

# Расширенный базисный конечный автомат.

## Часть I. Основные определения

Б. Ф. Мельников, А. А. Мельникова

**Аннотация**—В настоящей статье мы рассматриваем специальное расширение класса недетерминированных конечных автоматов. Целью этого рассмотрения является: во-первых, описание с помощью подобных автоматов различных «переборных» (brute-force) алгоритмов, в том числе – алгоритмов эквивалентного преобразования автоматов; во-вторых – применение описанного расширения в нескольких минимизационных проблемах, формулируемых для обычных недетерминированных конечных автоматов; в-третьих, с помощью расширенных автоматов мы можем упростить некоторые доказательства, также нужные для обычных конечных автоматов.

В статье мы рассматриваем определение расширенных конечных автоматов, приводим примеры, определяем расширенный базисный конечный автомат и рассматриваем некоторые его свойства. Также мы доказываем, что для расширенного базисного автомата все получаемые в построениях вспомогательные языки являются регулярными. Всё это позволяет упростить само определение расширенного базисного автомата.

Мы также показываем адекватность определения расширенного базисного конечного автомата – т.е. доказываем утверждение о том, что построенный на основе исходного регулярного языка расширенный базисный автомат действительно определяет этот язык. А на одном из примеров мы показываем следующий факт: существуют регулярные языки, в базисных автоматах которых у некоторых вершин нет циклов, соответствующих циклам соответствующих вершин других эквивалентных автоматов; именно такие циклы имеются в аналогичных вершинах расширенных базисных автоматов.

**Ключевые слова**—Регулярные языки, недетерминированные конечные автоматы, расширение автоматов, алгоритмы эквивалентного преобразования автоматов.

### I. ВВЕДЕНИЕ

В настоящей статье мы рассматриваем специальное расширение класса недетерминированных конечных автоматов – определяющее, однако, то же самое множество регулярных языков. Целью рассмотрения такого расширения *не* является описание очередного способа задания класса регулярных языков: просто потому, что уже было опубликовано множество работ, в которых описывались *похожие* способы его задания. Например, можно было бы подробно рассмотреть расширение класса конечных автоматов, в котором каждая дуга помечена некоторым регулярным выражением – и фактически мы рассматривали такой способ в [1]. Таким образом, в отличие от т.н. обобщённых автоматов (generalized automata, [2]), мы не стремимся описать как можно более широкое обобщение

класса «обычных» конечных автоматов, определяющее то же самое множество регулярных языков. Более того, мы специально накладываем сформулированные далее ограничения на регулярные выражения, которыми могут быть помечены дуги. В нашем расширении подобные дуги (т.е. помеченные регулярными выражениями) допускаются только для стартовых и финальных состояний, а также для дуг-петель.

Материал статьи [3] в настоящей статье существенно изменён и дополнен. Мы приводим все определения из той статьи (ранее на русском языке не публиковавшиеся) и, кроме того:

- рассматриваем примеры – которых в [3] практически не было;
- приводим некоторые связанные вспомогательные утверждения, опубликованные ранее в статьях, не связанных с тематикой расширенных автоматов;
- несколько упрощаем приведённые ранее доказательства;
- приводим дополнительные рисунки, иллюстрирующие их;
- заменяем схемы доказательств на непосредственные доказательства;
- доказываем, что в расширенном базисном автомате (как следует из названия, именно ему в первую очередь посвящена настоящая статья) все получаемые в доказательствах вспомогательные языки являются регулярными; отметим, что это не следовало из материала, приведённого в [3];
- показываем, как на основе приведённого здесь материала упростить задание расширенного базисного автомата.

Мы предполагаем вернуться к заданию регулярных языков с помощью расширенных конечных автоматов в статье-продолжении – поэтому нам необходима законченная терминология, связанная с этими автоматами.

Как уже было отмечено, мы в наших предыдущих работах рассматривали и другие расширения класса недетерминированных конечных автоматов – причём эти расширения иногда определяют «исходный» класс языков (регулярные языки<sup>1</sup>), а иногда – более широкие классы языков. По этому поводу, кроме уже процитированных статей [1], [3] отметим следующие формализмы.

- Уже упомянутые выше обобщённые автоматы. При этом сто́ит отметить, что в [2] мы привели *несколько*

Статья получена 19 июля 2019.

Борис Феликсович Мельников, Университет МГУ – ППИ в Шэньчжэне (bf-melnikov@yandex.ru).

Александра Александровна Мельникова, Димитровградский инженерно-технологический институт – филиал Национального исследовательского ядерного университета «МИФИ» (super-avahi@yandex.ru).

<sup>1</sup> В этом случае речь идёт как раз об «очередном» («1001-м») способе задания всего класса регулярных языков. Именно это мы имели в виду выше.

возможных «автоматных моделей»<sup>2</sup>, каждая из которых описывает обобщённые регулярные выражения ([4] и мн.др.).

- Т.н.  $\pi$ -подклассы класса КС-языков, рассматривавшиеся нами в [5]; как показано в той статье, один из подклассов совпадает с классом регулярных языков.
- Скобочные автоматы (bracketed automata) – см. [6], [7].

А из многочисленных «расширяющих» формализмов, описанных *не* авторами настоящей статьи, отметим только один, следующий.

- См. [8]; по нашему мнению, этот способ задания автоматов может много вариантов *практического* применения – однако дальнейшие публикации на эту тему нам неизвестны.

Можно сказать, *немного упрощая* рассмотрение всех упомянутых моделей, что в каждой из них дуги автомата помечены, вообще говоря, *не* элементами множества  $\Sigma \cup \{\varepsilon\}$ .

Мы уже упомянули выше, что очередной способ задания класса регулярных языков *не* является целью статьи, и вообще, не является целью описываемого нами обобщения конечных автоматов – но что же *является* этой целью? Наше обобщение автоматов может применяться, например, в следующих трёх случаях.

- Во-первых, это различные *точные алгоритмы* («переборные», brute-force algorithms), описываемые с помощью подобных автоматов. Они могут быть применены в нескольких минимизационных проблемах, рассмотренных нами, например в [9], [10]. См. также далее описание алгоритмов для возможности добавления дуги – это добавление часто даёт возможность использовать различные вспомогательные алгоритмы.
- Во-вторых, это различные *эвристические алгоритмы реального времени* – применяемые для тех же самых проблем минимизации; см., например, [11], [12].
- И, в-третьих, путём использования таких автоматов мы часто упрощаем некоторые доказательства; например, некоторые из доказательств, которые мы собираемся рассмотреть в части II, могут быть применены и для обычных НКА.

