The star-height of a finite automaton and some related questions

B. F. Melnikov

***Abstract*—The paper is related to the star-height for nondeterministic finite automata, not for the star-height problem for regular languages. We describe an alternative proof of Kleene's theorem, and then our version of the reduction of some problems related to the star-height to nondeterministic finite automata. For a given regular language, the corresponding finite automaton is constructed; the method we are considering has the property that the reverse construction gives the original regular expression.**

The publication of this not-so-complicated problem has two goals, the both are related to the future development of the topic under consideration. First, we assume the future development of the topic with the aim of describing a similar approach for generalized regular expressions. Secondly, another generalization is proposed, namely, consideration of the structures we describe for the so-called extended automata. Thus, the material of this article is necessary for the definition of extended generalized pseudo-automata, which the author proposes to cite in the next publication.

***Keywords*—nondeterministic finite automata, regular languages, Kleene's theorem, star-height problem.**

I. Introduction and motivation

This paper summarizes some previous publications of the author. Among these publications, we note papers [1], [2], [3] (in chronological order).

In [1], only some schemes of algorithms were published.

In [3], an algorithm was given that relates Kleene's theorem and the star-height. This material has for now been published in Russian only; besides, due to the termination of Samara State University in 2015, the website of the journal is now unavailable.

In the current paper, the author gives the English version of [3], where also some noted misprints are corrected, and for some statements simpler proofs are given.

And the paper [2] was published long ago, it contained *another* algorithm that also describes the connection between Kleene's theorem and the star-height.

The current paper is related to the star-height *for nondeterministic finite automata*, its subject is practically unrelated to much more complex problems, i.e., to both "ordinary" star-height problem and generalized star-height problem *for regular languages*.

For these problems for the languages, let us only make some remarks. The first of mentioned problems, i.e., the "ordinary" star-height problem, was set in 1963 in [4] and solved in 1988 in [5]; in [6], however, this solution was called "extremely difficult".

In 2005, there was published the much more understandable solution of D. Kirsten, see [7]. In 2015, this solution was some improved using an approach of the game theory, see [8]. And unlike the previous comment, the last proof was called "elegant". Although it does improve the understanding of the proof of D. Kirsten, *the number of automata* being examined due to this "elegance" *does not decrease*.

The second of these mentioned problems is the *generalized* star-height problem. For now, it is unknown the answer to the question whether or not this problem is decidable. The author also does not know significant papers on this subject published after [9] (1992). The connection between the generalized star-height problem and generalized pseudo-automata (see [10] for the last formalism) is to be considered in one of the following publications.

In this paper we describe, firstly, an alternative proof of Kleene's theorem, and secondly, our version of the reduction of some problems related to the star-height to nondeterministic finite automata. Both these topics represent an alternative to the particular version of the reduction proposed back in 1963 in the above-mentioned paper [4]. As we already noted above, the subject of this paper is practically *unrelated* to the complex star-height problems. However, the publication of this not-so-complicated problem has two goals, the both ones are related to the future development of the topic under consideration. First, we assume the future development of the topic with the aim of describing a similar approach for generalized regular expressions (see [11] etc.) and generalized automata (see [10]). Secondly, another generalization is proposed; namely, we shall describe the consideration of our structures for so-called extended automata (see [12]) and, accordingly, extended pseudo-automata. Let us note, that all concepts needed for such extensions are present in this paper.

The structure of this paper is as follows. Section II briefly describes the used notation. In Section III, we consider a special version of the proof of Kleene's theorem.

In Section IV, we define the main object of this paper, i.e., the star-height of a finite automaton. This definition is built on the basis of its transition graph only, and is not associated with the marks of its edges.

In Section V, for a given regular language, the corresponding finite automaton is constructed. Of course, this problem was solved more than 60 years ago, but the method we are considering has the property that the "reverse construction" gives the original regular expression.

In Conclusion (Section VI), we formulate the directions for the further research.

II. Preliminaries

We shall not describe in detail the notation used in this paper; they are exactly the same as those used in [13]. Like

Received May 12, 2018.

