

# Some more on the equivalent transformation of nondeterministic finite automata.

## Part I. Notation and the “combining” algorithm

B. F. Melnikov

**Abstract**—This paper can be viewed as a continuation of our following previous papers, where we considered some simple algorithms for combining states of the given nondeterministic finite automaton, the reduction some problems related to the star-height to considering automata, and possible classification of the states and loops of the given automaton.

In this paper, we shall describe an algorithm which combines some states of a given nondeterministic finite automaton. However, unlike algorithms published before, we have more stringent requirements for two combined states of the considered automaton. Besides, we obtain (after combining these two states) the automaton, which is not only equivalent to the given one, but also has the value of star-height which is no more than such value for the given automaton. We also consider an example of using the described combining algorithm.

In the following parts of this paper, we are going to describe the algorithms for deleting and adding a state. These algorithms will have the same features of transformations, i.e. the values of star-height for the obtained automata will be no more than such value for the given automaton.

**Keywords**—nondeterministic finite automata, regular languages, equivalent transformations, combining states, the star-height problem.

### I. INTRODUCTION AND MOTIVATION

This paper can be viewed as a continuation of our previous papers:

- [1]: we considered there some simple algorithms for combining states of the given nondeterministic finite automaton;
- [2]: we considered there some applications of such algorithms; in particular, we constructed any automaton for the given regular language using special equivalent transformations of the basis automaton (see [3] etc.) for this language;
- [4]: we described there an alternative proof of Kleene’s theorem; on the basis of this proof, we reduced some problems related to the star-height to nondeterministic finite automata;
- [5]: we considered there some questions of the possible classification of the states and loops of a nondeterministic finite automaton;
- [6]: we continued there the classification mentioned in the previous item; we also defined some terms associated with the loops of the transition graph of the basis automaton for the given regular language (i.e. we defined so-called including loops, partially complete loops and

complete cyclic states) and considered some properties of these objects; after that, we formulated an important hypothesis about them;

- [7]: we led there the possible application of this theory using the results of some papers, published before in Russian.

In the current paper, we change the process of presentation of the whole material, provide some other proofs and consider more examples.

By [4], we reformulated the star-height problem for regular languages in the following way: for the given regular language, we have to construct the equivalent finite automaton having the minimum possible star-height. After that, considering  $n!$  bijective functions (“order functions”; the value  $n$  is the number of states of this automaton), we construct corresponding regular expressions and choose the expression having the minimum possible star-height. Thus, a possible solution of the star-height problem for regular language is constructing such “minimum” automaton. To build such an automaton, we perform some equivalent transformations; they are described in the papers cited before and in this paper. These description of such transformations is the main subject of this paper.

It follows from the title of this paper (more precisely, its Part I), that we shall describe an algorithm which combines some states of a given nondeterministic finite automaton. However, unlike [1], [2]:

- we have more stringent requirements for the states of the considered automaton;
- on the other hand, we obtain (after combining two states) the automaton, which is:
  - not only equivalent to the given one,
  - but also has the value of star-height which is no more than such value for the given automaton.

In the remaining parts of the paper, we shall consider the similar features of transformations, but for other algorithms.

This paper has the following structure. In Section II, we describe in detail the notation used in this paper; this notation is almost the same as one used, for example, in [1], [4], [5], [8]. In Section III, we add some designations related with ordering functions used for the states of considered automaton. In Section IV, we add the notation connected with adding/deleting states/edges for the considered automaton.

In Section V, we describe an algorithm which combines two states of the considered automaton. After its using, we obtain new automaton which is equivalent to the given one and also has the value of star-height which is no more than such value for the given automaton. And in Section VI we

Received February 2, 2019.

Boris F. Melnikov, Russian State Social University & Lomonosov Moscow State University (email: bf-melnikov@yandex.ru).

consider an example for the combining algorithm described before.

In the following parts of this paper, we are going to describe the algorithms for deleting and adding a state.

## II. PRELIMINARIES

In this section, we shall describe in detail the notation used in this paper. This notation is almost the same as one used in [1], [4], [5], [8] etc.

Like most of the cited papers, the “main” automaton under consideration will be denoted by

$$K = (Q, \Sigma, \delta, S, F). \quad (1)$$

We shall consider transition function  $\delta$  of automaton (1) as

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q),$$

but not as

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q),$$

The language of this automaton is  $L = \mathcal{L}(K)$ .

