PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2019, **53**(2), p. 82–90

M a t h e m a t i c s

SOME RESULTS ON REGULAR EXPRESSIONS FOR MULTITAPE FINITE AUTOMATA

T. A. GRIGORYAN [*]

IT Educational and Research Center YSU, Armenia

We consider sets of word tuples accepted by multitape finite automata. We use the known notation for regular expressions that describes languages accepted by one-tape automata. Nevertheless, the interpretation of the "concatenation" operation is different in this case. The algebra of events for multitape finite automata is defined in the same way as for one-tape automata. It is shown that the introduced algebra is a Kleene algebra. It is also, shown that some known results for the algebra of events accepted by one-tape finite automata are valid in this case too.

MSC2010: 68Q45

***Keywords*:** regular expressions, algebra of events, multitape finite automata.

1. Introduction. Deterministic multitape finite automata were introduced by M. O. Rabin and D. Scott in 1959 [1]. Since then, there were attempts to consider regular expressions as well as the algebra of events accepted by multitape automata. Algebra of events accepted by one-tape automata was considered in [2]. Several attempts most relevant for our consideration are briefly considered below.

B. G. Mirkin [3] has considered a special coding for sets of word pairs accepted by multitape automata. The paper [3] notes the following: "For simplicity in the following discussion we will consider only D_2 and N_2", without any proof seems not enough for using the suggested coding for $n > 2$. It is necessary to use another coding for that case (for example, see [4]) published back in 1980. The latter coding clearly reflects all dependencies that exist simultaneously in fillings of tapes.

Another consideration of sets of word tuples accepted by multitape automata was given by P. Starke in [5]. For the deterministic multitape automata the following result holds true: "A n-ary relation R over $W(X)$ is representable by a finite deterministic n-tape automaton, if there exists an admissible regular expression T such that $R = Val_n(T)$". Nevertheless, in the same paper the author mentioned that "unfortunately there are non-admissible regular expressions T such that $Val_n(T)$ is

[*] E-mail: tigran.grigoryan1995@gmail.com

representable by a deterministic automaton". It will be shown below that for any set of tuples accepted by a deterministic multitape automaton the corresponding regular expression can be built based on definitions and the notation introduced in this paper.

At the same time it will be shown that there is no need to introduce a special notation for the regular expression model for deterministic multitape finite automata. So the existing notation for one-tape automata can be used. The only difference is the interpretation of the regular expression for the operation of concatenation that should be changed based on the coding of the alphabet symbols introduced in [4].

Algebra of events for the multitape finite automata and the systems of equations in the algebra will be defined similarly to the definitions of the algebra of events accepted by one-tape automata [6]. In particular, an alternative specification of a deterministic multitape finite automaton via systems of equations in the algebra of events is suggested. It is proved that the considered algebra is a Kleene algebra. Consequently, all the related results in [6] are also true in this case.

2. Free Partially Commutative Semigroups. Recall some definitions from [7]. If X is an alphabet, then the set of all words over X, including the empty word, will be denoted by F_X, and the set of all n-element tuples of words by F_X^n.

Let G be a semigroup with a unit generated by the set of generators $Y = \{y_1, y_2, \dots, y_n\}$. G is called free partially commutative semigroup, if it is defined by a finite set of definitive assumptions of type $y_i y_j = y_j y_i$.

Let $K : F_Y \to F^n_{\{0,1\}}$ be a homomorphism over the set F_Y, which maps words from F_Y to n-element vectors in binary alphabet $\{0,1\}$. The homomorphism K over the set of symbols of the set F_Y is defined by the equation:

$$K(y_i) = (a_{1i}, \dots, a_{ni}), \quad \text{where} \quad a_{ij} = \begin{cases} 1, & i = j, \\ e, & y_i y_j = y_j y_i, \\ 0, & y_i y_j \neq y_j y_i. \end{cases}$$

At the same time $K(e) = (e, \dots, e)$.