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

there, the "main" automaton under consideration will be denoted by

$$K = (\, Q, \Sigma, \delta, S, F\,). \tag{1}$$

We shall use the classical definition of the star-height *of the regular expression*, see [7], [11]. We define it (denoting by $\boldsymbol{\mathcal{SH}(r)}$ for expression r) by the induction in the following way:

- $\mathcal{SH}(\varnothing) = \mathcal{SH}(\varnothing^*) = \mathcal{SH}(a) = 0$ for each $a \in \Sigma$;
- for each regular expressions r and s, we set

$$\mathcal{SH}((r+s)) = \mathcal{SH}((r \cdot s)) = \max(\,\mathcal{SH}(r), \mathcal{SH}(s)\,);$$

- for each regular expression r (where $r \neq \varnothing$), we set

$$\mathcal{SH}((r^*)) = \mathcal{SH}(r) + 1.$$

Remark that sometimes, another definition is considered: $\mathcal{SH}(\varnothing^*)$ is defined as 1. There is easy to prove, that such difference is important (i.e., two different definitions for expressions give two different values of the star-height of the regular *languages*) for *some finite* languages only.

Besides, some new notation will be described as necessary; most of these notations are associated with paths in the transition graph of the considered automaton.

III. A special version of the proof of Kleene's theorem

Let us consider a special version of the proof of Kleene's theorem. For the given automaton (1), let us consider some injective "ordering" function

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

We shall also write, e.g., $p < r$ meaning $\tau(p) < \tau(r)$; also we shall use notation "$\max$", 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 τ in this section.

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, whose sequence of vertices is

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

such that

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

(We allow $p = r$. In this case, i.e., if $p = r$, it is a simple loop.) We shall denote the set of *all* such paths by $\boldsymbol{\Delta_q(p,r)}$.

We also set

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

Some more notation:

- if $\Delta_q(p,r) \neq \varnothing$, then we shall write $\boldsymbol{\mathcal{V}_q(p,r)}$;
- otherwise, $\boldsymbol{\overline{\mathcal{V}_q}(p,r)}$;
- if both $\mathcal{V}_q(p,r)$ and $\mathcal{V}_q(r,p)$, then we shall write $\boldsymbol{\mathcal{W}_q(p,r)}$.

For each states $s, f \in Q$ (we allow $s = f$), we shall consider automaton $\boldsymbol{K_{s\to f}}$, defining as follows:

$$K_{s\to f} = (\, Q_{s\to f}, \Sigma, \delta_{s\to f}, \{s\}, \{f\}), \tag{2}$$

where

$$Q_{s\to f} = \{s, f\} \cup Q_{sf},$$
$$Q_{sf} = \{\, q \in Q \,|\, q > \max(s,f)\,\},$$

and $\delta_{s\to f}$ is constructed in the following way:

$$p \xrightarrow[\delta_{s\to f}]{a} r$$

if and only if

$$p \xrightarrow[\delta]{a} r,\ p \in \{s\} \cup Q_{sf},\ r \in \{f\} \cup Q_{sf}.$$

We shall construct regular expressions corresponding languages of automata of the type (2) by the induction on the value $\tau(\min(s,f))$. We shall denote *these* expressions by $\boldsymbol{\rho_{s\to f}}$; if $s = f$, we shall denote the automaton and the expression by $\boldsymbol{K_s}$ and $\boldsymbol{\rho_s}$.

(Let us remark, that we consider only expressions, obtained in the way described below. Certainly, each automaton has also infinitely many other corresponding regular expressions. But we shall not consider them in this section.)

Thus, let us consider the induction formulated before. Its *basis* is the following. If

$$\min(s,f) = q_{\max} = \max(\{\, q \,|\, q \in Q\,\}),$$

then automaton (2) (i.e., $K_{q_{\max}}$) defines the language, defined also by regular expression

$$\rho_{q_{\max}} = \left(\left\{\, a \in \Sigma \,|\, q_{\max} \xrightarrow{a} q_{\max}\,\right\}\right)^*; \tag{3}$$

remark that anyway $\rho_{q_{\max}} \ni \varepsilon$.

