1$ and $g_k(x_1, \ldots, x_k)=1$, which is a contradiction. Thus Lemma 1 is proved.

As a corollary of Lemma 1 we obtain

Theorem 2. If $k>2$ and k is not prime then $v(S_\pi)=\mathfrak{c}$ for all precomplete classes S_π.

Lemma 2. Let $k>2$, and π be a permutation which contains at least one cycle of length $q \geqq 3$. Then a set of closed classes of cardinality $\mathfrak{c}$ preserving π can be constructed.

Proof. We will give a set of functions $\{t_i\}$ such that $t_k \notin \left[\bigcup_{i>k} t_i\right] = T_k$ and t_i preserves π.

Let

$$t_m(a_1, \ldots, a_m) = \begin{cases} b & \text{if } (a_1, \ldots, a_m) = (b, b, \ldots, b) \text{ or} \\ & (a_1, \ldots, a_{j-1}, a_{j+1}, \ldots, a_m) = (b, b, \ldots, b) \\ & \text{and } a_j = \pi^{-1}(b) \\ \pi^{-1}(b) & \text{if } \{a_1, \ldots, a_m\} \in \{\pi^{-1}(b), b\}^m \\ & \text{and } \{j \mid a_j = b\}| \subset m-1 \\ a_1 & \text{in all other cases.} \end{cases}$$

(b is an element of a cycle which has the length $q \geqq 3$).

The definition implies that t_m preserves π, and $t_m(\{\pi^{-1}(b), b\}^m) \in \{\pi^{-1}(b), b\}$. A vector $a=(a_1, \ldots, a_m)$ is called characteristic if

$$|\{j \mid a_j = b\}| = m-1$$

and

$$|\{j \mid a_j = \pi^{-1}(b)\}| = 1.$$

Let us suppose that $t_m(x_1, \ldots, x_m)=\mathfrak{A}$ where $\mathfrak{A}$ is a superposition over T_m. In this case we can choose a formula $\mathfrak{A}^*$ such, that $\mathfrak{A}^*=t_s(\mathfrak{B}_1, \ldots, \mathfrak{B}_s)$, $\mathfrak{A}^*$ equals b on all characteristic vectors and for every $\mathfrak{B}_i$ there is a characteristic vector a^i such that $\mathfrak{B}_i(a^i) \neq b$. (I.e. $\mathfrak{A}^*$ is "minimal".)

By the assumption we have $s>m$. Let v^k denote the characteristic vector with $x_k=\pi^{-1}(b)$. Consider the matrix

$$\begin{vmatrix} \mathfrak{B}_1(v^1) \ldots \mathfrak{B}_s(v^1) \\ \mathfrak{B}_1(v^2) \ldots \mathfrak{B}_s(v^2) \\ \vdots \\ \mathfrak{B}_1(v^k) \ldots \mathfrak{B}_s(v^k) \end{vmatrix}$$

By the "minimality" of $\mathfrak{A}^*$ every column of the matrix contains at least one occurrence of $\pi^{-1}(b)$. $s>m$ implies that at least one row of the matrix contains two or more occurrence of $\pi^{-1}(b)$. If the l-th row contains at least twice $\pi^{-1}(b)$ then $\mathfrak{A}^*(v^l)=\pi^{-1}(b)$ which is a contradiction as $t_m(v^j)=b$ for all $j\in\{1, 2, \ldots, m\}$. Thus Lemma 2 is proved.

As an immediate consequence of Lemmas 2 and 1 we have

Theorem 3. If $k\geqq 3$ then for all precomplete classes S_π, $v(S_\pi)\geqq\aleph_0$ holds. If k is not prime, then $v(S_\pi)=\mathfrak{c}$.

COMPUTER AND AUTOMATION INSTITUTE
HUNGARIAN ACADEMY OF SCIENCES
KENDE U. 13—17.
BUDAPEST, HUNGARY
H—1502

References

[1] BAGYINSZKI, J., *The lattice of closed classes of linear functions defined over a finite ring of square-free order*, K. Marx Univ. of Economics, Dept. of Math., Budapest, v. 2, 1979.
[2] DEMETROVICS, J., J. BAGYINSZKI, The latttice of linear classes in prime valued logics, Banach Center Publications, Warsaw, PWN, v. 8, 1979, in press.
[3] BAGYINSZKI, J., J. DEMETROVICS, Lineáris osztályok szerkezete prímszám értékű logikában, *Közl. — MTA Számítástech. Automat. Kutató Int. Budapest*, v. 16, 1976, pp. 25—53.
[4] JABLONSZKII, S. V., Functional constructions in k-valued logics, (Russian) *Trudy Mat. Inst. Steklov.*, v. 51, 1958, pp. 5—142.
[5] JANOV, JU. I., A. A. MUČNIK, Existence of k-valued closed classes without a finite basis, (Russian) *Dokl. Akad. Nauk. USSR*, v. 127, 1959, pp. 44—46.
[6] LAU, D., Über die Anzahl von abgeschlossenen Mengen von linearen Funktionen der n-wertigen Logik, *Elektron. Informationsverarb. Kybernet.*, v. 14, 1978, pp. 567—569.
[7] SALOMAA, A. A., On infinitely generated sets of operation in finite algebras, *Ann. Univ. Turku. Ser A. I*, v. 74, 1964, pp. 1—12.
[8] ROSENBERG, I. G., Structure des fonctions de plusieurs variables sur un ensemble fini, *C. R. Acad. Sci. Paris*, v. 260, 1965, pp. 3817—3819.
[9] SZENDREI, A., On closed sets of linear operations over a finite set of square-free cardinality, *Elektron. Informationsverarb. Kybernet.*, v. 14, 1978, pp. 547—559.
[10] POST, E., Introduction to a general theory of elementary propositions, *Amer. J. Mat.*, v. 93, 1921, pp. 163—185.

(Received April 3, 1979)

Recognition of monotone functions

By H.-D. O. F. GRONAU

Let $n, k, k_1, k_2, \ldots, k_n$ be integers with $n \geqq 1$, $k \geqq 1$ and $1 \leqq k_1 \leqq k_2 \leqq \ldots \leqq k_n$. Moreover, let $E = \{0, 1, \ldots, k\}$ and $E_i = \{0, 1, \ldots, k_i\}$ for $i = 1, 2, \ldots, n$. We consider functions

$$f(\underset{\sim}{x}) = f(x_1, x_2, \ldots, x_n): \; N = E_1 \times E_2 \times \ldots \times E_n \to E.$$

We always may assume that f takes each value of E. If $\underset{\sim}{x} = (x_1, x_2, \ldots, x_n)$ and $\underset{\sim}{y} = (y_1, y_2, \ldots, y_n)$ are vectors from N, let $\underset{\sim}{x} \leqq \underset{\sim}{y}$ if and only if $x_i \leqq y_i$ for $i = 1, 2, \ldots, n$. f is said to be monotonically increasing if $\underset{\sim}{x} \leqq \underset{\sim}{y}$ implies $f(\underset{\sim}{x}) \leqq f(\underset{\sim}{y})$. Let $M(k_1, k_2, \ldots, k_n, k)$ denote the set of all such monotone functions. $M(1, 1, \ldots, 1, 1)$ is the set of monotone Boolean functions.

Let $P(f)$ be a minimal set of vectors $\underset{\sim}{x}$ on which f has to be known for knowing the function completely. Let

$$\chi(k_1, k_2, \ldots, k_n, k) = \max_{f \in M(k_1, \ldots, k_n, k)} |P(f)|.$$

Furthermore, let $\varphi(k_1, k_2, \ldots, k_n, k)$ denote the minimal number of operations of the best algorithm for the recognition of an arbitrary function f of $M(k_1, k_2, \ldots, k_n, k)$. Clearly,

$$\varphi(k_1, k_2, \ldots, k_n, k) \geqq \chi(k_1, k_2, \ldots, k_n, k).$$

G. HANSEL [1] proved in case $k_n = k = 1$ that

$$\varphi(1, 1, \ldots, 1, 1) = \chi(1, 1, \ldots, 1, 1) = \left(\left[\frac{n}{2}\right]\right) + \left(\left[\frac{n}{2}\right] + 1\right).$$

It is conjectured that $\varphi = \chi$ is also true in the general case. Therefore, it is important to known χ exactly, not only a lower estimation. The aim of this note is to determine the exact value of χ. Let $m = \sum_{i=1}^{n} k_i$, $m(\underset{\sim}{x}) = \sum_{i=1}^{n} x_i$ and $S_m^l(N) = |\{\underset{\sim}{x}: \underset{\sim}{x} \in N, m(\underset{\sim}{x}) = l\}|$. We have

Theorem 1.

$$\chi(k_1, k_2, \ldots, k_n, k) = \textit{sum of the } 2k \textit{ largest values } S_m^1(N).$$

Proof. A chain $(\underset{\sim}{x}^1, \underset{\sim}{x}^2, \ldots, \underset{\sim}{x}^m)$ of length m is a sequence of m different vectors from N satisfying $\underset{\sim}{x}^1 \leqq \underset{\sim}{x}^2 \leqq \ldots \leqq \underset{\sim}{x}^m$. $P(f)$, where f is an arbitrary function belonging to M, contains no chain of length $2k+1$. Assume the contrary. Then there are 3 consecutive members x', x'', x''' of the chain satisfying $f(x')=f(x'')=f(x''')=i$, where $i \in \{1, 2, \ldots, k-1\}$, or we have $f(x^2)=0$ or $f(x^{m-1})=k$. Since $i=f(x') \leqq f(x'') \leqq f(x''')=i$, $f(x^1) \leqq f(x^2)=0$ or $f(x^m) \geqq f(x^{m-1}) \geqq k$, $f(x'')$, $f(x^1)$ or $f(x^m)$, respectively, would follow from the others immediately, i.e. x'', x^1 or x^m could be omitted in $P(f)$, in contradiction to our supposition that P is minimal. By J. SCHÖNHEIM's result ([2], Theorem 2) we obtain for each f:

$$|P(f)| \leqq \textit{sum of the } 2k \textit{ largest values } S_m^1(N).$$

Now we consider the function

$$f(\underset{\sim}{x}) = \begin{cases} k & \text{if } \left[\frac{m}{2}\right]+k \leqq m(\underset{\sim}{x}), \\ i & \text{if } \left[\frac{m}{2}\right]+2i-k \leqq m(\underset{\sim}{x}) \leqq \left[\frac{m}{2}\right]+2i-k+1 \quad (i=1, \ldots, k-1), \\ 0 & \text{if } m(\underset{\sim}{x}) \leqq \left[\frac{m}{2}\right]-k+1. \end{cases}$$

$f(\underset{\sim}{x})$, where $\left[\frac{m}{2}\right]-k+1 \leqq m(\underset{\sim}{x}) \leqq \left[\frac{m}{2}\right]+k$, cannot be infered by f of the other vectors.

J. SCHÖNHEIM's remarks ([2], Remarks 4 and 5) completes the proof. □

In case $k_n=1$ we obtain

Corollary 1.

$$\chi(1, 1, \ldots, 1, k) = \sum_{i=\left[\frac{n}{2}\right]-k+1}^{\left[\frac{n}{2}\right]+k} \binom{n}{i}.$$

In case $k_n=k=1$ we obtain partly G. HANSEL's result.