We shall write some edge  $\delta(q, a) \ni r$  in the form  $q \xrightarrow[\delta]{a} r$ , or, if it does not cause discrepancies, simply in the form  $q \xrightarrow{a} r$ . Sometimes, we shall consider the transition function  $\delta$  as *the set of such edges*.

The mirror automaton for the automaton given in (1), i.e.

$$(Q, \Sigma, \delta^R, F, S),$$

where

$$q' \xrightarrow[\delta^R]{a} q'' \quad \text{if and only if} \quad q'' \xrightarrow[\delta]{a} q',$$

will be denoted by  $K^R$ ; note that  $K^R$  defines the language  $L^R$ .

For the considered regular language  $L$ , its automaton of canonical form will be denoted by  $\tilde{L}$ . Let automata  $\tilde{L}$  and  $\tilde{L}^R$  for the given language  $L$  be as follows:

$$\tilde{L} = (Q_\pi, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi)$$

and

$$\tilde{L}^R = (Q_\rho, \Sigma, \delta_\rho, \{s_\rho\}, F_\rho).$$

We do not consider the language  $L = \emptyset$ , then both these automata have initial states.

Let us recall definitions of binary relation  $\#$  and state-marking functions  $\varphi^{in}$  and  $\varphi^{out}$ . Relation

$$\# \subseteq Q_\pi \times Q_\rho$$

is defined for pairs of states of automata  $\tilde{L}$  and  $\tilde{L}^R$  in the following way:  $A \# X$  if and only if

$$(\exists uv \in L) (u \in \mathcal{L}_{\tilde{L}}^{in}(A), v^R \in \mathcal{L}_{\tilde{L}^R}^{in}(X)).$$

State-marking function

$$\varphi_K^{in} : Q \rightarrow \mathcal{P}(Q_\pi)$$

is defined in the following way:

$$\varphi_K^{in}(q) \ni \tilde{q} \quad \text{if and only if} \quad \mathcal{L}_K^{in}(q) \cap \mathcal{L}_{\tilde{L}}^{in}(\tilde{q}) \neq \emptyset.$$

And state-marking function

$$\varphi_K^{out} : Q \rightarrow \mathcal{P}(Q_\rho)$$

is defined similarly for automata  $K^R$  (the mirror automaton for  $K$ ) and  $\tilde{L}^R$ .

The definition for basis automaton for the given regular language  $L$  could be found in [3] etc. In this paper, it will be denoted by

$$\mathcal{BA}(L) = (\hat{Q}, \Sigma, \hat{\delta}, \hat{S}, \hat{F}).$$

Binary relation  $\#$  defined in the above manner forms also the set of so-called *pseudo-grids*: namely, each of them is a pair  $(P, R)$  (where  $P \subseteq Q_\pi$  and  $R \subseteq Q_\rho$ ), such that for each pair of states  $p \in P$  and  $r \in R$  condition  $p \# r$  holds. Each of such pseudo-grids corresponds to the state of any particular automaton for the given language. Moreover, the necessary condition for defining the given language by a finite automaton is that the subset of pseudo-grids corresponding to the set of states of considered automaton cover all the items of the relation  $\#$ .

And if for some pseudo-grid  $(P, R)$  we can not extend neither set  $P$  nor the set  $R$  in order to not violate the definition of a pseudo-grid, then we call such a pseudo-grid by a *grid*. In fact, we use *pseudo-grids* in this paper.

Thus, we can assume that by considering some given regular language  $L$ , we simultaneously introduce the notation for the related language:

- two canonical automata (i.e.,  $\tilde{L}$  and  $\tilde{L}^R$ ), and also their states, their transition functions etc.;
- binary relation  $\#$  defined on pairs of states of automata  $\tilde{L}$  and  $\tilde{L}^R$ ;
- state-marking functions  $\varphi^{in}$  and  $\varphi^{out}$ ;
- equivalent basis automaton  $\mathcal{BA}(L)$ ;
- equivalent automaton  $\mathcal{COM}(L)$ .<sup>1</sup>

If the transition graph of automaton (1) includes a path from  $q' \in Q$  to  $q'' \in Q$ , we shall write  $\mathcal{V}(q', q'')$ . If

$$\mathcal{V}(q', q'') \quad \text{and} \quad \mathcal{V}(q'', q'),$$

then we shall write  $\mathcal{W}(q', q'')$ . We shall complete these definitions in the next section.