Step of induction. If $s \neq f$, then we write expression defining language of automaton $K_{s\to f}$ in the following way:

$$\rho_{s\to f} = \left\{\, a \in \Sigma \,|\, s \xrightarrow{a} f\,\right\} + \bigcup_{q > \max(s,f)} \rho_{s\to q} \cdot \rho_q \cdot \rho_{q\to f} \tag{4}$$

(for expression, $\cup$ symbolizes $+$'s). And if $s = f$, then we write expression defining language of automaton K_s in the following way:

$$\rho_s = \left(\left\{\, a \in \Sigma \,|\, s \xrightarrow{a} s\,\right\} + \bigcup_{q > s} \rho_{s\to q} \cdot \rho_q \cdot \rho_{q\to s}\right)^*. \tag{5}$$

By the hypothesis of inductions, all the expressions of the right parts of (4) and (5) are already constructed.

For some pair $s \in S$ and $f \in F$, denote

$$\mathcal{L}_{s\to f} = \mathcal{L}(K_{s\to f}) + \bigcup_{q < \min(s,f)} \mathcal{L}(K_{s\to q}) \cdot \mathcal{L}(K_q) \cdot \mathcal{L}(K_{q\to f}).$$

Then we can write language of the given automaton (1) (i.e., $\mathcal{L}(K)$) in the following way:

$$\mathcal{L}(K) = \bigcup_{s\in S,\ f\in F} \mathcal{L}(K_s) \cdot \mathcal{L}_{s\to f} \cdot \mathcal{L}(K_f).$$

Therefore, the corresponding regular expression is

$$\bigcup_{s\in S,\ f\in F} \rho_s \cdot \left(\rho_{s\to f} + \bigcup_{q < \min(s,f)} \rho_{s\to q} \cdot \rho_q \cdot \rho_{q\to f}\right) \cdot \rho_f. \tag{6}$$

We shall everywhere (i.e., both in this paper and in the future) use this formula (6). Although we note, that there also exists a simpler expression

$$\bigcup_{s\in S,\ f\in F} \left(\rho_s \cdot \Big(\bigcup_{q\in Q} \rho_{s\to q} \cdot \rho_q \cdot \rho_{q\to f}\Big) \cdot \rho_f\right);$$

where, for instance, the language defined by $\rho_{s\to f}$ is a subset of the language defined by

$$\bigcup_{q\in Q} \rho_{s\to q}\cdot\rho_q\cdot\rho_{q\to f}.$$

However, the last record, as is easy to verify, usually gives a much larger number of components ("terms") in the obtained regular expression.

IV. The star-height of a finite automaton

In this section, we define the main object of this paper, i.e., the star-height of a finite automaton It is important to note that this definition is built on the basis of its transition graph only, and is not associated with the marks of its edges, i.e., the letters.

Firstly, let us consider some propositions; they are simply the corollaries of the given before definitions of regular expressions ρ (i.e., (3)–(5)).

Proposition 1: $\mathcal{SH}(\rho_{q_{\max}})\le 1$. $\square$

Proposition 2:

$$\mathcal{SH}(\rho_{q_{s\to f}})=\max_{q\in\Delta_{\max(s,f)}(s,f)}\mathcal{SH}(\rho_q)\le\max_{q\in Q_{s\to f}}\mathcal{SH}(\rho_q).\ \square$$

Proposition 3: If $\Delta_s\neq\emptyset$, then

$$\mathcal{SH}(\rho_s)=\max_{q\in\Delta_s}\mathcal{SH}(\rho_q)+1\le\max_{q>s}\mathcal{SH}(\rho_q)+1.\quad\square$$

In previous section, we fixed K and τ. Then below, the obtained regular expression defining $\mathcal{L}(K)$ for the ordering function τ (i.e., (6)) will be denoted below by $\boldsymbol{\mathcal{R}(K,\tau)}$.

Let for automaton (1), its set Q consists of n states. We are able to consider $n!$ different injective functions τ (because for each pair of states $q,r\in Q$, only the value of predicate $\tau(q)<\tau(r)$ is important). These $n!$ different functions have, generally speaking, different regular expressions defined by (4) and (5).

