Some more on the basis finite automaton

Boris MELNIKOV
Russia, Samara State University
email: bormel@rambler.ru

Aleksandra MELNIKOVA
Russia, National Research Nuclear
University "MEPhI"
email: super-avahi@yandex.ru

Abstract. We consider in this paper the basis finite automaton and its some properties. We shall also consider some properties of special binary relation defined on the sets of states of canonical automata for the given language and for its mirror image. We shall also consider an algorithm of constructing the basis automaton defining the language which has a priory given variant of this relation.

1 Introduction

The basis automaton for the given regular language was firstly defined in [11]. And in [6], we considered an extension of the basis automaton, which can describe *all the possible* labels of inputs, outputs and loops for *any* state of *any* nondeterministic automaton defining the given language.

The basis automaton can be considered as a complete invariant of regular language, like automaton of canonical form and Conway's universal automaton ([2, 5]). But using the basis automaton, we can formulate some properties of regular language; using other formalisms, these properties could be formulated in a more complicated way. Some of such properties were already considered in [8, 9, 11].

In this paper, we shall consider some other such properties and some examples for them. Among other things, we shall consider some properties of special binary relation defined on the sets of states of two canonical automata: for the

given language and for its mirror image. We shall also consider an algorithm of constructing the basis automaton defining the language, for which there holds the *a priory given* variant of this relation.

2 Preliminaries

We shall use the notation and preliminaries of [6, 7]. Let us repeat the main ones for the ease of reading.

We shall consider nondeterministic finite automaton

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

without ε-edges, i.e., we consider transition function δ of automaton (1) as

$$\delta : Q \times \Sigma \to \mathcal{P}(Q).$$

Its language will be denoted by $\mathcal{L}(K)$; unless other fact is formulated, we shall suppose that $\mathcal{L}(K) = L$.

The input language of the state $q \in Q$, i.e., the language of automaton $(Q, \Sigma, \delta, S, \{q\})$, will be denoted by $\mathcal{L}_K^{in}(q)$. Similarly, the output language of the state $q \in Q$, i.e., the language of automaton $(Q, \Sigma, \delta, \{q\}, F)$, will be denoted by $\mathcal{L}_K^{out}(q)$.

$\widetilde{L}$ is the canonical automaton defining L, without the useless ("dead") state. Let automata $\widetilde{L}$ and $\widetilde{L^R}$ for the given language L be as follows:

$$\widetilde{L} = (Q_\pi, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi) \quad \text{and} \quad \widetilde{L^R} = (Q_\rho, \Sigma, \delta_\rho, \{s_\rho\}, F_\rho)$$

(where π and ρ are indexes which indicate languages of two canonical automata, i.e., languages L and L^R respectively).

Binary relation $\# \subseteq Q_\pi \times Q_\rho$ is defined in the following way. For some states $A \in Q_\pi$ and $X \in Q_\rho$, condition $A\#X$ holds if and only if there exist some words $u \in \mathcal{L}_{\widetilde{L}}^{in}(A)$ and $v \in \mathcal{L}_{\widetilde{L^R}}^{out}(X)$, such that $uv^R \in \mathcal{L}(K)$. In [7], we considered a simple algorithm for constructing this relation.

Also in [6, 8, 7], we considered state-marking functions φ^{in} and φ^{out} for automaton (1); those are the function of the type

$$\varphi_K^{in} : Q \to \mathcal{P}(Q_\pi) \quad \text{and} \quad \varphi_K^{out} : Q \to \mathcal{P}(Q_\rho)$$

defined in the following way. We set $\varphi_K^{in}(q) \ni A$ (where $q \in Q$ and $A \in Q_\pi$) if and only if

$$(\exists u \in \Sigma^*)\,(u \in \mathcal{L}_K^{in}(q) \,\&\, u \in \mathcal{L}_{\widetilde{L}}^{in}(A)), \quad \text{i.e.,} \quad \mathcal{L}_K^{in}(q) \cap \mathcal{L}_{\widetilde{L}}^{in}(A) \neq \emptyset.$$

Similarly, we set $\varphi_K^{out}(q) \ni X$ (where $q \in Q$ and $X \in Q_\rho$) if and only if

$$\left(\mathcal{L}_K^{out}(q)\right)^R \cap \mathcal{L}_{\widetilde{L^R}}^{in}(X) \neq \emptyset.$$

A simple algorithm for constructing these functions was also given in [7].

For language L defined by automaton (1), we define the equivalent basis automaton $\mathcal{BA}(L)$; in this paper, we use the version of its definition of [6]. Thus, for the given regular language L, this automaton will be denoted by

$$\mathcal{BA}(L) = (\hat{Q}, \Sigma, \hat{\delta}, \hat{S}, \hat{F}),\qquad\qquad(2)$$

where [1]:

- $\hat{Q}$ is the set of pairs of the type $\genfrac{}{}{0pt}{}{A}{X}$, such that $A \in Q_\pi$, $X \in Q_\rho$ and $A \# X$;

- transition function $\hat{\delta}$ is defined in the following way: for each $\genfrac{}{}{0pt}{}{A}{X}$, $\genfrac{}{}{0pt}{}{B}{Y} \in \hat{Q}$ and $a \in \Sigma$, we have $\genfrac{}{}{0pt}{}{A}{X} \xrightarrow[\hat{\delta}]{a} \genfrac{}{}{0pt}{}{B}{Y}$ if and only if $A \xrightarrow[\delta_\pi]{a} B$ and $Y \xrightarrow[\delta_\rho]{a} X$;

- $\hat{S} = \left\{ \genfrac{}{}{0pt}{}{s_\pi}{X} \,\middle|\, s_\pi \# X \right\}$;

- similarly, $\hat{F} = \left\{ \genfrac{}{}{0pt}{}{A}{s_\rho} \,\middle|\, A \# s_\rho \right\}$.