$K(y_i y_j)$, $i \neq j$, can be defined in one of the two alternative ways:

1. Right concatenation: $K(y_i y_j) = (a_{1i}a_{1j}, \dots, a_{ni}a_{nj})$;

2. Left concatenation: $K(y_i y_j) = (a_{1j}a_{1i}, \dots, a_{nj}a_{ni})$.

The first way was in fact used in [5]. The second way was used in [4] as well as in this publication. Left concatenation for semigroup elements $a = y_{i1} \dots y_{ik}$ and $b = y_{j1} \dots y_{jl}$ is defined in the following way:

$$K(ab) = K\big((a_{1ik} \dots a_{1i1}, \dots, a_{nik} \dots a_{ni1}), (a_{1jl} \dots a_{1j1}, \dots, a_{njl} \dots a_{nj1})\big) =$$
$$= (a_{1jl} \dots a_{1j1} a_{1ik} \dots a_{1i1}, \dots, a_{njl} \dots a_{nj1} a_{nik} \dots a_{ni1}).$$

As stated in [4], the homomorphism K can be considered as a mapping not only over the F_Y, but also over the free partially commutative semigroup G.

3. Regular Expressions. We will consider tupless of binary words for the element representation in a free partially commutative semigroup that were introduced in Section 2.

First, let us discuss $K(G)$. Due to the fact that K is a homomorphism, $K(G)$ itself is a semigroup with $X = K(Y)$ generators and identity $(e, \dots, e)$. By Lemma 2 from [4], any element in $K(G)$ can be represented in the form $k_{i_1}k_{i_2}\dots k_{i_m}$, where $k_{i_j} \in X\ \forall j$. Note that an element may have multiple representations of this form. Hence, we define an equivalence relation ρ_K over $K(F_Y)$. If $p,q \in K(F_Y)$, then we write $p\rho_K q$ if and only if they are the representations of a same element in $K(G)$. The relation ρ_K splits $K(F_Y)$ into disjoint classes. The class containing an element p will be denoted by $[p]$. Obviously, $[(e,\dots,e)] = \{(e,\dots,e)\}$ and $[k_i] = \{k_i\}\ \forall k_i \in X$. These classes will be used to define regular events and regular expressions.

The operation of multiplication for tuples of binary words introduced in the mentioned section will be considered further as an operation of concatenation for tuples. Based on the operation of concatenation, the operation of iteration over a tuple of binary words may be defined similarly to the operation of iteration for words [2]. The operation of union for a pair of tuples also can be defined similarly to the operation of union for words given in [2].

Having the operations of a union, a concatenation and an iteration, we can define notions of regular event (set) and regular expression [2].

A regular event in a partially commutative alphabet X is defined as follows:

1. $\emptyset$ (empty set) is a regular event in X;
2. $E = \{[(e,\dots,e)]\}$ is a regular event in X;
3. $\forall y \in Y\ \{[K(y)]\}$ is a regular event in X;
4. If P and Q are regular events in X, then so are
   i) $P + Q = P \cup Q$;
   ii) $PQ = \{[s]\ \mid\ s = pq,\ [p] \in P,\ [q] \in Q\}$, where pq is the left concatenation of p and q;
   iii) $P^* = \cup_{n\geq 0} P^n$, where $P^0 = E,\ P^n = PP^{(n-1)}$ for $n \geq 1$;
5. There are no any other regular events in X.

A regular expression in a partially commutative alphabet X is defined as follows:

1. $\emptyset$ is a regular expression, which denotes the regular event $\emptyset$;
2. $K(e) = (e,\dots,e)$ is a regular expression, which denotes the regular event E;
3. $\forall y \in Y\ K(y)$ is a regular expression, which denotes the regular event $\{[K(y)]\}$;
4. If p and q are regular expressions in X denoting regular events P and Q correspondingly, then so are:
   i) $(p+q)$, which denotes the regular event $P \cup Q$;
   ii) (pq), which denotes the regular event PQ;
   iii) $(p)^* = \cup_{n\geq 0}(p)^n$, where $(p)^0 = K(e),\ (p)^n = p(p)^{(n-1)}$ for $n \geq 1$, which denotes the regular event P^*.
5. There are no any other regular expressions in X.