Corollary 2.

$$\chi(1, 1, \ldots, 1, 1) = \binom{n}{\left[\frac{n}{2}\right]} + \binom{n}{\left[\frac{n}{2}\right]+1}.$$

Theorem 2.

$$\varphi(1, 1, \ldots, 1, k) = \sum_{i=\left[\frac{n}{2}\right]-k+1}^{\left[\frac{n}{2}\right]+k} \binom{n}{i}.$$

Proof. We use $\varphi \cong \chi$ and Corollary 1 on one side and the special symmetrical chain method by G. HANSEL on the other side. Let f be known on all chains having a length $\leqq a$. Furthermore, let c be an arbitrary chain of length $a+2$. Then f is known on many of the members of c immediately. More precisely, at most on 2 vectors of c we do not know if f takes the value 0 or a value of $\{1, \ldots, k\}$. Then at most on 2 vectors of c we do not know if f takes the value 1 or a value of $\{2, \ldots, k\}$; and so on. Finally, f is unknown at most on $2k$ members of c. By HANSEL's argument the theorem follows immediately. □

Finally, we want to mention that HANSEL's special symmetrical chain method cannot be generalized to the general case $k_n \geqq 2$. N is then partitionable too, but not in HANSEL's special symmetrical chains. This can be verified easily in the case $n=2$, $k_1=1$ and $k_2=2$.

WILHELM-PIECK-UNIVERSITÄT
SEKTION MATHEMATIK
DDR—25 ROSTOCK
UNIVERSITÄTSPLATZ 1

References

[1] HANSEL, G., Sur le nombre des fonctions booléennes monotones de n variables, *C. R. Acad. Sci. Paris*, v. 262, 1966, No. 20, pp. 1088—1090.
[2] SCHÖNHEIM, J., A generalization of results of P. Erdős, G. Katona, and D. J. Kleitman concerning Sperner's theorem, *J. Combinatorial Theory*, v. 11, 1971, pp. 111—117.

(Received April 3, 1979)

A method for minimizing partially defined Boolean functions

By F. MÓRICZ,* A. VARGA* and P. ECSEDI-TÓTH**

A simple procedure is presented for minimizing partially defined Boolean functions. A binary tree is constructed to every such function in a natural way, then certain subtrees are used to obtain a partially defined irredundant normal form, equivalent to the starting function.

§ 1. Preliminaries

The truth-values TRUE, FALSE, and UNDEFINED will be denoted by 1, 0, and $*$ (asterisk), respectively. The notion of partially defined Boolean function (truth function) and the notion of totally defined Boolean function (propositional formula) is used in the standard way. The collection of the partially defined Boolean functions will be denoted by $\mathscr{B}$, while the collection of the totally defined Boolean functions by $\mathscr{F}$. It is clear that $\mathscr{F} \subsetneqq \mathscr{B}$. We emphasize that the truth-values TRUE and FALSE are considered also as elements of $\mathscr{F}$, but UNDEFINED is not a partially defined Boolean function, i.e. we put

$$\{0, 1\} \subset \mathscr{F} \quad \text{and} \quad * \notin B.$$

Let $\varphi, \psi_1, \ldots, \psi_m$ be arbitrary formulae (in $\mathscr{F}$) and let $A_1, \ldots, A_m$ be arbitrary logical variables, not necessarily occurring in φ. If each occurrence of A_i in φ (if any) is substituted by ψ_i for $i=1, \ldots, m$, then the resulting formula will be denoted by $\varphi[A_1|\psi_1, \ldots, A_m|\psi_m]$, while the substitution process by $\langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle$. If $\{A_1, \ldots, A_m\}$ is the (full) set of the logical variables occurring in φ and each formula ψ_i $(1 \leqq i \leqq m)$ is identical with one of the truth-values TRUE and FALSE, then the substitution $\langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle$ will be called a *valuation*, and $\varphi[A_1|\psi_1, \ldots, A_m|\psi_m]$ is the *value* of φ under this valuation. The value of φ is clearly logically equivalent to one of the truth-values TRUE and FALSE.

It is well-known from the propositional calculus that the value of any totally defined function φ does not depend on the order of the logical variables in the valuation. In other words, to determine the value of a formula under a valuation it is indifferent that the substitution is executed simultaneously or successively.

We note that in certain cases the same definition works for $\varphi \in \mathscr{B}$, too. In more detail, if $\langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle$ is a valuation for a partially defined function φ, then $\varphi[A_1|\psi_1, \ldots, A_m|\psi_m]$, defined exactly as above, is either UNDEFINED or logically equivalent to one of the truth-values TRUE and FALSE.

§ 2. The tree constructing algorithm

For a partially defined Boolean function φ we have to know the set of those valuations for which φ is not defined, i.e. the set of undefined valuations. For any $\varphi \in \mathscr{B}$, let us write

$$\varphi[A_1|\psi_1, \ldots, A_m|\psi_m] = *$$

if

$$\text{neither} \quad \varphi[A_1|\psi_1, \ldots, A_m|\psi_m] = 1 \quad \text{nor} \quad \varphi[A_1|\psi_1, \ldots, A_m|\psi_m] = 0,$$

where $\{A_1, \ldots, A_m\}$ is the full set of the logical variables occurring in φ, and $\psi_1, \ldots, \psi_m$ are from the set {TRUE, FALSE}. We put

$$\varphi_* = \{\langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle \colon \varphi[A_1|\psi_1, \ldots, A_m|\psi_m] = *\}$$

and introduce the following two totally defined Boolean functions:

$$\varphi^0[A_1|\psi_1, \ldots, A_m|\psi_m] = \begin{cases} \varphi[A_1|\psi_1, \ldots, A_m|\psi_m] \\ \quad \text{if} \quad \langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle \notin \varphi_*, \\ 0 \quad \text{otherwise;} \end{cases}$$

$$\varphi^1[A_1|\psi_1, \ldots, A_m|\psi_m] = \begin{cases} \varphi[A_1|\psi_1, \ldots, A_m|\psi_m] \\ \quad \text{if} \quad \langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle \notin \varphi_*, \\ 1 \quad \text{otherwise.} \end{cases}$$

Finally, let us define a subset $[\varphi^0, \varphi^1]$ of $\mathscr{F}$ by

$$[\varphi^0, \varphi^1] = \{\psi \in \mathscr{F} \colon \text{if } \langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle \notin \varphi_*, \text{ then } \psi[A_1|\psi_1, \ldots, A_m|\psi_m] = \varphi[A_1|\psi_1, \ldots, A_m|\psi_m]\}.$$

Lemma 1. *Let the function f be defined as follows*
(i) $f\colon \mathscr{B} \to \mathscr{P}(\mathscr{F})$, *where $\mathscr{P}(\mathscr{F})$ is the power set of $\mathscr{F}$;*
(ii) $f(\varphi) = [\varphi^0, \varphi^1]$ *if* $\varphi \in \mathscr{B}$.
Then f is one-one.

Proof. Trivial. □

A somewhat stronger connection between the sets $\mathscr{B}$ and $\mathscr{P}(\mathscr{F})$ can be easily established, too. In this paper, however, we need only the statement of the above lemma, so we do not deal with this strengthening.

Now, it is clear that a partially defined Boolean function φ may be given by the set φ_* and an arbitrary but fixed element ψ of $[\varphi^0, \varphi^1]$. By our point of interest, the function φ in question does not depend on the choice of ψ. This is what we are going to demonstrate in the subsequent paragraphs of this section. The totally defined Boolean function ψ is called a *representative* of φ and denoted by $[\varphi]$.

We turn to the tree constructing algorithm.

Let $\varphi \in \mathscr{B}$, fix an order S of the logical variables occurring in φ, and fix a representative $[\varphi]$. Then the following process will yield a binary tree.

(i) Start with $[\varphi]$ as the initial vertex of the tree to be constructed.

(ii) Choose the first variable if we come from (i) or the subsequent variable if we come from (vii), say A, in the fixed order S at the vertex ψ actually treated.

(iii) Form the expressions $\psi[A|1]$ and $\psi[A|0]$.

(iv) Apply the so-called computational rules of the propositional calculus (listed, e. g., in [2], [3], from (1) to (19)) as many times as possible in order to eliminate the truth-values from $\psi[A|1]$ and $\psi[A|0]$, unless they are truth-values.

(v) If the elimination is terminated, then these truth-value free expressions, otherwise $\psi[A|1]$ and $\psi[A|0]$ themselves, will provide the two new vertices obtainable from ψ.

(vi) If ψ' is obtained from ψ by substituting 1 (resp. 0) for A, then connect ψ and ψ' by an edge labelled with A (resp. $\bar{A}$, the negation of A).

(vii) Stop if all the variables of S have been chosen, otherwise repeat from (ii).

As it was proved in [2], [3] this process, for every $[\varphi]$ and S, determines a unique binary tree denoted by $[\varphi]_S$. The vertices of this tree are formulae; in particular, all the lowest vertices are already truth-values, while the edges are labelled with literals, i.e. logical variables or negated logical variables. However, the tree $[\varphi]_S$ depends heavily on the representative $[\varphi]$; if another representative of φ is given, then the resulting tree will usually alter. Our next step is to remove this undesirable dependence. To this end, we recall the notion of end vertex and path (cf. [3]).

Consider the binary tree $[\varphi]_S$ constructed above. A vertex of $[\varphi]_S$ is called an *end vertex* if it is identical to a single truth-value (excluding UNDEFINED). By a *path* in $[\varphi]_S$ we mean a sequence of literals

(i) whose first element labels an edge starting from $[\varphi]$,

(ii) the subsequent elements of the sequence label edges adjacent in $[\varphi]_S$,

(iii) the last element of the sequence labels an edge terminating at an end vertex.

This end vertex will be called also the end vertex of the path in question.

It is clear that any path contains every literal at most once (i.e. there no loop can occur in $[\varphi]_S$).

We can easily establish a one-one connection between the set of all paths (of $[\varphi]_S$) and the set of all valuations (of the formula $[\varphi]$) in the following way:

a) Let $p=\langle \varphi_1, \ldots, \varphi_m \rangle$ be a path. Define the valuation $v_p=\langle A_1|\psi_1, \ldots, A_m|\psi_m \rangle$ by

$$\psi_j = \begin{cases} 0 & \text{if} \quad A_j = \bar{\varphi}_j, \\ 1 & \text{otherwise.} \end{cases}$$

b) Conversely, if $v=\langle A_1|\psi_1, \ldots, A_m|\psi_m \rangle$ is a valuation, then define the path $p_v=\langle \varphi_1, \ldots, \varphi_m \rangle$ by

$$\varphi_j = \begin{cases} A_j & \text{if} \quad \psi_j = 1, \\ \bar{A}_j & \text{otherwise.} \end{cases}$$

Lemma 2. *Let $\varphi\in\mathscr{B}$ and $\psi', \psi''\in\mathscr{F}$ be two representatives for φ. Let an order S of the logical variables occurring in φ and a valuation $v=\langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle$ be fixed. If the end vertices of the two paths, associated with v in the trees ψ'_S and ψ''_S respectively, are not identical, then we necessarily have $v\in\varphi_*$.*

Proof. Since $\psi'\in[\varphi^0, \varphi^1]$ and $\psi''\in[\varphi^0, \varphi^1]$, by definition, for each valuation $\tilde{v}=\langle A_1|\tilde{\psi}_1, \ldots, A_m|\tilde{\psi}_m\rangle \notin \varphi_*$ we have

$$\psi'[A_1|\tilde{\psi}_1, \ldots, A_m|\tilde{\psi}_m] = \varphi[A_1|\tilde{\psi}_1, \ldots, A_m|\tilde{\psi}_m] = \\ = \psi''[A_1|\tilde{\psi}_1, \ldots, A_m|\tilde{\psi}_m].$$

This implies immediately the statement of our Lemma. □

By virtue of Lemma 2, for every $\varphi\in\mathscr{B}$ and fixed order S of the logical variables in φ we can uniquely define a tree φ_{S*} as follows:

(i) Choosing an arbitrary representative for φ, let us construct the $[\varphi]_S$-tree in the way described above;

(ii) For each valuation $v\in\varphi_*$ let us put the sign $*$ at the end vertex of the path p_v associated with v.