Definition 1: The ***star-height of automaton*** (1) is defined in the following way:

$$\mathcal{SH}(K)=\min_{\tau\in\mathcal{T}}\mathcal{SH}(\mathcal{R}(K,\tau)),$$

where $\mathcal{T}$ is the set of all the *bijective* functions of the type $\tau: Q\to\{1,\dots,n\}$. $\square$

It is important to remark, that for defining $\mathcal{SH}(K)$, we have used exactly the way of constructing regular expressions, which was given in previous subsection. (Certainly, there exist other ways of constructing regular expressions by the given automaton.)

Let us consider a simple well-known example; the transition graph for the language of regular expression $(ab^*c)^*$ is given on Fig. 1. (The agreement on the use of single and double circles as a designation of states of the transition graph was given in [10].)

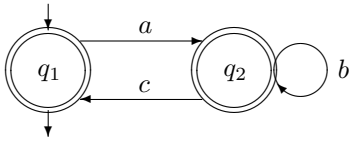

Fig. 1

To determine the star-height of this automaton, we have to consider 2 functions τ; let us call them τ_1 and τ_2. For the function

$$\big\{\,\tau_1(q_1)=1,\quad\tau_1(q_2)=2\,\big\},$$

we obtain, using (3)–(6) and some simplest equivalent transformations ([14] etc.), the following expressions:

$$\rho_{q_2}=b^*,\quad \rho_{q_2\to q_1}=\{c\},\quad \rho_{q_1\to q_2}=\{a\}$$

and

$$\rho_{q_1}=(ab^*c)^*,$$

and, therefore,

$$\mathcal{R}(K,\tau_1)=(ab^*c)^*\cdot(ab^*c)^*\cdot(ab^*c)^*. \tag{7}$$

Similarly, considering the function

$$\big\{\,\tau_2(q_1)=2,\quad\tau_2(q_2)=1\,\big\},$$

we obtain the following expressions:

$$\rho_{q_1}=\{\varepsilon\},\quad \rho_{q_1\to q_2}=\{a\},\quad \rho_{q_2\to q_1}=\{c\}$$

and

$$\rho_{q_2}=(ca+b)^*,$$

and, therefore,

$$\mathcal{R}(K,\tau_2)=\{\varepsilon\}\cdot\big(\{\varepsilon\}+a\cdot(ca+b)^*\cdot c\big)\cdot\{\varepsilon\}. \tag{8}$$

Counting the star-height of regular *expressions* (7) and (8), we obtain, that the star-height of considered *automaton* is equal to 1.

Proposition 4: For automaton without useless and inaccessible states and given ordering function τ,

$$\mathcal{SH}\big(\mathcal{R}(K,\tau)\big)=\max_{q\in Q}\mathcal{SH}(\rho_q).\quad\square$$

V. The finite automaton for the given language

In this section, for a given regular language, the corresponding finite automaton is constructed. As we already said, this problem was solved more than 60 years ago, but the method we are considering has the property that the "reverse construction" (that is, the standard algorithm for constructing a regular expression based on a given finite automaton) gives (after some simplest equivalent transformation) exactly the original regular expression.

We shall not formulate these transformations strictly, since *this is simply not necessary*. See, e.g., the example before, where we have obtained 3 factors $(ab^*c)^*$ in (7), but we can consider, e.g., $\tau=(ab^*c)^*$.

Thus, we shall formulate the main proposition of the paper as follows.

Proposition 5: For each regular expression r, there exists automaton

$$K_r=(\,Q_r,\Sigma,\delta_r,S_r,F_r\,)$$

and function τ_r for its states, such that:

(r1) $\mathcal{L}(K_r)=\mathcal{L}(r)$;

(r2) $\mathcal{SH}(K_r)\le\mathcal{SH}(\mathcal{R}(K_r,\tau_r))=\mathcal{SH}(r)$.

(Let us note once again, that we do not assert, that $r=\mathcal{R}(K,\tau)$.)

Proof. We shall prove this proposition in the usual way, i.e., considering the usual process of constructing the given regular expression; at the same time, we shall construct the equivalent automaton and corresponding function τ.