Приведём содержание настоящей статьи. В разделе II приведены некоторые обозначения, введённые в наших предыдущих статьях и используемые в настоящей. В разделе III даны основные определения, связанные с расширенными конечными автоматами; эти определения были впервые введены в [3]. Также в этом разделе приведено одно из свойств расширенных автоматов – которое фактически определяет их важный частный случай.

В разделе IV мы определяем расширенный конечный автомат  $\mathcal{K}(L)$  – основной предмет настоящей статьи. В разделе V мы рассматриваем пример такого автомата и на этом примере показываем следующий важный факт: существуют регулярные языки, в базисных автоматах которых у некоторых вершин нет циклов, соответствующих

циклам соответствующих вершин других эквивалентных автоматов.

(Отметим, что термин «соответствующие циклы» применялся нами в названии статьи [13] – это был основной предмет той статьи. А термин «соответствующие вершины» применялся нами для записей вида

$$q \in \langle (A, X) \rangle_K,$$

см. подробнее в следующем разделе. Существенно более сложная классификация вершин и циклов графа переходов недетерминированного конечного автомата была приведена нами в [14] – однако для настоящей статьи эта классификация вряд ли интересна. Таким образом, последнее предложение предыдущего абзаца вполне правильное – хотя оно и «не очень красивое».)

В разделе VI мы показываем адекватность определения расширенного базисного конечного автомата – т.е. доказываем утверждение о том, что построенный на основе исходного языка  $L$  автомат  $\mathcal{K}(L)$  действительно определяет этот язык<sup>3</sup>. А в разделе VII мы кратко рассматриваем алгоритмы добавления петли к некоторому состоянию автомата  $\mathcal{K}(L)$ ; подробнее эти алгоритмы будут рассмотрены в части II настоящей статьи.

## II. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ

Будем использовать обозначения из наших предыдущих работ, основные из этих обозначений связаны с каноническими конечными автоматами. А именно, пусть некоторый автомат

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

задаёт регулярный язык  $L$ . Соответствующий канонический автомат будем обозначать записью  $\tilde{L}$ .<sup>4</sup> Определим зеркальный конечный автомат как

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

где условие

$$\delta^R(q_1, q_2) \ni a$$

выполнено тогда и только тогда, когда

$$\delta(q_2, q_1) \ni a.$$

Будем обозначать автомат, зеркальный к автомату  $K$ , записью  $K^R$ .

Будем считать, что автоматы  $\tilde{L}$  и  $\tilde{L}^R$  для заданного регулярного языка  $L$  таковы:

$$\tilde{L} = (Q, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi) \quad \text{и} \quad \tilde{L}^R = (R, \Sigma, \delta_\rho, \{s_\rho\}, F_\rho).$$

Записью  $\langle L \rangle$  будем обозначать автомат  $(\tilde{L}^R)^R$ .

Базисный конечный автомат определяем согласно [3], [15]. При этом для некоторой пары состояний  $A \in Q$  и  $X \in R$  мы в наших предыдущих работах для соответствующего им состояния базисного автомата употребляли два варианта записи: либо  $\overset{A}{X}$ , либо  $(A, X)$  (при этом второй вариант, конечно, может также обозначать

<sup>3</sup> Конечно же, имеется в виду определение регулярного языка *не* на основе классических определений теории конечных автоматов, а на основе материала, изложенного в разделе III.

<sup>4</sup> При этом, как и в наших предыдущих работах, мы *не* требуем, чтобы канонический автомат был всюду определённым (total). Следовательно, в нём, вообще говоря, нет т.н. «дохлого состояния» (dead state).

<sup>2</sup> В упомянутой работе мы начали рассматривать *только одну* из этих моделей – и предполагаем вернуться к их рассмотрению в последующих публикациях.

упорядоченную пару состояний). В настоящей статье мы будем использовать оба варианта – в зависимости от того, какой именно из них более удобен в нужном месте.

Для автомата  $K$  (1), определяющего язык  $L$ , и состояния  $(A, X)$  эквивалентного базисного автомата  $\mathcal{BA}(L)$  будем писать

$$\langle (A, X) \rangle_K = \left\{ q \in Q \mid \varphi_K^{in}(q) \ni A \text{ и } \varphi_K^{out}(q) \ni X \right\}$$

(об этой записи кратко было упомянуто во введении, в тексте про соответствующие вершины).

Для дальнейшего важными являются следующие два утверждения. Они приведены в [16], здесь мы их повторяем – причём доказательство приведено с небольшими дополнениями.

**Утверждение 1:** Пусть  $A$  – некоторое состояние автомата  $\tilde{L}$ . Тогда для каждого состояния  $X$  автомата  $\tilde{L}^R$  выполнено условие

$$\mathcal{L}_{\tilde{L}}^{in}(A) = \mathcal{L}_{\mathcal{BA}(L)}^{in}((A, X)).$$

*Доказательство.* Условие

$$\mathcal{L}_{\mathcal{BA}(L)}^{in}((A, X)) \subseteq \mathcal{L}_{\tilde{L}}^{in}(A)$$

очевидно – оно просто является следствием определения автомата  $\mathcal{BA}(L)$ . Поэтому мы должны доказать выполнение включения

$$\mathcal{L}_{\tilde{L}}^{in}(A) \subseteq \mathcal{L}_{\mathcal{BA}(L)}^{in}((A, X));$$

докажем это утверждение по индукции.

Пусть  $u \in \mathcal{L}_{\tilde{L}}^{in}(A)$ . Базис индукции (т.е. случай  $u = \varepsilon$ ) очевиден. Докажем шаг индукции, т.е. рассмотрим случай  $|u| \geq 1$ . При этом мы можем записать  $u$  в виде  $u = va$ , где  $a \in \Sigma$  и  $v \in \Sigma^*$ . Пусть  $B$  – состояние автомата  $\tilde{L}$ , такое что  $v \in \mathcal{L}_{\tilde{L}}^{in}(B)$ .

