Transactions of the Institute for Informatics and Automation Problems of NAS RA.
Collected reports of participants of JAF-23.

THE FINITE-STATE RECOGNIZABILITY OF SEQUENCES OF INTEGERS[1]

K. V. Shahbazyan and Yu. H. Shoukourian

We consider the possibilities of on-line tessellation automata (OTA) to recognize the properties of sequences of natural numbers. Our constructions provide a simple, and yet powerful, way to automatical construction of OTA that recognize the properties defined by EMSO-sentences over information matrices of sequences.

INTRODUCTION

The aim of this paper is to investigate the possibilities of on-line finite state recognizability of sequences of natural numbers and to derive an implementation mode of automata that, processing the matrix code of sequences, captures the properties definable in existentional monadic second-order logic over information matrices of sequences. As a tool for solving this problem we consider a cellular acceptor, namely, two-dimensional on-line finite state tessellation automata (OTA) introduced by K. Inoue and A.Nakamura [1,2]. This automaton is an array of identical finite-state automata in which a transition wave passes once diagonally across the array.

We identify a sequence $w = w_1, ..., w_n$ of naturals with a square picture $MC(w)$ whose top row contains the sequence of unary codes of $w_i, i = 1, ..., n$ with zeroes as separators. So to any set W of natural's sequences we associate a language $\{MC(w)|w \in W\}$ of pictures, and transfer the notion of recognizability from picture languages to sequences. As is known[3], the recognizability by OTA of the set W is equivalent to the definability of the language $\{MC(w)|w \in W\}$ by the formulas over signature σ_{MC} with unary predicate.

We investigate two ways for defining the sets of sequences by formulas over signatures equipped with binary predicates defined over natural's that makes them more convenient

[1]This research is supported by ISTC - 823 grant and INTAS - 2001-447.

tool for describing the sets of sequences. In both cases we indicate classes of formulas defining such sets W that $\{MC(w)|w \in W\}$ is recognizable by on-line tessellation automata.

The first way is to define the sets of sequences by the sentences of Existentional Monadic Second-Order (EMSO) logic over the information matrices (IM) of sequences. Information matrix $IM(w)$ contains the values of some binary predicates for every pair (w_i, w_j). If the set W of natural's sequences is defined by EMSO-logic over information matrices, then this set is recognizable by a tessellation automaton processing the MC codes. More over there exists an effective algorithm that constructs the tessellation automaton that recognizes the set $\{MC(w)|w \in W\}$ (Theorem 1). This algorithm can be employed as a tool for OTA's automatical construction by EMSO- formula. An easy generalization of our results can be adapted to the problem of verification of message sequence charts [6,7].

The second way to define sequences is by the formulas of First-Order (FO) logic equipped with binary predicates. However, this brings to the recognizability for very restricted class of formulas (Theorems 2 and 3).

§1. PRELIMINARIES. DEFINITIONS AND NOTATIONS.

1.1. Languages of sequences of natural numbers.

In the following we write „*sequence*" as a short hand for „*sequence of natural numbers*". Let $w = w_1, ..., w_n$ be a sequence. A *code* of sequence w is a string $C(w) = 1^{w_1}01^{w_2}0...$-$01^{w_n}0$ over alphabet $\Sigma = \{0, 1\}$. A *matrix code* of w (denoted by $MC(w)$) is a square matrix over Σ that contains $C(w)$ as the first row. The entries of other rows are zeroes. We identify a sequence w with its matrix code $MC(w)$.

A matrix code $MC(w)$ is a model over the signature $\sigma_{MC} = (S_1, S_2, R)$ with universe $dom_{MC}(w) = \{1, ..., n + \sum_{i=1}^{n} w_i\}^2$, where S_1 and S_2 are horizontal and vertical successor relations of the points of $dom_{MC}(w)$, R is an unary predicate that gives the set of points of $dom_{MC}(w)$, where

$$R((i,j)) = \begin{cases} 1 & \text{if } i - 1,\ 1 + k + \sum_{i=1}^{k} w_i \leq j < k + \sum_{i=1}^{k+1} w_i,\ k = 0, ..., n-1, \\ 0 & \text{in other cases.} \end{cases}$$

Identifying a sequence language W with the language $MC(W) = \{MC(w)|w \in W\}$, we call the language W *MC-recognizable* if the language of matrices $MC(W)$ is recognizable by OTA. As is known [3], a language of matrices (pictures) is recognizable by an OTA iff it is definable in EMSO-logic over σ_{MC}.

We consider two other signatures for representing languages of natural's sequences.

Let binary predicates $P_1, \dots, P_r$ over naturals be fixed. We associate to any sequence $w = w_1, \dots, w_n$ a square (n,n)-matrix $IM(w)$ over Σ^r (where $\Sigma = \{0,1\}$) whose entries are r-bit vectors $(R^1, ..., R^r)$. If $R^1((i,j)) = P_1(w_i, w_j)$, ... , $R^r((i,j)) = P_r(w_i, w_j)$. This matrix is called an *information matrix* (denoted by $IM(w)$) of the sequence w. Notice that information matrix contains values of all considered predicates for every pair (w_i, w_j). $IM(w)$ can be viewed as a model over signature $\sigma_{IM}(P_1, ..., P_r) = (S_1, S_2, R^1, ..., R^r)$ with universe $dom_{IM}(w) = \{1, \dots, n\}^2$, and unary predicates $R^1, ..., R^r$. So a language W of sequences can be defined by a logical formula over $\sigma_{IM}(P_1, ..., P_r)$.

On the other hand if $P_1, \dots, P_r$ are fixed, we can represent a sequence $w = w_1, ..., w_n$ as a model $\Pi(w) = <\{1, ..., n\}, \pi, S, Q_1, ..., Q_r>$ over signature $\sigma_\Pi(P_1, ..., P_r) = (\pi, \leq, Q_1, ..., Q_r)$, with universe $dom_\Pi(w) = \{1, ..., n\}$ where $\leq$ is successor relation over $dom_\Pi(w)$, π is a function $\pi(i) = w_i$, and $Q_l(i,k) = P_l(\pi(i), \pi(k)) = P_l(w_i, w_k)$, $(l = 1, ..., r)$, are binary predicates. So a language W of sequences can be defined by a logical formula over $\sigma_\Pi(P_1, \dots, P_r)$.

Example 1. Let the sequence w be $w = (2,3)$. Then $C(w) = (1101110)$ and the matrix code of $MC(w)$ is (7,7)-matrix.

$$MC(w) = \begin{array}{|ccccccc|} \hline 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \end{array}$$

The information matrix $IM(w)$ can be defined only if the predicates are fixed.

Let us consider one binary predicate over naturals $P(x,y) = \begin{cases} 1 & \text{if } x = y, \\ 0 & \text{if } x \neq y. \end{cases}$

Then $IM(w) = $

1	0
0	1

As an example of sequence languages consider the language $W_=$ that consists of the sequences $w = w_1, \ldots, w_n$ where $w_i = w_j$ for $i, j \in \{1, ..., n\}$.

Then $\sigma_\Pi(P_=) = (\pi, S, Q_1)$, where the value of the binary predicate $Q_1(i,j)$ can be computed by means of the function π and the predicate $P_=$ by the formula $Q_1(i,j) = P_=(\pi(i), \pi(j))$. Here $i, j \in dom_\Pi(w) = \{1, ..., n\}$. The formula for the language $W_=$ over signature $\sigma_\Pi(P_=)$ is $\forall i, j\ Q_1(i,j)$.

Now pass to the signature $\sigma_{IM}(P_=) = (S_1, S_2, R_1)$ where R_1 is unary predicate: $R_1(x) = P_=(w_i, w_j)$ if $x = (i,j)$ is a point of $dom_{IM}(w) = \{1, ..., n\}^2$. The same language $W_=$ can be defined over $\sigma_{IM}(P_=)$ by the formula $\forall x\ R_1(x)$.

Relating to the signature σ_{MC}, we see that to write a formula defining $W_=$ is very difficult. We give here only an intuition about how to proceed. We define OTA where the computation of the matrix $IM(w)$ in a sparce form is combined with the computation of the realizability of the formula $\forall x R_1(x)$. This OTA is equivalent to an EMSO-formula over signature $\sigma_{MC}(R)$. ◇

Further we intend to investigate the relations among the classes of languages of sequences defined over mentioned three signatures.

1.2. Matrices.

In the following we consider a variant of pictures called k-bit matrices (for some $k > 0$), i.e., matrices with entries in alphabet Σ^k where $\Sigma = \{0, 1\}$. (The expressions *row, column, top* etc. are interpreted as in terminology of matrices). Given a (m,n)-matrix M over Σ^k we denote the components of k-bit vector $M(i,j)$ by superscripts: $M(i,j) = M^1(i,j) \ldots M^k(i,j)$, $(1 \le i \le m, 1 \le j \le n)$.