Thus, we can think that considering the given regular *language* L, we also have notation for *its*:

- two automata of canonical form (i.e., $\widetilde{L}$ and $\widetilde{L^R}$), and also their states, transition functionc etc;

- binary relation #;

- state-marking functions φ^{in} and φ^{out};

- basis automaton $\mathcal{BA}(L)$.

We also shall sometimes consider automaton $(\widetilde{L^R})^R$ which also defines language L.

---

[1] See [7] for some more details, e.g., for binary relation #.

3 The correctness of the definition $\mathcal{BA}(L)$: the complete proof

As we said before, the definition of the basis automaton was firstly given in [11], we use the equivalent definition of [6]. Also in [11], there was given the proof of the correctness of that definition. But that proof was incomplete: in fact, we have proved only that each word of the given language can be accepted by automaton $\mathcal{BA}(L)$. In this section, we consider the complete version of this proof. This complete version will be also used in Section 7.

Proposition 1 $\mathcal{L}\Big(\mathcal{BA}(L)\Big) = L.$

Proof.   We shall prove the equivalence of automata $\widetilde{L}$ and $\mathcal{BA}(L)$.

1. Firstly, let us consider some word $u \in L$, i.e., $u \in \mathcal{L}(\widetilde{L})$. Then $u^R \in L^R$, i.e., $u^R \in \mathcal{L}(\widetilde{L^R})$. Let $|u| = n$.

Let the accepting of the word u by automaton $\widetilde{L}$ is the following sequence of transitions beginning (the only) initial state s_π:

$$p_0 = s_\pi,\ p_1,\ p_2,\ \ldots,\ p_{n-2},\ p_{n-1},\ f_\pi = p_n\,, \tag{3}$$

where $p_i \in Q_\pi$ for each $i \in \{1, \ldots, n-1\}$ (i.e., each p_i is some state of automaton $\widetilde{L}$), and $f_\pi \in F_\pi$ (i.e., f_π is *some* final state of automaton $\widetilde{L}$). Because $\widetilde{L}$ is deterministic automaton, sequence (3) is the (unique) accepting run of $\widetilde{L}$ on u.

Similarly, automaton $\widetilde{L^R}$ reading letters of the word u^R has the following sequence of transitions:

$$r_0 = s_\rho,\ r_1,\ r_2,\ \ldots,\ r_{n-2},\ r_{n-1},\ f_\rho = r_n \tag{4}$$

(where $r_i \in Q_\rho$ for each $i \in \{1, \ldots, n-1\}$, and $f_\rho \in F_\rho$); as before, s_ρ is the only initial state. Sequence (4) is also defined by the given word u (or u^R) in the only way. The numbers of elements of the sequences (3) and (4) are the same; they are equal to $n + 1$. The sequence of transitions for automata $\widetilde{L}$, $\widetilde{L^R}$ and $(\widetilde{L^R})^R$ reading words u and u^R is shown on the following diagram:

$$
\begin{array}{ccccccccc}
\widetilde{L}: & s_\pi & \xrightarrow{} & p_1 & \xrightarrow{} & p_2 & \cdots\ p_{n-2} & \xrightarrow{} & p_{n-1} & \xrightarrow{} & f_\pi \\
 & & a_1 & & a_2 & & \cdots & & a_{n-1} & & a_n \\
 & f_\rho & \xleftarrow{} & r_{n-1} & \xleftarrow{} & r_{n-2} & \cdots\ r_2 & \xleftarrow{} & r_1 & \xleftarrow{} & s_\rho & : \widetilde{L^R} \\
(\widetilde{L^R})^R : f_\rho & & \xrightarrow{} & r_{n-1} & \xrightarrow{} & r_{n-2} & \cdots\ r_2 & & \xrightarrow{} & r_1 & & \xrightarrow{} & s_\rho
\end{array}
$$
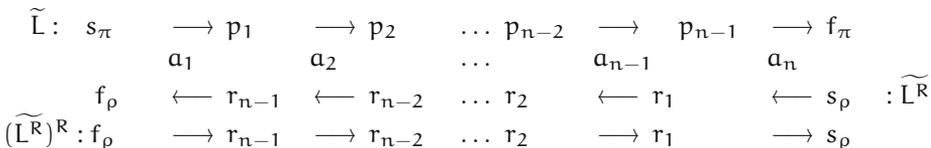
Let us rewrite the sequence (4) in the reverse order:

$$f_\rho, \ r_{n-1}, \ r_{n-2}, \ \ldots, \ r_2, \ r_1, \ s_\rho, \tag{5}$$

then let us combine elements of (3) and (5) in the following sequence of the pairs:

$$\frac{s_\pi}{f_\rho}, \quad \frac{p_1}{r_{n-1}}, \quad \frac{p_2}{r_{n-2}}, \quad \ldots, \quad \frac{p_{n-2}}{r_2}, \quad \frac{p_{n-1}}{r_1}, \quad \frac{f_\pi}{s_\rho}.$$

By the definition of relation #, each pair of this sequence can be considered as a state of automaton $\mathcal{BA}(L)$, because for each such pair (let it be $\frac{q_i}{r_{n-i}}$), we can write a word of L (i.e., the given word u) by $u = vw$, where

$$v = a_1 \ldots a_i \in \mathcal{L}_{\widetilde{L}}^{in}(p_i) \quad \text{and} \quad w = a_{i+1} \ldots a_n \in \left(\mathcal{L}_{\widetilde{L^R}}^{in}(r_{n-i}) \right)^R.$$

Besides, by definition of $\mathcal{BA}(L)$, state $\frac{s_\pi}{f_\rho}$ belongs to $\hat{S}$ (the set of initial states of automaton $\mathcal{BA}(L)$), and state $\frac{f_\pi}{s_\rho}$ belongs to $\hat{F}$ (the set of final states). And for each $i \in \{0, \ldots, n-1\}$, we have

$$\frac{p_i}{r_{n-i}} \xrightarrow[\mathcal{BA}(L)]{a_i} \frac{p_{i+1}}{r_{n-(i+1)}}$$

by definition of $\mathcal{BA}(L)$. Therefore $L \subseteq \mathcal{L}\left(\mathcal{BA}(L) \right)$.