Рассмотрим некоторое слово  $w \in \Sigma^*$ , такое что

$$w^R \in \mathcal{L}_{\tilde{L}^R}^{in}(X)$$

(т.е.  $uw = vaw$  является словом рассматриваемого языка  $L$ ). Такое слово существует – поскольку автомат  $\tilde{L}^R$  по определению не имеет ни бесполезных, ни недостижимых состояний.

Рассмотрим состояние (пусть  $Y$ ) автомата  $\tilde{L}^R$ , такое что

$$w^R a \in \mathcal{L}_{\tilde{L}^R}^{in}(Y).$$

Такое состояние существует – иначе слово  $uw = vaw$  не могло бы входить в заданный язык  $L$ .

Согласно определению бинарного отношения  $\#$ , выполнено условие  $B\#Y$ . А по определению автомата  $\mathcal{BA}(L)$  мы получаем, что

$$(B, Y) \xrightarrow{\mathcal{BA}(L)} (A, X).$$

Последний факт доказывает выполнение шага индукции.  $\square$

Аналогично доказывается следующий факт.

**Утверждение 2:** Пусть  $L$  – заданный регулярный язык, Пусть  $X$  – некоторое состояние автомата  $\tilde{L}^R$ .<sup>5</sup> Тогда для каждого состояния  $A$  автомата  $\tilde{L}$  выполнено условие

$$\mathcal{L}_{\langle A, X \rangle}^{out}(X) = \left( \mathcal{L}_{\tilde{L}^R}^{in}(X) \right)^R = \mathcal{L}_{\mathcal{BA}(L)}^{out}((A, X)). \quad \square$$

<sup>5</sup> А, следовательно, и автомата  $\langle L \rangle$ .

### III. ОПРЕДЕЛЕНИЕ РАСШИРЕННЫХ КОНЕЧНЫХ АВТОМАТОВ И СООТВЕТСТВУЮЩИХ РЕГУЛЯРНЫХ ЯЗЫКОВ

Рассмотрим основные определения, связанные с расширенными конечными автоматами. Заранее отметим, что для их функций переходов мы будем использовать обозначения, определённые ранее для автоматов «обычных»; например, для некоторого  $u \in \zeta(q)$  мы будем писать

$$q \xrightarrow[u]{\kappa} q.$$

**Определение 1:** Определим расширенный автомат как

$$\mathcal{K} = (Q, \Sigma, \delta, \zeta, \zeta^{in}, \zeta^{out}), \quad (2)$$

где  $K = (Q, \Sigma, \delta, S, F)$  – автомат (1), а остальные элементы являются функциями вида

$$\zeta, \zeta^{in}, \zeta^{out} : Q \rightarrow \mathcal{P}(\Sigma^*).$$

Кроме того:

- $\zeta(q) \ni \varepsilon$  для каждого  $q \in Q$ ;
- $\zeta^{in}(s) \ni \varepsilon$  для каждого  $s \in S$ ;
- $\zeta^{out}(f) \ni \varepsilon$  для каждого  $f \in F$ .  $\square$

**Определение 2:** Пусть  $q_1, q_2 \in Q$  – некоторые состояния расширенного автомата (2); допускаем случай  $q_1 = q_2$ . Определим язык  $\mathcal{L}_{\mathcal{K}}^{io}(q_1, q_2)$  следующим образом:

- если  $u \in \mathcal{L}_{\mathcal{K}}^{io}(q_1, q_2)$ , то мы считаем, что также

$$u \in \mathcal{L}_{\mathcal{K}}^{io}(q_1, q_2);$$

- если  $q_1 = q_2$ , то  $\zeta(q_1) \subseteq \mathcal{L}_{\mathcal{K}}^{io}(q_1)$ ;
- если для некоторого состояния  $q \in Q$  мы уже определили, что

$$u \in \mathcal{L}_{\mathcal{K}}^{io}(q_1, q), \quad v \in \mathcal{L}_{\mathcal{K}}^{io}(q) \quad \text{и} \quad w \in \mathcal{L}_{\mathcal{K}}^{io}(q, q_2),$$

то мы полагаем, что  $uvw \in \mathcal{L}_{\mathcal{K}}^{io}(q_1, q_2)$ ;

- ничто иное не является словом языка  $\mathcal{L}_{\mathcal{K}}^{io}(q_1, q_2)$ .  $\square$

**Определение 3:** Язык расширенного автомата (2) определяется как

$$CL(\mathcal{K}) = \bigcup_{q \in Q} \zeta^{in}(q) \cdot \mathcal{L}_{\mathcal{K}}^{io}(q) \cdot \zeta^{out}(q). \quad \square$$

Очевидно, что не каждый язык, который может быть задан согласно определению 3, является регулярным – просто потому, что множества  $\zeta$ ,  $\zeta^{in}$  и  $\zeta^{out}$  могут быть бесконечными. Однако мы будем рассматривать только такие случаи, когда эти множества являются регулярными языками – из чего следует, например, что язык  $\mathcal{L}(\mathcal{K})$  тоже регулярен. При этом следующее утверждение 3 описывает некоторые возможные элементы множества  $\zeta(q)$ , соответствующие реальным автоматам.

Для функции переходов расширенного автомата будем использовать обозначения, ранее введённые для «обычных» автоматов; например, для некоторого слова  $u \in \zeta(q)$  можно писать

$$q \xrightarrow[u]{\kappa} q.$$

**Утверждение 3:** Пусть для некоторого регулярного языка  $L$  и определяющего его автомата (1) выполнены следующие условия:

$$q \in Q, \quad q \in \langle (A, X) \rangle_K, \quad \mathcal{L}_K^{io}(q) \ni v, \\ \mathcal{L}_{\mathcal{BA}(L)}^{in}((A, X)) \ni u \quad \text{и} \quad \mathcal{L}_{\mathcal{BA}(L)}^{out}((A, X)) \ni w.$$

Тогда для каждого  $i \geq 0$  выполнено условие  $uv^i w \in L$ .<sup>6</sup>

*Доказательство.* Будем в этом доказательстве использовать утверждение 1 без специальных на него ссылок.