In addition to conditions (r1) and (r2) formulated before, we shall construct automata (let it be K_r) and corresponding function (let τ_r), for which two additional requirements are also met:

(r3) there exists a value $T_r \in \mathbb{R}^+$, such that:
- $\tau(s) < T_r$ for each $s \in S_r$;
- $\tau(q) > T_r$ for each $q \notin S_r$;

(r4) there is *no* edge of the type

$$s_1 \xrightarrow[\delta_r]{a} s_2,$$

where $s_1, s_2 \in S_r$.

Possible automata for the regular expressions $\varnothing$, ε and a (for each $a \in \Sigma$) are given on Fig. 2–4 respectively. It is evident, that for each of these expressions, we have $\mathcal{SH}(K_r) = 0$. Conditions (r3) and (r4) also hold.

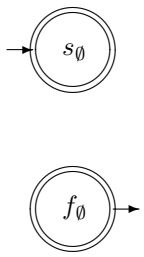

Fig. 2

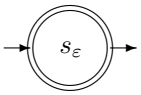

Fig. 3

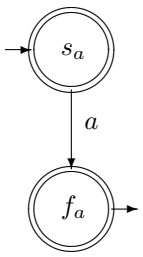

Fig. 4

Now, let us suppose that we *already have* automata corresponding to given regular expressions p and r; we mean that we suppose, that already have automata, which define the same regular languages and have the same star-height. Certainly, we can also suppose that we also have functions τ for these automata obtaining regular expressions which star-height is equal to the star-height of the given expressions (i.e., giving minimum possible star-height). I.e., we can suppose that conditions (r1–r4) hold. Let these automata be

$$K_p = (\, Q_p, \Sigma, \delta_p, S_p, F_p \,) \quad \text{and} \quad K_r = (\, Q_r, \Sigma, \delta_r, S_r, F_r \,);$$

and corresponding functions be

$$\tau_p : Q_p \to \{1, \ldots, |Q_p|\,\} \quad \text{and} \quad \tau_r : Q_r \to \{1, \ldots, |Q_r|\,\}.$$

Then we can use the following automata and ordering functions; let us remark in advance, that the transition functions are considered as the sets, and the facts of defining required regular expressions by these automata are evident.

For expression $(p + r)$, we can consider automaton

$$K_{(p+r)} = (\, Q_p \cup Q_r,\, \Sigma,\, \delta_p \cup \delta_r,\, S_p \cup S_r,\, F_p \cup F_r \,).$$

And considering values T_p, T_r and

$$M_p = \max_{q_p \in Q_p} \tau_p(q_p),$$

we obtain, that the ordering function can be the following one:

$$\tau_{(p+r)}(q) = \begin{cases} \tau_p(s_p), & \text{if } s_p \in S_p\,; \\ \tau_r(s_r) + T_p, & \text{if } s_r \in S_r\,; \\ \tau_p(q_p) + T_p + T_r, & \text{if } q_p \in Q_p \setminus S_p\,; \\ \tau_r(q_r) + T_p + T_r + M_p, & \text{if } q_r \in Q_r \setminus S_r\,. \end{cases}$$

Evidently,

$$\mathcal{SH}(\mathcal{R}(K_{(p+r)}, \tau_{(p+r)})) = \max\Big(\mathcal{SH}(\mathcal{R}(K_p, \tau_p)), \mathcal{SH}(\mathcal{R}(K_r, \tau_r))\Big).$$

Conditions (r1), (r3) and (r4) also hold.