Lemma 3. *Let $\varphi\in\mathscr{B}$ and fix an order S of the logical variables occurring in φ. Then the tree φ_{S*} just constructed is uniquely determined (not depending on the representative $[\varphi]$).*

Proof. The assertion clearly follows from the previous lemma. □

§ 3. Minimization

Two partially defined Boolean functions, φ and ψ are said to be *equivalent* if

(i) φ is defined under a valuation if and only if ψ is so;

(ii) whenever φ is defined under a valuation $\langle A_1|\psi_1, \ldots, A_m|\psi_m\rangle$, then

$$\varphi[A_1|\psi_1, \ldots, A_m|\psi_m] = \psi[A_1|\psi_1, \ldots, A_m|\psi_m].$$

A $\varphi\in\mathscr{B}$ is said to be in a *disjunctive* (resp. *conjunctive*) *normal form* if $[\varphi]$ is so. If $[\varphi]$ is irredundant, then φ is also said to be *irredundant*.

We note that somewhat other definitions of these notions are possible, too. We do not go into the details of this question, since the above definitions are quite appropriate for us concerning the minimization of partially defined Boolean functions.

Let $\varphi\in\mathscr{B}$, a representative of φ, and an order S of the logical variables in φ be fixed. Construct the $[\varphi]_S$-tree. If all the edges leading to the truth-value FALSE (resp. TRUE) are omitted in the $[\varphi]_S$-tree, we get the so-called truncated $[\varphi]^1_S$-tree (resp. $[\varphi]^0_S$-tree) (cf. [2]).

The notion of maximal simplifiable subtree was introduced in [3]. Here we recall the definition.

Let a $[\varphi]^i_S$-tree ($i=0$ or 1) and a path p in it be given. By a *maximal simplifiable subtree* (in abbreviation: MSST) of p we mean a subtree Φ of the $[\varphi]^i_S$-tree which satisfies the following four conditions:

(i) Φ contains p;

(ii) Each path in Φ starts from the initial vertex and terminates at an end vertex of the $[\varphi]_S^i$-tree;

(iii) Φ is of the form drawn in Fig. 1, where $A_1, A_2, \ldots, A_n$ are logical variables in φ and $\alpha_1, \ldots, \alpha_{n+1}$ are sequences of literals such that each of the literals in these sequences is different from $A_1, \bar{A}_1, \ldots A_n, \bar{A}_n$; and there exists at least one path, the end vertex of which is not identical with $*$;

(iv) The number of the paths in Φ is maximal in the sense that there exists no subtree of the $[\varphi]_S^i$-tree that also satisfies the conditions (i), (ii), and (iii) and contains more paths than Φ contains.

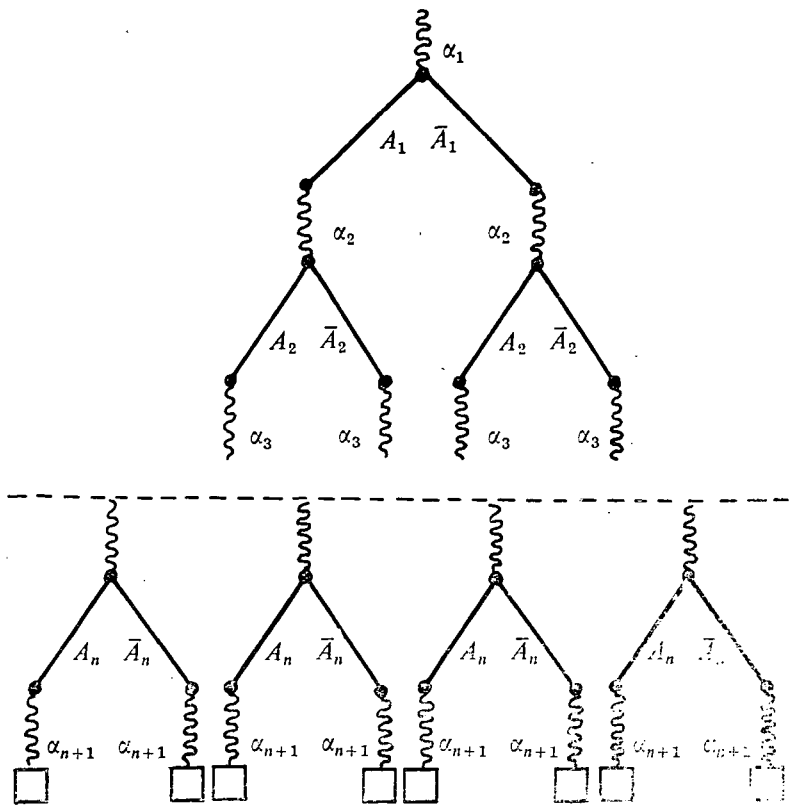


Fig. 1

These subtrees can be effectively generated by a simple way (see [2]) using an algorithm from [1].

By a *cover* of the truncated $[\varphi]_S^i$-tree we mean a set of maximal simplifiable subtrees such that every path in $[\varphi]_S^i$ belongs to at least one MSST from this set. A cover is said to be *irredundant* if every MSST in it contains at least one path which belongs only to this MSST.

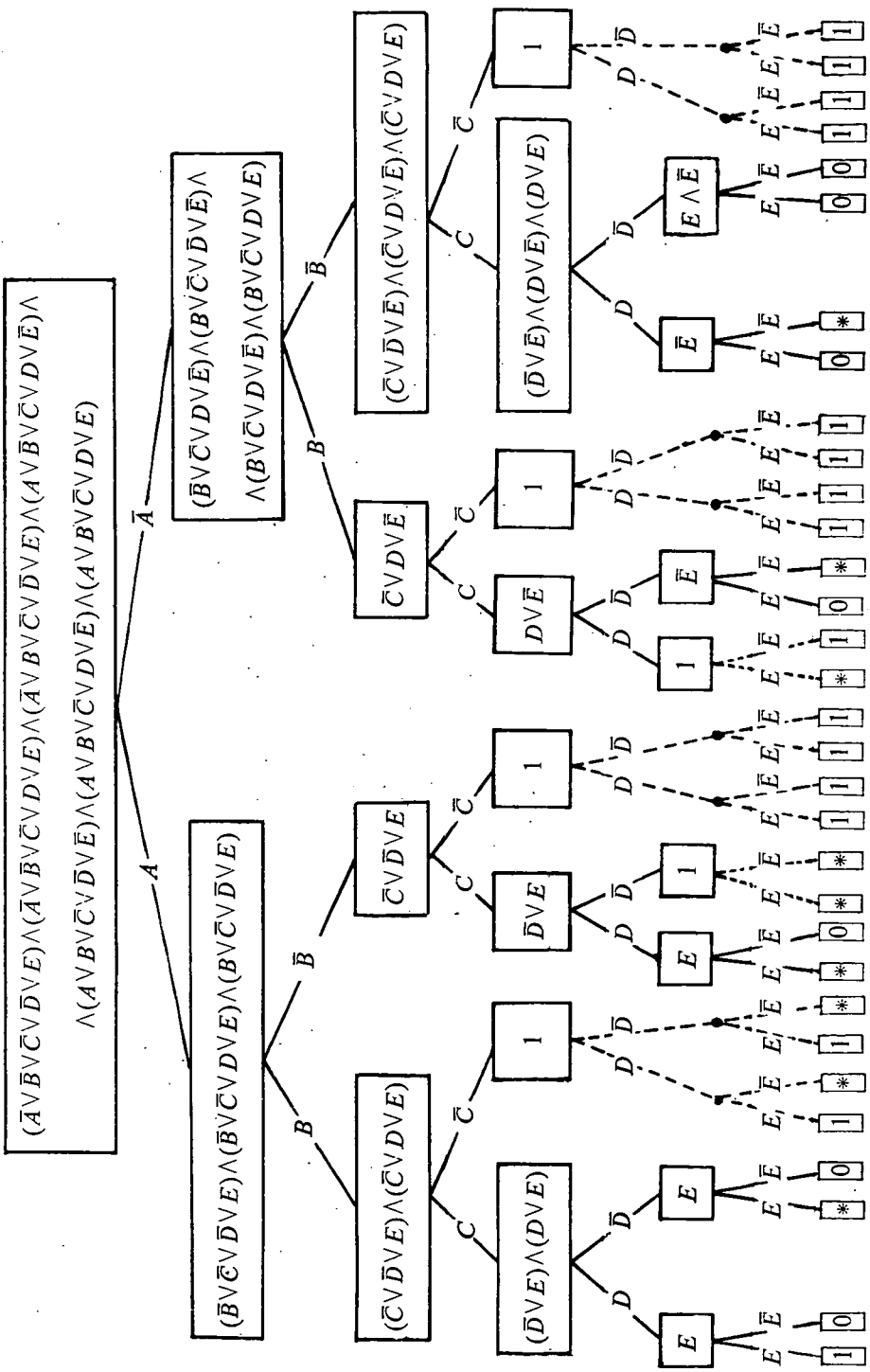

(Ā∨B̄∨C̄∨D̄∨E)∧(Ā∨B̄∨C̄∨D∨E)∧(Ā∨B∨C̄∨D̄∨E)∧(A∨B̄∨C̄∨D̄∨E)∧
∧(A∨B∨C̄∨D̄∨E)∧(A∨B∨C̄∨D∨E)∧(A∨B∨C̄∨D∨E)
A
Ā
(B̄∨C̄∨D̄∨E)∧(B̄∨C̄∨D∨E)∧(B̄∨C̄∨D∨E)
(B̄∨C̄∨D∨Ē)∧(B̄∨C̄∨D̄∨E)∧
∧(B̄∨C̄∨D∨E)∧(B∨C̄∨D∨E)
B̄
B
(C̄∨D̄∨E)∧(C̄∨D∨E)
C̄∨D∨E
(C̄∨D̄∨Ē)∧(C̄∨D∨Ē)∧(C̄∨D∨E)
C̄
C
(C̄∨D̄∨E)∧(C̄∨D∨E)
C̄∨D̄∨E
C̄∨D∨E
1
(D̄∨E)∧(D∨E)
D̄∨E
D̄∨E
(D̄∨Ē)∧(D∨Ē)∧(D∨E)
D̄
D
E
Ē
E∧Ē
0
1
*

Fig. 2

In [2] we presented an algorithm for constructing an irredundant cover of an arbitrary φ_S^i-tree, when φ is a totally defined Boolean function. This algorithm can be applied without any changes to the case of partially defined Boolean functions if the end vertices $*$ in the $[\varphi]_S^i$-tree are treated as if they were either 1 or 0 according to how it is more favourable to obtain an irredundant cover.

Thus, we find an irredundant cover of the $[\varphi]_S^i$-tree, whence, via the main theorem of [2], we get an irredundant normal form of the representative $[\varphi]$. At the same time, this irredundant normal form of $[\varphi]$ is a representative of a partially defined irredundant normal form of φ. Taking the set φ_* into account, a partially defined irredundant normal form of φ is completely determined.

Example. Let $[\varphi]$ and φ_* be given as follows:

$$[\varphi] = (\bar{A}\vee\bar{B}\vee\bar{C}\vee\bar{D}\vee E)\wedge(\bar{A}\vee\bar{B}\vee\bar{C}\vee D\vee E)\wedge(\bar{A}\vee B\vee\bar{C}\vee\bar{D}\vee E)\wedge$$
$$\wedge(A\vee\bar{B}\vee\bar{C}\vee D\vee\bar{E})\wedge(A\vee B\vee\bar{C}\vee\bar{D}\vee\bar{E})\wedge(A\vee B\vee\bar{C}\vee D\vee\bar{E})\wedge$$
$$\wedge(A\vee B\vee\bar{C}\vee D\vee E),$$
$$\varphi_* = \{\langle A, \bar{B}, C, \bar{D}, E\rangle;\ \langle A, B, \bar{C}, D, \bar{E}\rangle;\ \langle A, B, \bar{C}, \bar{D}, \bar{E}\rangle;$$
$$\langle A, \bar{B}, C, D, E\rangle;\ \langle A, \bar{B}, C, \bar{D}, E\rangle;\ \langle A, \bar{B}, C, \bar{D}, \bar{E}\rangle;$$
$$\langle \bar{A}, B, C, D, E\rangle;\ \langle \bar{A}, B, C, \bar{D}, \bar{E}\rangle;\ \langle \bar{A}, \bar{B}, C, D, \bar{E}\rangle\},$$

where A stands for $A|1$, while $\bar{A}$ stands for $A|0$; similarly for the other letters. In the case $S=\langle A, B, C, D, E\rangle$ the $[\varphi]_S$-tree is indicated in Fig. 2.