Выберем некоторое слово

$$u_1 \in \mathcal{L}_K^{in}(q),$$

такое что

$$u_1 \in \mathcal{L}_L^{in}(A);$$

такой выбор возможен в связи с тем, что

$$A \in \varphi_K^{in}(q).$$

Для автомата  $K$  мы можем записать

$$\frac{u_1}{K} \rightarrow q \xrightarrow{v^i} q;$$

поэтому существует соответствующий путь автомата  $\tilde{L}$  для некоторого его состояния  $B$ . Итак,

$$A \xrightarrow{\tilde{L}} B. \quad (3)$$

Поскольку

$$u \in \mathcal{L}_{\mathcal{BA}(L)}^{in}((A, X)),$$

мы получаем, что

$$\frac{u}{\tilde{L}} \rightarrow A.$$

А отсюда вследствие (3) выполняется следующий факт:

$$\frac{u}{\tilde{L}} \rightarrow A \xrightarrow{v^i} B. \quad (4)$$

Из условий

$$u \in \mathcal{L}_{\mathcal{BA}(L)}^{in}((A, X)), \quad q \in \langle (A, X) \rangle_K \quad \text{и} \quad v \in \mathcal{L}_K^{io}(q)$$

мы получаем, что

$$\frac{u}{K} \rightarrow q \xrightarrow{v^i} q. \quad (5)$$

Из (4) и (5) следует, что

$$\varphi_K^{in}(q) \ni B.$$

Кроме того, существует слово  $w$ , такое что

$$w \in \mathcal{L}_{\mathcal{BA}(L)}^{out}((A, X)),$$

то есть

$$w \in \mathcal{L}_{LR}^{in}(X).$$

Отсюда  $B\#X$ , и, кроме того,  $uv^i w \in L$ .  $\square$

Итак, в утверждении 3 мы фактически рассмотрели *специальный случай* расширенного автомата.

<sup>6</sup> Заметим, что, вообще говоря,  $u \notin \mathcal{L}_K^{in}(q)$  и  $w \notin \mathcal{L}_K^{out}(q)$ .

#### IV. РАСШИРЕННЫЙ БАЗИСНЫЙ КОНЕЧНЫЙ АВТОМАТ: ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ

Всюду в этом разделе считаем регулярный язык  $L$  заданным. Для него определим расширенный конечный автомат  $\mathcal{BE}(L)$ , элементы которого задаются согласно следующему определению.

**Определение 4:** Пусть  $(A, X)$  – некоторое состояние автомата  $\mathcal{BA}(L)$  для заданного регулярного языка  $L$ . Определим функции

$$\zeta_{\mathcal{BE}(L)}, \zeta_{\mathcal{BE}(L)}^{in}, \zeta_{\mathcal{BE}(L)}^{out} : \mathcal{T}(L) \rightarrow \mathcal{P}(\Sigma^*)$$

следующим образом.

- Пусть (1) – некоторый конечный автомат, определяющий  $L$ , а  $q \in Q$  – некоторое его состояние, такое что  $q \in \langle (A, X) \rangle_K$ . Пусть также

$$u \in \mathcal{L}_K^{io}(q).$$

Тогда будем считать, что

$$u \in \zeta_{\mathcal{BE}(L)}((A, X)).$$

Ничто иное словом языка

$$\zeta_{\mathcal{BE}(L)}((A, X))$$

не является.

- Пусть (1) – некоторый конечный автомат, определяющий язык  $L$ ,<sup>7</sup> а  $q \in Q$  – некоторое его состояние, такое что

$$q \in \langle (A, X) \rangle_K.$$

Пусть также

$$u \in \mathcal{L}_K^{in}(q).$$

Тогда будем считать, что

$$u \in \zeta_{\mathcal{BE}(L)}^{in}((A, X)).$$

Ничто иное словом языка

$$\zeta_{\mathcal{BE}(L)}^{in}((A, X))$$

не является.

- Пусть (1) – некоторый конечный автомат, определяющий  $L$ , а  $q \in Q$  – некоторое его состояние, такое что

$$q \in \langle (A, X) \rangle_K.$$

Пусть также

$$u \in \mathcal{L}_K^{out}(q).$$

Тогда будем считать, что

$$u \in \zeta_{\mathcal{BE}(L)}^{out}((A, X)).$$

Ничто иное словом множества

$$\zeta_{\mathcal{BE}(L)}^{out}((A, X))$$

<sup>7</sup> Важно отметить, что, вообще говоря, этот автомат отличается от рассматривавшегося в предыдущем пункте определения. То же – для следующего пункта. Ещё раз отметим, что в каждом из этих пунктов выбирается *какой-нибудь произвольный* автомат для языка  $L$ , причём для разных слов  $u$ , рассматриваемых в каждом из этих пунктов, *автоматы могут быть различными*. Иными словами – мы рассматриваем *все возможные варианты* автоматов и соответствующих слов.

не является.<sup>8</sup>

Расширенный конечный автомат  $\mathcal{BE}(L)$  определяется как

$$\mathcal{BE}(L) = (\mathcal{T}(L), \Sigma, \delta_B, \zeta_{\mathcal{BE}(L)}, \zeta_{\mathcal{BE}(L)}^{in}, \zeta_{\mathcal{BE}(L)}^{out}).$$

V. ПРИМЕР

В качестве примера рассмотрим пример языка  $L$ , заданного регулярным выражением

$$(a + ab + ba)^*. \quad (6)$$

Соответствующий базисный автомат изображён на рис. 1.

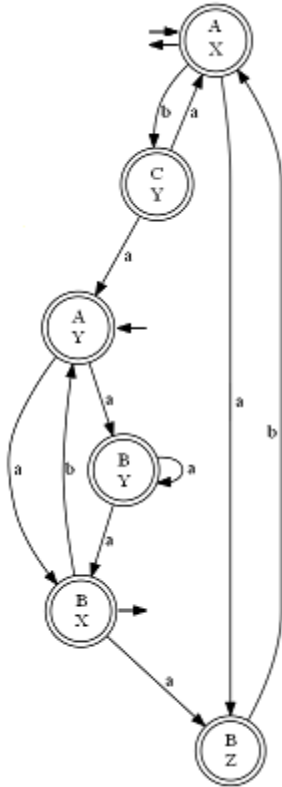


Рис. 1. Базисный автомат для языка, определяемого РВ (6)

