Some more on the equivalent transformation of nondeterministic finite automata. Part III. The “adding” algorithm

B. F. Melnikov

***Abstract*—This paper is a continuation of two previous parts. In them, we considered some simple algorithms for combining and removing (deleting) states of the given nondeterministic finite automaton, as well as the reduction some problems related to the star-height to considering automata. To do this, we used the possible classification of the states and loops of the given automaton.**

In this part of the paper, we shall describe an algorithm which adds some states and edges to the given nondeterministic finite automaton. This algorithm preserves the basic properties of automata: the languages of the given and the obtained automata are the same, and the value of star-height for the obtained automaton is no more than such value for the given automaton.

This algorithm is more complicated than the ones considered in the previous parts due to the following conditions. Here, we consider the case, when there exists the only direction for the paths between two considered states (we denoted them q and q_m in previous parts). I.e., the path in the direction of all edges, or, vice versa, in the reverse direction; but not in both directions at the same time. (Like previous parts, we consider only such paths that do not pass through edges with smaller numbers, which are less than the value of q_m; more strictly, all the considered vertices of such path should have the values of ω-function defined for the “minimum” automaton less than q has.)

The simple algorithms for combining these two states q and q_m (similar to the algorithm discussed in Part I) increase the $\mathcal{SH}$-value of the automaton under consideration. To prevent this increase, we use the removing (deletion) instead of combining, but, unlike Part II, we have to add some new elements (i.e. vertices and edges) to the transition graph of the automaton we transform.

***Keywords*—nondeterministic finite automata, regular languages, equivalent transformations, adding state, adding edge, the star-height problem.**

XIII. Introduction to Part III (Once more about the motivation)

This paper is the continuation of [1], [2], i.e. Parts I and II. [1] We continue the numeration of sections, equations, definitions, propositions, theorems, tables, and figures, but use the new numbers of references and footnotes.

As we said in [3] before, 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 “order” functions [2], we construct corresponding regular expressions and choose the one 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 auxiliary equivalent transformations. *The description of such transformations is the main subject of the all three parts of this paper.*

In this part, we shall consider the most difficult case. In it, there is any path along the edges of transition graph of the automaton from the previously fixed vertex to the vertex q under consideration *in exactly one direction.* (I.e., either in the direction of all edges, or, vice versa, in the reverse direction; but not in both directions at the same time. Like previous parts, the labels of the edges do not matter.) Before, we denoted this previously fixed vertex q_m, the designation for it coincides with ones fixed and used in Parts I and II; see also [4], [5]. At the same time, like previous parts, we consider only such paths that *do not pass through edges with smaller numbers* (which are less than the value of q_m; more strictly, all considered vertices of the path have less than q values of ω-function defined for the “minimum” automaton under consideration.)

The complication of Part III we mentioned before is that *the simple algorithms for combining these two states* (similar to the algorithms discussed in the previous parts) *increase the $\mathcal{SH}$-value*. To prevent this increase, we use the removing (deletion) instead of combining, but, unlike Part II, we have to add new elements (i.e. vertices and edges) to the transition graph of the automaton we transform. And, adding these elements, we must ensure that the $\mathcal{SH}$-value does not increase.

Thus, in Part III, we describe the algorithm for *special adding states.* [3] This algorithm will also have the same feature of transformations, i.e. the values of star-height for the obtained automata will be no more than such value for the given automaton. Like Part I (the combining case of Part I), we shall add *not only the edges, but also the states*. Like previous parts, we “improve the structure” of the automaton under consideration; this will help to subsequently apply one

Received October 7, 2019.

Boris F. Melnikov, Shenzhen MSU – BIT University (email: bf-melnikov @yandex.ru).

[1] In Part II, there was a misprint in the same section: it was said that Part II is not a continuation of Part I, but of the other paper. In fact, this is *partly true*, but, certainly, Part II is a continuation of Part I. We hope, that in the current Part III, there are no reference misprints.

[2] n is the number of the states of the “minimum” automaton.

[3] In the conclusion of Part II, we called this process: “special adding a state”. This is also true, because all states added in Part III (there are, generally speaking, several such states) duplicate the one considered state of the changed automaton. The text of the current part (“special adding states”) is some more accurate.

of the algorithms discussed in the first two parts. But this "improving", like before, does not imply the "complication" of the automaton.

Let us note in advance, that in this part of the paper hardly any examples are possible (of the type of ones we considered in the previous two parts). This can be explained by the fact that the situation described in Part III is hardly possible at all. Moreover, the author believes that a simple description of an example with such a possibility (or the proof that this is not possible) is the topic of a separate publication. But to describe completeness, we perform the transformations required to show how to transform an automaton in this situation.

XIV. The adding algorithm

As we said before, we shall here consider the case, when there exists any path along the edges of transition graph of the automaton from the previously fixed vertex q_m to the vertex q under consideration *in exactly one direction*. More strictly, all the considered vertices of such path should have the values of ω-function defined for the "minimum" automaton less than q has. In formulas, for each state $q \in \hat{T}_m$ (the set of states T_m for the considered state q_m was defined in Part I), we have

$$\begin{aligned}&\bigl(\overline{\mathcal{V}_{q_m}}(q_m,q)\;\&\;\mathcal{V}_{q_m}(q,q_m)\bigr)\\ \text{or}\;\;&\bigl(\mathcal{V}_{q_m}(q_m,q)\;\&\;\overline{\mathcal{V}_{q_m}}(q,q_m)\bigr).\end{aligned} \tag{17}$$

Thus, let us consider this case.

The modified automaton will be denoted by

$$\overline{K} = (\,Q \cup \overline{Q},\ \Sigma,\ \delta \cup \overline{\delta},\ S \cup \overline{S},\ F \cup \overline{F}\,),$$

and we shall describe new elements only (i.e., $\overline{Q}$, $\overline{\delta}$, $\overline{S}$, and $\overline{F}$)[4], i.e., elements which are *added* to the elements of K.

For the following description of changing ordering function[5], we choose a priori some values $\tau^{(q)}$ (for each state $q \in \hat{T}_m$). Choosing these values, we assume that they:

- are *different*;
- lie in interval $\bigl(0,\ \min\limits_{r,t\in Q,\ r\neq t} |\tau(r)-\tau(t)|\,\bigr)$.

We shall make the following transformations for *each* state $q \in \hat{T}_m$. At first, let us consider the first subcase of (17), changing there q for q'. I.e., we assume that for *some* $q' \in \hat{T}_m$, we have

$$\overline{\mathcal{V}_{q_m}}(q_m,q') \quad \& \quad \mathcal{V}_{q_m}(q',q_m)$$

(and we shall make the transformations for each such state q').

Firstly, we include in the set $\overline{\delta}$ the following elements:

$$\bigl\{\, r \xrightarrow[\overline{\delta}]{a} q_m \;\bigm|\; r \xrightarrow[\delta]{a} q' \,\bigr\}. \tag{18}$$

Secondly, we add in the set $\overline{Q}$ the states

$$\bigl\{\, r^{(q')} \bigm| r \in T'_m \,\bigr\}, \quad \text{where} \quad T'_m = T_m \setminus \{q'\}.$$

For these states, we assume that

$$\overline{S} \supseteq \bigl\{\, s^{(q')} \bigm| s \in S \,\bigr\} \quad \text{and} \quad \overline{F} \supseteq \bigl\{\, f^{(q')} \bigm| f \in F \,\bigr\}$$

(and the sets $\overline{S}$ and $\overline{F}$ contain *no other* elements having superscript $^{(q')}$). And we set values of the ordering function τ in the following way:

$$\tau\bigl(r^{(q')}\bigr) = \tau(r) + \tau^{(q')}$$

for the value $\tau^{(q')}$ chosen before.

Thirdly, we include in the set $\overline{\delta}$ the following elements:

$$\begin{aligned}&\bigl\{\, r^{(q')} \xrightarrow[\overline{\delta}]{a} t^{(q')} \bigm| r \in T'_m\,, t \in T'_m \cup Q_m,\ r \xrightarrow[\delta]{a} t \,\bigr\}\\ \cup\;\;&\bigl\{\, q_m \xrightarrow[\overline{\delta}]{a} t^{(q')} \bigm| t \in T'_m \cup Q_m,\ q' \xrightarrow[\delta]{a} t \,\bigr\}.\end{aligned} \tag{19}$$

The described process of adding edges can be demonstrated by Fig. 11 below. For this figure, let us make the following comments.

- When there is an opportunity, we tried to place the states (vertices) such that if $\tau(r) < \tau(t)$ then r lies higher than t. States (q' and q'') having the same values of both functions φ^{in}_K and φ^{out} as q_m has, are shown by shaded circles. The path passes q', then q_m, and then q'', shown by the firm line, is contained in automaton K. It symbolizes that we cannot have *two* the following paths which passes the vertices of T_m only: a path from q_m to a vertex having the same values of state-marking functions φ^{in}_K and φ^{out} and a back one; we can have only one of two such paths.
- The paths shown by chain lines are also contained in automaton K. These paths are kept in automaton $\overline{K}$, and for each of them (let it be ν), we also have a new path of automaton $\overline{K}$ (let it be $\overline{\nu}$). It is important to remark, that their words (i.e., the labels of these pairs of paths ν and $\overline{\nu}$) are different, but $\overline{\nu}$ (considered together with a needed path consisting of vertices of Q_m as a loop) includes the long loops corresponding to the loops of the basis automaton, which ν has. Furthermore, for the pair (A,X) (where state $\frac{A}{X}$ corresponds to q' in ν), we have *the only* entry of corresponding vertex for $\overline{\nu}$.
- Thus, the two new parts of such new path ($\overline{\nu}$) are shown by the dotted lines and marked on Fig. 11 by the word "new". Its first part (from vertex r) is an edge and is defined by (18); and its second part is defined by (19).

For another subcase, let us describe transformations briefly; they are similar. Thus, in the second subcase of (17), we change q for q'', assuming that for some $q'' \in \hat{T}_m$, we have

$$\mathcal{V}_{q_m}(q_m,q'') \quad \& \quad \overline{\mathcal{V}_{q_m}}(q'',q_m)$$

(and we also shall make the transformations for each such state q''). Firstly, we include in the set $\overline{\delta}$ the following elements: $\bigl\{\, q_m \xrightarrow[\overline{\delta}]{a} t \bigm| q'' \xrightarrow[\delta]{a} t \,\bigr\}$. Secondly, we include in the set $\overline{Q}$ the states

$$\bigl\{\, r^{(q'')} \bigm| r \in T''_m \,\bigr\},$$

where

$$T''_m = T_m \setminus \{q''\} \quad \text{and} \quad \tau\bigl(r^{(q'')}\bigr) = \tau(r) + \tau^{(q'')},$$

adding also

$$\overline{S} \supseteq \bigl\{\, s^{(q'')} \bigm| s \in S \,\bigr\} \quad \text{and} \quad \overline{F} \supseteq \bigl\{\, f^{(q'')} \bigm| f \in F \,\bigr\}.$$

Thirdly, we include in the set $\overline{\delta}$ the following elements:

$$\begin{aligned}&\bigl\{\, r^{(q'')} \xrightarrow[\overline{\delta}]{a} t^{(q'')} \bigm| r \in T''_m \cup Q_m\,, t \in T''_m,\ r \xrightarrow[\delta]{a} t \,\bigr\} \quad \cup\\ &\bigl\{\, t^{(q'')} \xrightarrow[\overline{\delta}]{a} q_m \bigm| t \in T''_m \cup Q_m,\ t \xrightarrow[\delta]{a} q'' \,\bigr\}.\end{aligned}$$

[4] As before, we consider transition functions as the *sets* of their elements.

[5] For both automata, i.e., for K and $\overline{K}$, we shall denote this functions by τ, without subscripts etc.

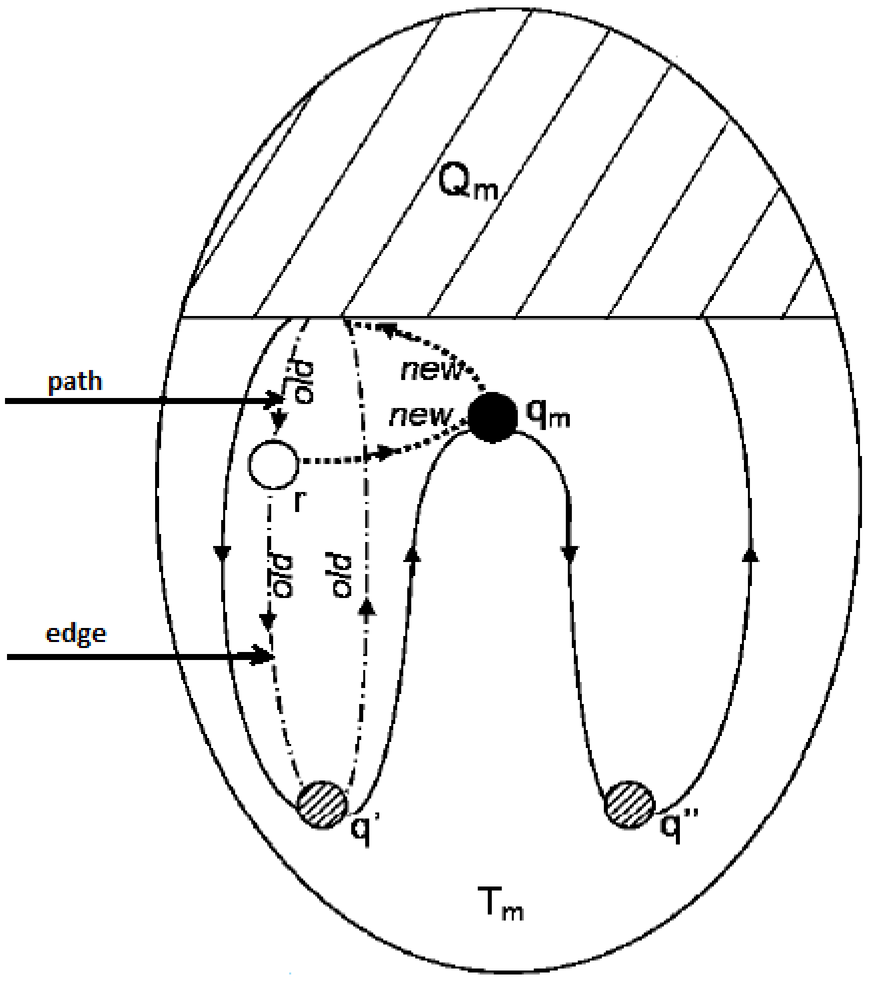

Figure 11.

Proposition 11: If the case of (17) holds, then:

- $\mathcal{L}(\overline{K}) = \mathcal{L}(K)$;
- $\mathcal{SH}(\overline{K}, \tau) = \mathcal{SH}(K, \tau)$.

Proof. K can be obtained from $\overline{K}$ by removing some states and edges, then K is a quasi-subset of $\overline{K}$ (see Part I, Definition 5).

Vice versa, each of added edge satisfies conditions of Proposition 8 (see Part II), then also by Definition 5, automaton $\overline{K}$ is a quasi-subset of K. Then by Proposition 1 (see Part I), we obtain equation $\mathcal{L}(\overline{K}) = \mathcal{L}(K)$.

Owing to the process of adding edges, if two vertices have different superscripts, then each path from the first in the second of them has to pass vertices of Q_m. And each vertex of Q_m have the value of the ordering function τ which is less than such value of each vertex having superscript; then changing automaton, we increase value of $\mathcal{SH}(q)$ neither for $q \in Q_m$ nor for $q \notin Q_m$. (Because for the vertices having superscripts, the value of their ordering function is the same as the one of corresponding vertices not having superscripts.) Therefore, we do not increase value of $\mathcal{SH}$ for the automaton in the process of considered transformations. □

XV. Conclusion

Thus, after considering the case of (17), we can claim that state q_m is also an *important* one (for important states, see Part I, Definition 3). In fact, we already have *informally* explained this statement before, describing added states $r^{(q')}$ and corresponding edges. I.e., we have formulated there, why we can choose the expected long loop having *the only* state (i.e., q_m), such that:

- it has values of state-marking functions $\varphi_K^{in}(q_m)$ and $\varphi_K^{out}(q_m)$;[6]
- it corresponds to at least 1 state of the basis automaton, which was *not* included in set $\hat{\Psi}_m$ before.

Therefore, we can repeat the modifications (i.e., consider one of three possible cases once again) for new number m which is 1 more that previous value.

Considering all the cases of Parts I–III, we increase the number of elements in each of the sub-cases considered in them for the sets $\hat{\Psi}(q_i)$, at least for one of the possible i. However, only a finite number of such increases are possible. Therefore, the number of executions of the case of (17) is also limited.

After the transformations we have described, all the states of automaton K are important. Therefore *we can in advance limit the number of states* that have to be considered to obtain the automaton, which has no states with the same values of both state-marking functions.

A very rough upper bound on the number of states of the transformed automaton can be obtained as follows. In [4],

[6] The obtained automaton is equivalent to the given one, then we *can* use such notation.

[6], [7], [8], we considered grids (blocks) and pseudo-grids; see also [9].

Let $n = \min\{\,|Q_\pi|, |Q_\rho|\,\}$ (i.e., the minimum of the numbers of states of the 2 considered canonical automata for the given language). Let $M(n)$ be the n-th Dedekind number, see [10], [11] etc. Then by [9], the number of grids (which is no more than the number of states of the corresponding universal automaton, or automaton $\mathcal{COM}(L)$) can be limited by the value $M(n)$.

Each grid can form no more than $(2^n-1)^2$ pseudo-grids; the total number of the pseudo-grids can be limited by the value $2^{2n} \cdot M(n)$. Each pseudo-grid can include no more than n^2 elements [7]; each such element corresponds to a state of automaton $\mathcal{BA}(L)$.

Consider pairs, where each pair consists of a pseudo-grid and an *corresponding state* of $\mathcal{BA}(L)$; for corresponding states, see [4]. Then the number of pairs, which are necessary for the consideration of important states, is no more than

$$2^{2n} \cdot M(n) \cdot n^2.$$

References

[1] Melnikov B. Some more on the equivalent transformation of nondeterministic nite automata. Part I. Notation and the "combining" algorithm // International Journal of Open Information Technologies. 2019, vol. 7, no. 4, pp. 1–5.

[2] Melnikov B. Some more on the equivalent transformation of nondeterministic nite automata. Part II. The "deleting" algorithm // International Journal of Open Information Technologies. 2019, vol. 7, no. 9, pp. 1–6.

[3] 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.

[4] Melnikov B., 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.

[5] Melnikov B., 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.

[6] Melnikov B., Sciarini-Guryanova N. Possible edges of a finite automaton defining a given regular language // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). 2002, vol. 9, no. 2, pp. 475–485.

[7] Melnikov B. The complete finite automaton // International Journal of Open Information Technologies. 2017, vol. 5, no. 10, pp. 9–17.

[8] Melnikov B. Regular languages and nondeterministic finite automata. – Russian State Social University Ed., 2018, 180 p. (in Russian)

[9] Lombardy S., Sakarovitch J. The universal automaton // Logic and Automata. Amsterdam University Press, 2008, pp. 457–504.

[10] Dedekind R. Über Zerlegungen von Zahlen durch ihre größten gemeinsamen Teiler // Gesammelte Werke 2. Braunschweig, 1897, pp. 103–148. (in German)

[11] Korshunov A. On the number of monotonous Boolean function // The Problems of Cybernetics. 1981, vol. 38, pp. 5–108. (in Russian)

Boris Feliksovich MELNIKOV,
Professor of Shenzhen MSU – BIT University, China
(`http://szmsubit.ru/`),
Professor of Russian State Social University
(`http://www.rgsu.net/`),
email: `bf-melnikov@yandex.ru`,
mathnet.ru: `personid=27967`,
elibrary.ru: `authorid=15715`,
scopus.com: `authorId=55954040300`,
ORCID: `orcidID=0000-0002-6765-6800`.

[7] In fact, no more than $n^2/4$ elements.