Let us consider the $[\varphi]_S^1$-tree (Fig. 3).

An irredundant cover of the $[\varphi]_S^1$-tree can be obtained by means of the maximal simplifiable subtrees indicated in Fig. 4.

Hence a representative of a partially defined irredundant normal form of φ will be

$$\bar{C}\vee(A\wedge E)\vee(\bar{A}\wedge B\wedge\bar{E}).$$

Finally, we should like to mention that from a practical point of view we have a considerable number of experimental evidences produced by two different PL/I

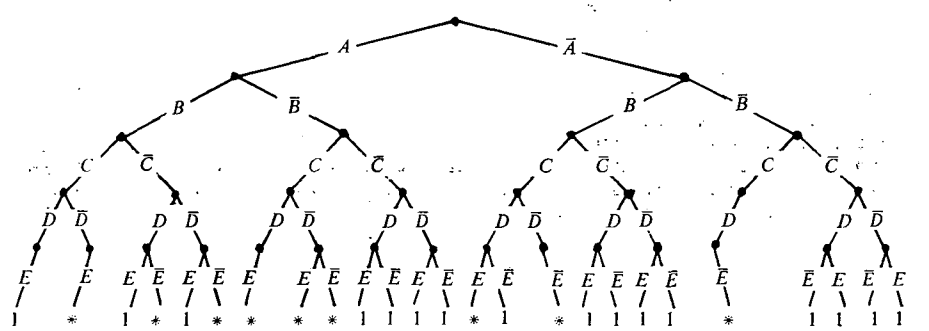


Fig. 3

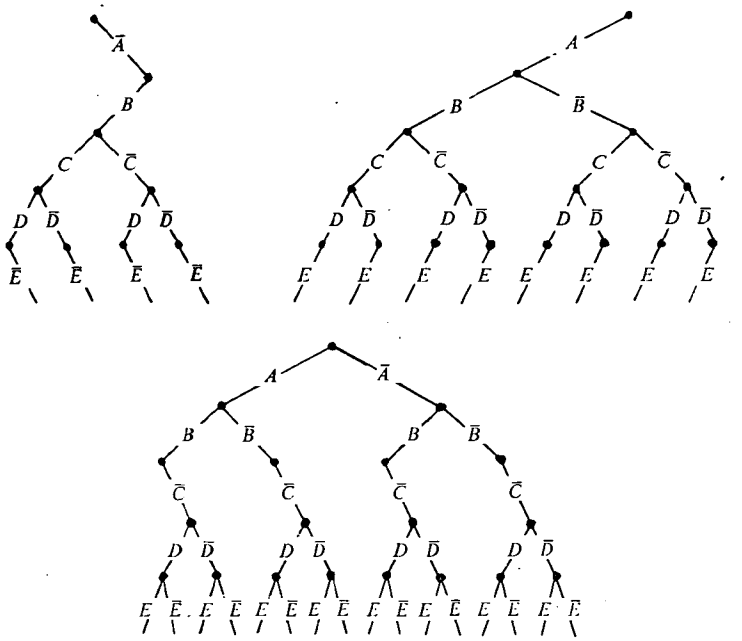


Fig. 4

realizations of our algorithm that provide minima on the average cost of other well-known algorithms such as e.g. the algorithm due to Quine—McCluskey. From a theoretical point of view, however, the proper nature of the minimization has not been understood yet.

* BOLYAI INSTITUTE OF THE
ATTILA JÓZSEF UNIVERSITY
ARADI VÉRTANÚK TERE 1.
SZEGED, HUNGARY
H—6720

** RESEARCH GROUP ON THEORY OF AUTOMATA
HUNGARIAN ACADEMY OF SCIENCES
SOMOGYI U. 7.
SZEGED, HUNGARY
H—6720

References

[1] ECSEDI-TÓTH, P., F. MÓRICZ and A. VARGA, A note on symmetric Boolean functions, *Acta Cybernet.*, v. 3, 1978, pp. 321—326.

[2] VARGA, A., P. ECSEDI-TÓTH and F. MÓRICZ, On a connection between algebraic and graph-theoretic minimization of truth functions, *Proceedings of the International Colloquium on Algebraic Methods in Graph Theory*, Szeged, 1978, Akadémiai Kiadó and North Holland Company.

[3] VARGA, A., P. ECSEDI-TÓTH and F. MÓRICZ, A heuristic method for speeding up the manual optimization of Boolean functions, *Acta Tech. Acad. Sci. Hungar.*, to appear.

(Received Dec. 13, 1978)

Modal logics with function symbols

By K. TÓTH

We prove completeness theorems for modal logics with function symbols. These logics are generalizations of the well-known non-classical logical systems.

Our work was deeply influenced by a paper of K. SCHÜTTE [2].

§ 1. Preliminaries

We shall use the following symbols: parantheses, commas, variables, function symbols, relation symbols, logical symbols $(\sim, \wedge, \Box, \forall)$. The set of terms is defined by the usual recurrence:

(i) If x is a variable, then x is a term.
(ii) If f is an n-ary function symbol and $\tau_1, \ldots, \tau_n$ are terms, then $f(\tau_1, \ldots, \tau_n)$ is also a term. In the case of $n=0$ the parantheses will be omitted.

The set of atoms is defined in the standard way: if r is an n-ary relation symbol and $\tau_1, \ldots, \tau_n$ are terms, then $r(\tau_1, \ldots, \tau_n)$ is an atom. Also, definition of the set of formulae is well-known:

(i) If $\mathscr{A}$ is an atom, then $\mathscr{A}$ is a formula.
(ii) If $\mathscr{A}, \mathscr{B}$ are formulae, then so are $(\mathscr{A} \wedge \mathscr{B})$, $\sim\mathscr{A}$ and $\Box\mathscr{A}$.
(iii) If $\mathscr{A}$ is a formula and x is a variable, then $\forall x\mathscr{A}$ is a formula.

We shall use the abbreviations: $(\mathscr{A} \vee \mathscr{B})$ for $\sim(\sim\mathscr{A} \wedge \sim\mathscr{B})$; $(\mathscr{A} \to \mathscr{B})$ for $\sim(\sim\mathscr{B} \wedge \mathscr{A})$; $\Diamond\mathscr{A}$ for $\sim\Box\sim\mathscr{A}$; $\exists x\mathscr{A}$ for $\sim\forall x\sim\mathscr{A}$. Parantheses will be omitted if no confusion can occur. If $\mathscr{K}$ is a formula or term x is a free variable (defined in the well-known way) and τ is a term, then $\mathscr{K}[x/\tau]$ will denote the result of substitution of τ for x everywhere in $\mathscr{K}$. By a classical model A we shall mean a function if it associates

(i) a non-empty set $|A|$ to 0 (zero),
(ii) a function $f_A: |A|^n \to |A|$ to each n-ary function symbol f,
(iii) a relation $r_A \subseteq |A|^n$ to each n-ary relation symbol r.

Definition. By a modal model we mean a quintuple $\langle S, N, O, R, P\rangle$ where S is an arbitrary set, $N\subseteq S$, $O\in S$, $R\subseteq S\times S$, P is a function with domain S and $P(A)$ is a classical model, provided $A\in S$, furthermore $|P(A)|\subseteq|P(B)|$ if $A, B\in S$, ARB.

Definition. A modal model is simple if for every n-ary function symbol f, there exists a function $\bar{f}$ with domain $\bigcup_{A\in S}|P(A)|$, such that f_A is a restriction of $\bar{f}$ to $|P(A)|$ where $A\in S$.

Definition. If $|P(A)|=|P(B)|$ for every $A, B\in S$, ARB, then the model is called stable.

Let $\langle S, N, O, R, P\rangle$ be a modal model. By an interpretation we mean a function k such that to each variable k associates an element of $\bigcup_{A\in S}|P(A)|$.

Let a model $\langle S, N, O, R, P\rangle$ and an interpretation k be given. By a valuation $\varkappa$ a partial function is meant with the following properties:

(i) $\varkappa(x, A)=k(x)$ if $A\in S$ and x is variable such that $k(x)\in|P(A)|$.
(ii) $\varkappa(f(\tau_1, \ldots, \tau_n), A)=f_{P(A)}(\varkappa(\tau_1, A), \ldots, \varkappa(\tau_n, A))$ if $A\in S$ and $\tau_1, \ldots, \tau_n$ are terms such that for every variable x occuring in any of them, $k(x)\in|P(A)|$.
(iii) $\varkappa(\tau, A)$ is undefined if $A\in S$ and there exists a variable x in the term τ such that $k(x)\notin|P(A)|$.

Let $\mathscr{A}$ be an expression (i.e. a term or formula) and assume a model $\langle S, N, O, R, P\rangle$ is given. Let us fix $A\in S$ and an interpretation k. We say that $\mathscr{A}\in\mathscr{H}_k(A)$ if for every variable x occuring free in $\mathscr{A}$ we have $k(x)\in|P(A)|$.

Let $A\in S$, $\mathscr{A}$ be a formula and k an interpretation. We define the satisfaction relation $A\models\mathscr{A}[k]$ by the following clauses:

(i) $A\models r(\tau_1, \ldots, \tau_n)[k]$ if and only if $\tau_1, \ldots, \tau_n\in\mathscr{H}_k(A)$ and $r_{P(A)}(\varkappa(\tau_1, A), \ldots, \varkappa(\tau_n, A))$;
(ii) $A\models(\mathscr{A}\wedge\mathscr{B})[k]$ if and only if $A\models\mathscr{A}[k]$ and $A\models\mathscr{B}[k]$;
(iii) $A\models\sim\mathscr{A}[k]$ if and only if $\mathscr{A}\in\mathscr{H}_k(A)$ and $A\models\mathscr{A}[k]$ does not hold;
(iv) $A\models\Box\mathscr{A}[k]$ if and only if $A\in N$ and for every $B\in S$, ARB implies $B\models\mathscr{A}[k]$;
(v) $A\models\forall x\mathscr{A}[k]$ if and only if for every interpretation k', such that $k'(x)\in|P(A)|$ and $k'(y)=k(y)$ if $y\neq x$, we have $A\models\mathscr{A}[k']$.

We put $\mathscr{I}$ into the set of relation symbols with the following meaning:

$A\models\mathscr{I}(\tau_1, \tau_2)[k]$ if and only if $\tau_1, \tau_2\in\mathscr{H}_k(A)$ and $\varkappa(\tau_1, A)=\varkappa(\tau_2, A)$, i.e. $\mathscr{I}$ denotes the identity.

Let $\mathscr{A}$ be a formula, $\langle S, N, O, R, P\rangle$ a modal model and k an interpretation. $\mathscr{A}$ is valid in $\langle S, N, O, R, P\rangle$ under the interpretation k if $O\models\mathscr{A}[k]$.

The reader can consult with [1] for notions and notations not explained here.

§ 2. Modal logics

To give a modal logic we have to give a classical formula $\mathscr{T}$ with the properties:

(i) no free variable occurs in $\mathscr{T}$,

(ii) $\mathscr{T}$ is in the classical language of the following non-logical symbols: o, 0-ary function symbol; n, unary relation symbol; r, binary relation symbol; i, binary relation symbol.

This classical formula, called parameter of the logic, is meant to formalize a property of the structure $\langle S, N, O, R\rangle$ provided $\langle S, N, O, R, P\rangle$ is a model of the intended modal logic.

If we restrict ourselves to modal logic with only simple/stable models then we call them simple/stable modal logics.