В [3] мы отмечали, что каждый из языков

$$\zeta_{\mathcal{BE}(L)}((B, Y)), \quad \zeta_{\mathcal{BE}(L)}^{in}((B, Y)) \quad \text{и} \quad \zeta_{\mathcal{BE}(L)}^{out}((B, Y))$$

соответствующих состоянию  $(B, Y)$  базисного автомата, содержит слово  $ab$  – несмотря на то, что этого слова не содержит ни один из аналогичных языков

$$\mathcal{L}_{\mathcal{BA}(L)}^{io}((B, Y)), \quad \mathcal{L}_{\mathcal{BA}(L)}^{in}((B, Y)) \quad \text{и} \quad \mathcal{L}_{\mathcal{BA}(L)}^{out}((B, Y))$$

(для того же самого состояния базисного автомата  $(B, Y)$ , рис. 1). Для первого из упомянутых языков, т.е.  $\zeta_{\mathcal{BE}(L)}((B, Y))$ , это можно показать, например, следующим образом:

- состояние  $q_1$  соответствующего этому языку автомата, приведённого на рис. 2, содержит цикл, имеющий пометку  $ab$ ;
- мы показывали в [17] и других статьях, что для рассматриваемого языка выполнено следующее:

$$\varphi_K^{in}(q_1) \ni B \quad \text{и} \quad \varphi_K^{out}(q_1) \ni Y;$$

<sup>8</sup> Условия двух последних пунктов можно переписать в виде

$$\varphi_K^{in}(q) \ni A \quad \text{и} \quad \varphi_K^{out}(q) \ni X$$

соответственно.

- к последним двум фактам применяем определение 4.

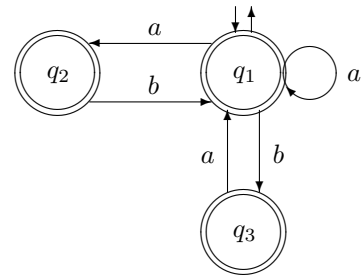


Рис. 2. Автомат для языка, определяемого РВ (6)

Таким образом, «всё не так просто»: на этом примере мы показываем, что существуют регулярные языки, в базисных автоматах которых у некоторых вершин нет циклов, соответствующих циклам соответствующих вершин других эквивалентных автоматов<sup>9</sup>. Но, несмотря на это, выполняется важная теорема 1, приведённая в следующем разделе; она фактически показывает адекватность определения расширенного базисного конечного автомата – т.е. тот факт, что построенный на основе исходного языка  $L$  автомат  $\mathcal{BE}(L)$  действительно определяет этот язык. А в качестве некоторого дополнения к теореме 1 можно рассматривать материал раздела VII; к этим вопросам мы вернёмся в части II настоящей статьи.

VI. РАСШИРЕННЫЙ БАЗИСНЫЙ КОНЕЧНЫЙ АВТОМАТ: АДЕКВАТНОСТЬ ОПРЕДЕЛЕНИЯ

Докажем основное утверждение этого раздела.

**Теорема 1:**  $\mathcal{L}(\mathcal{BE}(L)) = L$ .

*Доказательство.* Достаточно доказать, что

$$\mathcal{L}(\mathcal{BE}(L)) \subseteq L$$

(обратное утверждение очевидно).

Рассмотрим некоторое слово  $w \in \mathcal{L}(\mathcal{BE}(L))$ . Поскольку  $w$  получено согласно определениям 3 и 4, мы можем записать его в виде

$$w = u_0(v_1)^{i_1}u_1(v_2)^{i_2}u_2 \dots u_{n-1}(v_n)^{i_n}u_n,$$

где:

- $v_j \in \zeta_{\mathcal{BE}(L)}(X_j^j)$  и  $i_j \geq 0$  для каждого  $j \in \{1, \dots, n\}$ ;
- $u_0 \in \zeta_{\mathcal{BE}(L)}^{in}(X^1)$  для состояния

$$X^1 \in \mathcal{T}(L),$$

соответствующего слову  $v_1$ ; отсюда согласно определению 4 мы можем выбрать состояние

$$X^0 \in \mathcal{T}(L),$$

такое что

$$X^0 \xrightarrow{\mathcal{BA}(L)}^{u_0} X^1,$$

где  $A^0 = s_\pi$  – стартовое состояние автомата  $\tilde{L}$ ;

- аналогично,

$$X^n \xrightarrow{\mathcal{BA}(L)}^{u_n} X^{n+1},$$

<sup>9</sup> Замечание о «не очень красивой» терминологии было приведено во введении.

где  $X^{n+1} = s_\rho$  – стартовое состояние автомата  $\widetilde{L}^R$ ;

- следовательно,  $u_0 u_1 u_2 \dots u_n$  – слово, являющееся пометкой пути автомата  $\mathcal{BA}(L)$ , то есть

$$\begin{matrix} A^0 & \xrightarrow{u_0} & A^1 & \xrightarrow{u_1} & A^2 & \xrightarrow{u_2} & \dots & \xrightarrow{u_{n-1}} & A^n & \xrightarrow{u_n} & A^{n+1} \\ X^0 & \xrightarrow{\mathcal{BA}(L)} & X^1 & \xrightarrow{\mathcal{BA}(L)} & X^2 & \xrightarrow{\mathcal{BA}(L)} & \dots & \xrightarrow{\mathcal{BA}(L)} & X^n & \xrightarrow{\mathcal{BA}(L)} & X^{n+1} \end{matrix} \quad (7)$$

для некоторых состояний

$$\begin{matrix} A^0 & A^1 & A^2 & & A^n & A^{n+1} \\ X^0 & X^1 & X^2 & \dots & X^n & X^{n+1} \end{matrix} \in \mathcal{T}(L).$$

Заметим, что для каждого возможного  $j$  мы допускаем случай  $u_j = \varepsilon$ .