2. Secondly, let us consider some word $u \in \mathcal{L}\left(\mathcal{BA}(L) \right)$. Let also $|u| = n$, and $u = a_1 a_2 \ldots a_n$. Then for $\mathcal{BA}(L)$ and u, we can write the following sequence of transitions:

$$\frac{p_0}{r_n} \xrightarrow[\mathcal{BA}(L)]{a_1} \frac{p_1}{r_{n-1}} \xrightarrow[\mathcal{BA}(L)]{a_2} \frac{p_2}{r_{n-2}} \quad \ldots \quad \frac{p_{n-2}}{r_2} \xrightarrow[\mathcal{BA}(L)]{a_{n-1}} \frac{p_{n-1}}{r_1} \xrightarrow[\mathcal{BA}(L)]{a_n} \frac{p_n}{r_0}. \tag{6}$$

By definition of $\mathcal{BA}(L)$ we obtain, that $p_0 = s_\pi$, and also

$$p_i \xrightarrow[\widetilde{L}]{a_{i+1}} p_{i+1} \quad \text{for each} \quad i \in \{0, \ldots, n-1\}.$$

We have to prove, that $p_n \in F_\pi$.

Let $p_n \notin F_\pi$. By definition of relation # (see [7]) and pair $\frac{p_n}{r_0}$ for it, *there exists* a word $u' \in L$, such that $u' = vw$, where

$$v \in \mathcal{L}_{\widetilde{L}}^{in}(p_n) \quad \text{and} \quad w^R \in \mathcal{L}_{\widetilde{L^R}}^{in}(r_0).$$

By (6), we can think that $v = u$; then $uw \in L$, and therefore $u^R \in \mathcal{L}_{\widetilde{L^R}}^{out}(r_0)$. Then (because $\varepsilon \in \mathcal{L}_{\widetilde{L^R}}^{in}(r_0)$ and automaton $\widetilde{L^R}$ is deterministic) $u^R \in L^R$, therefore $u \in L$. $\square$

4   Some properties of the state-marking functions

In this section we consider some properties of the state-marking functions. They were formulated in [10], and afterwards were used in some other our papers. In this section we shall prove them without other facts, i.e., using the definitions only. All these properties combine in the common expressions the values of input and output languages of the states (i.e., of $\mathcal{L}^{in}$ and $\mathcal{L}^{out}$, see [7]):

- of *any* nondeterministic finite automaton K defining considered regular language;

- of canonical automata for languages $\mathcal{L}(\mathsf{K})$ and $\mathcal{L}(\mathsf{K}^{\mathsf{R}})$.

It is important to remark, that by following Propositions 6 and 7, corresponding languages are also input and output languages of the states of the equivalent *basis* automaton.

The first proposition of this section formulates the *sufficient* condition of the given word: whether or not it belongs to the corresponding output language.

Proposition 2
$$\mathcal{L}_{\mathsf{K}}^{out}(\mathsf{q}) \subseteq \bigcup_{\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{in}(\mathsf{q})} \mathcal{L}_{\widetilde{\mathsf{L}}}^{out}(\widetilde{\mathsf{q}})\,.$$

Proof.   Let for some word v and state of canonical automaton $\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{in}(\mathsf{q})$ condition $v \notin \mathcal{L}_{\widetilde{\mathsf{L}}}^{out}(\widetilde{\mathsf{q}})$ holds. Then we prove, that $v \notin \mathcal{L}_{\mathsf{K}}^{out}(\mathsf{q})$.

Consider some word
$$u \in \mathcal{L}_{\mathsf{K}}^{in}(\mathsf{q}) \cap \mathcal{L}_{\widetilde{\mathsf{L}}}^{in}(\widetilde{\mathsf{q}})\,;$$

such a word u exists by definition of the function φ^{in}. Automaton $\widetilde{\mathsf{L}}$ is deterministic, then $uv \notin \mathcal{L}(\mathsf{K})$. Therefore, condition $v \in \mathcal{L}_{\mathsf{K}}^{in}(\mathsf{q})$, which is equivalent to $uv \in \mathcal{L}(\mathsf{K})$, contradicts the equality $\mathcal{L}(\widetilde{\mathsf{L}}) = \mathcal{L}(\mathsf{K})$.   □

The "mirror" fact is the following

Proposition 3
$$\mathcal{L}_{\mathsf{K}}^{in}(\mathsf{q}) \subseteq \Big(\bigcap_{\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{out}(\mathsf{q})} \mathcal{L}_{\widetilde{\mathsf{L}^{\mathsf{R}}}}^{out}(\widetilde{\mathsf{q}}) \Big)^{\mathsf{R}}\,.   □$$

In the two following propositions we consider subsets of some language using output languages of the states.