Let a modal logic be given. A formula $\mathscr{A}$ is satisfiable if there exist a model $\langle S, N, O, R, P\rangle$ and a interpretation k such that:

(i) $\langle S, N, O, R, P\rangle$ is simple/stable if the given logic is simple/stable;
(ii) the parameter of the logic is valid in the classical model A defined by: $|A|=S$, $O_A=o$, $n_A(B)\Leftrightarrow B\in N$, if $B, C\in S$ then $r_A(B, C)\Leftrightarrow BRC$ and $i_A(B, C)\Leftrightarrow B=C$;
(iii) $O\models\mathscr{A}[k]$.

A formula $\mathscr{A}$ is a tautology if $\sim\mathscr{A}$ is not satisfiable.

In this paper we treat some special logics, the parameter of which is an arbitrary (may be empty) conjunction of the following formulae:

K1. $\forall x n(x)$
K2. $\forall x r(x, x)$
K3. $\forall x\forall y\forall z\big(r(x, y)\wedge r(y, z)\rightarrow r(x, z)\big)\wedge\forall x\big(n(x)\rightarrow\forall y\big(r(x, y)\rightarrow n(y)\big)\big)$.

These logics will be axiomatized with suitable subsets of the following axioms:

A1. $\mathscr{A}\rightarrow\mathscr{A}\wedge\mathscr{A}$
A2. $\mathscr{A}\wedge\mathscr{B}\rightarrow\mathscr{A}$
A3. $(\mathscr{A}\rightarrow\mathscr{B})\rightarrow\big(\sim(\mathscr{B}\wedge\mathscr{C})\rightarrow\sim(\mathscr{C}\wedge\mathscr{A})\big)$
A4. $\forall x(\mathscr{A}\rightarrow\mathscr{B})\rightarrow(\forall x\mathscr{A}\rightarrow\forall x\mathscr{B})$
A5. $\mathscr{A}\rightarrow\forall x\mathscr{A}$ where x is not free in $\mathscr{A}$;
A6.a. $\forall x\mathscr{A}\rightarrow\mathscr{A}[x/y]$ where y is a variable and it is free with respect to x in $\mathscr{A}$;
A6.b. $\forall x\mathscr{A}\rightarrow\mathscr{A}[x/\tau]$ where τ is a term and it is free with respect to x in $\mathscr{A}$ and $\mathscr{A}$ is a classical formula;
A7. $\Box(\mathscr{A}\rightarrow\mathscr{B})\rightarrow(\Box\mathscr{A}\rightarrow\Box\mathscr{B})$
A8. $\Box(\mathscr{A}\rightarrow\mathscr{A})$ if K1 appears in the parameter of the logic as a conjunct;
A9. $\Box\mathscr{A}\rightarrow\mathscr{A}$ if K2 is a conjunct in the parameter of the logic;
A10. $\Box\mathscr{A}\rightarrow\Box\Box\mathscr{A}$ if K3 occurs in the parameter of the logic;
A11. $\forall x\mathscr{I}(x, x)$
$\forall x\forall y\big(\mathscr{I}(x, y)\rightarrow(\mathscr{A}[x/y]\rightarrow\mathscr{A})\big)$
$\Box\mathscr{A}\rightarrow\forall x\forall y\big(\mathscr{I}(x, y)\rightarrow\Box\mathscr{I}(x, y)\big)$
$\Box\mathscr{A}\rightarrow\forall x\forall y\big(\sim\mathscr{I}(x, y)\rightarrow\Box\sim\mathscr{I}(x, y)\big)$ if $\mathscr{I}$ occurs in the logic;
A12. $\forall x\Box\mathscr{A}\rightarrow\Box\forall x\mathscr{A}$ if the logic is stable.

If the logic is simple, then axioms A6.a and A6.b are replaced by the more general axiom.

A6. $\forall x\mathscr{A} \to \mathscr{A}[x/\tau]$ where τ. is a term free with respect to x in $\mathscr{A}$ and $\mathscr{A}$ is arbitrary.

We fix the following rules of inference:

R1. From $\mathscr{A}$ and $\mathscr{A} \to \mathscr{B}$ we infer $\mathscr{B}$.
R2. From $\mathscr{A}$ we infer $\forall x\mathscr{A}$.
R3. From $\mathscr{A} \to \mathscr{B}$ we infer $\Box\mathscr{A} \to \Box\mathscr{B}$.

This last rule can be used in a logic in which $\forall x(r(o,x) \wedge \mathscr{T} \to \mathscr{T}[o/x])$ is a tautology, where $\mathscr{T}$ is the parameter of the logic. This holds for K1, K2, K3.
The notion of derivability is used in the usual sense (denotation: $\vdash$).

Theorem 1. *(Soundness.)* Let a modal logic be given. If a formula $\mathscr{A}$ is derivable in this modal logic, then it is a tautology.

Proof. Trivial.

§ 3. Metatheorems

The proofs of metatheorems will only be sketched.

Assertion 1. Every tautology of classical sentential logic is derivable.

Proof. A1—A3 and R1 is a complete formalization of classical sentential logic.

Assertion 2. $\vdash \Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{A} \wedge \Box\mathscr{B}$.

Proof.

$\vdash \mathscr{A} \wedge \mathscr{B} \to \mathscr{A}$	(classical theorem)
$\vdash \Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{A}$	(R3)
$\vdash \mathscr{A} \wedge \mathscr{B} \to \mathscr{B}$	(classical theorem)
$\vdash \Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{B}$	(R3)
$\vdash (\Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{A}) \to ((\Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{B}) \to (\Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{A} \wedge \Box\mathscr{B}))$	(classical theorem)
$\vdash \Box(\mathscr{A} \wedge \mathscr{B}) \to \Box\mathscr{A} \wedge \Box\mathscr{B}$	(R1)

Assertion 3. $\vdash \Box\mathscr{A} \wedge \Box\mathscr{B} \to \Box(\mathscr{A} \wedge \mathscr{B})$.

Proof.

$\vdash \mathscr{A} \to (\mathscr{B} \to \mathscr{A} \wedge \mathscr{B})$	(classical theorem)
$\vdash \Box\mathscr{A} \to \Box(\mathscr{B} \to \mathscr{A} \wedge \mathscr{B})$	(R3)
$\vdash \Box(\mathscr{B} \to \mathscr{A} \wedge \mathscr{B}) \to (\Box\mathscr{B} \to \Box(\mathscr{A} \wedge \mathscr{B}))$	(A7)
$\vdash \Box\mathscr{A} \to (\Box\mathscr{B} \to \Box(\mathscr{A} \wedge \mathscr{B}))$	(by classical theorems)
$\vdash \Box\mathscr{A} \wedge \Box\mathscr{B} \to \Box(\mathscr{A} \wedge \mathscr{B})$	(by classical theorems)

Assertion 4. $\vdash \Box\mathscr{A} \vee \Box\mathscr{B} \to \Box(\mathscr{A} \vee \mathscr{B})$.

Proof.

$\vdash \mathscr{A} \to \mathscr{A} \vee \mathscr{B}$	(classical theorem)
$\vdash \Box\mathscr{A} \to \Box(\mathscr{A} \vee \mathscr{B})$	(R3)
$\vdash \Box\mathscr{B} \to \Box(\mathscr{A} \vee \mathscr{B})$	(similarly)
$\vdash \Box\mathscr{A} \vee \Box\mathscr{B} \to \Box(\mathscr{A} \vee \mathscr{B})$	(by classical theorems)

Theorem 2. If $\vdash \mathscr{A} \to \mathscr{B}$ and $\vdash \mathscr{B} \to \mathscr{A}$ then $\mathscr{A}$ can be replaced by $\mathscr{B}$ in an arbitrary formula without loss of its derivability.

Proof. One can proceed by induction from the following facts:
$\vdash \mathscr{A} \to \mathscr{B}$ implies that $\vdash \sim\mathscr{B} \to \sim\mathscr{A}$
$\vdash \mathscr{A} \to \mathscr{B}$ implies that $\vdash \mathscr{A} \wedge \mathscr{C} \to \mathscr{B} \wedge \mathscr{C}$ and $\vdash \mathscr{C} \wedge \mathscr{A} \to \mathscr{C} \wedge \mathscr{B}$
$\vdash \mathscr{A} \to \mathscr{B}$ implies that $\vdash \Box\mathscr{A} \to \Box\mathscr{B}$
$\vdash \mathscr{A} \to \mathscr{B}$ implies that $\vdash \forall x\mathscr{A} \to \forall x\mathscr{B}$.

Assertion 5. A8 and R3 can be omitted if the following rule is added to the system: If $\vdash \mathscr{A}$ then $\vdash \Box\mathscr{A}$.

Proof. (a) Let $\vdash \mathscr{A}$. Since $\vdash A \to ((\mathscr{B} \to \mathscr{B}) \to \mathscr{A})$ implies $\vdash (\mathscr{B} \to \mathscr{B}) \to \mathscr{A}$, by R3, we have $\vdash \Box(\mathscr{B} \to \mathscr{B}) \to \Box\mathscr{A}$. By A8, $\vdash \Box(\mathscr{B} \to \mathscr{B})$, i.e., $\vdash \Box\mathscr{A}$.
(b) Let $\vdash \mathscr{A} \to \mathscr{B}$, then $\vdash \Box(\mathscr{A} \to \mathscr{B})$. By A7, $\vdash \Box\mathscr{A} \to \Box\mathscr{B}$. But $\vdash \mathscr{A} \to \mathscr{A}$, so $\vdash \Box(\mathscr{A} \to \mathscr{A})$ holds.

Assertion 6. $\vdash \Box \forall x\mathscr{A} \to \forall x\Box\mathscr{A}$.

Proof.

$\vdash \forall x\mathscr{A} \to \mathscr{A}$	(A6.a)
$\vdash \Box\forall x\mathscr{A} \to \Box\mathscr{A}$	(by R3)
$\vdash \forall x\Box\forall x\mathscr{A} \to \forall x\Box\mathscr{A}$	(by R2 and A4)
$\vdash \Box\forall x\mathscr{A} \to \forall x\Box\forall x\mathscr{A}$	(A5)
$\vdash \Box\forall x\mathscr{A} \to \forall x\Box\mathscr{A}$	(by classical theorems)

Assertion 7. $\vdash \Diamond \forall x\mathscr{A} \to \forall x\Diamond\mathscr{A}$.

Proof.

$\vdash \sim\mathscr{A} \to \exists x \sim\mathscr{A}$	(from A6.a)
$\vdash \Box\sim\mathscr{A} \to \Box\exists x \sim\mathscr{A}$	(by R3)
$\vdash \Diamond\forall x\mathscr{A} \to \Diamond\mathscr{A}$	(by classical theorems)
$\vdash \Diamond\forall x\mathscr{A} \to \forall x\Diamond\mathscr{A}$	(similarly).

§ 4. Completeness theorems

A set of formulae α is consistent if for every $\mathscr{A}_1, \ldots, \mathscr{A}_n \in \alpha$, $\sim(\mathscr{A}_1 \wedge \ldots \wedge \mathscr{A}_n)$ is not a theorem.
We introduce the following notation: $\alpha^+ = \{\mathscr{A} : \Box\mathscr{A} \in \alpha\}$.

Theorem 3. Let α be a consistent set of formulae and assume $\alpha^+ \neq \emptyset$. If $\Diamond\mathscr{B} \in \alpha$, then $\alpha^+ \cup \{\mathscr{B}\}$ is consistent.

Proof. Assume the contrary, i.e. $\alpha^+ \cup \{\mathscr{B}\}$ is not consistent. Then there exist $\mathscr{A}_1, \ldots, \mathscr{A}_n \in \alpha^+$ such, that $\vdash \sim(\mathscr{A}_1 \wedge \ldots \wedge \mathscr{A}_n \wedge \mathscr{B})$. It means that $\vdash \mathscr{A}_1 \wedge \ldots \wedge \mathscr{A}_n \to \sim\mathscr{B}$ (using the hypothesis $\alpha^+ \neq \emptyset$ and that $\vdash \sim\mathscr{A}$ implies $\vdash \sim(\mathscr{A} \wedge \mathscr{C})$ for arbitrary $\mathscr{C}$). By R3, we obtain $\vdash \Box\mathscr{A}_1 \wedge \ldots \wedge \Box\mathscr{A}_n \to \Box\sim\mathscr{B}$, i.e. $\vdash \sim(\Box\mathscr{A}_1 \wedge \ldots \wedge \Box\mathscr{A}_n \wedge \Diamond\mathscr{B})$. This contradicts the assumption, that α is consistent.
If α is a set of formulae, then let us denote the set of variables occuring in α by $\Pi(\alpha)$.