Определим

$$\begin{aligned} x_k &= u_0 (v_1)^{i_1} u_1 (v_2)^{i_2} u_2 \dots u_{k-1} (v_k)^{i_k} u_k, \\ x'_k &= u_0 u_1 \dots u_k, \quad \text{и} \quad y_k = u_{k+1} \dots u_n. \end{aligned}$$

Докажем по индукции по  $k \in 0, 1, \dots, n$ , что:

- $x_k y_k \in L$ ;
- для (единственного) пути автомата  $\mathcal{BA}(L)$  вида

$$\begin{matrix} x_k & \xrightarrow{t} & y_k \\ \mathcal{BA}(L) & & \mathcal{BA}(L) \end{matrix}$$

выполнено следующее условие:

$$t = \begin{matrix} B^{k+1} \\ X^{k+1} \end{matrix};$$

заметим, что мы используем заранее зафиксированное состояние  $X^{k+1}$  автомата  $\widetilde{L}^R$ , а также некоторое состояние  $B^{k+1}$  автомата  $\widetilde{L}$ .<sup>10, 11</sup>

Базис индукции (т.е. случай  $k = 0$ ) очевиден – при этом мы рассматриваем  $w$  как слово автомата  $\mathcal{BA}(L)$ , а не  $\mathcal{BE}(L)$ ; таким образом,

$$i_1 = i_2 = \dots = i_n = 0.$$

Докажем шаг индукции.

Рассмотрим слово

$$w'_k = x'_k (v_{k+1})^{i_{k+1}} y_k.$$

Согласно утверждению 3,  $w'_k \in L$ .<sup>12</sup> Кроме того, вследствие (7),  $x'_k y_k \in L$ . Также на основе (7) мы получаем, что

$$\begin{matrix} (y_k) \\ \widetilde{L}^R \end{matrix} \xrightarrow{R} X^{k+1},$$

поэтому

$$\begin{matrix} x'_k & \xrightarrow{\mathcal{BA}(L)} & A^{k+1} & \xrightarrow{(v_{k+1})^{i_{k+1}}} & C^{k+1} & \xrightarrow{\mathcal{BA}(L)} & y_k \\ & & X^{k+1} & & X^{k+1} & & \mathcal{BA}(L) \end{matrix} \quad (8)$$

для некоторого состояния  $C^{k+1}$  автомата  $\widetilde{L}$ .<sup>13</sup>

По предположению индукции и (8) мы для некоторого состояния  $D^{k+1}$  автомата  $\widetilde{L}$  получаем следующий путь

$$\begin{matrix} x_k & \xrightarrow{\mathcal{BA}(L)} & B^{k+1} & \xrightarrow{(v_{k+1})^{i_{k+1}}} & D^{k+1} & \xrightarrow{\mathcal{BA}(L)} & y_k \\ & & X^{k+1} & & X^{k+1} & & \mathcal{BA}(L) \end{matrix} \quad (9)$$

<sup>10</sup> Отметим, что возможен случай  $B^{k+1} = A^{k+1}$ .

<sup>11</sup> Процесс доказательства по индукции мы иллюстрируем рисунком 3, на котором приведена работа автомата  $\mathcal{BA}(L)$ . Некоторые комментарии к рисунку будут приведены далее – также в сносках.

<sup>12</sup> Путь автомата  $\mathcal{BA}(L)$ , принимающий это слово, отмечен на рисунке малыми кружками, каждый из которых имеет номер 1. Отметим также, что автомат  $\mathcal{BA}(L)$  является однозначным – поэтому выражение *путь* (а не «один из путей») верно.

<sup>13</sup> Для одного из последних шагов индукции, а именно для  $k + 1 = n - 1$ , т.е.  $k = n - 2$ , состояние  $C^{k+1}$  отмечено знаком вопроса – внутри состояния базисного автомата, помеченного малым кружком с номером 2. Знак вопроса при этом используется потому, что таким же образом может быть отмечено и состояние  $D^{k+1}$ , которое будет введено в следующем абзаце.

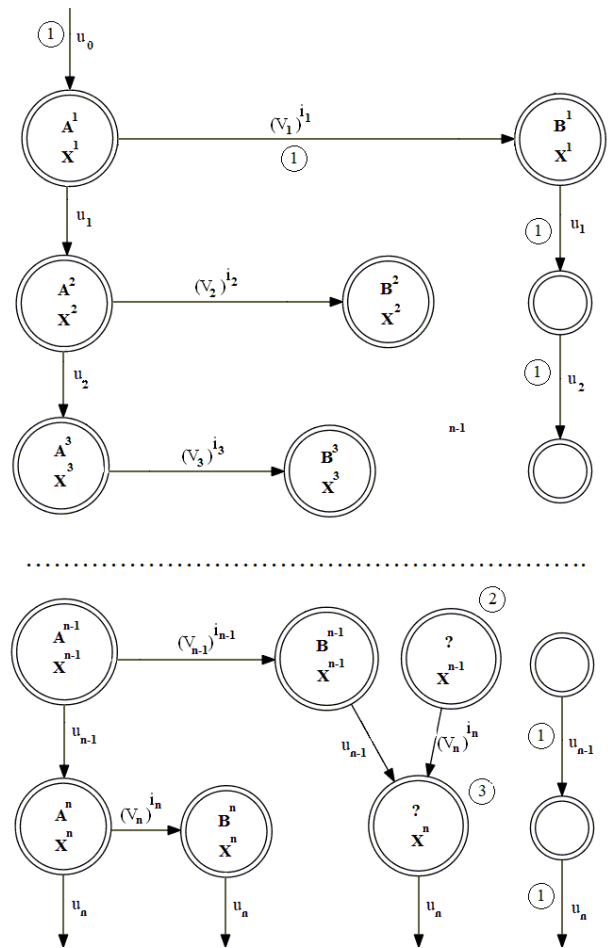


Рис. 3. Иллюстрация к доказательству теоремы

Кроме того, для  $k < n$  мы можем записать (9) в виде

$$\begin{matrix} x_k & \xrightarrow{\mathcal{BA}(L)} & B^{k+1} & \xrightarrow{(v_{k+1})^{i_{k+1}}} & D^{k+1} & \xrightarrow{\mathcal{BA}(L)} & y_{k+1} \\ & & X^{k+1} & & X^{k+1} & & \mathcal{BA}(L) \end{matrix} \quad (10)$$

Состояние  $X^{k+2}$  было зафиксировано ранее, поэтому состояние  $B^{k+2}$  может быть выбрано на основе (10).<sup>14</sup> А существование требуемого состояния  $B^{k+2}$  доказывает шаг индукции для  $k < n$ .