Let M be a square (m,m)-matrix over Σ^k where $k \ge 2$, and $T(M) = \{i_1, ..., i_s\}$ be a set of integers such that $0 < i_1 < i_2 < ... < i_s = m$ and

$$M^1(i,j) = \begin{cases} 0 & \text{if } j \in I(M), \\ 1 & \text{in other cases,} \end{cases}$$

$$M^2(i,j) = \begin{cases} 0 & \text{if } i \in I(M), \\ 1 & \text{in other cases.} \end{cases}$$

$M =$

11	01	11	11	11	01
10	00	10	10	10	00
11	01	11	11	11	01
11	01	11	11	11	01
11	01	11	11	11	01
10	00	10	10	10	00

Figure 1: Cell-like matrix M.

A *separating row (column)* of M is the i-th row (corresp. column) of M if $i \in T(M)$. An element $M(i,j)$ is a *crossing* if $i,j \in T(M)$. A matrix M is *cell-like* if $T(M) \neq \emptyset$. A *cell* of matrix M is a block of M bounded by the columns i_k, i_{k+1} and the rows i_j, i_{j+1} (denoted by $M < i_{k+1}, i_{l+1} >$), i.e.,

$$M(i,j) \in M < i_k, i_l > \Leftrightarrow i_{k-1} < i < i_k,\ i_{l-1} < j < i_l.$$

(m,m)-matrices M_1 and M_2 are *similar* if $T(M_1) = T(M_2)$.

Example 2. A cell-like matrix M over Σ^2 is represented in the Figure 1.

The cells are shown by bold-face lines. The crossings are shaded. ◇

Suppose M is a cell-like (m,m)-matrix and $card(T(M)) = s$. A *skeleton* of cell-like matrix M is a (s,s)-matrix $S(M)$ composed of all crossings of M, i.e.,

$$S(M)(l,k) = M(i_l, i_k),\ \ i_l, i_k \in T(M),\ \ 1 \leq l,k \leq s.$$

A *projection* $\mathcal{P}_\tau(M)$ of matrix M by the tuple of integers $\tau = (\tau_1, ..., \tau_r)$, where $0 < \tau_1 < ... < \tau_r \leq n$ is a matrix M_1 such that $M_1(i,j) = M^{\tau_1}(i,j)...M^{\tau_r}(i,j)$ for every $(i,j) \in dom(M)$. The same notation will be used for the projection of vectors.

Let M_1 and M_2 be (m,n)-matrices over alphabets Σ^{l_1} and Σ^{l_2}. The *element-wise concatenation* of M_1 and M_2 (denoted by $M = M_1 \odot M_2$) is the (m,n)-matrix M over $\Sigma^{l_1+l_2}$ where $M(i,j)$ is the concatenations of vectors $M_1(i,j)$ and $M_2(i,j)$.

1.3. OTA and EMSO-logic.

We consider two-dimensional on-line tessellation automata (OTA) introduced by K Inoue and A.Nakamura [1,2]. This class of automata recognizes the matrix (picture) languages definable by existentional expressions of monadic second-order logic (EMSO-logic) [3].

A non-deterministic two-dimensional on-line tessellation automaton (OTA) is defined by the tuple $\mathcal{A} = (X, Q, q_0, F, \delta)$ where

X is a finite input alphabet,

Q is a finite set of states,

$q_0 \in Q$ is an initial state,

$F \subset Q$ is the set of accepting states,

$\delta \subset Q \times Q \times A \to Q$ is the set of transitions.

Let M be a (m,n)-matrix over alphabet X. The *run* of the automaton $\mathcal{A} = (X, Q, q_0, F, \delta)$ over M is a (m,n)-matrix M_1 over alphabet Q such that the transitions

$q_0, q_0, M(1,1) \to M_1(1,1)$,

$M_1(1, j-1), q_0, M(1,j) \to M_1(1,j),\ j = 1, ..., n$,

$q_0, M_1(i-1, 1), M(i,1) \to M_1(i,1),\ i = 1, ..., m$,

$M_1(i, j-1), M_1(i-1, j), M(i,j) \to M_1(i,j),\ i = 1, ..., m,\ j = 1, ..., n$

are the transitions of automaton $\mathcal{A}$, i.e., belong to δ. The automaton $\mathcal{A}$ *accepts* the matrix M if there exists a run M_1 such that $M_1(m,n) \in F$. We say that automaton $\mathcal{A}$ *maps* the matrix M to the matrix M_1 (denoted by $M_1 = \mathcal{A}(M)$).

We need the following three operations over OTA.

Let $\mathcal{A}_1 = (X, Q, q_0, F_1, \delta_1)$ and $\mathcal{A}_2 = (X, T, q_0, F_2, \delta_2)$ be OTA.

1) The *disjunction* of automata $\mathcal{A}_1$ and $\mathcal{A}_2$ (denoted by $\mathcal{A}_1 \vee \mathcal{A}_2$) is defined as $\mathcal{A}_1 \vee \mathcal{A}_2 = (X, Q \times T,\ (q_0, q_0),\ (F_1 \times Q) \cup (Q \times F_2),\ \delta(\delta_1, \delta_2))$ where
$\delta(\delta_1, \delta_2) = \{(q_1, t_1), (q_2, t_2), \sigma \to (q_3, t_3) |\ q_1, q_2, \sigma \to q_3 \in \delta_1,\ \ t_1, t_2, \sigma \to t_3 \in \delta_2\}$. (1) Here $\sigma \in X,\ \ q_1, q_2, q_3 \in Q,\ \ t_1, t_2, t_3 \in T$.
OTA $\mathcal{A}_1 \vee \mathcal{A}_2$ accepts a matrix M iff either $\mathcal{A}_1$ or $\mathcal{A}_2$ accepts M. The automaton $\mathcal{A}_1 \vee \mathcal{A}_2$ maps M into $M_1 = (\mathcal{A}_1 \vee \mathcal{A}_2)(M) = \mathcal{A}_1(M) \odot \mathcal{A}_2(M)$.

2)The *conjunction* of $\mathcal{A}_1$ and $\mathcal{A}_2$ (denoted by $\mathcal{A}_1 \wedge \mathcal{A}_2$) is defined as $\mathcal{A}_1 \wedge \mathcal{A}_2 = (X,\ Q \times T,\ (q_0, q_0), F_1 \times F_2,\ \delta(\delta_1, \delta_2))$ where $\delta(\delta_1, \delta_2)$ is defined by (1).
OTA $\mathcal{A}_1 \wedge \mathcal{A}_2$ accepts a matrix M iff both $\mathcal{A}_1$ and $\mathcal{A}_2$ accept M. The automaton $\mathcal{A}_1 \wedge \mathcal{A}_2$ transforms M into $M_1 = (\mathcal{A}_1 \wedge \mathcal{A}_2)(M) = \mathcal{A}_1(M) \odot \mathcal{A}_2(M)$, i.e., $\mathcal{A}_1 \wedge \mathcal{A}_2$ and $\mathcal{A}_1 \vee \mathcal{A}_2$ define the same transformation.

3)The *composition* of $\mathcal{A}_1$ and $\mathcal{A}_2$ (denoted by $\mathcal{A}_1 \circ \mathcal{A}_2$) is defined as $\mathcal{A}_2 \circ \mathcal{A}_1 = (X,\ Q \times T,\ (q_0, q_0),\ F_1 \times F_2,\ \delta)$ where δ is defined by (2):

$$\delta = \{(q_1, t_1), (q_2, t_2), \sigma \to (q_3, t_3) \mid q_1, q_2, \sigma \to q_3 \in \delta_1,\ \ t_1, t_2, q_3 \to t_3 \in \delta_2\} \quad (2)$$

OTA $\mathcal{A}_2 \circ \mathcal{A}_1$ accepts a matrix M iff $\mathcal{A}_1$ accepts M, and $\mathcal{A}_2$ accepts $M_1 = \mathcal{A}_1(M)$. OTA $\mathcal{A}_2 \circ \mathcal{A}_1$ maps M to the matrix $\overline{M} = (\mathcal{A}_2 \circ \mathcal{A}_1)(M) = M_1 \odot \mathcal{A}_2(M_1)$.

1.4. Recognizable binary predicates.

We begin with almost literally citing of [3, section 9]. Consider matrix languages over one-letter alphabet. In this case a matrix can be identified with a pair (m, n) of natural numbers where m is the number of rows of the matrix and n is the number of columns. On the other hand, every binary predicate P defined over the set of pairs of natural numbers is actually a set of such pairs and can be viewed as a matrix language over one-letter alphabet.