Definition. Let α be a set of formulae. α is complete if the following conditions are satisfied:

(i) α is consistent;
(ii) If $\mathscr{A}$ contains variables from $\Pi(\alpha)$ only, then either $\mathscr{A}\in\alpha$ or $\sim\mathscr{A}\in\alpha$;
(iii) Let $\mathscr{A}$ contain variables from $\Pi(\alpha)$ only and let x be the only variable occuring free in $\mathscr{A}$. If $\exists x\mathscr{A}\in\alpha$ then there exists a such that $a\in\Pi(\alpha)$ and a is free for x, moreover $\mathscr{A}[x/a]\in\alpha$.

Theorem 4. Let α be a complete set of formulae. Then

(i) $\mathscr{A}\in\alpha$ and $\mathscr{A}\to\mathscr{B}\in\alpha$ imply $\mathscr{B}\in\alpha$;
(ii) $\mathscr{A}\wedge\mathscr{B}\in\alpha$ if and only if $\mathscr{A}\in\alpha$ and $\mathscr{B}\in\alpha$;
(iii) $\forall x\mathscr{A}\in\alpha$ if and only if for every $a\in\Pi(\alpha)$ free for x we have $\mathscr{A}[x/a]\in\alpha$, where x is the only variable occuring free in $\mathscr{A}$.
(iv) If $\alpha^+\cup\{\mathscr{A}\}$ is consistent and $\mathscr{A}$ contains variables from $\Pi(\alpha)$ only, then $\Diamond\mathscr{A}\in\alpha$.

Proof. (i) If $\mathscr{B}\notin\alpha$ then $\sim\mathscr{B}\in\alpha$ by completeness. But it means α is not consistent since $\vdash\sim(\mathscr{A}\wedge(\mathscr{A}\to\mathscr{B})\wedge\sim\mathscr{B})$.

(ii) Since $\vdash\sim((\mathscr{A}\wedge\mathscr{B})\wedge\sim\mathscr{A})$, $\vdash\sim((\mathscr{A}\wedge\mathscr{B})\wedge\sim\mathscr{B})$ and $\vdash\sim(\mathscr{A}\wedge\mathscr{B}\wedge\sim\sim(\mathscr{A}\wedge\mathscr{B}))$ hold, it is trivial.
(iii) $\vdash\sim(\forall x\mathscr{A}\wedge\sim\mathscr{A}[x/a])$, so if $\forall x\mathscr{A}\in\alpha$, then $\mathscr{A}[x/a]\in\alpha$. Conversely, if $\forall x\mathscr{A}\notin\alpha$, then $\sim\forall x\mathscr{A}\in\alpha$ by completeness, i.e. $\exists x\sim\mathscr{A}\in\alpha$. Thus there exists $a\in\Pi(\alpha)$ such that $\sim\mathscr{A}[x/a]\in\alpha$, i.e., $\mathscr{A}[x/a]\notin\alpha$.
(iv) If $\Diamond\mathscr{A}\notin\alpha$, then $\Box\sim\mathscr{A}\in\alpha$, by completeness, and $\sim\mathscr{A}\in\alpha^+$. But it means $\alpha^+\cup\{\mathscr{A}\}$ is not consistent since $\vdash\sim(\mathscr{A}\wedge\sim\mathscr{A})$.

Theorem 5. If α is consistent, then there exists a complete β such that $\alpha\subseteq\beta$.

Proof. It is easy by using the following three lemmata since we can assume that the set of variables has enough elements.

Lemma A. Let α be consistent. Then at least one of $\alpha\cup\{\mathscr{A}\}$ and $\alpha\cup\{\sim\mathscr{A}\}$ will also be consistent.

Proof. Suppose both $\alpha\cup\{\mathscr{A}\}$ and $\alpha\cup\{\sim\mathscr{A}\}$ are inconsistent; that means there exist $\mathscr{B}_1,\ldots,\mathscr{B}_n\in\alpha$ for which $\vdash\sim(\mathscr{B}_1\wedge\ldots\wedge\mathscr{B}_n\wedge\mathscr{A})$ and $\vdash\sim(\mathscr{B}_1\wedge\ldots\wedge\mathscr{B}_n\wedge\wedge\sim\mathscr{A})$. Then $\vdash\sim(\mathscr{C}\wedge\mathscr{A})\to(\sim(\mathscr{C}\wedge\sim\mathscr{A})\to\sim\mathscr{C})$ entails $\vdash\sim(\mathscr{B}_1\wedge\ldots\wedge\mathscr{B}_n)$ so α is inconsistent. This completes the proof of Lemma A.

Lemma B. If $a\notin\Pi(\alpha\cup\{\mathscr{A}\})$ and α is consistent, then $\alpha\cup\{\exists x\mathscr{A}\to\mathscr{A}[x/a]\}$ is also consistent.

Proof. Suppose the contrary. Then there exist $\mathscr{A}_1,\ldots,\mathscr{A}_n\in\alpha$ such that $\vdash\sim(\mathscr{A}_1\wedge\ldots\wedge\mathscr{A}_n\wedge(\exists x\mathscr{A}\to\mathscr{A}[x/a]))$. By applying R2 we arrive at $\vdash\sim(\mathscr{A}_1\wedge\ldots\ldots\wedge\mathscr{A}_n\wedge\sim(\exists x\mathscr{A}\wedge\forall a\sim\mathscr{A}[x/a]))$. Since $\vdash\sim(\exists x\mathscr{A}\wedge\forall a\sim\mathscr{A}[x/a])$, we have $\vdash\sim(\mathscr{A}_1\wedge\ldots\wedge\mathscr{A}_n)$ which contradicts the assumptions. This completes the proof of Lemma B.

Lemma C. If α_n is consistent and $\alpha_n \subseteq \alpha_{n+1}$ for every n, then $\bigcup_{n=1}^{\infty} \alpha_n$ is also consistent.

Proof. Trivial.

Definition. The system of sets of formulae M is said to be complete if the following conditions are satisfied:

(i) Each $\alpha \in M$ is complete.
(ii) If $\alpha \in M$, $\alpha^+ \neq \emptyset$ and $\Diamond \mathscr{A} \in \alpha$, then there exists $\beta \in M$ such that $\alpha^+ \cup \cup \{\mathscr{A}\} \subseteq \beta$.
(iii) If the logic has equality symbol, then

(a) If $\alpha, \beta \in M$, $a \in \Pi(\alpha) \cap \Pi(\beta)$, then there exist natural numbers $n, m \geqq 0$ and sets of formulae $\alpha_0, \ldots, \alpha_n, \beta_0, \ldots, \beta_m \in M$ such that $\alpha_0 = \beta_0$, $\alpha_n = \alpha$, $\beta_m = \beta$, $a \in \Pi(\alpha_0), \alpha_i^+ \neq \emptyset$ and $\alpha_i^+ \subseteq \alpha_{i+1} (i = 0, \ldots, n-1), \beta_i^+ \neq \emptyset$ and $\beta_i^+ \subseteq \beta_{i+1} (i = 0, \ldots, m-1)$.
(b) If $\alpha_i^+ \neq \emptyset$, $\alpha_i^+ \subseteq \alpha_{i+1}$ $(i = 0, \ldots, n-1)$, $\beta_i^+ \neq \emptyset$, $\beta_i^+ \subseteq \beta_{i+1}$ $(i = 0, \ldots, m-1)$, $\alpha_n = \beta_m$, then there exist $\gamma_0, \ldots, \gamma_k \in M$ such that $\gamma_i^+ \neq \emptyset$ and $\gamma_i^+ \subseteq \gamma_{i+1} (i = 0, \ldots, k-1)$ and either $\gamma_0 = \alpha_0$, $\gamma_k = \beta_0$ or $\gamma_0 = \beta_0$, $\gamma_k = \alpha_0$ are true.

Theorem 6. If α is complete, then there exists a complete system of sets of formulae M such that $\alpha \in M$.

Proof. Let $M_0 = \{\alpha\}$. Assume that M_n is a set of complete sets of formulae. Let $\beta \in M_n$, $\beta^+ \neq \emptyset$, $\Diamond \mathscr{A} \in \beta$. Then, to $(\beta, \mathscr{A})$ we associate a set of variables. This set is disjoint from $\bigcup_{\gamma \in M_n} \Pi(\gamma)$ and different pairs have disjoint associated sets of variables. There exists a complete set γ such that $\gamma \in M_{n+1}$, $\beta^+ \cup \{\mathscr{A}\} \subseteq \gamma$ and $\Pi(\gamma) \setminus \Pi(\beta)$ is associated to the pair $(\beta, \mathscr{A})$. It is trivial, that $\bigcup_{n=0}^{\infty} M_n$ is a complete system of sets of formulae.

Theorem 7. *(Completeness Theorem.)* Let us suppose that a simple non-stable and equality free modal logic is given. If a formula $\mathscr{A}$ cannot be derived then $\sim \mathscr{A}$ is satisfiable.

Proof. We can assume without loss of generality, that no free variable occurs in $\mathscr{A}$. Since not $\vdash \mathscr{A}$, we have not $\vdash \mathscr{A}$ i.e., $\{\sim \mathscr{A}\}$ is consistent. There exists a complete set of formulae α and a complete system of sets of formulae M such that $\sim \mathscr{A} \in \alpha$ and $\alpha \in M$.

Let us define the following notions: $N = \{\beta : \beta \in M$ and $\beta^+ \neq \emptyset\}$; if $\beta, \gamma \in M$, then $\beta R \gamma \leftrightarrow ((\beta^+ \subseteq \gamma$ and $\beta^+ \neq \emptyset)$ or $(\beta^+ = \emptyset$ and $\gamma = \beta))$; $|P(\beta)| = \{\tau :$ all variables occuring in τ are from $\Pi(\beta)\}$; $f_{P(\beta)}(\tau_1, \ldots, \tau_n) = f(\tau_1, \ldots, \tau_n)$ where $\tau_1, \ldots, \tau_n \in |P(\beta)|$; $r_{P(\beta)}(\tau_1, \ldots, \tau_n) \leftrightarrow r(\tau_1, \ldots, \tau_n) \in \beta$ if $\tau_1, \ldots, \tau_n \in |P(\beta)|$. It is easy to see that $\langle M, N, \alpha, R, P \rangle$ is a simple model.

Let k be an interpretation and $\varkappa$ the corresponding valuation. The following two assertions can easily be proved by a simple induction.

If $\tau \in \mathscr{H}_k(\beta)$, $\beta \in M$, then $\varkappa(\tau, \beta) = \tau[x_1, \ldots, x_n / k(x_1), \ldots, k(x_n)]$ where $x_1, \ldots, x_n$ are all the variables occuring in τ, and $\tau[x_1, \ldots, x_n / \tau_1, \ldots, \tau_n]$ is the result of the substitutions $[x_1/\tau_1], \ldots, [x_n/\tau_n]$ executed simultaneously.

If $\mathscr{B} \in \mathscr{H}_k(\beta)$, $\beta \in M$ and $x_1, \ldots, x_n$ are all the variables occuring in $\mathscr{B}$, then $\beta \models \mathscr{B}[k] \Leftrightarrow \mathscr{B}[x_1, \ldots, x_n/k(x_1), \ldots, k(x_n)] \in \beta$. Hence, if for every a, $k(a)=a$, then $\alpha \models \sim\mathscr{A}[k]$. Let us suppose that $\mathscr{B}$ contains variables only from $\Pi(\beta)$, where β is a complete set.