Proposition 4

$$\mathcal{L}_{\mathsf{K}}^{in}(\mathsf{q}) \cdot \Big(\bigcap_{\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{in}(\mathsf{q})} \mathcal{L}_{\widetilde{\mathsf{L}}}^{out}(\widetilde{\mathsf{q}}) \Big) \subseteq \mathcal{L}(\mathsf{K}).$$

Proof. Consider any word $u \in \mathcal{L}_{\mathsf{K}}^{in}(\mathsf{q})$. By definition of function φ^{in}, for some state $\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{in}(\mathsf{q})$, the word u belongs to the language $\mathcal{L}_{\widetilde{\mathsf{L}}}^{in}(\widetilde{\mathsf{q}})$. For each word

$$v \in \bigcap_{\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{in}(\mathsf{q})} \mathcal{L}_{\widetilde{\mathsf{L}}}^{out}(\widetilde{\mathsf{q}}),$$

condition $v \in \mathcal{L}_{\widetilde{\mathsf{L}}}^{out}(\widetilde{\mathsf{q}})$ holds. Therefore,

$$uv \in \mathcal{L}_{\widetilde{\mathsf{L}}}^{in}(\widetilde{\mathsf{q}}) \cdot \mathcal{L}_{\widetilde{\mathsf{L}}}^{out}(\widetilde{\mathsf{q}}) \subseteq \mathcal{L}(\widetilde{\mathsf{L}}) = \mathcal{L}(\mathsf{K}),$$

and the last condition proves the proposition. □

The "mirror" fact is the following

Proposition 5

$$\Big(\bigcap_{\widetilde{\mathsf{q}} \in \varphi_{\mathsf{K}}^{out}(\mathsf{q})} \mathcal{L}_{\widetilde{\mathsf{L}^{\mathsf{R}}}}^{out}(\widetilde{\mathsf{q}}) \Big)^{\mathsf{R}} \cdot \mathcal{L}_{\mathsf{K}}^{out}(\mathsf{q}) \subseteq \mathcal{L}(\mathsf{K}). \quad □$$

5   The first example

In this section, we shall consider an example for Proposition 4. For this thing, we shall use the automaton considered in detail in [7, Sect. 3].

For the ease of reading, let us give this figure once again (Fig. 1). And the equivalent canonical automaton (i.e., $\widetilde{\mathsf{L}}$) is given on Fig. 2.
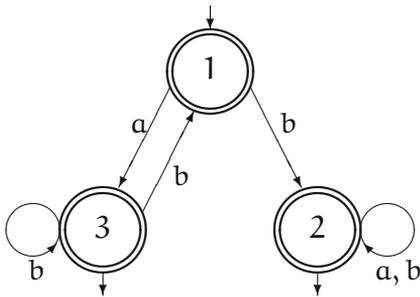

Figure 1
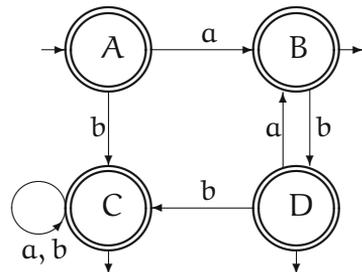
Figure 2

For automaton on Fig. 1, we shall consider state 3. By simple definitions of [7], input language of this state can be defined by the following automaton of Fig. 3:
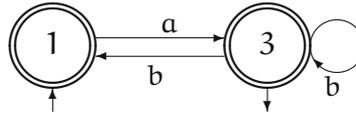

Figure 3

Then it can also be defined by regular expression

$$a(b + ba)^*. \tag{7}$$

For state 3, this language is the first factor of condition of Proposition 4.

By [7, Sect. 3], $\varphi_K^{in}(3) = \{B, C, D\}$. There is evident, that $\mathcal{L}_{\widetilde{L}}^{out}(C) = \Sigma^*$. Then we have to define the intersection of languages $\mathcal{L}_{\widetilde{L}}^{out}(B)$ and $\mathcal{L}_{\widetilde{L}}^{out}(D)$.

The automaton defining language of their intersection is shown on the following Fig. 4. It can be simply constructed using (deterministic) automaton of Fig. 2. E.g., the (initial) state marked $B \cap D$ symbolized the intersection of languages $\mathcal{L}_{\widetilde{L}}^{out}(B)$ and $\mathcal{L}_{\widetilde{L}}^{out}(D)$. We have

$$B \xrightarrow[\widetilde{L}]{b} D \quad \text{and} \quad D \xrightarrow[\widetilde{L}]{b} C,$$

then in constructed automaton, we have $B \cap D \xrightarrow{b} C \cap D$, etc.
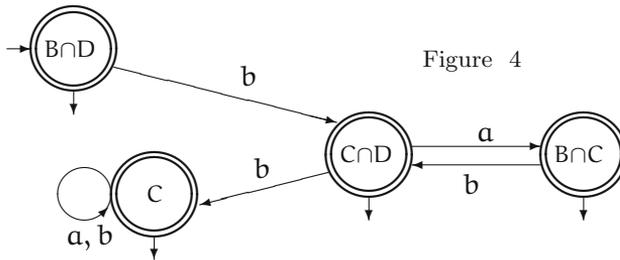
Figure 4

The language of this automaton can also be defined be regular expression

$$\varepsilon + b(ab)^*(a + \varepsilon + b(a + b)^*)).$$

Then considering the last expression and (7), we obtain, that each word defined by the following expression

$$a(b + ba)^* \cdot (\varepsilon + b(ab)^*(a + \varepsilon + b(a + b)^*)))$$

belongs to L.

6 Input and output languages of the states of the basis automaton

In this section we consider properties of input and output languages of the states of the basis automaton; these properties also can be called by the properties of the table of binary relation #. Those are properties, which combine:

- input and output languages of the basis automaton;

- and also input and output languages of two canonical automata (i.e., of automata $\widetilde{L}$ and $\widetilde{L^R}$).

The canonical automaton $\widetilde{L^R}$ contains no more than 2^m-1 states (where m is the number of states of automaton $\widetilde{L}$).[2] Then we can limit by this value the number of possible columns of the table of binary relation #, which has m rows. Besides, this table cannot have duplicate rows and duplicate columns.

The next propositions 6–9 are proved by the definition of the basis automaton.