Given a predicate P, the matrix language *associated to* P is defined as

$$L_P = \{M \mid M \text{ is } (m,n)\text{-matrix},\ n, m \in N,\ P(m,n) = 1\}.$$

A predicate P is *recognizable* if the associated matrix language L_P is recognizable by OTA. For example, the following predicates are recognizable: $=$, $\leq$, $\geq$, $n = m + c$, $n = cm$, $n = m^c$, $n = c^m$, $n : m$, where c is a constant integer.

§2. THE PROBLEM.

In this paper we are interested in on-line finite state MC-recognizability of the sequence languages. As a recognizer we consider OTA. It is evident that the class of sequence languages definable by EMSO-formulas over the signature σ_{MC} is the widest class of sequence languages recognizable by OTA. Our problem is: what classes of formulas over signatures σ_{Π} and σ_{IM} define the MC-recognizable languages. We present below the investigation of this problem and an effective algorithm that constructs the corresponding OTA. In this connection, our arguments are based on the existence of effective algorithm that for every EMSO-formula constructs corresponding OTA [3,4] and vice versa.

Now we sketch our approach to the problem. Suppose that the predicates $P_1, \ldots, P_r$ and their complements $\overline{P_1}, \ldots, \overline{P_r}$ are represented by corresponding OTA's $\mathcal{A}_{P_1}, \mathcal{A}_{\overline{P_1}}, \ldots, \mathcal{A}_{P_r}, \mathcal{A}_{\overline{P_r}}$, and the formula $\Phi_{\sigma_{IM}}$ is represented by OTA $\mathcal{A}_{\Phi_{\sigma_{IM}}}$.

We proceed as follows.

First we define OTA $\mathcal{A}_{prel}$ that carries out preliminary processing of $MC(w)$. For matrix code $MC(w)$ of the sequence $w = w_1, ..., w_n$ OTA $\mathcal{A}_{prel}$ constructs a cell-matrix $M = \mathcal{A}_{prel}(MC(w))$ where each cell $M < i_k, i_j >$ is (w_k, w_j)-matrix.

Observe that for each pair (w_k, w_j) the value of predicate $P_s(w_k, w_j)$, $s = 1, ..., r$ can be computed on the cell $M < i_k, i_j >$ by means of $\mathcal{A}_{P_s}, \mathcal{A}_{\overline{P_s}}$. To do this we define the operator O_{cell} that constructs an OTA $\mathcal{L}$ which imitates on each cell of M the runs of all OTA's $\mathcal{A}_{P_s}, \mathcal{A}_{\overline{P_s}}$ $s = 1, ..., r$. Besides $\mathcal{L}$ transfers values of predicates computed on the cell $M < i_k, i_j >$ to crossing $M(i_k, i_j)$. So the information matrix $IM(w)$ is found in skeleton matrix $S(\mathcal{L}(M))$.

Then the operator O_{skel} constructs OTA $O_{skel}(\Phi_{\sigma_{IM}})$ that imitates on the matrix $S(\mathcal{L}(M))$ the run of $\mathcal{A}_{\Phi_{\sigma_{IM}}}$ over $IM(w)$.

At last OTA $\mathcal{D}$ realizes a composition of $\mathcal{A}_{prel}, \mathcal{L}, \mathcal{A}_{\Phi_{\sigma_{IM}}}$ that processes $MC(w)$. OTA $\mathcal{D}$ recognizes $MC(w)$ iff $\mathcal{A}_{\Phi_{\sigma_{IM}}}$ recognizes $IM(w)$.

§3. THE COMPUTATION OF INFORMATION MATRIX $IM(w)$.

Let the predicates $P_1, \dots P_r$ be given. Suppose that these predicates and their complements $\overline{P_1}, ..., \overline{P_r}$ are recognizable by OTA. This assumption implies the existence of OTA $\mathcal{A}_{P_1}, \mathcal{A}_{\overline{P_1}}, ..., \mathcal{A}_{P_r}, \mathcal{A}_{\overline{P_r}}$ recognizing these predicates.

Lemma. There exists a nondeterministic OTA $\mathcal{L}$ and a tuple of integers $\tau = (\tau_1, ..., \tau_r)$ such that for every sequence $w = w_1, ..., w_n$ OTA $\mathcal{L}$ maps the matrix code $MC(w)$ to a cell matrix M such that

$$\mathcal{L}(\text{MC}(w)) = M,$$

where

$$IM(w) = \mathcal{P}_\tau(S(M)). \tag{3}$$

Proof: Without loss of generality we assume that the automata $\mathcal{A}_{P_1}, \mathcal{A}_{\overline{P_1}}, ..., \mathcal{A}_{P_r}, \mathcal{A}_{\overline{P_r}}$ have common dimension ν of the state vectors. First of all let us describe several auxiliary automata and $\mathcal{A}_{prel}$.

1°. Deterministic automaton $\mathcal{A}_{down} = (\Sigma, \Sigma, q_0, \{0\}, \delta_{down})$ where $\Sigma = \{0, 1\}$ copies the first row of input matrix in all remaining rows, i.e., if $M_1 = \mathcal{A}_{down}(M)$ where M is (m, n)-matrix, then $M_1^1(i, j) = M^1(1, j)$ for $j = 1, ..., n$.

The automaton $\mathcal{A}_{down}$ accepts any matrix iff it is the matrix code of a sequence.

2°. Automaton $\mathcal{A}_{column} = (\Sigma, \Sigma, q_0, \Sigma, \delta_{column})$ nondeterministically guesses the states of the leftmost column and copies it in all other columns, i.e., if $M_1 = \mathcal{A}_{column}(M)$ where M is a (m,n)-matrix, then $M_1^1(i,j) = M_1^1(i,1)$ for $i = 1, ..., m$.

The automaton $\mathcal{A}_{column}$ accepts all matrices.

3°. Deterministic automaton $\mathcal{A}_{diag} = (\Sigma^2, \Sigma^2, q_0, \{(11)\}, \delta_{diag})$ accepts a square (m,m)-matrix M iff $M(i,i) = (0,0)$ or $(1,1)$. If $M_1 = \mathcal{A}_{diag}(M)$ then

$$M_1(i,j) = \begin{cases} (11), & \text{if } i = j \\ (10), & \text{if } i < j \\ (01), & \text{if } i > j \end{cases}$$

4°. Nondeterministic automaton $\mathcal{A}_{prel} = \mathcal{A}_{diag} \circ (\mathcal{A}_{down} \wedge \mathcal{A}_{column}) = (\Sigma, \Sigma^4, q_0, \{0\} \times \Sigma \times \{11\}, \delta)$, accomplishes a single-valued *preliminary* transformation of matrix codes. $\mathcal{A}_{prel}$ maps $MC(w)$ to a cell-like (N,N)-matrix M where $N = n + \sum_{i=1}^{n} w_i$, and $T(M) = \{k + \sum_{i=1}^{k} w_i\}_{k=1,...,n}$ such that:

1) $M^1(i,j) = M^1(1,j) = MC^1(1,j), \quad 1 \le i,j \le N$, i.e., the first components of rows are copies of the first row of $MC(w)$, i.e., copies of $C(w)$;

2)$M^1(1,j) = M^2(j,1), \quad 1 \le j \le N,$

i.e., the second components of the first columns copy the first components of the first row. Automaton $\mathcal{A}_{column}$ must guess the column equal to $C(w)$. If other column is guessed then $\mathcal{A}_{prel}$ has no run.

3) $M^2(i,j) = M^2(i,1), \quad 1 \le i,j \le N,$

i.e., the second components of columns are copies of the second components of the first column which are guessed by $\mathcal{A}_{column}$;

4) $M^3(i,j)$, $M^4(i,j)$ are obtained by $\mathcal{A}_{diag}$:

$$M^3(i,j) = \begin{cases} 0 & \text{if } i < j, \\ 1 & \text{if } i \ge j, \end{cases}$$

$$M^4(i,j) = \begin{cases} 0 & \text{if } i > j, \\ 1 & \text{if } i \le j. \end{cases}$$

Notice that each cell $M < i_l, i_k >$ is (w_l, w_k)-matrix over the alphabet $\{(11)\} \times \Sigma^2$.

5°. Define now an operator O_{cell} that transforms arbitrary automaton $\mathcal{A}$ to the automaton $O_{cell}(\mathcal{A})$. The automaton $O_{cell}(\mathcal{A})$ imitates the runs of the automaton $\mathcal{A}$ on all cells of input matrix M.

Let $\mathcal{A} = (\Sigma^l, \Sigma^\nu, q_0, F, \delta_\mathcal{A})$. The automaton $O_{cell}(\mathcal{A}) = (\Sigma^{l+2}, \Sigma^{\nu+2}, q_0, \Sigma^{\nu+2}, \delta_{cell})$ maps a cell-like matrix M over alphabet Σ^{l+2} to the matrix M_1 over the alphabet $\Sigma^{\nu+2}$ so that M and M_1 are similar and

$$\mathcal{A}(\mathcal{P}_{3,\dots,l+2}(M < i_s, i_k >)) = \mathcal{P}_{3,\dots,\nu+2}(M_1 < i_s, i_\nu >), \quad i_s, i_k \in T(M).$$

The automaton $O_{cell}(\mathcal{A})$ realizes an arbitrary run (i.e., not necessarily the accepting run) of $\mathcal{A}$ on each cell of M.