If $\vdash \mathscr{B}$, then $\mathscr{B} \in \beta$, since in the opposite case we have $\sim\mathscr{B} \in \beta$, i.e. β is inconsistent.

K1. If $\Box(\mathscr{B} \to \mathscr{B})$ is an axiom, then for every $\beta \in M$, $\Box(\mathscr{B} \to \mathscr{B}) \in \beta$ provided no variable occurs in $\mathscr{B}$. Thus, $\beta^+ \neq \emptyset$ and $N = M$.

K2. Let β be an arbitrary formula for which $\mathscr{B} \in \beta^+$. From $\Box\mathscr{B} \in \beta$ and $\Box\mathscr{B} \to \mathscr{B} \in \beta$ we infer $\mathscr{B} \in \beta$, i.e. $\beta^+ \subseteq \beta$, $\beta R \beta$.

K3. Let $\beta R \gamma$ and $\gamma R \delta$, moreover $\mathscr{B} \in \beta^+$. Then $\Box\mathscr{B} \in \beta$, $\Box\mathscr{B} \to \Box\Box\mathscr{B} \in \beta$, so $\Box\Box\mathscr{B} \in \beta$. By definition of R, $\Box\mathscr{B} \in \gamma$ and $\mathscr{B} \in \delta$ follow. We obtain $\beta R \delta$. Let $\beta \in N$, then for some $\mathscr{B}$, $\Box\mathscr{B} \in \beta$. If $\beta R \gamma$ then $\Box\mathscr{B} \in \gamma$, so $\gamma \in N$.

This completes the proof of Theorem 7.

In what follows we assume that a non-stable modal logic with equality is given. Let M be a complete system of sets of formulae, let N and R be defined analogouosly to the ones in the proof of the previous theorem. We denote the reflexive and transitive closure of R by $\bar{R}$.

By these notations we redefine the third clause of the last definition in the following simple way:

(iii)′ If the logic has equality symbol, then

(a) If $\alpha, \beta \in M$, $a \in \Pi(\alpha) \cap \Pi(\beta)$ then there exists $\gamma \in M$ such that $\gamma \bar{R} \alpha, \gamma \bar{R} \beta$ and $a \in \Pi(\gamma)$.

(b) If $\alpha, \beta, \gamma \in M$, $\alpha \bar{R} \gamma$ and $\beta \bar{R} \gamma$, then either $\alpha \bar{R} \beta$ or $\beta \bar{R} \alpha$ is true; in other words, $\bar{R}$ is trichotom on the set $\{\alpha: \alpha \bar{R} \gamma\}$.

We prove some simple assertions:

Assertion 8. If $\beta \bar{R} \gamma$, then $\Pi(\beta) \subseteq \Pi(\gamma)$.

Proof. Trivial.

Assertion 9. If $\beta \bar{R} \gamma$ and $a, b \in \Pi(\beta)$, then $\mathscr{I}(a, b) \in \beta \Leftrightarrow \mathscr{I}(a, b) \in \gamma$.

Proof. If $\beta \bar{R} \gamma$, then $\Box\mathscr{I}(a, b) \vee \Diamond \mathscr{A} \in \beta$. If $\beta \bar{R} \gamma$ and $\gamma \neq \beta$, then $\beta^+ \neq \emptyset$, e.g. $\Box \sim \mathscr{A}_1 \in \beta$. If $\mathscr{A}$ is replaced by $\mathscr{A}_1$, then $\Box \sim \mathscr{A}_1 \to \Box\mathscr{I}(a, b) \in \beta$ and so $\Box\mathscr{I}(a, b) \in \beta$. That means $\mathscr{I}(a, b) \in \beta^+$ and $\mathscr{I}(a, b) \in \gamma$ by induction.

If $\sim\mathscr{I}(a, b) \in \beta$, then by an analogouos argument we can obtain the other direction. This completes the proof the Assertion 9.

Let a, b be two variables. $a \equiv b$ if and only if there exist $\alpha, \beta, \gamma \in M$ and $c \in \Pi(\gamma)$ such that $\gamma \bar{R} \alpha, \gamma \bar{R} \beta, \mathscr{I}(a, c) \in \alpha$ and $\mathscr{I}(b, c) \in \beta$. Obviously, $\equiv$ is a reflexive and symmetric relation. We shall prove that it is transitive, as well.

Assertion 10. If $\gamma \bar{R} \alpha, \gamma \bar{R} \beta, c \in \Pi(\gamma), \mathscr{I}(c, a) \in \alpha$ and $\mathscr{I}(a, b) \in \beta$, then $\mathscr{I}(c, b) \in \beta$.

Proof. It is clear, that $a \in \Pi(\alpha) \cap \Pi(\beta)$. By (iii)′ there exists $\delta \in M$ such that $a \in \Pi(\delta)$ and $\delta \bar{R} \alpha, \delta \bar{R} \beta$. Also by this definition we have either $\gamma \bar{R} \delta$ or $\delta \bar{R} \gamma$. By Assertion 8 either $a, c \in \Pi(\delta)$ or $a, c \in \Pi(\gamma)$ and so either $\mathscr{I}(c, a) \in \delta$ or $\mathscr{I}(c, a) \in \gamma$. In both cases $\mathscr{I}(c, a) \in \beta$ by Assertion 9. Then $\mathscr{I}(c, b) \in \beta$ by transitivity of equality.

Assertion 11. Let $\alpha_i \bar{R} \beta_i$ $(i=1, \ldots, n)$ and $\alpha_{i+1} \bar{R} \beta_i$ $(i=1, \ldots, n)$. There exists a k such that $1 \leqq k \leqq n+1$ and $\alpha_k \bar{R} \alpha_i$ for every i $(1 \leqq i \leqq n+1)$; furthermore, there exists an l such that $1 \leqq l \leqq n+1$ and $\alpha_i \bar{R} \alpha_l$ for every i $(1 \leqq i \leqq n+1)$.

Proof. Readily follows by definitions.

Assertion 12. The relation $\equiv$ is transitive.

Proof. Assume $a \equiv b$ and $b \equiv c$. Then, there exist d, e and $\alpha_1, \alpha_3, \beta_1, \beta_2, \beta_3, \beta_4$ such that $\alpha_1 \bar{R} \beta_1$, $\alpha_1 \bar{R} \beta_2$, $\mathscr{J}(d, a) \in \beta_1$, $\mathscr{J}(d, b) \in \beta_2$, $d \in \Pi(\alpha_1)$, $\alpha_3 \bar{R} \beta_3$, $\alpha_3 \bar{R} \beta_4$, $\mathscr{J}(e, b) \in \beta_3$, $\mathscr{J}(e, c) \in \beta_4$, $e \in \Pi(\alpha_3)$. By (iii)′ (a), there exists an α_2 such that $\alpha_2 \bar{R} \beta_2$, $\alpha_2 \bar{R} \beta_3$, $b \in \Pi(\alpha_2)$. By the previous assertion, for some i, $\alpha_i \bar{R} \alpha_j$ $(j=1, 2, 3)$ and $\alpha_i \bar{R} \beta_j$ $(j=1, 2, 3, 4)$. It is known, that $d \in \Pi(\alpha_1)$, $b \in \Pi(\alpha_2)$, $e \in \Pi(\alpha_3)$. Let f be that variable among (of d, b, e), which is in $\Pi(\alpha_i)$. We have $\mathscr{J}(a, d) \in \beta_1$, $\mathscr{J}(d, b) \in \beta_2$, $\mathscr{J}(b, e) \in \beta_3$, $\mathscr{J}(e, c) \in \beta_4$. By Assertion 10, we obtain $\mathscr{J}(a, f) \in \beta_1$, $\mathscr{J}(f, c) \in \beta_4$, that means $a \equiv c$.

Assertion 13. If $a, b \in \Pi(\beta)$ and $a \equiv b$, then $\mathscr{J}(a, b) \in \beta$.

Proof. Let $\beta_1 = \beta_4 = \beta$. Since $a \equiv b$ there exist $\beta_2, \beta_3, \alpha_2 \in M$ and $c \in \Pi(\alpha_2)$ such that $\mathscr{J}(a, c) \in \beta_2$, $\mathscr{J}(b, c) \in \beta_3$, $\alpha_2 \bar{R} \beta_2$, $\alpha_2 \bar{R} \beta_3$. By (iii)′ (a) there exist α_1 and α_3 for which $\alpha_1 \bar{R} \beta_1$, $\alpha_1 \bar{R} \beta_2$, $d \in \Pi(\alpha_1)$, $\alpha_3 \bar{R} \beta_3$, $\alpha_3 \bar{R} \beta_4$, $\beta \in \Pi(\alpha_3)$. By Assertion 11, there exists an i such that $\alpha_i \bar{R} \beta_j$ $(j=1, 2, 3, 4)$. Obviously $\mathscr{J}(a, a) \in \beta_1$, $\mathscr{J}(a, c) \in \beta_2$, $\mathscr{J}(c, b) \in \beta_3$, $\mathscr{J}(b, b) \in \beta_4$. Let d be that variable among a, b, c which is in $\Pi(\alpha_i)$. Applying Assertion 10, we have $\mathscr{J}(a, d) \in \beta_1$ and $\mathscr{J}(d, b) \in \beta_4$, i.e. $\mathscr{J}(a, b) \in \beta$.

Theorem 8. *(Completeness Theorem.)* Let a non-stable modal logic with equality be given. If $\mathscr{A}$ is not derivable, then $\sim \mathscr{A}$ is satisfiable.

Proof. Let M be a complete system of sets of formulae, N, R as defined in the proof of Theorem 7. Let us define P by the following causes: for $\beta \in M$ $|P(\beta)| = \{\bar{a} : a \in \Pi(\beta)\}$, where $\bar{a} = \{b : a \equiv b\}$; if $a_1, \ldots, a_n, a \in \Pi(\beta)$, then

$$f_{P(\beta)}(\bar{a}_1, \ldots, \bar{a}_n) = \bar{a} \Leftrightarrow \mathscr{J}(f(a_1, \ldots, a_n), a) \in \beta;$$

(By definition of completeness, this function is defined and it is unique by last assertion.) $r_{P(\beta)}(\bar{a}_1, \ldots, \bar{a}_n) \Leftrightarrow r(a_1, \ldots, a_n) \in \beta$. For an arbitrary $\beta \in M$, $\langle M, N, \beta, R, P \rangle$ is a model.

If A 6 is an axiom of the given logic, then this model is simple. We have to prove that

if $a_1, \ldots, a_n, a \in \Pi(\beta), b_1, \ldots, b_n, b \in \Pi(\gamma), a_1 \equiv b_1, \ldots, a_n \equiv b_n$, $\mathscr{J}(f(a_1, \ldots, a_n), a) \in \beta$ and $\mathscr{J}(f(b_1, \ldots, b_n), b) \in \gamma$, then $a \equiv b$. Let $1 \leqq i \leqq n$ be given. By definition of $\equiv$ and clause (iii)′ (a) we can assume that $\beta_1 = \beta$, $\beta_n = \gamma$, $\alpha_1 \bar{R} \beta_1$, $\alpha_1 \bar{R} \beta_2$, $\alpha_2 \bar{R} \beta_2$, $\alpha_2 \bar{R} \beta_3$, $\alpha_3 \bar{R} \beta_3$, $\alpha_3 \bar{R} \beta_4$, $a_i \in \Pi(\alpha_1)$, $\mathscr{J}(a_i, c) \in \beta_2$, $c \in \Pi(\alpha_2)$, $\mathscr{J}(c, b_i) \in \beta_3$, $b_i \in \Pi(\alpha_3)$. Let γ_i denote the first element among $\alpha_1, \alpha_2, \alpha_3$ under $\bar{R}$. Using methods from proofs of Assertion 9—13, we get $c_i \in \Pi(\gamma_i)$, $\mathscr{J}(a_i, c_i) \in \beta$, $\mathscr{J}(c_i, b_i) \in \gamma$, $\gamma_i \bar{R} \beta$, $\gamma_i \bar{R} \gamma$. Since for every i, $\gamma_i \bar{R} \beta$, applying (iii)′ (b) we obtain that there exists an i such that $\gamma_j \bar{R} \gamma_i$ for every j. By Assertion 8, for this i we have $c_1, \ldots, c_n \in \Pi(\gamma_i)$. So there exists a c for which $\mathscr{J}(f(c_1, \ldots, c_n), c) \in \gamma_i$. Generalizing the method used in proof of Assertion 9, we arrive to $\mathscr{J}(f(c_1, \ldots, c_n), c) \in \beta$ and $\mathscr{J}(f(c_1, \ldots, c_n), c) \in \gamma$. From $\mathscr{J}(f(a_1, \ldots, a_n), a) \in \beta$ and $\mathscr{J}(f(b_1, \ldots, b_n), b) \in \gamma$, it follows that $\mathscr{J}(a, c) \in \beta$, $\mathscr{J}(b, c) \in \gamma$ and so $a \equiv b$.