Итак, мы доказали, что  $w = x_n y_n \in L$ ; отсюда получаем, что  $\mathcal{L}(\mathcal{BE}(L)) = L$ .  $\square$

## VII. ОБ АЛГОРИТМАХ ДОБАВЛЕНИЯ ПЕТЛИ И О ЧАСТИ II НАСТОЯЩЕЙ СТАТЬИ

Как мы уже отмечали во введении, алгоритмы, которые могут быть созданы на основе описанных в настоящем разделе утверждений, на первый взгляд вряд ли могут быть названы «алгоритмами упрощения» автомата. Однако во многих задачах преобразования конечных автоматов эти алгоритмы могут быть вспомогательными «для временного небольшого ухудшения структуры автомата с целью её существенного последующего улучшения» – т.е. для последующего удаления некоторых состояний автомата и/или его дуг; таким образом, эти алгоритмы

<sup>14</sup> Процесс выбора состояния  $B^{k+2}$  – сначала неизвестного, поэтому также первоначально обозначенному знаком вопроса, – помечен на рисунке малым кружком с номером 3.

действительно могут являться вспомогательными алгоритмами для минимизационных проблем. Например, алгоритмы, сформулированные в [18] для описания множества возможных дуг, являются существенно более быстрыми, чем простейшие переборные алгоритмы<sup>15</sup>, – однако всё же вряд ли могут быть использованы на практике. Поэтому, рассматривая конкретный конечный автомат, мы можем «постараться» описать множество «недостающих в нём» возможных дуг<sup>16</sup>.

**Утверждение 4:** Пусть  $q$  – некоторое состояние автомата  $\mathcal{BA}(L)$ , причём

$$q \xrightarrow{u} q.$$

Тогда  $u \in \zeta_{\mathcal{BA}(L)}(q)$ . □

Мы не будем строго доказывать это утверждение – а именно, мы не будем доказывать, что существует автомат вида (1), для которого требуется слово  $u$  может быть выбрано на основе определения 4; мы приведём такое доказательство в части II, в нём мы определим необходимые элементы автомата (1). Кроме того, следующим обстоятельством может даже являться аргументом, что подобное доказательство малоинтересно. Если мы просто добавим слово  $u$  в язык  $\zeta_{\mathcal{BA}(L)}$  (т.е. если мы таким образом дополним определение 4), то очевидно, что мы получим автомат, эквивалентный заданному; этот факт можно рассматривать как следствие определения 3. Например, для регулярного выражения (6) на основе показанного выше факта

$$\zeta_{\mathcal{BA}(L)}((B, Y)) \ni ab$$

и отмеченного выше мы можем утверждать, что к состоянию  $(B, Y)$  автомата  $\mathcal{BA}(L)$  можно добавить цикл<sup>17</sup> с пометкой  $ab$ .

#### Список литературы

- [1] Melnikov B., 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. – P. 495–506.
- [2] Melnikov B., Melnikova A. *Pseudo-automata for generalized regular expressions* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 1. – P. 1–8.
- [3] Melnikov B. *Extended nondeterministic finite automata* // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 255–265.
- [4] Саломая А. *Жемчужины теории формальных языков*. М.: Мир, 1986. – 160 с.
- [5] Мельников Б.Ф. *Об одной классификации последовательностных контекстно-свободных языков и грамматик* // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. – 1993. – № 3. – С. 64–67.
- [6] Выхиток А.А., Зубова М.А., Мельников Б.Ф. *Об одном расширении класса конечных автоматов для задания контекстно-свободных языков* // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. – 2013. – № 1. – С. 39–45.
- [7] Generalova T. V., Melnikov B. F., Vylitok A. A. *On the problem of vertex optimization of bracketed automata* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 9. – P. 1–8.
- [8] Han Y.-S., Wood D. *The generalization of generalized automata: expression automata* // Proceedings of The 9th International Conference on Implementation and Application of Automata. – 2004. – P. 114–122.
- [9] Мельников Б.Ф., Мельникова А.А. *Многоаспектная минимизация недетерминированных конечных автоматов (Часть I. Вспомогательные факты и алгоритмы)* // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2011. – № 4 (20). – С. 59–69.
- [10] Мельников Б.Ф., Мельникова А.А. *Многоаспектная минимизация недетерминированных конечных автоматов (Часть II. Основные алгоритмы)* // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2012. – № 1 (21). – С. 31–43.
- [11] Баумгертнер С.В., Мельников Б.Ф. *Мультиэвристический подход к проблеме звездо-высотной минимизации недетерминированных конечных автоматов* // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2010. – № 1. – С. 5–7.
- [12] Melnikov B., Tsyganov A. *The state minimization problem for nondeterministic finite automata: The parallel implementation of the truncated branch and bound method* // Proceedings of the International Symposium on Parallel Architectures, Algorithms and Programming, PAAP-2012. Taipei, Taiwan. – 2012. – P. 194–201.
- [13] 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. – P. 9–14.
- [14] 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. – P. 1–6.
- [15] Melnikov B. *Once more on the edge-minimization of nondeterministic finite automata and the connected problems* // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 267–283.
- [16] Мельникова А.А. *Использование функций разметки при работе с циклами базисного конечного автомата* // Вестник Тамбовского университета. Серия: Естественные и технические науки. – 2015. – Т. 20. № 5. – С. 1310–1312.
- [17] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [18] 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. – P. 475–485.
- [19] Мельников Б.Ф., Сайфуллина М.Р. *О некоторых алгоритмах эквивалентного преобразования недетерминированных конечных автоматов* // Известия высших учебных заведений. Математика. – 2009. – № 4. – С. 67–72.

Борис Феликсович МЕЛЬНИКОВ,  
профессор Университета МГУ–ППИ в Шэнчжэне  
(<http://szmsubit.ru/>),  
email: [bf-melnikov@yandex.ru](mailto:bf-melnikov@yandex.ru),  
[mathnet.ru](mailto:bf-melnikov@mathnet.ru): personid=27967,  
[elibrary.ru](http://elibrary.ru): authorid=15715,  
[scopus.com](http://scopus.com): authorId=55954040300,  
ORCID: [orcidID=0000-0002-6765-6800](http://orcid.org/0000-0002-6765-6800).