As an illustrating example we consider the following input alphabet: $X = \{x_1, x_2, x_3\}$, where $x_2x_3 = x_3x_2$ and $x_1x_2 \neq x_2x_1,\ x_1x_3 \neq x_3x_1$.

Using the binary coding considered in the Section 3, the following correspondence between the words in F_X and three-element vectors in the binary alphabet

$\{0,1\}$ will be obtained:

$$K(x_1) = (1,0,0),\ K(x_2) = (0,1,e),\ K(x_3) = (0,e,1).$$

If we consider the regular expression

$$((1,0,0)+(0,1,e)+(0,e,1))^*(1,0,0) = (x_1+x_2+x_3)^*x_1,$$

then we will obtain the set of all words ending with x_1.

The regular event corresponding to a given regular expression R will be denoted as $E(R)$.

For the regular expressions we use the same notation both for one-tape and multitape automata, but they will have different interpretations.

4. Algebra of Events. Similarly to [2], it can be considered an algebra of regular events in the case of multitape automata, where the elements of events are words in the alphabet $K(X)$. This algebra will be denoted with A_X.

The tuple $\tilde{e}$ of empty binary words $(e,\dots,e)$ will be named the empty tuple and the set $E = \{[\tilde{e}]\}$ is the empty event.

In addition, let us define a partial order in a natural way. If P and Q are regular events in a partially commutative alphabet X, then $P \le Q$ denotes the set relation $P \subseteq Q$.

It is evident that the operation of concatenation for tuples, introduced in the Section 2, will preserve the relations between events that we have in the algebra of events for the one tape automata [6]. Thus we come to the following theorem.

Theorem . *The algebra of regular events for multitape automata is a Kleene algebra. In particular, for any $P,Q,S,Z \in (A_X,\ +,\ \cdot,\ ^*,\ \emptyset,\ E)$ all 15 Kleene axioms hold:*

1. $P+(Q+S) = (P+Q)+S.$
2. $P+Q = Q+P.$
3. $P+\emptyset = P.$
4. $P+P = P.$
5. $P(QS) = (PQ)S.$
6. $EP = P.$
7. $PE = P.$
8. $P(Q+S) = PQ+PS.$
9. $(P+Q)S = PS+QS.$
10. $\emptyset P = \emptyset.$
11. $P\emptyset = \emptyset.$
12. $E+PP^* \le P^*.$
13. $E+P^*P \le P^*.$
14. $Q+PZ \le Z \to P^*Q \le Z.$
15. $Q+ZP \le Z \to QP^* \le Z.$

Each of this axioms will be proved below.

Proof. The axioms 1–4 directly follow from the definition of $+$ operation:

1. $P+(Q+S) = P\cup(Q\cup S) = P\cup Q\cup S,$
   $(P+Q)+S = (P\cup Q)\cup S = P\cup Q\cup S;$
2. $P+Q = P\cup Q = Q\cup P = Q+P;$
3. $P+\emptyset = P\cup\emptyset = P;$
4. $P+P = P\cup P = P.$

To prove the 5th axiom we will use the associativity of the left concatenation operation.