Let k be an interpretation and $\varkappa$ the corresponding valuation. If for a variable x, $k(x)\in|P(\beta)|$, then $k(x)\cap\Pi(\beta)\neq\emptyset$. Let $x^*\in k(x)\cap\Pi(\beta)$. We extend the operation * for arbitrary expressions: $\mathscr{K}^*=\mathscr{K}[x_1, \ldots, x_n/x_1^*, \ldots, x_n^*]$, where $x_1, \ldots, x_n$ are all the variables occuring in $\mathscr{K}$. By a simple induction, the following statements are easy to prove:

(i) $\varkappa(\tau, \beta)=\bar{a}$ and $a\in\Pi(\beta)\Leftrightarrow\mathscr{I}(\tau^*, a)\in\beta$;
(ii) If $\mathscr{A}$ contains variables from $\Pi(\beta)$ only then $\beta\models\mathscr{A}[k]\leftrightarrow\mathscr{A}^*\in\beta$.

From (ii) the theorem follows.

§ 5. Connections with classical logics

Let us suppose that a modal logic is given; i.e., the sets of relation symbols, function symbols and set of variables are fixed. We also suppose that the following symbols do not occur in these sets: o, s, n, r, p, i, z, z'. Furthermore the parameter $\mathscr{T}$ of this logic is fixed. Also we know if this logic is simple, stable or so.

Now we define a classical theory. The language of this theory contains all the relation symbols and function symbols of the modal language but if a symbol has arity m in the modal language we use it with arity $m+1$ in the classical one. Also we shall use the following symbols: o: 0-ary function symbol, s and n both unary relation symbols, r, p, i all of them are binary relation symbols, and two new variables: z and z'.

We define a mapping [ ] from the set of modal expressions into the set of classical ones:

(i) if x is a variable, then $[x]=x$;
(ii) $[f(\tau_1, \ldots, \tau_m)]=f([\tau_1], \ldots, [\tau_m], z)$ if f is an m-ary function symbol in the modal language, $\tau_1, \ldots, \tau_m$ are terms;
(iii) $[r(\tau_1, \ldots, \tau_m)]=r([\tau_1], \ldots, [\tau_m], z)$ if r is an m-ary relation symbol in the modal language, $\tau_1, \ldots, \tau_m$ are terms; in particular $[\mathscr{I}(\tau_1, \tau_2)]=i([\tau_1],[\tau_2])$;
(iv) $[\sim\mathscr{A}]=\sim[\mathscr{A}]$; $[\mathscr{A}\wedge\mathscr{B}]=[\mathscr{A}]\wedge[\mathscr{B}]$;
(v) $[\forall x\mathscr{A}]=\forall x(p(x, z)\rightarrow[\mathscr{A}])$;
(vi) $[\Box\mathscr{A}]=\forall z'(r(z, z')\rightarrow[\mathscr{A}][z/z'])\wedge n(z)$.

Let $\mathscr{A}^*=p(x_1, z)\wedge\ldots\wedge p(x_m, z)\wedge[\mathscr{A}]$, where $x_1, \ldots, x_m$ are all the free variables of $\mathscr{A}$.

Let M be a classical model in which the following formulae are valid: $s(o)$; $s(z)\rightarrow\exists x p(x, z)$; $p(x, z)\wedge r(z, z')\rightarrow p(x, z')$; $s(z)\wedge r(z, z')\rightarrow s(z')$; $s(z)\rightarrow p(f(x_1, \ldots, x_m, z), z)$ for every function symbol.

Let $0=o_M$, $S=\{a: a\in|M|$ and $s_M(a)\}$, $N=\{a: a\in S$ and $n_M(a)\}$, $aRb\Leftrightarrow a, b\in S$ and $r_M(a, b)$, $|P(a)|=\{b: p_M(b, a)\}$, if $a\in S$, for $a_1, \ldots, a_m\in|P(a)|$, $f_{P(a)}(a_1, \ldots, a_m)=f_M(a_1, \ldots, a_m, a)$ and $q_{P(a)}(a_1, \ldots, a_m)\Leftrightarrow q_M(a_1, \ldots, a_m, a)$.

It is obvious, that by these definitions $\langle S, N, O, R, P\rangle$ is a modal model.

Let k be an interpretation for M such that k associates an element of S to z and z', and k associates an element of $\bigcup_{a\in S}|P(a)|$ to every variable other then z or z'. It is clear, that k is also an interpretation for $\langle S, N, O, R, P\rangle$. Let the corresponding valuations be K in M and $\varkappa$ in $\langle S, N, O, R, P\rangle$.

Theorem 9. Let τ be a term, $\mathscr{A}$ a formula and suppose z, z' do not occur in them. Then

(i) $\varkappa(\tau, k(z)) = K([\tau])$, provided $\varkappa(\tau, k(z))$ is defined;
(ii) $k(z) \models \mathscr{A}[k] \Leftrightarrow M \models \mathscr{A}^*[k]$.

Proof. The easy induction is left to the reader.

Now we give the inverse of the mapping

$$M \to \langle S, N, O, R, P \rangle.$$

Let $\langle S, N, O, R, P \rangle$ be an arbitrary modal model. We define

$$|M| = S \cup \left(\bigcup_{a \in S} |P(a)|\right); \quad o_M = 0; \quad s_M(a) \Leftrightarrow a \in S;$$

$$n_M(a) \Leftrightarrow a \in N; \quad p_M(a, b) \Leftrightarrow b \in S \quad \text{and} \quad a \in |P(b)|;$$

$$r_M(a, b) \Leftrightarrow a, b \in S \quad \text{and} \quad aRb, \; i_M(a, b) \Leftrightarrow a = b;$$

$$f_M(a_1, \ldots, a_m, a) = \begin{cases} f_{P(a)}(a_1, \ldots, a_m), & \text{if } a \in S \text{ and } a_1, \ldots, a_m \in |P(a)| \\ \text{arbitrary element of } |P(a)| & \text{otherwise;} \end{cases}$$

$$q_M(a_1, \ldots, a_m, a) \Leftrightarrow q_{P(a)}(a_1, \ldots, a_m).$$

Theorem 10. Let A, B classical models and $|A| \subseteq |B|$. There are the same symbols in the languages of A and B the only exeption is s, which is used only in the language of B as a unary relation symbol. Let

$$f_A(a_1, \ldots, a_m) = f_B(a_1, \ldots, a_m), \quad \text{if} \quad a_1, \ldots, a_m \in |A|;$$

$$q_A(a_1, \ldots, a_m) \Leftrightarrow q(a_1, \ldots, a_m), \quad \text{if} \quad a_1, \ldots, a_m \in |A|;$$

$$s_B(b) \Leftrightarrow b \in |A|.$$

We define the mapping $\mathscr{H}$ on the set of formulae not containing the symbol s:
$\mathscr{H}(\mathscr{A}) = \mathscr{A}$, if $\mathscr{A}$ is an atom;
$\mathscr{H}(\mathscr{A} \wedge \mathscr{B}) = \mathscr{H}(\mathscr{A}) \wedge \mathscr{H}(\mathscr{B})$
$\mathscr{H}(\sim \mathscr{A}) = \sim \mathscr{H}(\mathscr{A})$
$\mathscr{H}(\forall x \mathscr{A}) = \forall x(s(x) \to \mathscr{H}(\mathscr{A}))$.
Let k be an interpretation the range of which is in $|A|$. Then

$$A \models \mathscr{A}[k] \Leftrightarrow B \models \mathscr{H}(\mathscr{A})[k].$$

Proof. Trivial.

If $M \to \langle S, N, O, R, P \rangle$ is the mapping defined above, $\mathscr{T}$ is the parameter of the logic, then we have:
(i) the modal model has property $\mathscr{T}$ if and only if $M \models \mathscr{H}(\mathscr{T})$;
(ii) the modal model is simple if and only if for every function symbol f

$$M \models \forall x_1 \ldots \forall x_m \forall z \forall z' (p(x_1, z) \wedge p(x_1, z') \wedge \ldots \wedge p(x_m, z) \wedge p(x_m, z') \to$$

$$\to i(f(x_1, \ldots, x_m, z), f(x_1, \ldots, x_m, z'))).$$

(iii) the modal model is stable if and only if

$$M \models \forall z \forall z' (s(z) \wedge s(z') \to \forall x (p(x, z) \to p(x, z'))).$$

Let $\mathscr{A}$ be a modal formula and assume a modal logic is given. The formula $\mathscr{A}$ is satisfiable (in modal sense) if and only if the following formula is classically satisfiable: $s(o) \wedge (s(z) \to \exists x p(x, z)) \wedge \forall x \forall z \forall z' \ (p(x, z) \wedge r(z, z') \to p(x, z')) \wedge \wedge \forall z \forall z' (s(z) \wedge r(z, z') \to s(z')) \wedge \forall z \forall x_1, \ldots, \forall x_{m_1} (s(z) \to p(f_1(x_1, \ldots, x_{m_1}, z), z)) \wedge \ldots \ldots \wedge \forall z \forall x_1 \ldots \forall x_{m_k} (s(z) \to p(f_k(x_1, \ldots, x_{m_k}, z), z)) \wedge \mathscr{P} \wedge \mathscr{S} \wedge \mathscr{H}(\mathscr{T}) \wedge \mathscr{A}^*[z/o]$, where $f_1, \ldots, f_k$ are all the function symbols occuring in $\mathscr{A}$; $\mathscr{P}$ is true if the logic is not simple, otherwise it is the following:

$$\bigwedge_{j=1}^{k} (\forall x_1, \ldots \forall x_{m_j} \forall z \forall z' (p(x_1, z) \wedge p(x_1, z') \wedge \ldots \wedge p(x_{m_j}, z) \wedge p(x_{m_j}, z) \to i(f(x_1, \ldots, x_{m_j}, z), f(x_1, \ldots, x_{m_j}, z'))));$$

$\mathscr{S}$ is true if the logic is not stable, otherwise it is the formula

$$\forall z \forall z' (s(z) \wedge s(z') \to \forall x (p(x, z) \to p(x, z'))).$$

References

[1] Feys, R., *Modal logics*, Paris, 1965.
[2] Schütte, K., *Vollständige Systeme modaler und intuitionistischer Logik*, Springer Varlag, Berlin, 1970.

(Received Oct. 2, 1978)

INDEX — TARTALOM

ISSN 0324—721 X

Felelős szerkesztő és kiadó: Gécseg Ferenc
A kézirat a nyomdába érkezett: 1979. június hó
Megjelenés: 1979. december hó
Példányszám: 1025. Terjedelem: 5,42 (A/5) ív
Készült monószedéssel, íves magasnyomással
az MSZ 5601 és az MSZ 5602—55 szabvány szerint
79-3166 — Szegedi Nyomda — F. v.: Dobó József igazgató