Proposition 6 *Let there are given some regular language* L *and some state* $A \in Q_\pi$ *of automaton* $\widetilde{L}$. *Then for each state* X *of automaton* $\widetilde{L^R}$, *such that* $A\#X$, *the following condition holds:*

$$\mathcal{L}^{in}_{\mathcal{BA}(L)}\left(\begin{smallmatrix}A\\X\end{smallmatrix}\right) = \mathcal{L}^{in}_{\widetilde{L}}(A).$$

Proof. Condition

$$\mathcal{L}^{in}_{\mathcal{BA}(L)}\left(\begin{smallmatrix}A\\X\end{smallmatrix}\right) \subseteq \mathcal{L}^{in}_{\widetilde{L}}(A)$$

is the direct consequence of the definition of automaton $\mathcal{BA}(L)$. Let us prove the reverse inclusion

$$\mathcal{L}^{in}_{\widetilde{L}}(A) \subseteq \mathcal{L}^{in}_{\mathcal{BA}(L)}\left(\begin{smallmatrix}A\\X\end{smallmatrix}\right),$$

i.e., that for every word $w \in \Sigma^*$, the following fact holds:

$$w \in \mathcal{L}^{in}_{\widetilde{L}}(A) \quad \text{implies} \quad w \in \mathcal{L}^{in}_{\mathcal{BA}(L)}\left(\begin{smallmatrix}A\\X\end{smallmatrix}\right).$$

We shall prove this fact by induction by $|w|$.

The basis of induction (i.e., $w = \varepsilon$) is evident, because if $\varepsilon \in \mathcal{L}^{in}_{\mathcal{BA}(L)}\left(\begin{smallmatrix}A\\X\end{smallmatrix}\right)$ then $\varepsilon \in \mathcal{L}^{in}_{\widetilde{L}}(A)$. Then let us prove *the step of induction*.

---

[2] Remark once again, that we consider canonical automaton without possible "dead" state.

Let $w = w'a$, where $w' \in \Sigma^*$ and $a \in \Sigma$; let also $w \in \mathcal{L}_{\widetilde{L}}^{in}(A)$. Because $A \# X$, there exist words $u, v \in \Sigma^*$, such that

$$uv \in L, \quad u \in \mathcal{L}_{\widetilde{L}}^{in}(A) \quad \text{and} \quad v^R \in \mathcal{L}_{\widetilde{L^R}}^{in}(X).$$

Because w also belongs to the language $\mathcal{L}_{\widetilde{L}}^{in}(A)$, we obtain, that $wv \in L$, i.e., $w'av \in L$. Let also:

- B be some state of automaton $\widetilde{L}$, such that $w' \in \mathcal{L}_{\widetilde{L}}^{in}(B)$;

- Y be some state of automaton $\widetilde{L^R}$, such that $v^R a \in \mathcal{L}_{\widetilde{L^R}}^{in}(Y)$;

both the states (B and Y) do exist, because $w'av \in L$. Then $B \# Y$, and, by the induction hypothesis, $w' \in \mathcal{L}_{\mathcal{BA}(L)}^{in}((B, Y))$. And using the fact $\delta_{\mathcal{T}}((B, Y), a) \ni {}_X^A$, we obtain that $w \in \mathcal{L}_{\mathcal{BA}(L)}^{in}(_X^A)$. $\quad\square$

The "mirror" fact is the following

Proposition 7 *Let there are given some regular language* L *and some state* $X \in Q_\rho$ *of automaton* $\widetilde{L^R}$*. Then for each state* A *of automaton* $\widetilde{L}$*, the following condition holds:*

$$\mathcal{L}_{\mathcal{BA}(L)}^{out}(_X^A) = \left(\mathcal{L}_{\widetilde{L^R}}^{in}(X) \right)^R = \mathcal{L}_{(\widetilde{L^R})^R}^{out}(X). \quad \square$$

Proposition 8 *Let there is given some regular language* L*. Then for each state* $A \in Q_\pi$ *of automaton* $\widetilde{L}$*, the following condition holds:*

$$\mathcal{L}_{\widetilde{L}}^{out}(A) = \bigcup_{X \in Q_\pi} \mathcal{L}_{\mathcal{BA}(L)}^{out}(_X^A).$$

Proof. Consider some word $uv \in L$, such that

$$u \in \mathcal{L}_{\widetilde{L}}^{in}(A) \quad \text{and} \quad v \in \mathcal{L}_{\widetilde{L}}^{out}(A).$$

By [11] (and also by consequence of the proof of Proposition 1), automaton $\mathcal{BA}(L)$ has the only accepting path for the word uv, and for some $x \in Q_\rho$ the following conditions hold:

$$u \in \mathcal{L}_{\mathcal{BA}(L)}^{in}(_X^A) \quad \text{and} \quad v \in \mathcal{L}_{\mathcal{BA}(L)}^{out}(_X^A).$$

Then combining all the possible ν, we obtain, that

$$\mathcal{L}_{\widetilde{L}}^{out}(A) \subseteq \bigcup_{X \in Q_\pi} \mathcal{L}_{\mathcal{BA}(L)}^{out}\left(\begin{smallmatrix} A \\ X \end{smallmatrix}\right).$$

The reverse inclusion, i.e., that

$$\bigcup_{X \in Q_\pi} \mathcal{L}_{\mathcal{BA}(L)}^{out}\left(\begin{smallmatrix} A \\ X \end{smallmatrix}\right) \subseteq \mathcal{L}_{\widetilde{L}}^{out}(A),$$

is evident.   □

The "mirror" fact is the following

Proposition 9 *Let there is given some regular language* L. *Then for each state* $X \in Q_\rho$ *of automaton* $\widetilde{L^R}$, *the following condition holds:*

$$\left(\mathcal{L}_{\widetilde{L^R}}^{out}(X)\right)^R = \mathcal{L}_{(\widetilde{L^R})^R}^{in}(X) = \bigcup_{A \in Q_\rho} \mathcal{L}_{\mathcal{BA}(L)}^{in}\left(\begin{smallmatrix} A \\ X \end{smallmatrix}\right). \quad □$$

Proposition 10 *Let canonical automaton* $\widetilde{L}$ *for the given regular language* L *has at least 2 states, and* $A, B \in Q_\pi$ *is a pair of such states. Then there exists a state* $X \in Q_\rho$ *of automaton* $\widetilde{L^R}$, *such that automaton* $\mathcal{BA}(L)$ *contains exactly one state of the following 2 ones:* $\begin{smallmatrix} A \\ X \end{smallmatrix}$ *and* $\begin{smallmatrix} B \\ X \end{smallmatrix}$.