The automaton $O_{cell}(\mathcal{A})$ accepts a cell-like matrix M iff $\mathcal{A}$ has a run on the indicated projection of each cell of M. The automaton $O_{cell}(\mathcal{A})$ is deterministic if $\mathcal{A}$ is deterministic.

Example 3. Consider the automaton $\mathcal{A}$ that realizes the following map of $(2,2)$-matrices:

$$\mathcal{A}\left(\begin{array}{|c|c|}\hline x_{11} & x_{12} \\ \hline x_{21} & x_{22} \\ \hline\end{array}\right) = \begin{array}{|c|c|}\hline q_{11} & q_{12} \\ \hline q_{21} & q_{22} \\ \hline\end{array} \qquad \mathcal{A}\left(\begin{array}{|c|c|}\hline x_{14} & x_{15} \\ \hline x_{24} & x_{25} \\ \hline\end{array}\right) = \begin{array}{|c|c|}\hline q_{14} & q_{15} \\ \hline q_{24} & q_{25} \\ \hline\end{array} \qquad (4.a)$$

$$\mathcal{A}\left(\begin{array}{|c|c|}\hline x_{41} & x_{42} \\ \hline x_{51} & x_{52} \\ \hline\end{array}\right) = \begin{array}{|c|c|}\hline q_{41} & q_{42} \\ \hline q_{51} & q_{52} \\ \hline\end{array} \qquad \mathcal{A}\left(\begin{array}{|c|c|}\hline x_{44} & x_{45} \\ \hline x_{54} & x_{55} \\ \hline\end{array}\right) = \begin{array}{|c|c|}\hline q_{44} & q_{45} \\ \hline q_{54} & q_{55} \\ \hline\end{array} \qquad (4.b)$$

$x_{ij} \in \Sigma^l, \qquad q_{i,j} \in \Sigma^\nu.$

The transformation by $O_{\text{cell}}(\mathcal{A})$ of cell-like matrix M is represented in Fig. 2.

The automaton $O_{cell}(\mathcal{A})$ replaces the projection of each cell of matrix M by the cell defined in (4.a), (4.b).◇

The set of transitions of $O_{cell}(\mathcal{A})$ consists of five sets:

$$\delta_{cell} = \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4 \cup \Delta_5$$

$$\Delta_1 = \{q_0, q_0, (11x) \to (11q) \mid q_0, q_0, x \to q \in \delta_\mathcal{A}\} \cup$$

$$\cup \{q_0, (11q), (11x) \to (11q') \mid q_0, q, x \to q' \in \delta_\mathcal{A}\} \cup$$

$$\cup \{(11q), q_0, (11x) \to (11q') \mid q, q_0, x \to q' \in \delta_\mathcal{A}\} \cup$$

$$\cup \{(11q), (11q'), (11x) \to (11q'') \mid q, q', x \to q'' \in \delta_\mathcal{A}\}.$$

$M =$

$11x_{11}$	$11x_{12}$	$01x_{13}$	$11x_{14}$	$11x_{15}$	$01x_{16}$
$11x_{21}$	$11x_{22}$	$01x_{23}$	$11x_{24}$	$11x_{25}$	$01x_{26}$
$10x_{31}$	$10x_{32}$	$00x_{33}$	$10x_{34}$	$10x_{35}$	$00x_{36}$
$11x_{41}$	$11x_{42}$	$01x_{43}$	$11x_{44}$	$11x_{45}$	$01x_{46}$
$11x_{51}$	$11x_{52}$	$01x_{53}$	$11x_{54}$	$11x_{55}$	$01x_{56}$
$10x_{61}$	$10x_{62}$	$00x_{63}$	$10x_{64}$	$10x_{65}$	$00x_{66}$

$\downarrow O_{cell}(\mathcal{A})$

$M_1 =$

$11q_{11}$	$11q_{12}$	$01q'_0$	$11q_{14}$	$11q_{15}$	$01q'_0$
$11q_{21}$	$11q_{22}$	$01q'_0$	$11q_{24}$	$11q_{25}$	$01q'_0$
$10q'_0$	$10q'_0$	$00q'_0$	$10q'_0$	$10q'_0$	$00q'_0$
$11q_{41}$	$11q_{42}$	$01q'_0$	$11q_{44}$	$11q_{45}$	$01q'_0$
$11q_{51}$	$11q_{52}$	$01q'_0$	$11q_{54}$	$11q_{55}$	$01q'_0$
$10q'_0$	$10q'_0$	$00q'_0$	$10q'_0$	$10q'_0$	$00x'_0$

Figure 2: The transformation of M by the automaton $O_{cell}(\mathcal{A})$.

The transitions from the set Δ_1 are used for imitating $\mathcal{A}$ on the left top cell $M_1 < i_1, i_1 >$.

Let $q'_0 \in \Sigma^\nu$ be a fixed symbol, for example $q'_0 = 0^\nu$. This symbol is used for computing the states of separating rows and columns of M_1.

$$\Delta_2 = \{(11q), q_0, (01x) \to (01q'_0) \mid q \in \Sigma^\nu,\ x \in \Sigma^l\} \cup$$
$$\cup \{(11q), (01q'_0), (01x) \to (01q'_0) \mid q \in \Sigma^\nu,\ x \in \Sigma^l\} \cup$$
$$\cup \{(11q), (00q'_0), (01x) \to (01q'_0) \mid q \in \Sigma^\nu,\ x \in \Sigma^l\}.$$

The transitions of Δ_2 compute the states of separating columns in M_1, except of the states of crossings. These states are $(01q'_0)$.

$$\Delta_3 = \{q_0, (11q), (10x) \to (10q'_0) \mid q \in \Sigma^\nu,\ x \in \Sigma^l\} \cup$$
$$\cup \{(10q'_0), (11q), (10x) \to (10q'_0) \mid q \in \Sigma^\nu,\ x \in \Sigma^l\} \cup$$
$$\cup \{(00q'_0), (11q), (10x) \to (10q'_0) \mid q \in \Sigma^\nu,\ x \in \Sigma^l\}.$$

The transitions of Δ_3 compute the states of separating rows in M_1, except the states of crossings. These computed states are $(10q_0')$.

$$\Delta_4 = \{(10q_0'), (01q_0'), (00x) \to (00q_0') \mid x \in \Sigma^\nu\}.$$

The transitions of Δ_4 compute the states of crossings. These computed states are $(00q_0')$.

$$\begin{aligned}\Delta_5 = \{&(01q_0'), (10q_0'), (11x) \to (11q);\ (01q_0'), q_0, (11x) \to (11q);\\ &q_0, (10q_0'), (11x) \to (11q) \mid q_0, q_0, x \to q \in \delta_A\} \cup\\ &\cup\{(11q), (11q'), (11x) \to (11q'') \mid q, q', x \to q'' \in \delta_A\} \cup\\ &\cup\{(11q), (10q_0'), (11x) \to (11q') \mid q, q_0, x \to q' \in \delta_A\} \cup\\ &\cup\{(01q_0'), (11q), (11x) \to (11q') \mid q_0, q, x \to q' \in \delta_A\}.\end{aligned}$$

The transitions of Δ_5 imitate the automaton $\mathcal{A}$ on all cells of M_1 except the sell $M_1 < i_1, i_1 >$.

The operator O_{cell} is fully defined.

Now we return to the proof of the Lemma. Let a predicate P and its complement $\overline{P}$ be OTA-recognizable, i.e., there exist two automata

$$\mathcal{A}_P = (\Sigma^2, \Sigma^\nu, q_0, F_P, \delta_P),$$
$$\mathcal{A}_{\overline{P}} = (\Sigma^2, \Sigma^\nu, q_0, F_{\overline{P}}, \delta_{\overline{P}}),$$

which accept (m,n)-matrix over Σ^2 if $P(m,n) = 1$ (corresp. $\overline{P}(m,n) = 1$). Without loss of generality we assume that automata $\mathcal{A}_P$ and $\mathcal{A}_{\overline{P}}$ have a run on each input matrix.

Consider the automaton $\mathcal{U}_P = \mathcal{A}_P \vee \mathcal{A}_{\overline{P}} = (\Sigma^2,\ \Sigma^{2\nu},\ q_0,\ (F_P \times \Sigma^\nu) \cup (\Sigma^\nu \times F_{\overline{P}}),\ \delta_{\mathcal{U}_P})$, and the automaton $O_{cell}(\mathcal{U}_P) = (\Sigma^4, \Sigma^{2\nu+2}, q_0, \Sigma^{2\nu+2}, \delta_{cell})$ that has a run $M_1 = O_{cell}(\mathcal{U}_P)(M)$ on the every cell-like matrix M. This run computes for each cell $M < i_k, i_l >$, where $i_k, i_l \in T(M)$, the values of the predicate $P(w_k, w_l)$ and its complement $\overline{P}(w_k, w_l)$. The accepting (or rejecting) values of the cell are ν-dimensional vectors: $t_1 = \mathcal{P}_{(3,\dots,\nu+2)}(M_1(i_k - 1, i_l - 1))$ and $t_2 = \mathcal{P}_{(\nu+3,\dots,2\nu+2)}(M_1(i_k - 1, i_l - 1))$. In every final point of cells of M there are the following alternatives:

$t_1 \in F_P, t_2 \notin F_{\overline{P}}$, i.e., $\mathcal{A}_P$ accepts the cell $M < i_k, i_l >$ and $P(w_k, w_l) = 1$.

$t_1 \notin F_P, t_2 \in F_{\overline{P}}$, i.e., $\mathcal{A}_{\overline{P}}$ accepts the cell $M < i_k, i_l >$ and $\overline{P}(w_k, w_l) = 1$.

$t_1 \notin F_P, t_2 \notin F_{\overline{P}}$, i.e., both $\mathcal{A}_P$ and $\mathcal{A}_{\overline{P}}$ do not accept the cell $M < i_k, i_l >$.

In the last case the value of predicate P is not obtained and the run on M must be interrupted. However, notice that for each M there exists such a run of $O_{cell}(\mathcal{U}_P)$ that successfully computes the values of both predicates for all cells of M.

Now we define an auxiliary automaton $\mathcal{L}_P$ that (if it is applied to $M_1 = O_{cell}(\mathcal{U}_P)(M)$) computes and transports the values of P and $\overline{P}$ from the final position of each cell to the nearest crossing.

Let us consider the following function f_P defined on pairs $t_1, t_2 \in \Sigma^\nu$:

$$f_P(t_1,t_2) = \begin{cases} (10) & \text{if } t_1 \in F_P, t_2 \notin F_{\overline{P}}, \\ (01) & \text{if } t_1 \notin F_P, t_2 \in F_{\overline{P}}, \\ (00) & \text{if } t_1 \notin F_P, t_2 \notin F_{\overline{P}}, \\ (11), & \text{if } t_1 \in F_P, t_2 \in F_{\overline{P}}, \end{cases}$$

and the automaton

$$\mathcal{L}_P = (\Sigma^{2\nu+2}, \Sigma^2, q_0, \Sigma^2, \delta_{\mathcal{L}_P}).$$

Here $\delta_{\mathcal{L}_P} = \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4$

$$\begin{aligned} \Delta_1 &= \{q_1, q_2, (11t_1t_2) \to f_P(t_1,t_2) \mid q_1, q_2 \in \Sigma^2 \cup \{q_0\}\}, \\ \Delta_2 &= \{q_1, q_2, (01t_1t_2) \to (q_1) \mid q_1, q_2 \in \Sigma^2 \cup \{q_0\}\}, \\ \Delta_3 &= \{q_1, q_2, (10t_1t_2) \to (10) \mid q_1, q_2 \in \Sigma^2 \cup \{q_0\}\}, \\ \Delta_4 &= \{q_1, (01), (00t_1t_2) \to (01) \mid q_1 \in \Sigma^2 \cup \{q_0\}\} \cup \\ &\cup \{q_1, (10), (00t_1t_2) \to (10) \mid q_1 \in \Sigma^2 \cup \{q_0\}\}. \end{aligned}$$

The automaton $\mathcal{L}_P$ transforms a cell-like matrix M in the following way. Notice that the elements of the cells of M are $(2\nu+2)$-dimensional vectors and therefore can be represented as $(11t_1t_2)$ where t_1, t_2 are ν-dimensional vectors.

The transitions of Δ_1 replace all elements of every cell by $f_P(t_1,t_2)$.

The transitions of Δ_2 transport the elements of the right column of every cell to the nearest right separating column. So the state of the point (i_k-1, i_l-1) gets into the point (i_k-1, i_l) for $i_k, i_l \in I(M)$.

The transitions of Δ_3 compute the elements of separating rows except the crossings.

The transitions of Δ_4 transport the state of (i_k-1, i_l) to the point (i_k, i_l), i.e., to the corresponding crossing.

Observe that if the transported value is $(0,0)$ then there is no correspondent transition in Δ_4 and this run on the matrix M is interrupted.

Consider the composition

$$\mathcal{L}_{cell}(P) = \mathcal{L}_P \circ O_{cell}(\mathcal{U}_P) = (\Sigma^4, \Sigma^{2\nu+4}, q_0, \Sigma^{2\nu+4}, \delta').$$

It is evident, that for every sequence w if $M = \mathcal{A}_{prel}(MC(w))$ and $M_1 = \mathcal{L}_{cell}(P)(M)$, then each crossing of M contains the value of the predicate P as $(3+2\nu)$-th component, i.e.,

$$M_1^{3+2\nu}(i_k, i_l) = \begin{cases} 1 & \text{if } P(w_k, w_l) = 1, \\ 0 & \text{if } P(w_k, w_l) = 0. \end{cases}$$

Consider the automaton $\mathcal{L} = (\mathcal{L}_{cell}(P_1) \wedge ... \wedge \mathcal{L}_{cell}(P_r)) \circ \mathcal{A}_{pred} = (\Sigma, \Sigma^{2r\nu+4r+4}, q_0, \Sigma^{2r\nu+4r+4}$ and the tuple of integers

$$\tau(\nu, r) = (2\nu+7, 4\nu+11, ..., 2r\nu+4r+3). \tag{5}$$

Evidently, for each sequence w the automaton $\mathcal{L}$ maps the matrix code $MC(w)$ to the cell-like matrix $M = \mathcal{L}(MC(w))$ where the required condition (3) from Lemma is satisfied: $\mathcal{P}_{\tau(\nu,r)}(S(M)) = IM(w)$. The Lemma is proved. ◇

Thus for fixed predicates $P_1 \dots P_r$ and for every $w = w_1 \dots w_n$:

1. $\mathcal{L}$ processes the matrix code $MC(w)$;

2 The matrix $IM(w)$ with respect to $P_1 \dots P_r$ is found in the sparce form in the skeleton of cell-like matrix $\mathcal{L}(MC(w))$.

§4. MC-RECOGNITION OF LANGUAGES DEFINED OVER THE SIGNATURE σ_{IM}.

Theorem 1. Let $P_1, \dots, P_r$ and their complements be OTA-recognizable and $\Phi_{\sigma_{IM}}$ be an EMSO-formula over signature $\sigma_{IM} = \sigma_{IM}(P_1, \dots, P_r)$. Then

1) There is EMSO-formula $\Psi_{\sigma_{MC}}(\Phi_{\sigma_{IM}})$ over signature σ_{MC} such that for every sequence w: $IM(w) \models \Phi_{\sigma_{IM}} \iff MC(w) \models \Psi_{\sigma_{MC}}(\Phi_{\sigma_{IM}})$.

2) There exists an algorithm that effective constructs OTA which:

2.1) processes the matrix code $MC(w)$;

2.2) accepts $MC(w)$ if $IM(w) \models \Phi_{\sigma_{IM}}$.

Proof. We begin with definition of an operator O_{skel}. Given an arbitrary OTA $\mathcal{A} = (\Sigma^k, \Sigma^t, q_0, F, \delta_{\mathcal{A}})$ a tuple $\tau = (\tau_1, ..., \tau_k)$ and an integer $l \geq \tau_i$ $(i = 1, ..., k)$ operator O_{skel} constructs OTA $\mathcal{A}' = O_{skel}(\mathcal{A}, \tau, l)$ such that

1) $\mathcal{A}'$ processes any cell-like matrix M over alphabet Σ^{l+2},

2) $\mathcal{A}'$ imitates OTA $\mathcal{A}$ on the projection $\mathcal{P}_{\tau'}(S(M))$ where $\tau' = (\tau_1 + 2, ..., \tau_k + 2)$.