5. $P(QS) = \{[pr] \mid [p] \in P,\ [r] \in QS\}$
$= \{[pr] \mid [p] \in P,\ [r] \in \{[qs] \mid [q] \in Q,\ [s] \in S\}\}$
$= \{[pr] \mid [p] \in P,\ [r] = [qs],\ [q] \in Q,\ [s] \in S\}$
$= \{[pqs] \mid [p] \in P,\ [q] \in Q,\ [s] \in S\},$
$(PQ)S = \{[rs] \mid [r] \in PQ,\ [s] \in S\}$
$= \{[rs] \mid [r] \in \{[pq] \mid [p] \in P,\ [q] \in Q\},\ [s] \in S\}$
$= \{[rs] \mid [r] = [pq],\ [p] \in P,\ [q] \in Q,\ [s] \in S\}$
$= \{[pqs] \mid [p] \in P,\ [q] \in Q,\ [s] \in S\}.$

Next:

6. $EP = \{[ep] \mid [e] \in E,\ [p] \in P\} = \{[\tilde{e}p] \mid [p] \in P\} = \{[p] \mid [p] \in P\} = P.$
7. $PE = \{[pe] \mid [p] \in P,\ [e] \in E\} = \{[p\tilde{e}] \mid [p] \in P\} = \{[p] \mid [p] \in P\} = P.$
8. $P(Q+S) = \{[pr] \mid [p] \in P,\ [r] \in Q+S\}$
$= \{[pr] \mid [p] \in P,\ [r] \in Q \text{ or } [r] \in S\}$
$= \{[pr] \mid ([p] \in P,\ [r] \in Q) \text{ or } ([p] \in P,\ [r] \in S)\}$
$= \{[pr] \mid [p] \in P,\ [r] \in Q\} \cup \{[pr] \mid [p] \in P,\ [r] \in S\}$
$= PQ \cup PS = PQ + PS.$
9. $(P+Q)S = \{[rs] \mid [r] \in P+Q,\ [s] \in S\}$
$= \{[rs] \mid [r] \in P \text{ or } [r] \in Q,\ [s] \in S\}$
$= \{[rs] \mid ([r] \in P,\ [s] \in S) \text{ or } ([r] \in Q,\ [s] \in S)\}$
$= \{[rs] \mid [r] \in P,\ [s] \in S\} \cup \{[rs] \mid [r] \in Q,\ [s] \in S\}$
$= PS \cup QS = PS + QS.$
10. $\emptyset P = \{[op] \mid [o] \in \emptyset,\ [p] \in P\} = \emptyset.$
11. $P\emptyset = \{[po] \mid [p] \in P,\ [o] \in \emptyset\} = \emptyset.$

To prove the 12th and 13th equations it is enough to prove that any element in the left expression is contained in P^* too.

12. $\forall p \in E + PP^* \Rightarrow p \in E$ or $p \in PP^*$,
[1] $p \in E \Rightarrow p \in P^*$,
[2] $p \in PP^* \Rightarrow p \in \{[p_1p_2] \mid [p_1] \in P,\ [p_2] \in P^*\}$
$\Rightarrow p \in \{[p_1p_2] \mid [p_1] \in P,\ [p_2] \in \cup_{n\geq 0}P^n\}$
$\Rightarrow p \in \{[p'] \mid [p'] \in \cup_{n\geq 1}P^n\} \subseteq \{[p'] \mid [p'] \in \cup_{n\geq 0}P^n\}$
$\Rightarrow p \in P^*.$
13. $\forall p \in E + P^*P \Rightarrow p \in E$ or $p \in P^*P$,
[1] $p \in E \Rightarrow p \in P^*$,
[2] $p \in P^*P \Rightarrow p \in \{[p_1p_2] \mid [p_1] \in P^*,\ [p_2] \in P\}$
$\Rightarrow p \in \{[p_1p_2] \mid [p_1] \in \cup_{n\geq 0}P^n,\ [p_2] \in P\}$
$\Rightarrow p \in \{[p'] \mid [p'] \in \cup_{n\geq 1}P^n\} \subseteq \{[p'] \mid [p'] \in \cup_{n\geq 0}P^n\}$
$\Rightarrow p \in P^*.$

It was proved in [8] that if the previous 13 axioms hold, then axioms 14 and 15 are equivalent to:

16. $PZ \leq Z \to P^*Z \leq Z$.
17. $ZP \leq Z \to ZP^* \leq Z$.

respectively.