Proof.   This proposition can be considered as a consequence of the classical algorithm of constructing of canonical automaton (which includes imperative combining equivalent states) and also [7, Th. 4.1].   □

The "mirror" fact is the following

Proposition 11 *Let canonical automaton* $\widetilde{L^R}$ *for the mirror image of the given regular language* L *has at least 2 states, and* $X, Y \in Q_\rho$ *is a pair of such states. Then there exists a state* $A \in Q_\pi$ *of automaton* $\widetilde{L}$, *such that automaton* $\mathcal{BA}(L)$ *contains exactly one state of the following 2 ones:* $\begin{smallmatrix} A \\ X \end{smallmatrix}$ *and* $\begin{smallmatrix} A \\ Y \end{smallmatrix}$.
□

Two the following propositions (we briefly talked about them at the beginning of this section) are the direct consequences of two previous propositions. They formulate facts about the possible size of the table of binary relation $\#$ for the given regular language.

Proposition 12 *Let regular language* L *be given, and automaton* $\widetilde{L}$ *contains exactly* $\mathfrak{n}$ *states (i.e.,* $|Q_\pi| = \mathfrak{m}$*). Then automaton* $\widetilde{L^R}$ *contains no more than* $2^{\mathfrak{m}}{-}1$ *states (i.e.,* $|Q_\rho| \leq 2^{\mathfrak{m}}{-}1$*).* $\square$

Proposition 13 *Let regular language* L *be given, and automaton* $\widetilde{L^R}$ *contains exactly* $\mathfrak{n}$ *states (i.e.,* $|Q_\rho| = \mathfrak{n}$*). Then automaton* $\widetilde{L}$ *contains no more than* $2^{\mathfrak{n}}{-}1$ *states (i.e.,* $|Q_\pi| \leq 2^{\mathfrak{n}}{-}1$*).* $\square$

In the next section we shall obtain, that such maximums (i.e., the values $2^{\mathfrak{m}}{-}1$ and $2^{\mathfrak{n}}{-}1$) can be achieved.

7 On the possible set of the states of the basis automaton

In this section we shall formulate some properties for *all the possible* states of a basis automaton (or, in other words, all the possible variants of binary relation #). We shall obtain, that if there hold all the limitations formulated in previous section for the table of the binary relation #, then such table can really describe such relation for some regular language. Besides, such proof is *constructive*, i.e., we obtain *an algorithm* of constructing the basis automaton for regular language having such table of *a priori given* binary relation #.

Proposition 14 *Let binary relation* # *be given, and for it, all the limitations formulated before hold.*[3] *Then there exists a regular language, for which corresponding binary relation* # *coincides with the given one.*

Proof. Thus, we think, that the sets of states Q_π (where $|Q_\pi| = \mathfrak{m}$) and Q_ρ (where $|Q_\rho| = \mathfrak{n}$) are already given. Binary relation $\# \subseteq Q_\pi \times Q_\rho$ is also given. These objects satisfy all the limitations formulated before.

---

[3] Let us formulate them once again, for the table of #. Let $|Q_\pi| = \mathfrak{m}$ and $|Q_\rho| = \mathfrak{n}$. Then:

- $1 \leq \mathfrak{m} \leq 2^{\mathfrak{n}}{-}1$;
- $1 \leq \mathfrak{n} \leq 2^{\mathfrak{m}}{-}1$;
- there are no duplicate rows;
- there are no duplicate columns;
- there are no empty rows (i.e., there are no row A, such that A#X holds for none column X);
- there are no empty columns.

Consider the following alphabet:

$$\Sigma_\# = \left\{ a_{\substack{A \\ X}} \;\middle|\; A \in Q_\pi,\, X \in Q_\rho \right\}.$$

Over this alphabet, consider the arbitrary states $s_\pi \in Q_\pi$ and $s_\rho \in Q_\rho$. For them, consider following automaton

$$K^{\#s_\pi s_\rho} = (\, Q_\pi, \Sigma_\#, \delta_\pi, \{s_\pi\}, F_\pi \,)$$

(or, briefly, $K^\#$, when s_π and s_ρ are meant), where:

- $F_\pi = \{\, f_\pi \in Q_\pi \mid f_\pi \# s_\rho \,\};\,$[4]

- transition function δ_π is defined in the following way:

    for each $A, B \in Q_\pi$ and $X \in Q_\rho$, $\;\delta_\pi(A, a_{\substack{B \\ X}}) = \begin{cases} \{B\}, & \text{if } A \# X; \\ \emptyset, & \text{otherwise} \end{cases}$

    (we allow for the possibility $A = B$).

Let us prove that the language $\mathcal{L}(K^\#)$ is the desired one.

By the construction, automaton $\mathcal{L}(K^\#)$ is deterministic. The conditions of Proposition 10 hold, because we made automaton using relation corresponding #; therefore automaton $K^\#$ has no pairs of equivalent states. Also by the construction, the transition graph of is automaton is strongly connected ([3]); then it contains no useless states. Therefore, automaton $K^\#$ is canonical (for its language), i.e.,

for the language $\;\; L^\# = \mathcal{L}(K^\#)$, we have $\;\; K^\# = \widetilde{L^\#}$.