Example 4. Consider OTA $\mathcal{A} = (\Sigma, \Sigma, q_0, \{1\}, \delta)$ and let

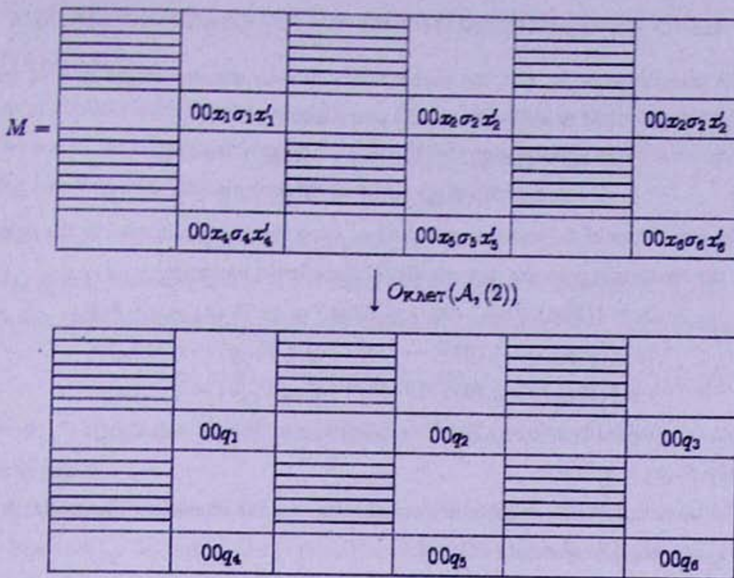

Figure 3: The transformation of the matrix M by $O_{skel}(\mathcal{A}, (2), 3)$

$$\mathcal{A}\left(\begin{array}{|c|c|c|}\hline \sigma_1 & \sigma_2 & \sigma_3 \\ \hline \sigma_4 & \sigma_5 & \sigma_6 \\ \hline \end{array}\right) = \begin{array}{|c|c|c|}\hline q_1 & q_2 & q_3 \\ \hline q_4 & q_5 & q_6 \\ \hline \end{array} \qquad (6)$$

where $\sigma_i, q_i \in \Sigma$

In the Figure 3 a cell-like matrix M is presented where only the elements of skeleton $S(M)$ are indicated. The cells of M are shaded.

The OTA $O_{skel}(\mathcal{A}, (2), 3)$ replaces each crossing of the input matrix M in accordance to (6) and accepts M if the final state is (001). ◇

Let us define $O_{skel}(\mathcal{A}, \tau, l) = (\Sigma^{l+2},\ \Sigma^{l+2},\ (11q_0),\ F_{skel},\ \delta_{skel})$, where $F_{skel} = \Sigma^2 \times F_{\mathcal{A}}$, $\delta_{skel} = \Delta_1 \cup \Delta_2 \cup \Delta_3 \cup \Delta_4$.

Here

$$\begin{aligned}\Delta_1 = \{&(11q_0), (11q_0), (11\sigma) \to (1^{l+2}) \mid \sigma \in \Sigma^l\} \cup \\ \cup\ \{&(11q_0), (1^{l+2}), (11\sigma) \to (1^{l+2}) \mid \sigma \in \Sigma^l\} \cup \\ \cup\ \{&(1^{l+2}), (11q_0), (11\sigma) \to (1^{l+2}) \mid \sigma \in \Sigma^l\} \cup\end{aligned}$$

$$\cup \{(1^{l+2}), (1^{l+2}), (11\sigma) \to (1^{l+2}) \mid \sigma \in \Sigma^l\}.$$

The transitions of Δ_1 put the vector (1^{l+2}) in each element of the cells of matrix $O_{skel}(\mathcal{A}, \tau, l)(M)$ if M is a cell-like matrix over alphabet $\Sigma^{(l+2)}$.

$$\Delta_2 = \{(10q_0), (1^{l+2}), (10\sigma) \to (10q_0) \mid \sigma \in \Sigma^l\} \cup$$
$$\cup \{(1^{l+2}), (01q_0), (01\sigma) \to (01q_0) \mid \sigma \in \Sigma^l\}.$$

The transitions of Δ_2 translate the state q_0 along the separating rows to the right and along the separating columns downwards till the nearest crossings.

$$\Delta_3 = \{(10q_0), (01q_0), (00\sigma) \to (00q) \mid q_0, q_0, \mathcal{P}_{\tau'}(\sigma) \to q \in \delta_{\mathcal{A}}\} \cup$$
$$\cup \{(10q_0), (01q), (00\sigma) \to (00q') \mid q_0, q, \mathcal{P}_{\tau'}(\sigma) \to q' \in \delta_{\mathcal{A}}\} \cup$$
$$\cup \{(10q), (01q_0), (00\sigma) \to (00q') \mid q, q_0, \mathcal{P}_{\tau'}(\sigma) \to q' \in \delta_{\mathcal{A}}\} \cup$$
$$\cup \{(10q_1), (01q_2), (00\sigma) \to (00q_3) \mid q_1, q_2, \mathcal{P}_{\tau'}(\sigma) \to q_3 \in \delta_{\mathcal{A}}\}.$$

Here $\mathcal{P}_{\tau}(\sigma) = \sigma^{\tau_1} \ldots \sigma^{\tau_k}$.

The transitions of Δ_3 realize the computations on the crossings. The transitions are agreed upon the transitions of OTA $\mathcal{A}$.

$$\Delta_4 = \{(00q), (1^{l+2}), (10\sigma) \to (10q) \mid \sigma \in \Sigma^l\} \cup$$
$$\cup \{(1^{l+2}), (00q), (01\sigma) \to (01q) \mid \sigma \in \Sigma^l\} \cup$$
$$\cup \{(1^{l+2}), (01q), (01\sigma) \to (01q) \mid \sigma \in \Sigma^l\} \cup$$
$$\cup \{((10q), 1^{l+2}), (10\sigma) \to (10q) \mid \sigma \in \Sigma^l\}.$$

The transitions of Δ_4 translate the computed state from each crossing to the nearest crossings to the right and down. The OTA $O_{skel}(\mathcal{A}, \tau, l)$ is fully determined.

Suppose that the predicates $P_1, \ldots, P_r$ and their complements satisfy the conditions of Theorem 1 and let $\Phi_{\sigma_{IM}}$ be an EMSO-formula over signature $\sigma_{IM}(P_1, \ldots, P_r)$.

For the sentence $\Phi_{\sigma_{IM}}$ there exist an OTA $\mathcal{A}_{\Phi_{\sigma_{IM}}} = (\Sigma^r, \Sigma^t, q_0, F, \delta_{\mathcal{A}})$ which accepts a sequence w iff $IM(w) \models \Phi_{\sigma_{IM}}$. Given the OTA $\mathcal{A}_{\Phi_{\sigma_{IM}}}$ and the tuple $\tau(\nu, r)$ from (5) let us define the OTA $\mathcal{D} = O_{skel}(\mathcal{A}_{\Phi_{\sigma_{IM}}}, \tau(\nu, r), 2r\nu + 4r + 4) \circ \mathcal{L}$ where OTA $\mathcal{L}$ is defined by Lemma. For an arbitrary sequence w OTA $\mathcal{D}$ accepts $MC(w)$ iff $IM(w) \models \Phi_{\sigma_{IM}}$.

Therefore, an EMSO-formula $\Psi_{\sigma_{MC}}(\Phi_{\sigma_{IM}})$ exists such that D accepts $MC(w)$ iff $MC(w) \models \Psi_{\sigma_{MC}}(\Phi_{\sigma_{IM}})$.

The statement 1) of the Theorem 1 is proved. The proof of the statement 2) arise from the construction of the OTA D.◇

§5. THE RECOGNIZABILITY OF LANGUAGES DEFINED OVER THE SIGNATURE σ_Π.

Let us consider signature $\sigma_\Pi = \sigma_\Pi(P_1, \dots, P_r)$ for definition of sequence languages.

Theorem 2. Let binary predicates $P_1, \dots, P_r$ and their complements be OTA-recognizable. Let Φ_{σ_Π} be a first-order formula over the signature σ_Π in the form of

$$\Phi_{\sigma_\Pi} = \exists x_1 ... x_n \psi(x_1 ... x_k) \in \textstyle\sum_0, \tag{7}$$

i.e., Φ_{σ_Π} is EFO-formula where ψ is a boolean formula with free variables $x_1, ..., x_k$. Then there exists EMSO-sentence $\Xi_{\sigma_{IM}}(\Phi_{\sigma_\Pi})$ over signature $\sigma_{IM} = \sigma_{IM}(P_1 \dots P_r)$ such that

$$\Pi(w) \models \Phi_{\sigma_\Pi} \iff \mathrm{I}(w) \models \Xi_{\sigma_{IM}}(\Phi_{\sigma_\Pi}) \tag{8}$$

Proof: We begin the proof by enumeration of several formulas that define predicates we use in sequel.

1. Predicate „x belongs to the top row of matrix": $\varphi_t(x) := \neg\exists y\; yS_1x$.

2. Predicate „x belongs to the bottom row of matrix": $\varphi_b(x) := \neg\exists y\; xS_1y$.

3. Predicate „x belongs to the right column of matrix": $\varphi_r(x) := \neg\exists y\; xS_2y$.

4. Predicate „x belongs to the left column of matrix": $\varphi_l(x) := \neg\exists y\; yS_2x$.

5. Predicate „x and y coincide": $\varphi_=(x, y) := \forall z\; (zS_1x \Leftrightarrow zS_1y) \wedge (xS_1z \Leftrightarrow yS_1z) \wedge$ $\wedge (zS_2x \Leftrightarrow zS_2y) \wedge (xS_2z \Leftrightarrow yS_2z)$

6. Predicate „the set X is the main diagonal":

$\varphi_d(X) := \forall y[\; X(y) \to (\varphi_t(y) \Leftrightarrow \varphi_l(y)) \wedge (\varphi_b(y) \Leftrightarrow \varphi_r(y))] \wedge \forall y \forall z \forall t\; (X(y) \wedge yS_1z \wedge zS_2t) \to X(t))$.

Predicates „x is over the main diagonal" (denoted by $\varphi_{>d}$) and „x is under the main diagonal" (denoted by $\varphi_{<d}$)are defined analogously.

7. Predicate „the set X is a row":

$$\varphi_h(X) := \forall y, z((X(y) \wedge (yS_2z \vee zS_2y) \Rightarrow X(z))\wedge$$
$$\forall y, z[(X(y) \wedge X(z) \wedge \varphi_l(y) \wedge \varphi_l(z)) \Rightarrow \varphi_=(y, z)]\wedge$$
$$\forall y, z[(X(y) \wedge X(z) \wedge \varphi_r(y) \wedge \varphi_r(z)) \Rightarrow \varphi_=(y, z)].$$

The predicate „the set X is a column": $\varphi_v(X)$ is defined analogously.

Let Φ_{σ_Π} be a formula (7). We define the transformation process of this formula to a formula $\Xi_{\sigma_{IM}}(\Phi_{\sigma_\Pi})$ such that (8) is realised.

This transformation replaces the subformulas of Φ_{σ_Π} by formulas over signature σ_{IM}.

a) We replace the prefix $\exists x_1, ..., x_k$ by a formula that has the following meaning: „there exist k rows $H_1, \dots, H_k$ and k columns $V_1, \dots, V_k$ such that the intersections of H_i and V_i $(i = 1, ..., k)$ are on the main diagonal".

The point of this intersection we correspond to the variable x_i. So a pair H_i, V_i defines x_i $(i = 1, \dots, k)$.

The desired formula is:

$$\exists H_1...H_kV_1...V_k : \varphi_h(H_1) \wedge ... \wedge \varphi_h(H_k) \wedge \varphi_v(V_1) \wedge ... \wedge \varphi_v(V_k) \wedge$$
$$\exists z : (H_1(z) \wedge V_1(z) \wedge \varphi_d(z)) \wedge ... \wedge \exists z : (H_k(z) \wedge V_k(z) \wedge \varphi_d(z)).$$

b) The elementary subformulas of $\psi(x_1, ..., x_n)$ are of the form $P_l(\pi(x_i), \pi(x_j))$, $\overline{P_l}(\pi(x_i), \pi(x_j))$, $x_i \leq x_j$.

We replace the subformula $P_l(\pi(x_i), \pi(x_j))$ by $\exists z : (H_i(z) \wedge V_j(z) \wedge R_l(z))$.

The subformula $\overline{P_l}(\pi(x_i), \pi(x_j))$ we replace by $\exists z : (H_i(z) \wedge V_j(z) \wedge \overline{R_l}(z))$.

The subformula $x_i \leq x_j$ we replace by $\exists z : (H_i(z) \wedge V_j(z) \wedge (\varphi_{>d}(z) \vee \varphi_d(z))$.

The formula obtained by all replacements is denoted by $\Xi_{\sigma_{IM}}(\Phi_{\sigma_\Pi})$. Evidently, this is EMSO-formula over signature σ_{IM} and (8) is satisfied.◇

Corollary. Let binary predicates $P_1 \dots P_r$ and their complements be OTA-recognizable. Then for each EFO-formula Φ_{σ_Π} over signature σ_Π there is a EMSO-formula $\Psi_{\sigma_{MC}}(\Phi_{\sigma_\Pi})$ over signature σ_{MC} such that for each sequence w

$$\Pi(w) \models \Phi_{\sigma_\Pi} \iff MC(w) \models \Psi_{\sigma_{MC}}(\Phi_{\sigma_\Pi}). \quad \diamond$$

Theorem 3. There is a binary OTA-recognizable predicate P and a formula $\Phi^0_{\sigma_\Pi}$ over the signature $\sigma_\Pi(P) = (\pi, S, P)$ in the form of

$$\Phi^0_{\sigma_\Pi} \equiv \forall x_1...x_k\psi(x_1, ..., x_k) \in \prod_0$$

where ψ is a boolean formula with free variables $x_1, \dots, x_k$ such that the languages $\{IM(w)| \Pi(w) \models \Phi^0_{\sigma_\Pi}\}$ and $\{MC(w)| \Pi(w) \models \Phi^0_{\sigma_\Pi}\}$ are not recognizable by OTA.

Proof: First of all we prove two statements.

Statement 1. The predicate

$$P_{pr}(m, n) = \begin{cases} 1 & \text{if } m, n \text{ are relatively prime,} \\ 0 & \text{in inverse case} \end{cases}$$

and its complement are OTA-recognizable.

Proof: We use here the equivalence between OTA recognizability and the recogniz-

1	b	b	b	b	b	b	b	b	b	3	1	b	b	b	b	b	b	b	b	b	3	1	b	b	3
a	*									c	a	*									c	a	*	0	c
a		*								c	a		*								c	a	0	*	c
a			*							c	a			*							c	2	d	d	4
a				*						c	a				*						c	1	b	b	3
a			0		*		0			c	a			0		*		0			c	a	*	0	c
a						*				c	a						*				c	a	0	*	c
a							*			c	a							*			c	2	d	d	4
a								*		c	a								*		c	1	b	3	1
a									*	c	a									*	c	a	*	c	a
2	d	d	d	d	d	d	d	d	d	4	2	d	d	d	d	d	d	d	d	d	4	2	d	4	2

Figure 4: The tiling system for $(11, 26)$-matrix.

ability by tiling systems [3] . The tiling system recognizing P_{pr} for the case $m < n$ is shown in Fig.4.

It is the tiling system over the alphabet $\{0, 1, 2, 3, 4, a, b, c, d, *, \#\}$.

For the case $m > n$ there exist analogous tiling system. Analogous tiling system exists for the complement $\overline{P_{pr}}$. ◇

Statement 2. Consider the signature $\sigma_{IM}(P_{pr}) = (S_1, S_2, R^1)$. For each square symmetric matrix M over the alphabet $\Sigma = \{0, 1\}$ there exists a naturals sequence $w = w_1, \dots, w_n$ such that $IM(w) = M$.

Proof: Let M be a symmetric (n, n)-matrix over Σ. Consider a symmetric (n, n)-matrix A of prime numbers such that if $i < j$, $k < l$ and $(i, j) \neq (k, l)$ then $A(i, j) \neq A(k, l)$. Define a sequence w where $w_i = \prod_{j=1}^{n} \overline{M}(i, j) A(i, j)$, where

$$\overline{M}(i,j) = \begin{cases} 1 & \text{if } M(i,j) = 0 \\ 0 & \text{if } M(i,j) = 1 \end{cases}$$

Evidently,

$$P_{pr}(w_i, w_j) = \begin{cases} 1 & \text{if } M(i,j) = 1 \\ 0 & \text{if } M(i,j) = 0 \end{cases}$$

This gives us our statement. ◇

Now complete the proof of Theorem 3. Consider the formula

$$\Phi^0_{\sigma_\Pi} = \forall x_1 x_2 y_1 y_2 P_{pr}(x_1, y_1) \wedge P_{pr}(x_2, y_2) \wedge P_{pr}(x_2, y_1) \Rightarrow P_{pr}(x_1, y_2)$$

over $\sigma_\Pi(P_{pr})$. It is easy to see that

$$\Pi(w) \models \Phi^0_{\sigma_\Pi} \Longleftrightarrow \mathrm{IM}(w) \in \mathrm{CORNERS}$$

where the language CORNERS , defined in [5], is the set of matrices over Σ such that whenever $M(i,j) = M(i',j) = M(i,j') = 1$ then also $M(i',j') = 1$. This language is not recognizable by OTA. Therefore, it is sufficient to prove that CORNERS is not recognizable in the universe of symmetric matrices over Σ. We use the construction proposed in [5]. To each partition $\mathcal{P}$ of the set $\{1, \ldots, 2k\}$ on two-element sets can be associated a $(2k, k)$-matrix $B_{\mathcal{P}}$ [5] so that

$$B_{\mathcal{P}} B_{\mathcal{P}'} \in CORNERS \Longleftrightarrow \mathcal{P} = \mathcal{P}' \qquad (9)$$

The number of these matrices is $k!$.

Now consider for each partition $\mathcal{P}$ symmetric $(3k, 3k)$-matrices $M_{\mathcal{P}}$ in the form

$$M_{\mathcal{P}} = \begin{array}{|c|c|} \hline 0 & B_{\mathcal{P}} \\ \hline S(B_{\mathcal{P}}) & 0 \\ \hline \end{array}$$

where $S(B_{\mathcal{P}})$ is a matrix, symmetric to $B_{\mathcal{P}}$. The number of these matrices is $k!$ as before. From the consideration of $(4k, 4k)$-matrix $A(\mathcal{P}, \mathcal{P}')$ where

$$A(\mathcal{P}, \mathcal{P}') = \begin{array}{|c|c|c|} \hline 0 & 0 & S(B_{\mathcal{P}'}) \\ \hline 0 & 0 & S(B_{\mathcal{P}}) \\ \hline B_{\mathcal{P}'} & B_{\mathcal{P}} & 0 \\ \hline \end{array}$$

and from (9) follows that $A(\mathcal{P}, \mathcal{P}') \in CORNERS \Longleftrightarrow \mathcal{P} = \mathcal{P}'$. The language $CORNERS$ defines the corresponding syntactic equivalence denoted by $\sim_{CORNERS}$ [3,5]. It is easy to see that $M_{\mathcal{P}} \sim_{CORNERS} M_{\mathcal{P}'} \Longleftrightarrow \mathcal{P} = \mathcal{P}'$. Therefore the number of classes of syntactic equivalence $f_{CORNERS}(3k, 3k) \geq k!$ and $\{IM(w) | \Pi(w) \models \Phi^0_{\sigma_\Pi}\}$ is not recognizable. Evidently, the language $\{MC(w) | \Pi(w) \models \Phi^0_{\sigma_\Pi}\}$ is not recognizable.$\diamondsuit$

§6. EXAMPLES.

Here are three examples of OTA-recognizable languages of sequences. Recall that it is sufficient to prove the recognizability of corresponding languages of information matrices. In the following examples only one predicate is used - the equality predicate $P_=$ and the signature $\sigma^1_{IM} = \sigma^1_{IM}(P_=) = (S_1, S_2, R_1)$, where $R_1((i,j)) = P_=(w_i, w_j)$.

Example 5. Let $W_1 = \{w_1, ..., w_n \mid w_i \neq w_j \text{ if } i \neq j,\ w_i \in N,\ n \in N\}$ be the set of sequences of pairwise different numbers. If $w \in W_1$ then $IM(w)$ over signature σ^1_{IM} is a square (n,n)-matrix over $\Sigma = \{0,1\}$ in the form

$$\mathrm{IM}(w) = \begin{array}{|ccccc|} \hline 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & \vdots \\ 0 & 0 & \ddots & 0 & \vdots \\ \vdots & \vdots & 0 & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 1 \\ \hline \end{array} \quad \text{for } w \in W_1.$$

and the language $L_1 = \{IM(w) | w \in W_1\}$ can be defined by the formula

$$\Phi_1 := \forall x\ (\varphi_d(x) \Leftrightarrow R_1(x)).$$

Hence, the set W_1 is MC-recognizable. ◇

Example 6. Let $W_2 = \{w_1, ..., w_n \mid w_i = w_{n-i+1},\ i = 1, ..., n\}$.

W_2 is the set of palindromes. For each $w \in W_2$ the matrix $IM(w)$ has the form

$$IM(w) = \begin{array}{|cccccc|} \hline & & & & & 1 \\ & & & & 1 & \\ & & & \cdot & & \\ & & \cdot & & & \\ & \cdot & & & & \\ & 1 & & & & \\ 1 & & & & & \\ \hline \end{array} \quad \text{for } w \in W_2,$$

i.e., the language $L_2 = \{(w) \mid w \in W_2\}$ is the set of square matrices that have 1 on indicated diagonal. As is known the language of square matrices with 1 on the main diagonal is OTA- recognizable. On the other hand, the set of OTA-recognizable languages is closed under rotation [3]. Therefore the set W_2 is MC-recognizable. ◇

Example 7. Let W_3 be a set of sequences w that can be obtained as a result of Shuffle operation, i.e., $w = w_1 ... w_n$ such that there exist two sequences $0 < i_1 < ... < i_{\frac{n}{2}} \leq n$ and $0 < j_1 < ... < j_{\frac{n}{2}} \leq n$ such that

1) $i_s \neq j_k$ $(1 \leq s, k \leq \frac{n}{2})$.

2) $w_{i_k} = w_{j_k}$ $(1 \leq k \leq \frac{n}{2})$.

Let us prove the MC-recognizability of W_3. In this connection we need some auxiliary automata.

$\mathcal{A}_{column} = (\Sigma, \Sigma, q_0, \Sigma, \delta_{column})$ -which is defined in section 3. $\mathcal{A}_{column}$ guesses the first column and copies it in all other columns.

$\mathcal{A}_{row} = (\Sigma, \Sigma, q_0, \Sigma, \delta_{row})$ -an automaton that guesses the first row analogously to $\mathcal{A}_{column}$ and copies it in all other rows.

$\mathcal{A}_{diag} = (\Sigma^2, \Sigma^2, q_0, \{(10), (01)\}, \delta_{diag})$ accepts square matrices M over Σ^2 if $M(i,i) = (10)$ or (01) for $i = 1, ..., n$.

$\mathcal{A}_{id} = (\Sigma, \Sigma, q_0, \Sigma, \delta_{id})$ accepts any matrix over Σ and $\mathcal{A}_{id}(M) = M$.

$\mathcal{A}'_{diag} = (\Sigma, \Sigma^2, q_0, \{11\}, \delta'_{diag})$ accepts M if

1) M is a square matrix,

2) $M^1(i,i) = 1$ for $i = 1, ..., n$.

Evidently, the OTA

$\mathcal{A}_0 = (\mathcal{A}_{diag} \circ (\mathcal{A}_{row} \wedge \mathcal{A}_{column})) \circ \mathcal{A}_{id} = (\Sigma, \Sigma^5, q_0, \Sigma^5, \delta_0)$. transforms arbitrary square matrix M to the matrix M_1 where $M_1^5(i,j) = M(i,j)$ and $M_1(i,i) = 01x$ or $10x$ for $x \in \Sigma^3$. Hence the OTA $\mathcal{A}_{W_3} = \mathcal{O}_{skel}(\mathcal{A}'_{diag}, (5), 5) \circ \mathcal{A}_0$ recognizes the set $\{IM(w) | \, w \in W_3\}$, and due to Theorem 1 the language $\{MC(w) | \, w \in W_3\}$ is OTA-recognizable. So W_3 is MC-recognizable. ◇

§7. CONCLUSION.

We have investigated the sequences of naturals. Our investigations can easily be generalized for the sequences of words $w = w_1|w_2|w_3|w_n$ where w_i are words over alphabet A (the separator $| \notin A$).

The notion of OTA-recognizability of binary predicate $P(w, w')$ can be transferred for the case $w, w' \in A^*$. Then if $P_1, P_2, \ldots, P_r$ are OTA-recognizable predicates there exists OTA that accepts the $MC(w)$ iff $IM(w)$ is EMSO-definable over signature

$(S_1, S_2, P_1, \dots, P_r)$. As an example, we refer to the problem of recognizing send-receive protocols by cellular automata [6,7].

REFERENCES

1. K.Inoue and A.Nakamura. Some properties of two-dimensional on-line tessellation acceptors. *Information Sciences*, vol.13, p.95-121, 1977.

2. K.Inoue and A.Nakamura. A Survey of two-dimensional automata theory. In *Proc. 5th Int. Meeting of Young Computer Scientists*. J Dasson and J.Kelemen (Eds.), p. 72-91. Lecture Notes in Computer Science 381, Springer-Verlag, Berlin, 1990.

3. D.Giammarresi and A.Restivo. Two-dimensional languages. In *Handbook of Formal Laguage Theory*, v.3, Springer-Verlag, N.Y.,1996.

4. D.Giammarresi, A.Restivo, S.Siebert and W.Thomas. Monadic second order logic over rectangular pictures and recognizability by tiling systems. *Information and computation*. Vol.125, No. 1, p. 32-45, 1996.

5. O.Matz. On Piecewise Testable, Starfree,and Recognizable Picture Langauges. *Foundation of Software Science and Computation Structures*, v.1378, LNCS, p.203-210, Springer, 1998.

6. ITU-TS. ITU-TS Recommendation Z.120: Message Sequence Chart 1999 (MSC 99). Tech. Rep., Geneva, 1999.

7. B.Bollig, M.Leucker. Message-Passing Automata are expressively equivalent to EMSO-logic. RWTH Aachen. Depart. Of Comp. Sci., Tech. Rep. 2004.

30 August 2005

Institute for Informatics and Automation Problems

National Academy of Sciences of Armenia

E-mail: shahb@ipia.sci.am, shouk@sci.am