To prove these two axioms the mathematical induction will be used.

16. [1] $PZ \leq Z$.
    [2] Let us assume that $P^{n-1}Z \leq Z$.
    [3] Let us proof that $P^nZ \leq Z$.
    Indeed:

$$\begin{cases} P^{n-1}Z \leq Z \\ PZ \leq Z \end{cases} \Rightarrow P^{n-1}(PZ) \leq Z \Leftrightarrow P^nZ \leq Z,$$

$P^*Z = (E+P+P^2+\cdots+P^n+\cdots)Z = EZ+PZ+P^2Z+\cdots$
Obviously, $EZ \leq Z$, and by the induction: $P^kZ \leq Z \ \forall k \geq 1$. Thus, $P^*Z \leq Z$.

17. [1] $ZP \leq Z$.
    [2] Let us assume that $ZP^{n-1} \leq Z$.
    [3] Let us proof that $ZP^n \leq Z$.
    Indeed:

$$\begin{cases} ZP^{n-1} \leq Z \\ ZP \leq Z \end{cases} \Rightarrow (ZP)P^{n-1} \leq Z \Leftrightarrow ZP^n \leq Z,$$

$ZP^* = Z(E+P+P^2+\cdots+P^n+\cdots) = ZE+ZP+ZP^2+\cdots$
Obviously, $ZE \leq Z$, and by the induction: $ZP^k \leq Z \ \forall k \geq 1$. Thus, $ZP^* \leq Z$. □

The Theorem is proved and so A_X is a Kleene algebra. It was shown in [9] that the algebra of events for one-tape automata is complete over Kleene axioms, in other words, any property in the algebra of events for one-tape automata can be induced from the mentioned 15 axioms.

Corollary . *Any equation in the algebra of events for one-tape automata is also true in the algebra of regular events for multitape automata. The opposite is not always true.*

Now let us consider a system of equations of regular events for multitape automata similarly to the system of equations of regular events for one-tape automata discussed in [6].

$$\begin{cases} X_1 = X_1S_{11}+X_2S_{21}+\cdots+X_nS_{n1}+R_1, \\ X_2 = X_1S_{12}+X_2S_{22}+\cdots+X_nS_{n2}+R_2, \\ \vdots \\ X_n = X_1S_{1n}+X_2S_{2n}+\cdots+X_nS_{nn}+R_n, \end{cases} \tag{1}$$

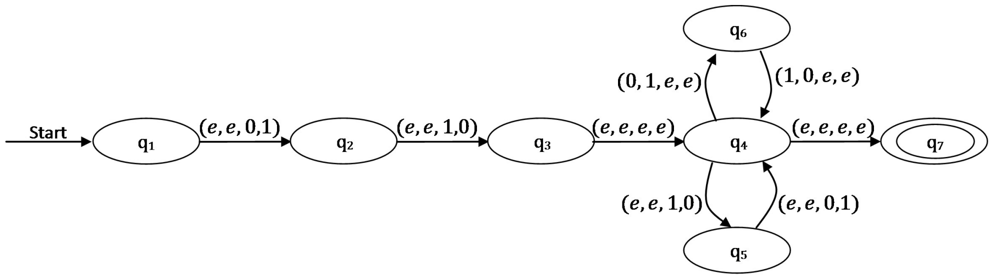
where S_{ij} and R_i are given regular events and X_i are unknown events.

Proposition 1. *If $[\tilde{e}] \notin S$, then the equation $X = XS+R$ has a unique solution, and it can be expressed as $X = R(S)^*$.*

Proposition 2. *If the solution of the system of equations (1) is unique, then it can be found by a successive elimination of unknown variables.*

Both Propositions are analogous to the theorems in the case of regular expressions for one-tape automata from [6]. They may be proved using the same technique discussed there, since all the steps are done using the Kleene axioms or their corollaries.