Over the same alphabet $\Sigma_\#$, consider also automaton

$$K_{\#s_\pi s_\rho} = (\, Q_\rho, \Sigma_\#, \delta_\rho, \{s_\rho\}, F_\rho \,)$$

(or, briefly, $K_\#$, when s_π and s_ρ are meant), where:

- $F_\rho = \{\, f_\rho \in Q_\pi \mid s_\pi \# f_\rho \,\};$

---

[4] Such a choice is possible, because of limitations formulated before. Remark also, that choosing various $s_\pi \in Q_\pi$ and $s_\rho \in Q_\rho$, we obtain a *set* of languages having the given binary relation #.

- transition function δ_ρ is defined in the following way:

$$\text{for each } B \in Q_\pi \text{ and } X, Y \in Q_\rho\,, \quad \delta_\rho(Y, a_{\underset{X}{B}}) = \begin{cases} \{X\}, & \text{if} \quad B\#Y\,; \\ \emptyset, & \text{otherwise} \end{cases}$$

(we allow for the possibility $X = Y$).

Like automaton $K^\#$ we can prove, that automaton $K_\#$ is canonical (for its language $L_\# = \mathcal{L}(K_\#)$, we have $K_\# = \widetilde{L_\#}$.

Let us prove, that $L_\# = (L^\#)^R$. For this thing, consider any word $u \in L^\#$. Let

$$u = a_{\underset{X_1}{A_1}} a_{\underset{X_2}{A_2}} \dots a_{\underset{X_k}{A_k}}.$$

Then we can write the following sequence of transitions of canonical automaton $K^\# = \widetilde{L^\#}$ while readind the word u:

$$s_\pi \xrightarrow[K^\#]{a_{\underset{X_1}{A_1}}} A_1 \xrightarrow[K^\#]{a_{\underset{X_2}{A_2}}} A_2 \dots A_{k-1} \xrightarrow[K^\#]{a_{\underset{X_k}{A_k}}} A_k\,, \quad \text{where} \quad A_k \in F_\pi\,.$$

Since $A_k \in F_\pi$, we have $A_k \# s_\rho$. Then for this sequence, we can construct the following sequence of transitions of automaton $K_\# = \widetilde{L_\#}$:

$$s_\rho \xrightarrow[K_\#]{a_{\underset{X_k}{A_k}}} X_{k-1} \dots X_2 \xrightarrow[K_\#]{a_{\underset{X_2}{A_2}}} X_1 \xrightarrow[K_\#]{a_{\underset{X_1}{A_1}}} X_0$$

for the sequence of states $X_{k-1}, \dots, X_2, X_1$ selected before and some new state X_0. Like the proof of Proposition 1, we obtain that $X_0 \in F_\rho$.

We proved that $L^\# \subseteq (L_\#)^R$. The reverse inclusion, i.e., $(L_\#)^R \subseteq L^\#$, can be proved similarly.

Thus, automata $K^\#$ and $K_\#$ are canonical automata for the languages $L^\#$ and $(L^\#)^R = L_\#$. Then for them, we can construct the following *basis* automaton

$$\mathcal{BA}(L^\#) = (\mathcal{T}, \Sigma_\#, \delta_T, S_T, F_T)\,,$$

(over the alphabet $\Sigma_\#$ defined before), where:

- $\mathcal{T} = \left\{ \genfrac{}{}{0pt}{}{A}{X} \,\middle|\, A \in Q_\pi,\, X \in Q_\rho,\, A\#X \right\};$

- for each $A, B \in Q_\pi$ and $X \in Q_\rho$, such that $A\#X$ (we admit the possibility of $A = B$), we set

$$\delta_T\left(\genfrac{}{}{0pt}{}{A}{X}, a_{\underset{X}{B}}\right) = \left\{ \genfrac{}{}{0pt}{}{B}{Y} \,\middle|\, (\exists Y \in Q_\rho)\,(B\#Y) \right\};$$

- for each other cases $a \in \Sigma_\#,$ [5] $A, B \in Q_\pi$ and $X, Y \in Q_\rho$, we set

$$\delta_T(^A_X, a) = \emptyset;$$

- $S_T = \left\{ \, ^{s_\pi}_X \, \middle| \, s_\pi \# X \, \right\};$

- $S_T = \left\{ \, ^A_{s_\rho} \, \middle| \, A \# s_\rho \, \right\};$

(where the states s_π and s_ρ were also previously selected). This automaton is $\mathcal{BA}(L^\#)$ by the process of its constructing. And also by its constructing, its set of states $\mathcal{T}$ forms the given binary relation $\#$. □

As a consequence of the Proposition 14 we obtain, that these maximums of the number of states of two canonical automata (i.e., the values 2^m-1 and 2^n-1) can be achieved. But there is important to remark the following thing. In some books ([1] etc.), there are examples, when the given automaton contains n states, and the equivalent canonical automaton contains 2^n-1 states. [6] This fact (i.e., the possible fulfilment the upper bound 2^n-1) is *not* a consequence of these results: it proved there for *arbitrary* nondeterministic finite automata (having no limitations), and we consider it for automata which are mirror automata for canonical ones. E.g. we can say, that these automata are *not* deterministic, but they are unambiguous.

8 The second example

Let us consider a simple example for automata defined in previous section. However, we describe the whole process of constructing detailed.

Thus, let us consider the following binary relation $\#$:

#	X	Y
A	#	–
B	#	#

Table 1

Certainly, this relation satisfies all the limitations formulated before.

---

[5] Remark once again, that we consider automata without ε-edges.

[6] Or 2^n states, when we assume the possible "dead" state, considering the canonical automaton as a total automaton.

By the previous section, corresponding alphabet $\Sigma_\#$ is the following:

$$\Sigma_\# = \left\{ a_{\substack{A \\ X}},\ a_{\substack{B \\ X}},\ a_{\substack{A \\ Y}},\ a_{\substack{B \\ Y}} \right\}.$$

Let $s_\pi = A$, $s_\rho = Y$ (also by the previous section, we can choose such states). Then $F_\pi = \{B\}$, $F_\rho = \{X\}$, and we obtain the following canonical automaton for the language $L^\#$:

$L^\#$	$a_{\substack{A\\X}}$	$a_{\substack{B\\X}}$	$a_{\substack{A\\Y}}$	$a_{\substack{B\\Y}}$
$\rightarrow$ A	A	B	–	–
$\leftarrow$ B	A	B	A	B

Table 2

For the convenience, let us rename the letters in the following way:

$$a_{\substack{A\\X}} = a,\quad a_{\substack{B\\X}} = b,\quad a_{\substack{A\\Y}} = c,\quad a_{\substack{B\\Y}} = d.$$

Then we can rewrite the considered automaton by the following Table 3 or Fig. 5:

| $L^\#$	a	b	c	d
$\rightarrow$ A	A	B	–	–
$\leftarrow$ B	A	B	A	B

Table 3
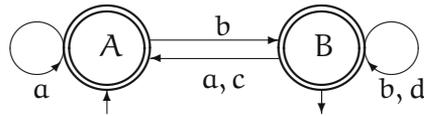
Figure 5

The mirror automaton $(L^\#)^R$ is the following (Table 4 or Fig. 6; for the table of nondeterministic automaton, we use the agreements of [7]):

$(L^\#)^R$	a	b	c	d
$\leftarrow$ A	A, B	–	B	–
$\rightarrow$ B	–	A, B	–	B

Table 4
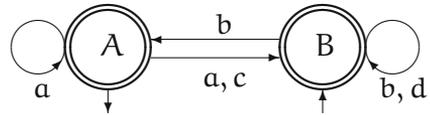
Figure 6

The process of determinization is given by the following table:

		a	b	c	d
→	B	−	A, B	−	B
←	A, B	A, B	A, B	B	B

Table 5

Renaming {B} for Y and {A, B} for X, we obtain the following automaton $\widetilde{L}' = L_{\#}$ (where $L' = (L^{\#})^R$; see Table 6 or Fig. 7):

L_#	a	b	c	d
→ Y	−	X	−	Y
← X	X	X	Y	Y

Table 6
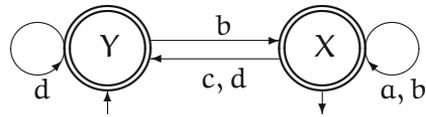
Figure 7

And using the last automaton, we obtain automata $(L_{\#})^R$ and then $\mathcal{B}\!\mathcal{A}(L^{\#})$ (Fig. 8 and 9).
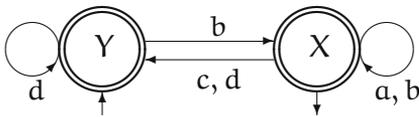

Figure 8
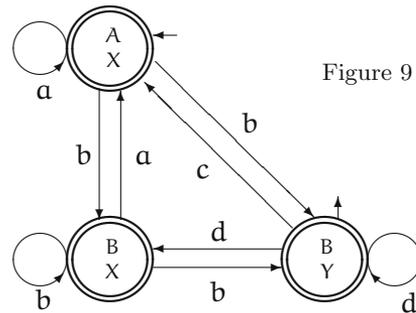
Figure 9

Certainly, they both also define the language $L^{\#}$. (Compare the given Table 1 and the obtained Fig. 9.)

9 Conclusion

Let us describe some possible problems for the future solution. Thus, we are going to:

- to show the relationship between the basis automaton and Conway's universal automaton;

- to use some propositions proved in this paper to describe the minimization algorithms for nondeterministic finite automata;

- vice versa, to use automaton $\mathsf{K}^{\#}$ to describe algorithms of automatic constructing some counter-examples for the algorithms of state minimization (the most famous example of such a counter-example is so called automaton Waterloo, [4]).

References

[1] W. Brauer, *Automatentheorie. Eine Einführung in die Theorie endlicher Automaten.* Teubner, 1984. ⇒ 241

[2] J. Conway, *Regular Algebra and Finite Machines.* Chapman and Hall, 1971. ⇒ 227

[3] F. Harary, *Graph Theory.* Addison-Wesley, 1969. ⇒ 239

[4] T. Kameda, P. Weiner, On the state minimization of nondeterministic finite automata, *IEEE Trans. on Comp.,* **C-19,** 7 (1970) 617–627. ⇒ 244

[5] S. Lombardy, J. Sakarovitch, The Universal Automaton, *Logic and Automata, Texts in Logic and Games, Amsterdam Univ. Press,* **2** (2008) 457–504. ⇒ 227

[6] B. Melnikov, On an expansion of nondeterministic finite automata, *J. of Applied Math. and Computing (The Korean J. of Comp. and Appl. Math.),* **24,** 1–2 (2007) 155–165. ⇒ 227, 228, 229, 230

[7] B. Melnikov, Once more on the edge-minimization of nondeterministic finite automata and the connected problems, *Fundamenta Informaticae,* **104,** 3 (2010) 267–283. ⇒ 228, 229, 231, 232, 233, 234, 237, 242

[8] B. Melnikov, A. Melnikova, Some properties of the basis finite automaton, *J. of Applied Math. and Computing (The Korean J. of Comp. and Appl. Math.),* **9,** 1 (2002) 131–150. ⇒ 227, 228

[9] B. Melnikov, A. Melnikova, A new algorithm of constructing the basis finite automaton, *Informatica (Lithuanian Acad. Sci. Ed.),* **13,** 3 (2002) 299–310. ⇒ 227

[10] B. B. Melnikov, N. Sciarini-Guryanova, Possible edges of a finite automaton defining a given regular language, *J. of Applied Math. and Computing (The Korean J. of Comp. and Appl. Math.),* **9,** 2 (2002) 475–485. ⇒ 232

[11] A. Vakhitova, The basis automaton for the given regular language, *J. of Applied Math. and Computing (The Korean J. of Comp. and Appl. Math.),* **6,** 3 (1999) 617–624. ⇒ 227, 230, 236