Александра Александровна МЕЛЬНИКОВА,  
доцент Димитровградского инженерно-технологического  
института – филиала Национального исследовательского  
ядерного университета «МИФИ»  
(<https://diti-mephi.ru/>),  
email: [super-avahi@yandex.ru](mailto:super-avahi@yandex.ru),  
[mathnet.ru](mailto:super-avahi@mathnet.ru): personid=148963,  
[elibrary.ru](http://elibrary.ru): authorid=143351.

<sup>15</sup> Сложность алгоритмов мы в настоящей статье не рассматриваем. Однако, по-видимому, в данной ситуации очевидно, какой из подходов даёт более простые алгоритмы.

<sup>16</sup> Понятно, что последнее приведённое предложение с математической точки зрения не является строгим. Однако при построении эвристических алгоритмов нам удаётся применять подобные методы – поскольку мы не обязаны постоянно проверять рассматриваемые автоматы на эквивалентность. См. краткое описание «алгоритмов синтеза автоматов» в [19] – к этой теме мы также предполагаем вернуться.

<sup>17</sup> Согласно определению расширенных автоматов, такой цикл можно назвать петлёй. Если же мы рассматриваем обычные НКА, термин «петля» употреблять нельзя.

# Extended basis finite automaton.

## Part I. The basic definitions

Boris Melnikov, Aleksandra Melnikova

**Abstract**—In this paper, we consider a special extension of the class of non-deterministic finite automata. The purpose of this consideration is the following: first, to describe using such automata various brute-force algorithms of equivalent transformation of automata; secondly, the application of the described extension in several minimization problems for ordinary non-deterministic finite automata; thirdly, with the help of these automata we can simplify some proofs, which are also necessary for ordinary finite automata.

In the paper, we consider the definition of extended finite automata, give examples, define the extended basic finite automaton for the given regular language and consider some properties of this automaton. We also prove, that for the extended basis automaton all the auxiliary languages obtained in the constructions are regular. All this allows to simplify the definition of an extended basic automaton.

We also show the adequacy of the definition of an extended basis finite automaton, i.e. we prove the assertion that the extended basic automaton constructed on the basis of the source regular language really defines this language. And on one of the examples we show the following fact: there are regular languages, in whose basic automata some vertices do not have cycles corresponding to the cycles of the corresponding vertices of other equivalent automata; such cycles exist in similar vertices of extended basic automata.

**Keywords**—Regular languages, nondeterministic finite automata, the extension of automata, algorithms of equivalent transformation for automata.

### References

- [1] Melnikov B., 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. – P. 495–506.
- [2] Melnikov B., Melnikova A. *Pseudo-automata for generalized regular expressions* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 1. – P. 1–8.
- [3] Melnikov B. *Extended nondeterministic finite automata* // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 255–265.
- [4] Salomaa A. *Jewels of formal language theory*. – Maryland, USA: Computer Science Press, Inc., 1981. – 144 p.
- [5] Melnikov B. *On a classification of sequential context-free languages and grammars* // Vestnik of Moscow University. Series 15: Computational Mathematics and Cybernetics. – 1993. – No. 3. – P. 64–67.
- [6] Vylitok A., Zubova M., Melnikov B. *On an extension of the class of finite automata for defining context-free languages* // Vestnik of Moscow University. Series 15: Computational Mathematics and Cybernetics. – 2013. – No. 1. – P. 39–45.
- [7] Generalova T., Melnikov B., Vylitok A. *On the problem of vertex optimization of bracketed automata* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 9. – P. 1–8.
- [8] Han Y.-S., Wood D. *The generalization of generalized automata: expression automata* // Proceedings of The 9th International Conference on Implementation and Application of Automata. – 2004. – P. 114–122.
- [9] Melnikov B., Melnikova A. *Multidimensional minimization of non-deterministic finite automata (Part I. Auxiliary facts and algorithms)* // News of Higher Educational Institutions. Volga Region. Physical and Mathematical Sciences. – 2011. – No 4 (20). – P. 59–69.
- [10] Melnikov B., Melnikova A. *Multidimensional minimization of non-deterministic finite automata (Part II. Basic algorithms)* // News of Higher Educational Institutions. Volga Region. Physical and Mathematical Sciences. – 2012. – No 1 (21). – P. 31–43.
- [11] Baumgärtner S., Melnikov B. *Multiheuristic approach to the problem of the star-height minimization of nondeterministic finite automata* // Vestnik of Voronezh State University. Series: System Analysis and Information Technologies. – 2010. – No. 1. – P. 5–7.
- [12] Melnikov B., Tsyganov A. *The state minimization problem for nondeterministic finite automata: The parallel implementation of the truncated branch and bound method* // Proceedings of the International Symposium on Parallel Architectures, Algorithms and Programming, PAAP-2012. Taipei, Taiwan. – 2012. – P. 194–201.
- [13] 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. – P. 9–14.
- [14] 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. – P. 1–6.
- [15] Melnikov B. *Once more on the edge-minimization of nondeterministic finite automata and the connected problems* // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 267–283.
- [16] Melnikova A. *Using state-marking functions when working with the loops of the basis finite automaton* // Vestnik of Tambov State University. Series: System Analysis and Information Technologies. – 2015. – Vol. 20. No 5. – P. 1310–1312.
- [17] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [18] 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. – P. 475–485.
- [19] Melnikov B., Sayfullina M. *On some algorithms for the equivalent transformation of nondeterministic finite automata* // News of Higher Educational Institutions. Mathematics. – 2009. – No 4. – P. 67–721.

Boris Feliksovich MELNIKOV,  
 Professor of Shenzhen MSU–BIT University  
 (<http://szmsubit.ru/>),  
 email: [bf-melnikov@yandex.ru](mailto:bf-melnikov@yandex.ru),  
 mathnet.ru: personid=27967,  
 elibrary.ru: authorid=15715,  
 scopus.com: authorId=55954040300,  
 ORCID: [orcidID=0000-0002-6765-6800](https://orcid.org/0000-0002-6765-6800).

Aleksandra Aleksandrovna MELNIKOVA,  
 Associated Professor (Docent)  
 of Dimitrovgrad Engineering and Technology Institute –  
 Branch of National Research Nuclear University “MEPhI”  
 (<https://diti-mephi.ru/>),  
 email: [super-avahi@yandex.ru](mailto:super-avahi@yandex.ru),  
 mathnet.ru: personid=148963,  
 elibrary.ru: authorid=143351.