To illustrate the technique described above, a finite multitape automaton in Figure is considered.



NMFA for $(e,e,10,01)((e,e,01,10)+(10,01,e,e))^{*}$.

The transition function of the considered automaton can be transformed to a form, displayed in the Table.

Transformed transition function

	q_1	q_2	q_3	q_4	q_5	q_6	q_7
q_1	$\emptyset$	$(e,e,1,0)$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q_2	$\emptyset$	$\emptyset$	$(e,e,1,0)$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
q_3	$\emptyset$	$\emptyset$	$\emptyset$	(e,e,e,e)	$\emptyset$	$\emptyset$	$\emptyset$
q_4	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$(e,e,1,0)$	$(0,1,e,e)$	(e,e,e,e)
q_5	$\emptyset$	$\emptyset$	$\emptyset$	$(e,e,0,1)$	$\emptyset$	$\emptyset$	$\emptyset$
q_6	$\emptyset$	$\emptyset$	$\emptyset$	$(1,0,e,e)$	$\emptyset$	$\emptyset$	$\emptyset$
q_7	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

It can be rewritten as a system of equations, where the equation i corresponds to the column i of the Table $(i=1,\ldots,7)$.

$$\begin{cases} X_1 = X_1\emptyset + X_2\emptyset + X_3\emptyset + X_4\emptyset + X_5\emptyset + X_6\emptyset + X_7\emptyset + \tilde{e}, \\ X_2 = X_1(e,e,0,1) + X_2\emptyset + X_3\emptyset + X_4\emptyset + X_5\emptyset + X_6\emptyset + X_7\emptyset, \\ X_3 = X_1\emptyset + X_2(e,e,1,0) + X_3\emptyset + X_4\emptyset + X_5\emptyset + X_6\emptyset + X_7\emptyset, \\ X_4 = X_1\emptyset + X_2\emptyset + X_3\tilde{e} + X_4\emptyset + X_5(e,e,0,1) + X_6(1,0,e,e) + X_7\emptyset, \\ X_5 = X_1\emptyset + X_2\emptyset + X_3\emptyset + X_4(e,e,1,0) + X_5\emptyset + X_6\emptyset + X_7\emptyset, \\ X_6 = X_1\emptyset + X_2\emptyset + X_3\emptyset + X_4(0,1,e,e) + X_5\emptyset + X_6\emptyset + X_7\emptyset, \\ X_7 = X_1\emptyset + X_2\emptyset + X_3\emptyset + X_4\tilde{e} + X_5\emptyset + X_6\emptyset + X_7\emptyset. \end{cases}$$

Solving the system of equations by successive elimination of unknown variables we obtain:

$$\begin{cases} X_1 = \tilde{e}, \\ X_2 = X_1(e,e,0,1), \\ X_3 = X_2(e,e,1,0), \\ X_4 = X_3\tilde{e} + X_5(e,e,0,1) + X_6(1,0,e,e), \\ X_5 = X_4(e,e,1,0), \\ X_6 = X_4(0,1,e,e), \\ X_7 = X_4\tilde{e}; \end{cases} \qquad \begin{cases} X_1 = e, \\ X_2 = (e,e,0,1), \\ X_3 = (e,e,0,1)(e,e,1,0), \\ X_4 = (e,e,0,1)(e,e,1,0) + \\ \qquad + X_4(e,e,1,0)(e,e,0,1) + \\ \qquad + X_4(0,1,e,e)(1,0,e,e), \\ X_5 = X_4(e,e,1,0), \\ X_6 = X_4(0,1,e,e), \\ X_7 = X_4. \end{cases}$$

It follows from Proposition 1 that:

$$X_4 = (e,e,0,1)(e,e,1,0)((e,e,1,0)(e,e,0,1) + (0,1,e,e)(1,0,e,e))^* =$$
$$= (e,e,10,01)((e,e,01,10) + (10,01,e,e))^*.$$

Since the final state of the automaton is q_7, the value of X_7 is the regular expression, which describes the considered DFMA. Since $X_7 = X_4$, we obtain: $X_7 = (e,e,10,01)((e,e,01,10) + (10,01,e,e))^*$.