For the further, also the papers [5], [6] are important. We considered there some questions of the possible classification of the states and loops of a nondeterministic finite automaton. For this, we consider the basis finite automaton for the given regular language ([1] etc.) and the paths and loops of its transition graph. We also consider the paths and loops of the transition graph of another nondeterministic automaton that defines the same language; that is (1) in our notation. On the basis of this, we define corresponding paths and loops of two mentioned automata and the questions of their classification.

We also described the following objects there. For each state of the basis automaton, we consider the states of (1) corresponding to this state of the basis automaton, and give their classification as a function of the loops passing through the same state of the basis automaton. Their subset is the set of so-called including loops, on the basis of which we determine so-called partially complete loops. For any chosen vertex of the basic automaton, we call the vertices of the considered nondeterministic automaton, through which all possible partially complete loops pass, by complete cyclic states. (See in [5], [6] the more detailed classification of states and loops.)

<sup>1</sup> Directly, automaton  $\mathcal{COM}(L)$  is not used in this paper. However, the “minimum” automaton considered in this paper can be defined as the sub-automaton of  $\mathcal{COM}(L)$ .

In the end of the section, we shall describe the procedure of combining two states (see this notation in [1, Sect. 2]). According to the name of the part of the paper, this designation is one of the main.

**Definition 1:** For the automaton (1) and some its states  $q_1, q_2 \in Q$ , let us denote by  $\mathcal{J}^{q_1 q_2}(K)$  automaton, which transition graph is obtained from the transition graph of automaton  $K$  in the following way<sup>2</sup>:

- for each vertex  $r \in Q$ , we change the sets of edges of the type  $\gamma(q_1, r)$  for the sets  $\gamma(q_1, r) \cup \gamma(q_2, r)$ ;
- for each vertex  $r \in Q$  we change the sets of edges of the type  $\gamma(r, q_1)$  for the sets  $\gamma(r, q_1) \cup \gamma(r, q_2)$ ;<sup>3</sup>
- $q_2$  is deleted, together with all the corresponding elements of the function  $\gamma$ .

In the obtained automaton, the vertex  $q_1$  is an initial (a final) vertex if and only if at last one of two vertices  $q_1, q_2$  was an initial (a final) one for automaton  $K$ .  $\square$

### III. NOTATIONS ASSOCIATED WITH THE ORDER FUNCTION

In this section, we add some designations related with ordering functions used for the states of considered automaton. For the further, like [4], we shall use the order function  $\tau$  for the given automaton (1) of the type

$$\tau : Q \rightarrow \mathbb{R}^+.$$

Using such functions, we defined in [4] the star-height of the finite automaton. We shall not repeat it in this paper (it is long), although we shall use it below.

We shall suppose below, that the set  $Q$  is the following:

$$Q = \{q_1, q_2, \dots, q_n\},$$

where

$$\tau(q_1) < \tau(q_2) < \dots < \tau(q_n).$$

We shall also write, e.g.,  $p < r$  meaning  $\tau(p) < \tau(r)$ , use “max” meaning

$$\max(p, r) = \begin{cases} p, & \text{if } \tau(p) \geq \tau(r) \\ r, & \text{if } \tau(p) < \tau(r), \end{cases}$$

etc. Let us fix  $K$  and  $\tau$ . Using these agreements and some value  $m \in \{1, \dots, n\}$ , let us define the following sets, which complete the definitions of the previous section.

**Definition 2:** For some states  $q, p, r \in Q$ , where  $p \geq q$  and  $r \geq q$ , let us consider some simple path from  $p$  to  $r$ ,<sup>4</sup> whose sequence of vertices is

$$(p, q_1, q_2, \dots, q_s, r)$$

such that

$$s \geq 0 \quad \text{and} \quad (\forall i \in \{1, \dots, s\}) (q_i > q).$$

We shall denote the set of *all* such paths by  $\Delta_q(p, r)$ .

<sup>2</sup> We consider this definition for the transition graph (having vertices marked by inputs and outputs), but not for elements of the 5-tuple (1). However, this definition is strict. If necessary, it can be reformulated for the objects of (1).

<sup>3</sup> Thus, for each of 4 opportunities  $q', q'' \in \{q_1, q_2\}$ , each edge  $\gamma(q', q'') \ni a$  becomes the loop  $\gamma(q_1, q_1) \ni a$ .

<sup>4</sup> We allow  $p = r$ . In this case, i.e., if  $p = r$ , it is a simple loop.

We set

$$\Delta^q = \{q' \text{ is a state of a path of } \Delta_q(q, q) \quad \text{and} \quad q' \neq q\}.$$

If  $\Delta_q(p, r) \neq \emptyset$ , then we shall write  $\mathcal{V}_q(p, r)$  (otherwise  $\overline{\mathcal{V}_q(p, r)}$ ).

If  $\mathcal{V}_q(p, r)$  and  $\mathcal{V}_q(r, p)$ , then we shall write  $\mathcal{W}_q(p, r)$ .  $\square$

**Definition 3:** Let us define the following sets

$$Q_m = \{q_1, q_2, \dots, q_{m-1}\}$$

and

$$T_m = \{q_{m+1}, q_{m+2}, \dots, q_n\},$$

and also their subsets

$$\hat{Q}_m = \{q \in Q_m \mid \Psi(q) = \Psi(q_m)\}$$

and

$$\hat{T}_m = \{q \in T_m \mid \Psi(q) = \Psi(q_m)\}.$$

Let us define also the following sets of pairs:

$$\hat{\Psi}_m = \bigcup_{q \in \hat{Q}_m} \hat{\Psi}(q_i).$$

A state is called *important state*, if the following condition holds:

$$(\exists (A, X) \in \hat{\Psi}(q_m)) ((A, X) \notin \hat{\Psi}_m). \quad \square$$

### IV. ADDITIONAL NOTATION

We also shall use the following notation. It is connected with adding/deleting states/edges for the considered automaton.

**Definition 4:** For a state  $q \in Q$ , let us define automaton  $K_{-q}$ , Obtained from automaton (1) by deleting the state  $q$ .<sup>5</sup>

For the edge  $r \xrightarrow{a} t$  (this edge may be contained or may not be contained in the given transition function  $\delta$ ), let us define automaton

$$K_{+(r \xrightarrow{a} t)},$$

obtained from automaton (1) by adding this edge.<sup>6</sup>  $\square$

**Definition 5:** Let automaton (1) be given, besides,  $\mathcal{L}(K) = L$ ; then automaton

$$K' = (Q', \Sigma, \delta', S', F')$$

will be called a *quasi-subset* of automaton  $K$ , if for some function  $\omega : Q' \rightarrow Q$  the following conditions hold:

- for each state  $q' \in Q'$ :
  - if  $q' \in S'$ , then  $\omega(q') \in S$ ;
  - if  $q' \in F'$ , then  $\omega(q') \in F$ ;<sup>7</sup>
- for each pair of states  $q', r' \in Q'$  and letter  $a \in \Sigma$ :
  - if  $q' \xrightarrow{\delta'} r'$ , then  $\omega(q') \xrightarrow{\delta} \omega(r')$ .  $\square$

Let us remark, that if an automaton is a subset of another one (i.e., it can be obtained by removing some states and edges), then it can be considered as its quasi-subset.

**Proposition 1:** Let automaton  $K'$  be a quasi-subset of automaton  $K$ . Then  $\mathcal{L}(K') \subseteq \mathcal{L}(K)$ .  $\square$

<sup>5</sup> We shall not formulate the strict definition of automata  $K_{-q}$  (for various states  $q$ ), i.e. we shall not define their initial and final states and the transition functions: these definitions are obvious.

<sup>6</sup> This designation will be used in the next parts of this paper.

<sup>7</sup> It means that the state  $q'$  can be an initial (a final) state, if the same property has the state  $\omega(q')$ . Similarly for the next item.

V. THE COMBINING ALGORITHM

In this section, we shall describe an algorithm which combines two states of the considered automaton. After its using, we obtain new automaton which is equivalent to the given one and also has the value of star-height which is no more than such value for the given automaton.

Thus, for the states of the given automaton (1) we shall use notation of Section III. Let also the order function  $\tau$  and the number  $m$  be given.

**Theorem 1:** Let  $q \in T_m$ . Let also

$$\mathcal{W}_{q_m}(q_m, q) \tag{2}$$

or

$$\overline{\mathcal{V}_{q_m}(q_m, q)} \ \& \ \overline{\mathcal{V}_{q_m}(q, q_m)}. \tag{3}$$

Consider automaton

$$K' = \mathcal{J}^{q_m q}(K).$$

Then

$$\mathcal{L}(K') = \mathcal{L}(K) \ \text{and} \ \mathcal{SH}(K', \tau) \leq \mathcal{SH}(K, \tau).$$

**Proof.** The first equality is a consequence of the theorem [1, Th. 1].

Let us prove the second fact. First, we note the following. According to the definitions given in [1], using notation  $\mathcal{J}^{q_m q}$ , we mean that the state  $q$  is deleted (and the state  $q_m$  is kept). Besides, because the inequality  $\tau(q_m) < \tau(q)$  holds (according to the agreements entered before, i.e.  $q \in T_m$ ), then after such deleting, we shall consider that state of two ones ( $q$  and  $q_m$ ), for which the value  $\tau$  is less.

For the first subcase (i.e. (2)), condition  $q \in \Delta^{q_m}$  holds, then  $\Delta^q \subseteq \Delta^{q_m}$ . Therefore by [4, Sect.III], we do not increase the star-height of the considered automaton when combining  $\mathcal{J}^{q_m q}$ .

For the second subcase (i.e. (3)), condition  $\Delta^q \cap \Delta^{q_m} = \emptyset$  holds. Then, also combining the states  $q$  and  $q_m$ , we obtain for the resulting automaton  $K' = \mathcal{J}^{q_m q}$ : for each pair consisting of states  $q_1 \in \Delta^q$  and  $q_2 \in \Delta^{q_m}$ , the following condition holds<sup>8</sup>:

$$\overline{\mathcal{W}_{q_m}(q_m, q)}$$

(otherwise, for the given automaton  $K$  one of the conditions would be met:  $\mathcal{V}_{q_m}(q_1, q_2)$  or  $\mathcal{V}_{q_m}(q_2, q_1)$ ); therefore, we also do not increase the value of star-height of the considered automaton.  $\square$

Thus, having considered this case, as a result of the described transformations, we reduce the number of its states and do not increase its star-height. (In other words, the “combining” algorithm *does not “complicate”* the considered automaton.)

VI. AN EXAMPLE OF THE COMBINING ALGORITHM

In this section, we shall consider an example for the combining algorithm described before. All the examples that we consider in this part of the paper and are supposed to be considered in the following parts are made on the basis of the basis automaton for the language of the regular expression

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

<sup>8</sup> Automaton  $K'$  does not contain the state  $q$ ; but in spite of this, we can consider the set of states  $\Delta^q$  selected before.

More precisely, we shall consider a modified basis automaton, for which we “doubled” 3 states; we already considered it in [5, Fig. 9].

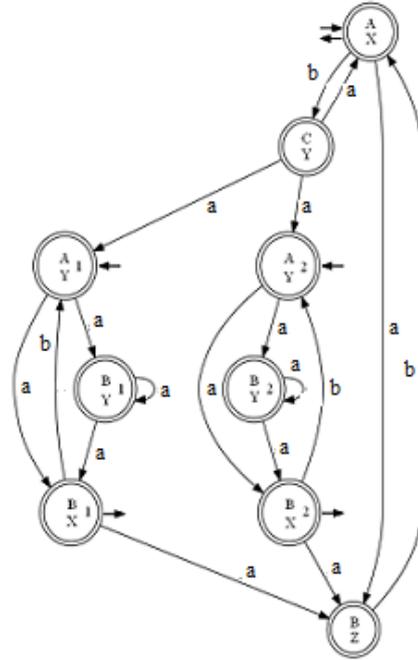


Figure 1. The given automaton (for the language  $(a + ab + ba)^*$ )

Certainly, dealing with the basis automata (and even automata obtained from them by “doubling” states) we cannot consider the complex examples: for instance, all the sets  $\Psi$  and  $\tilde{\Psi}$  for such examples will consist of 1 element only. However, as can be understood from the consideration of the example below, more complex examples are unlikely to be considered in the length of the article.

The next figure shows a possible function  $\tau$ :

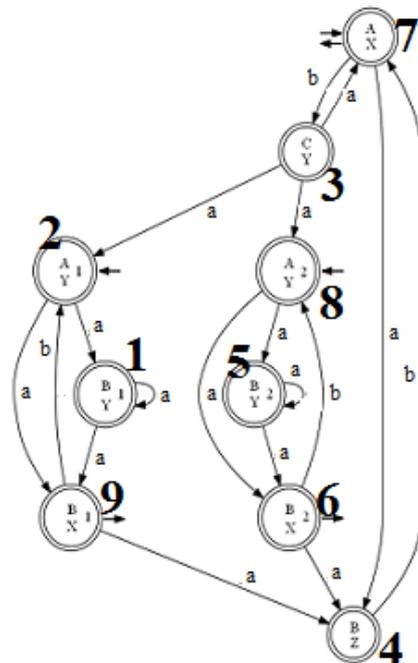


Figure 2. The previous automaton and an order function for it

i.e.,

$$\tau\left(\begin{smallmatrix} B \\ Y \end{smallmatrix} 1\right) = 1, \tau\left(\begin{smallmatrix} A \\ Y \end{smallmatrix} 1\right) = 2, \dots, \tau\left(\begin{smallmatrix} B \\ X \end{smallmatrix} 1\right) = 9.$$

Owing to previous notation,

$$\begin{smallmatrix} B \\ Y \end{smallmatrix} 1 = q_1, \begin{smallmatrix} A \\ Y \end{smallmatrix} 1 = q_2, \dots, \begin{smallmatrix} B \\ X \end{smallmatrix} 1 = q_9.$$

For this automaton and this function, we obtain the following values of star-height. This value is counted for the automaton including the specified states only and order function  $\tau$  given before. Let us especially remark, that this is  $\mathcal{SH}(K, \tau)$  (for *different* automata  $K$ ), but not  $\mathcal{SH}(K)$ .

Tab. 1

states of automaton	$\mathcal{SH}(K, \tau)$
{q <sub>9</sub> }	0
{q <sub>8</sub> , q <sub>9</sub> }	0
{q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	0
{q <sub>6</sub> , q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	1
{q <sub>5</sub> , q <sub>6</sub> , ..., q <sub>9</sub> }	2
{q <sub>4</sub> , q <sub>5</sub> , ..., q <sub>9</sub> }	2
{q <sub>3</sub> , q <sub>4</sub> , ..., q <sub>9</sub> }	3
{q <sub>2</sub> , q <sub>3</sub> , ..., q <sub>9</sub> }	4
{q <sub>1</sub> , q <sub>2</sub> , ..., q <sub>9</sub> }	5

(we defined  $\mathcal{SH}(\emptyset^*) = 0$ , see [4] etc.).

Remark once again, that we consider not the star-height of the considered automata, and certainly not star-height of its languages<sup>9</sup>, but values  $\mathcal{SH}(K, \tau)$ .

Then let  $m = 1$ , then  $q_1 = \begin{smallmatrix} B \\ Y \end{smallmatrix} 1$  and  $q = q_5 = \begin{smallmatrix} B \\ Y \end{smallmatrix} 2$  (i.e., we shall combine these states). After combining, we obtain the following automaton:

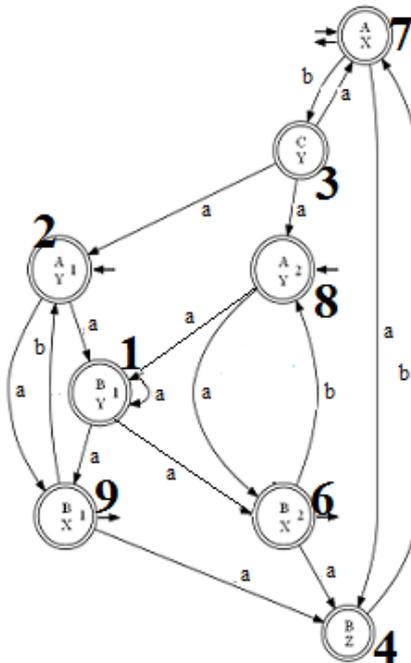


Figure 3. The previous automaton after combining  $q_1 = \begin{smallmatrix} B \\ Y \end{smallmatrix} 1$  and  $q_5 = \begin{smallmatrix} B \\ Y \end{smallmatrix} 2$

For the last automaton, let us obtain the similar table for the values  $\mathcal{SH}(K, \tau)$  (we can use the same function  $\tau$ , despite the automaton does not contain the state  $q_5$ ):

<sup>9</sup> As we know, the star-height of the language  $(a + ab + ba)^*$  is equal to 1. We also can prove, that the star-height of *each language* of 9 automata considered before is no more than 1.

Tab. 2

states of automaton	$\mathcal{SH}(K, \tau)$
{q <sub>9</sub> }	0
{q <sub>8</sub> , q <sub>9</sub> }	0
{q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	0
{q <sub>6</sub> , q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	1
{q <sub>4</sub> , q <sub>6</sub> , q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	1
{q <sub>3</sub> , q <sub>4</sub> , q <sub>6</sub> , q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	2
{q <sub>2</sub> , q <sub>3</sub> , q <sub>4</sub> , q <sub>6</sub> , q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	3
{q <sub>1</sub> , q <sub>2</sub> , q <sub>3</sub> , q <sub>4</sub> , q <sub>6</sub> , q <sub>7</sub> , q <sub>8</sub> , q <sub>9</sub> }	4

Certainly, the value  $\mathcal{SH}(K, \tau)$  does not increase compared to the same value for Tab. 1.

ON THE FOLLOWING PARTS OF THIS PAPER

As we said in Introduction, we are going to describe the algorithms for deleting and adding a state in the following parts of this paper. It is important to note the following: from our point of view, even the “adding” algorithm *does not “complicate”* the considered automaton; the details will be described later.

REFERENCES

- [1] Melnikov B. *Once more on the edge-minimization of nondeterministic finite automata and the connected problems*. Fundamenta Informaticae. 2010, vol. 104, no. 3, pp. 267–283.
- [2] Melnikov B., and Sayfullina M. *On some algorithms for the equivalent transformation of nondeterministic finite automata*. Izvestiya of Higher Educational Institutions. Mathematics. 2009, no. 4, pp. 67–72. (in Russian, <https://elibrary.ru/item.asp?id=11749888>)
- [3] Melnikov B., and Melnikova A. *Some properties of the basis finite automaton*. The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). 2002, vol. 9, no. 1, pp. 135–150.
- [4] Melnikov B. *The star-height of a finite automaton and some related questions*. International Journal of Open Information Technologies. 2018, vol. 6, no. 7, pp. 1–5. (<http://injoit.org/index.php/j1/article/view/581>)
- [5] Melnikov B., and Melnikova A. *An approach to the classification of the loops of finite automata. Part I: Long corresponding loops*. International Journal of Open Information Technologies. 2018, vol. 6, no. 9, pp. 9–14. (<http://injoit.org/index.php/j1/article/view/613>)
- [6] Melnikov B., and Melnikova A. *An approach to the classification of the loops of finite automata. Part II: The classification of the states based on the loops*. International Journal of Open Information Technologies. 2018, vol. 6, no. 11, pp. 1–6. (<http://injoit.org/index.php/j1/article/view/620>)
- [7] Melnikov B. *On the star-height of a regular language. Part IV: The detailed description of the algorithm of equivalent transformation*. Heuristic algorithms and distributed calculations. 2014, vol. 1, no. 5, pp. 57–75. (in Russian, <https://elibrary.ru/item.asp?id=23033967>)
- [8] Melnikov B. *Extended nondeterministic finite automata*. Fundamenta Informaticae. 2010, vol. 104, no. 3, pp. 255–265.

Boris Feliksovich MELNIKOV,  
 Professor of Russian State Social University  
 (<http://www.rgsu.net/>),  
 Professor of Lomonosov Moscow State University  
 (<https://www.msu.ru/>),  
 email: bf-melnikov@yandex.ru,  
 mathnet.ru: personid=27967,  
 elibrary.ru: authorid=15715,  
 scopus.com: authorId=55954040300.