For expression $(p \cdot r)$, we can consider automaton

$$K_{(p\cdot r)} = (\, Q_p \cup Q_r,\, \Sigma,\, \delta_p \cup \delta_r \cup \delta',\, S_p,\, F_r \,),$$
where $\delta'(q\,, a) \ni s_r$ if and only if $\delta(q\,, a) \ni f_p$
for all possible $f_p \in F_p\,,\ s_r \in S_r\,,\ a \in \Sigma$

(δ' contains no other elements). The ordering function $\tau_{(p\cdot r)}$ can be defined in the following way:

$$\tau_{(p+r)}(q) = \begin{cases} \tau_p(q_p), & \text{if } q_p \in Q_p\,; \\ \tau_r(q_r) + M_p, & \text{if } q_r \in Q_r\,. \end{cases}$$

Like previous case,

$$\mathcal{SH}(\mathcal{R}(K_{(p\cdot r)}, \tau_{(p\cdot r)})) = \max\Big(\mathcal{SH}(\mathcal{R}(K_p, \tau_p), \mathcal{SH}(\mathcal{R}(K_r, \tau_r))\Big),$$

and conditions (r1), (r3) and (r4) also hold.

And for expression (r^*), we can consider automaton

$$K_{(r^*)} = (\, Q_r \cup \{q'\},\, \Sigma,\, \delta_r \cup \delta',\, S_r \cup \{q'\},\, F_r \cup \{q'\} \,),$$
where $\delta'(q\,, a) \ni s_r$ if and only if $\delta(q\,, a) \ni f_r$
for all possible $f_r \in F_r\,,\ s_r \in S_r\,,\ a \in \Sigma$

(δ' contains no other elements). The ordering function $\tau_{(r^*)}$ can be defined in the following way:

$$\tau_{(r^*)}(q) = \begin{cases} \tau_r(q), & \text{if } \quad q \in Q_r\,; \\ \tau_r(q') = \Big(\min\limits_{q \in Q_r} \tau_r(q)\Big) \,/\, 2\,. \end{cases}$$

Conditions (r1), (r3) and (r4) are evident; let us prove (r2).

By the way of construction $K_{(r^*)}$, we obtain the following fact: each path of its transition graph, which belongs to $K_{(r^*)}$ and does *not* belong to K_r, has to contain a state $s_r \in S_r$. Then for any states $q, q', q'' \in Q_r$, where $q \notin S_r$, we obtain the coincidence of the sets $\Delta_q(q', q'')$ for automata $K_{(r^*)}$ and K_r. And only for states $s_r \in S_r$, we can obtain some *new* paths of sets $\Delta_{s_r}(s_r, q)$ and $\Delta_{s_r}(q, s_r)$. We mean the paths which *belong* to automaton $K_{(r^*)}$ and *does not belong* to K_r.

For three the following objects:
- automaton K_r;
- corresponding (defined in this section) function τ_r;
- and some its state q,

we denote the defined in previous section value $\mathcal{SH}(\rho_q)$ by $\boldsymbol{\mathcal{SH}_r(q)}$. Then by Proposition 3 and condition (r3) for automaton $K_{(r^*)}$, we obtain, that:

$$\begin{cases} \mathcal{SH}_{(r^*)}(q) = \mathcal{SH}_r(q), & \text{if } q \in Q_r \setminus S_r\,; \\ \mathcal{SH}_{(r^*)}(s) \le \mathcal{SH}_r(q) + 1, & \text{if } s \in S_r\,; \\ \mathcal{SH}_{(r^*)}(q') = 0\,. \end{cases}$$

Then by condition (r4) for automaton $K_{(r^*)}$, we obtain condition (r2). □

VI. Conclusion

Using Proposition 5, we can *reformulate the star-height problem for regular languages* in the following way:

> *for the given regular language, we have to construct the equivalent finite automaton K having the minimum possible star-height.*

After that, considering $n!$ bijective functions of the type $\tau : Q \to \{1, \ldots, n\}$ (where $n = |Q|$), we construct regular expressions $\mathcal{R}(K, \tau)$ and choose the expression having the minimum possible star-height.

In addition to this ("natural") continuation of the topic of this paper, there are others, not so obvious. As we said before, the publication of this not-so-complicated problem has some different goals, they are related to the future development of the topic under consideration here.

- The first topic is the consideration of the *generalized* star-height problem. (As we said before, the connection between the generalized star-height problem and generalized pseudo-automata is to be considered in one of the following publications.)
- The second topic is the consideration of *extended* automata, see [15].
- And the third topic is precisely the transition to the star-height problem *for the languages*.

For all such topics, problems close to those considered in this article are possible. Besides, it is very important to note that the possible developments of the topic described here can be treated completely independently of each other ("movements" towards the complication: "vertically", "horizontally" and "upwards"): for example, it is possible to consider the generalized star-height problem (i.e., the star-height problem for generalized regular expressions) with approach using extended pseudo-automata.

The problems considered in this paper are also related to a topic that is set aside from the ones described here: we should to describe an algorithm, answering the question whether or not a proper subset of the set of edges forms the equivalent automaton.

(See some related questions in [16], [17]. Certainly, we do not mean an exhaustive algorithm. The brute force method consists in this case in the complete application of the algorithm for constructing two canonical automata and their subsequent comparison.)

The author hopes, that the joint application of algorithms mentioned above can give a faster than available determining the star-height of the given regular language.

References

[1] Melnikov B. *Ob odnoy klassifikacii kontekstno-svobodnyh ... [On a classification of sequentional context-free languages and grammars]*. Vestnik of Moscow University. Series 15: Computational Mathematics and Cybernetics. 1993, no. 3, pp. 64–69. (in Russian, https://elibrary.ru/title_about.asp?id=8373)

[2] Melnikov B. and Vakhitova A. *Some more on the finite automata*. The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). 1998, vol. 5, no. 3. pp. 495–506.

[3] Melnikov B. *O zvyozdnoy vysote regulyarnogo yazyka ... [On the star-height of a regular language. Part III: The star-height of an automaton and the scheme of the transformation algorithm]*. Heuristic algorithms and distributed computations. 2014, no. 3, pp. 60–76. (in Russian, https://elibrary.ru/item.asp?id=22376180)

[4] Eggan L. *Transitions graphs and the star height of regular events*. Michigan Mathematical Journal. 1963, vol. 10, pp. 385–397.

[5] Hashiguchi K. *Algorithms for determining relative star height and star height*. Information and Computation. 1988, vol. 78, pp. 124–169.

[6] Perrin D. *Finite Automata. Handbook of theoretical computer science, Vol. A*. MIT Press Cambridge, MA, USA, 1990, 57 p.

[7] Kirsten D. *Distance desert automata and the star height problem*. Informatique Théorique et Applications. 2005, vol. 39, no. 3. pp. 455–509.

[8] Bojanczyk M. *Star height via games*. ACM/IEEE Symposium on Logic in Computer Science (LICS). 2015, pp. 214–219.

[9] Pin J.-E., Straubing H., Thérien D. *Some results on the generalized star-height problem*. Information and Computation. 1992, vol. 101, no. 2, pp. 219–250.

[10] Melnikov B. and Melnikova A. *Pseudo-automata for generalized regular expressions*. International Journal of Open Information Technologies. 2018, vol. 6, no. 1. pp. 1–8.

[11] Salomaa A. *Jewels of Formal Language Theory*. Rockville (Maryland): Computer Science Press, Inc. 1981, 144 p.

[12] Melnikov B. *Extended nondeterministic finite automata*. Fundamenta Informaticae. 2010, vol. 104, no. 3. pp. 255–265.

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

[14] Melnikov B. and Sayfullina M. *O nekotoryh algoritmah ekvivalentnogo preobrazovaniya nedeterminirovannyh konechnyh avtomatov [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)

[15] Melnikov B. *Extended nondeterministic finite automata*. Fundamenta Informaticae. 2010, vol. 104, no. 3. pp. 255–265.

[16] Dolgov V. and Melnikov B. *Postroenie universal'nogo konechnogo avtomata ... [The construction of a universal finite automaton. Part I: From theory to practical algorithms]*. Vestnik of Voronezh State University. 2013, no. 2, pp. 173–181. (in Russian, https://elibrary.ru/item.asp?id=20267924)

[17] Dolgov V., Melnikov B. and Melnikova A. *Cikly grafa perehodov bazisnogo avtomata ... [Cycles of the transition graph of a basic automaton and related questions]*. Vestnik of Voronezh State University. 2016, no. 4, pp. 95–111. (in Russian, https://elibrary.ru/item.asp?id=27257800)