Received 01.04.2019
Reviewed 19.04.2019
Accepted 26.04.2019

REFERENCES

1. Rabin M.O., Scott D. Finite Automata and Their Decision Problems. *IBM Journal of Research and Development*, **3** : 2 (1959), 114–125.
2. Glushkov V.M. The Abstract Theory of Automata. *Russian Mathematical Surveys*, **16** : 5 (1961).
3. Mirkin B.G. On the Theory of Multitape Automata. *Cybernetics and Systems Analysis* **2** : 5 (1966), 9–14.
4. Godlevskii A.B., Letichevskii A.A., Shukuryan S.K. Reducibility of Program-Scheme Functional Equivalence on a Nondegenerate Basis of Rank Unity to the Equivalence of Automata with Multidimensional Tapes. *Cybernetics and Systems Analysis*, **16** : 6 (1980), 793–799.
5. Starke P.H. *On the Representability of Relations by Deterministic and Nondeterministic Multi-Tape Automata.* International Symposium on Mathematical Foundations of Computer Science (1975), 114–124.
6. Bodnarchuk V.G. Systems of Equations in the Algebra of Events. *USSR Computational Mathematics and Mathematical Physics*, **3** : 6 (1963), 1470–1487.
7. Letichevsky A.A., Shoukourian A.S., Shoukourian S.K. *The Equivalence Problem of Deterministic Multitape Finite Automata: A New Proof of Solvability Using a Multidimensional Tape.* International Conference on Language and Automata Theory and Applications (2010), 392–402.

8. Pratt V. *Dynamic Algebras as a Well-Behaved Fragment of Relation Algebras.* International Conference on Algebraic Logic and Universal Algebra in Computer Science (1988), 77–110.
9. Kozen D. A Completeness Theorem for Kleene Algebras and the Algebra of Regular Events. *Information and Computation*, **110** : 2 (1994), 366–390.

Տ. Ա. ԳՐԻԳՈՐՅԱՆ

ԲԱԶՄԱԺԱՊԱՎԵՆ ԱՎՏՈՄԱՏՆԵՐԻ ՀԱՄԱՐ ԿԱՆՈՆԱՎՈՐ ԱՐՏԱՀԱՅՏՈՒԹՅՈՒՆՆԵՐԻ ՎԵՐԱԲԵՐՅԱԼ ՈՐՈՇ ԱՐԴՅՈՒՆՔՆԵՐ

Դիտարկվում են բազմաժապավեն վերջավոր ավտոմատների կողմից ճանաչվող բառերի կորտեժների բազմություններ: Օգտագործվում է մեկ ժապավենանոց ավտոմատների կողմից ճանաչվող լեզուները նկարագրող կանոնավոր արտահայտությունների հայտնի գրելաձևը: Սակայն, այս դեպքում “կոնկատենացիա” գործողության մեկնաբանությունն այլ է: Սահմանվել է բազմաժապավեն վերջավոր ավտոմատների պատահույթների հանրահաշիվը մեկ ժապավենանոց ավտոմատների պատահույթների հանրահաշվի սահմանման նմանությամբ: Ցույց է տրվել, որ ներմուծված հանրահաշիվը Քլինի հանրահաշիվ է: Ինչպես նաև ցույց է տրվել, որ մեկ ժապավենանոց ավտոմատների պատահույթների հանրահաշվի վերաբերյալ որոշ արդյունքներ ճիշտ են նաև այս հանրահաշվի դեպքում: