

Лепестковые конечные автоматы: основные определения, примеры и их связь с полными автоматами. Часть I

Б. Ф. Мельников

Аннотация—В статье рассматриваются недетерминированные конечные автоматы специального вида; в литературе на русском языке мы не нашли публикаций, определяющих название для таких автоматов, — поэтому предлагаем для них новое название, «лепестковые автоматы». (В единственной найденной публикации в англоязычной литературе употреблён термин “semi-flower automata” — по-видимому, не очень удачный.)

Исследование подобных автоматов очень важно для нескольких тем (тематик), рассмотренных в наших предыдущих публикациях. Прежде всего это связь с алгоритмами решения задач, относящихся к различным вариантам исследования равенства бесконечных итераций двух конечных языков, — в частности, к описанию алгоритмов для решения этих задач в виде специального вида недетерминированных конечных автоматов (т.н. автоматов PRI и NSPRI). Близки к этой теме и другие задачи, которые в принципе также могут быть решены путём перебора всех подмножеств множества потенциальных корней слов некоторого заданного языка (например, задача извлечения корня некоторой степени из языка) — но очень важная вспомогательная цель заключается в том, чтобы решить эти задачи с помощью алгоритмов, менее сложных, чем экспоненциальные.

В качестве ещё двух возможных тематик применения лепестковых автоматов приведём такие. Возможно применение полученных результатов во вспомогательных алгоритмах, необходимых для более общих алгоритмов вершинной и дуговой минимизации недетерминированных конечных автоматов. Кроме того, исследование лепестковых автоматов можно связать с исследованием самой таблицы бинарного отношения # — поскольку варианты таких таблиц разбивают всё множество регулярных языков на специальные классы эквивалентности.

Возможность применения лепестковых автоматов в последней тематике вытекает из основного результата настоящей статьи — а именно, из приведённого в ней описания алгоритма построения такого лепесткового автомата, таблицы бинарного отношения # которого совпадает с заданной таблицей — возможно, с некоторым единственным специально добавленным правым столбцом.

В предлагаемой первой части настоящей статьи приводятся основные определения и на примере рассматривается начальная фаза требуемого алгоритма построения лепесткового автомата. Эта фаза связана со специальными преобразованиями полного автомата для заданной таблицы бинарного отношения # — преобразованиями, вообще говоря, неэквивалентными, но в то же время сохраняющими это бинарное отношение.

Ключевые слова—формальные языки, итерации языков, морфизмы, бинарные отношения, лепестковые автоматы, полные автоматы.

I. ВВЕДЕНИЕ И МОТИВАЦИЯ

В настоящей статье рассматриваются (недетерминированные) конечные автоматы специального вида; в литературе на русском языке автор не нашёл публикаций, рассматривающих такие автоматы, — поэтому для них предлагается новое название, «лепестковые автоматы». Более того, и в англоязычной литературе фактически удалось найти только одну публикацию про такие автоматы — [1] (остальные найденные в Интернете публикации — уровня PhD-диссертаций и статей на arXiv.org). Однако название “semi-flower automata”, употребляемое в той статье (и связанных PhD-публикациях), не представляется удачным¹, и именно поэтому мы выбрали не кальку этого названия, а другое название — хотя и «из той же предметной области». Определение таких автоматов уже встречалось в наших предыдущих публикациях — некоторые подробности см. далее, в этом и следующем разделах.

Кратко общая схема графа переходов лепесткового автомата показана на рис. 1:

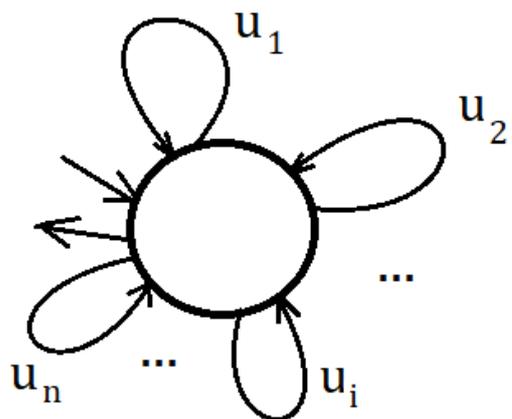


Рис. 1. Общая схема графа переходов лепесткового автомата.

(Про применяемые в наших статьях рисунки графов переходов отметим следующее. В них никогда — в отличие от очень многих классических публикаций — двойные окружности для обозначения выходных состояний не используются: для таковых мы всегда используем маленькие выходящие стрелки. Кроме того, в подавляющем большинстве публикаций мы не различаем такие

Статья получена 26 апреля 2022 г.
Борис Феликсович Мельников, Университет МГУ–ППИ в Шэньчжэне (bf-melnikov@yandex.ru).

¹ А также возникает «вопрос без ответа»: а что такое “flower automata”?

двойные окружности и окружности обычные – а исключением из последнего являются только статьи, посвящённые обобщённым недетерминированным автоматам, например, [2], [3]; но про такое возможное различие мы всегда специально предупреждаем в начале статьи. И, конечно, входные состояния автоматов всегда помечаются маленькими входящими стрелками.)

Кроме того, по поводу приведённого на рис. 1 автомата сразу отметим два очевидных факта. Во-первых, обозначая

$$A = \{u_1, u_2, \dots, u_i, \dots, u_n\}, \quad (1)$$

где $u_1, \dots, u_i, \dots, u_n \in \Sigma^*$,

мы получаем, что этот автомат определяет язык A^* (над заданным конечным алфавитом Σ)². Во-вторых, некоторые из слов языка A могут иметь общие префиксы (в частности, совпадающие первые буквы) – и поэтому подобные автоматы, вообще говоря, не являются детерминированными.

Более подробный рисунок, описывающий такие автоматы (рис. 2, побуквенное представление самих «лепестков»), приведён в конце настоящей статьи. Отметим, что таких ситуаций (когда рисунок или таблица приводятся в конце текста) далее будет много – и в основном тексте мы всегда будем писать про эти ситуации следующим образом:

(Рис. 2 приведён в конце текста статьи.)

Вернёмся к рис. 2. Для большей аналогии на нём справа приведён и схематичный рисунок лепестков реальной ромашки – конечно же, пожой на рис. 1 и левую часть рис. 2.

Выше автор уже упоминал про свои предыдущие публикации, посвящённые таким автоматам. В работах [4], [5], [6], [7], [8], [9] приводились немного различающиеся определения (обычно для заданного конечного языка $A \subseteq \Sigma^*$ введённые автомат, обозначавшийся $\mathcal{K}(A)$); при этом, как уже было отмечено, в некоторых статьях автоматы иногда задавали не язык A^* , а язык $\text{Pref}(A^*)$. Кроме того, в некоторых из указанных публикаций (а именно – в [4], [6]), а также в [10], [11], [12], [13], [14], приводилась связь итераций языков и соответствующих им автоматов с *бесконечными* итерациями (ω -итерациями) – и также с соответствующими им ω -автоматами. Более того, как название, так и *основной предмет* статьи [12] связаны именно с ω -автоматами³. Однако при этом в указанных выше статьях *подробного* исследования таких автоматов и ω -автоматов не делалось – и мы предполагаем заняться этими вопросами начиная с настоящей статьи.

В качестве ещё трёх возможных «пунктов мотивации» приведём такие.

- Во-первых – это связь с тематикой наших последних работ [9], [15], [16]⁴: ведь лепестковые автоматы

² В одной из предыдущих публикациях рассматривались (недетерминированные) конечные автоматы, определяющие язык $\text{Pref}(A^*)$; немного подробнее – ниже в этом разделе.

³ Часть II этой статьи предполагается к публикации в будущем. Более-менее подробный план этой предполагаемой части II упомянутой статьи – мы приведём в части II статьи настоящей (в разделе VIII, «Заключение»).

⁴ А также – работ, процитированных в трёх частях этой статьи.

(или определяемые далее автоматы $\mathcal{K}(A)$) возникают именно при описании нескольких различных алгоритмов проверки рассматриваемого в этих статьях бинарного отношения \trianglelefteq .

- Во-вторых – связь с алгоритмами вершинной и дуговой минимизации недетерминированных конечных автоматов. ([17], [18] и мн. др.). *Очень неформально*⁵ эту связь можно объяснить так: «причина существования»
 - как исследованного в [19] автомата Ватерлоо (в частности),
 - так и подобных ему «валибадов»⁶ (вообще) – заключается в том, что при построении покрывающего автомата ([20] и др.) в автомат-ответ попадают *не все* циклы, соответствующие циклам базисного автомата (см. [21], [22] и особенно [23]), – и, следовательно, могут «теряться» входы и/или выходы автомата универсального.
- В-третьих – связь с исследованием самой таблицы бинарного отношения $\#$ (а также – определённого в [24] бинарного отношения \mathcal{R} , практически не исследованного в той статье). Ведь как в настоящей статье, так и в [24] мы фактически применяем *неэквивалентные упрощения* заданного автомата (оставляя неизменным само бинарное отношение $\#$) – что, конечно, должно помочь подробно исследовать и отношение \mathcal{R} .

Немного подробнее про эти «пункты мотивации» будет сказано в заключении (раздел VIII, часть II настоящей статьи).

Основной результат статьи – алгоритм построения такого лепесткового автомата, у которого:

- либо таблица бинарного отношения $\#$;
- либо такая таблица с удалённым правым столбцом совпадает с некоторой заданной таблицей.

Приведём содержание статьи по разделам.

В разделе II приведены необходимые определения и обозначения. Для заданного языка A мы определяем недетерминированный конечный автомат $\mathcal{K}(A)$, языком которого является A^* , – и при этом небольшая модификация этого автомата даёт недетерминированный автомат с языком $\text{Pref}(A^*)$. Кроме того, в этом же разделе II приведены необходимые сведения о полном автомате (complete automaton) – взятые, в основном, из [24].

В разделах III и V подробно рассмотрен пример возможного алгоритма построения подходящего лепесткового автомата – на основе заданных таблицы бинарного отношения $\#$ и линейного порядка на множестве состояний. Сразу специально отметим, что рассмотрение примера приведено *до* самого описания алгоритма – поскольку, по-видимому, в данном случае именно такое изложение более удачное.

А ещё важнее следующее замечание. Как уже было отмечено, мы *не* строим лепестковый автомат, у которого таблица отношения $\#$ *в точности* совпадает с исходной – допускается один «новый» столбец. Понятно, что он соответствует возможному новому состоянию канониче-

⁵ «С философской точки зрения».

⁶ Waterloo-like badness'ов.

ского автомата для зеркального языка (в наших обычных обозначениях – \bar{L}^R)⁷.

Как уже сказано, рассмотрение примера разбито на два раздела; при этом разделы начиная с IV помещены в часть II нашей статьи. «Граница между частями» выбрана таким образом, чтобы в части II была рассмотрена вся работа с непосредственно лепестковыми автоматами (в части I такая работа начата, но приведено только определение таких автоматов).

Конкретно, в разделе III мы *начинаем* подробное рассмотрение примера, а именно – проводим специальные предварительные преобразования полного автомата, соответствующего заданному бинарному отношению #. В последующем, в разделе V части II, рассмотрение этого примера *продолжается*: приведены непосредственные (неэквивалентные) преобразования, дающие в итоге (детерминированный) автомат, также имеющий заданную таблицу бинарного отношения # – и плюс к этому являющийся лепестковым. А до этого, в разделе IV части II, приведено «продолжение мотивации»: описана связь рассматриваемых нами лепестковых автоматов с другими темами теории регулярных языков и недетерминированных конечных автоматов.

В упомянутых выше разделах III и V нами применена общая (сквозная) нумерация подразделов – по-видимому, это также более удобно, поскольку только одно рассмотрение этого примера могло бы представить предмет отдельной статьи. Конкретно по подразделам:

- A) ещё раз сформулирована цель проделываемых построений (специальных неэквивалентных преобразований автомата) и введены некоторые дополнительные обозначения; задаётся рассматриваемое в примере бинарное отношение #;
- B) для этого бинарного отношения приведён соответствующий ему (исходный для будущих преобразований) детерминированный автомат $K^\#$ и некоторые связанные с ним автоматы;
- C) задан один конкретный порядок вершин автомата $K^\#$ и приведено первое неэквивалентное упрощение этого автомата;
- D) приведено второе неэквивалентное упрощение автомата;
- E) приведено полное исследование полученного автомата – в частности, показано, что его таблица отношения # совпадает с исходной.

Итак, в конце проведения преобразований и рассмотрения примера – т. е. в конце раздела V – получается (детерминированный) автомат, соответствующий заданному отношению #; а то, что он является лепестковым, следует из проделанного построения.

Формальное описание и доказательство корректности подобного алгоритма (того, который рассматривается на примере в разделах III и V) приведено в разделах VI и VII. А раздел VIII – заключение: в нём показаны несколько вариантов связи между рассматриваемыми в статье лепестковыми автоматами и другими темами наших недавних публикаций, а также приведены некоторые варианты дальнейшей работы в этом направлении.

⁷ Ниже мы обычно не будем специально оговаривать возможное наличие нового столбца (нового состояния канонического автомата для зеркального языка).

II. ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ОБОЗНАЧЕНИЯ

Способ задания конечных автоматов и основные связанные с ними понятия и обозначения (в частности, функции φ^{in} и φ^{out} , а также уже упомянутое выше бинарное отношение #) совпадают с применяемыми в наших предыдущих публикациях – подробное их описание приведено, например, в [24], [25].

Далее, путь задан некоторый конечный язык A (1), причём $A \not\equiv \varepsilon$. Пусть также

$$u_i = a_{i,1} a_{i,2} \dots a_{i,l_i} \quad (l_i \geq 1)$$

для каждого $i = 1, 2, \dots, n$.

Для этого языка A рассмотрим недетерминированный конечный автомат

$$\mathcal{K}(A) = (Q, \Sigma, \delta, \{s\}, \{s\}), \quad (2)$$

где

$$Q = \{s\} \cup \bigcup_{i \leq n, j < l_i} q_{i,j},$$

а функция переходов δ для каждого $i = 1, \dots, n$ определена описанным далее способом.

- Если $|u_i| = 1$, то $\delta(s, a_{i,1}) \ni s$.
- Иначе⁸:
 - $\delta(s, a_{i,1}) \ni q_{i,1}$;
 - $\delta(q_{i,l_i-1}, a_{i,l_i}) = \{s\}$;
 - для каждого j , такого что $2 \leq j \leq l_i - 1$, полагаем $\delta(q_{i,j-1}, a_{i,j}) = \{q_{i,j}\}$.

Такая запись в точности описывает автомат, приведённый на рис. 2; также очевидно, что автомат (2) определяет язык A^* . Кроме того, при замене в (2) множества выходных состояний с $\{s\}$ на всё ранее определённое множество Q , т. е. при рассмотрении

$$\hat{\mathcal{K}}(A) = (Q, \Sigma, \delta, \{s\}, Q),$$

мы получаем автомат, определяющий язык $\text{Pref}(A^*)$.

Таким образом, мы получаем (пользуясь терминологией статей [9], [15], [16], а также нескольких работ по ссылкам из них), что для заданных конечных языков $A, B \subseteq \Sigma^*$ условие $A \trianglelefteq B$ может быть проверено путём определения возможной эквивалентности автоматов $\hat{\mathcal{K}}(A)$ и $\hat{\mathcal{K}}(B)$.

(Конечно же, последнее может быть сделано проверкой возможной эквивалентности канонических автоматов, соответствующих им, – т. е.

$$\widetilde{\hat{\mathcal{K}}}(A) \quad \text{и} \quad \widetilde{\hat{\mathcal{K}}}(B).$$

При этом, конечно, не каждый канонический автомат задаёт некоторый итерационный язык. Отсюда можно увидеть некоторую связь с алгоритмом определения, равна ли значению 1 звёздная высота регулярного языка, заданного, конечно, не автоматами вида $\mathcal{K}(A)$, а соответствующими им каноническими автоматами – см. [26], [27] и др.)

Переходим к определению полного автомата – мы здесь упрощаем определения, приведённые в [24]. Сначала формально определим *возможные для рассмотрения* бинарные отношения # – для каждого возможного отношения соответствующий ему регулярный язык L считаем

⁸ Выше мы отмечали, что $A \not\equiv \varepsilon$ – поэтому $|u_i| \geq 1$.

заданным⁹. При этом, упрощая теорию из [24], сразу приведём определение на множествах состояний двух канонических конечных автоматов (для языков L и L^R) – у этих автоматов, обычно обозначаемых нами \tilde{L} и \tilde{L}^R , состояния обозначаются Q_π и Q_ρ соответственно.

Итак, для бинарного отношения $\# \subseteq Q_\pi \times Q_\rho$ должны выполняться следующие ограничения.

- Таблица бинарного отношения $\#$ не имеет пустых строк:

$$(\forall p \in Q_\pi) (\exists r \in A_\rho) (p\#r).$$

- Таблица не имеет пустых столбцов:

$$(\forall r \in Q_\rho) (\exists p \in A_\pi) (p\#r).$$

- Таблица не имеет двух одинаковых строк:

$$(\forall p_1, p_2 \in Q_\pi, p_1 \neq p_2) (\exists r \in Q_\rho) ((p_1\#r \& p_2\#r) \vee (\overline{p_1\#r} \& \overline{p_2\#r})).$$

- Таблица не имеет двух одинаковых столбцов:

$$(\forall r_1, r_2 \in Q_\rho, r_1 \neq r_2) (\exists p \in Q_\pi) ((p\#r_1 \& p\#r_2) \vee (\overline{p\#r_1} \& \overline{p\#r_2})).$$

Доказательства того, что именно такие (и только такие) бинарные отношения $\#$ возможны для произвольных регулярных языков, см. в [20], [21], [28] и др., – будем далее (в том числе в части II) называть такие отношения *корректными*.

Далее, для заданного (корректного) бинарного отношения

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

и двух заданных состояний

$$s_\pi \in Q_\pi \text{ и } s_\rho \in Q_\rho$$

рассмотрим автомат

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

(или, кратко, $K^\#$ – если состояния s_π и s_ρ подразумеваются или не имеют значения), где:

- $F_\pi = \{f_\pi \in Q_\pi \mid f_\pi\#s_\rho\}$,¹⁰
- функция переходов $\delta^\#$ определяется следующим образом:

$$\delta^\#(A, \alpha_{\frac{B}{x}}) = \begin{cases} \{B\}, & \text{если } A\#X; \\ \emptyset, & \text{в противном случае} \end{cases}$$

(допускаем возможность $A = B$).

В [24] мы показали, что построенный таким образом конечный автомат $K^{\#s_\pi s_\rho}$ (или – любой из автоматов множества $K^\#$) действительно определяет регулярный язык, бинарным отношением $\#$ которого является то, которое задано для его определения.

Автомат (множество автоматов) $K_\#$ строится таким же образом для зеркального языка.

⁹ Если это предположение воспринимать как логическую ошибку (ведь мы определяем бинарное отношение $\#$ для регулярного языка – при том, что существование самого этого языка, а также соответствующего ему автомата, мы ещё не доказали) – то полную версию определения возможных для рассмотрения бинарных отношений $\#$ можно найти в процитированной статье.

¹⁰ Выбирая различные варианты

$$s_\pi \in Q_\pi \text{ и } s_\rho \in Q_\rho,$$

мы получаем *множество* языков, каждый из которых соответствует заданному отношению $\#$.

III. ПОДРОБНЫЙ ПРИМЕР ПОЛУЧЕНИЯ ЛЕПЕСТКОВОГО АВТОМАТА – НАЧАЛО РАССМОТРЕНИЯ

В этом разделе и в разделе V подробно рассмотрен пример возможного алгоритма построения подходящего лепесткового автомата – на основе заданных таблицы бинарного отношения $\#$ и линейного порядка на состояниях. Специально отметим, что рассмотрение примера приведено до самого описания алгоритма – поскольку, по-видимому, в данном случае такое изложение более удачное.

Рассмотрение примера разбито на два раздела: настоящий раздел III, а также помещённый уже в часть II нашей статьи раздел V. Здесь, в разделе III, мы начинаем рассматривать пример, а именно – проводим описанные далее предварительные преобразования специального полного автомата, соответствующего заданному бинарному отношению $\#$. А в разделе V рассмотрение *продолжается*: приведены непосредственные (неэквивалентные) преобразования, приводящие в итоге к (детерминированному) автомату, также имеющему заданную таблицу бинарного отношения $\#$ – и плюс к этому являющемуся лепестковым.

A. Рассматриваемый пример: введение, мотивация и дополнительные обозначения

Ещё раз отметим *цель* проделываемых в настоящей статье построений – неэквивалентных преобразований автомата (рассматриваемых как в этом примере, так и в последующем формальном описании общего случая): построить *лепестковый* автомат, обладающий:

- либо заданной таблицей отношения $\#$;
- либо такой же таблицей – но с добавленным столбцом (новым состоянием автомата L^R).

Из предыдущего можно заключить, что как сам исходный автомат, так и задаваемый им регулярный язык¹¹, для материала рассматриваемого примера не нужны – поэтому в следующей таб. 1 мы приводим только само бинарное отношение $\#$; вот эта таблица:

Таб. 1. Заданная таблица бинарного отношения $\#$.

#	X	Y	Z	U	#	X	Y	Z	U
A	—	#	#	—	A	0	1	1	0
B	#	—	#	—	B	1	0	1	0
C	#	#	#	#	C	1	1	1	1
D	#	#	#	—	D	1	1	1	0

При этом, как обычно, мы приводим одну и ту же таблицу (одно и то же отношение $\#$) двумя способами¹². Отметим, что именно это отношение $\#$ (а также соответствующие ему языки, автоматы, функции разметки

¹¹ Обычно обозначаемые в рассматриваемой нами теории K и L соответственно.

¹² Снова можно было бы увидеть логическую «ошибку»: мы рассматриваем таблицу бинарного отношения $\#$ для языка (или автомата) – в то время как сам язык мы не указали. Однако выше мы уже фактически ответили на такую возможную претензию: таковым автоматом можно считать строимый нами $K^\#$.

состояний и т. д.) многократно рассматривалось в наших предыдущих публикациях – см. [10], [24] и мн. др.¹³

Мы повторим (с большими изменениями и добавлениями) некоторую часть материала статьи [24], необходимую для наших текущих построений – это, в основном, подраздел В ниже: в дальнейших подразделах мы будем сравнивать те построения с построениями лепесткового автомата. И ещё раз повторим, что *всюду* ниже мы не отмечаем входы и выходы: согласно нашей же терминологии ([13], [14]), все рассматриваемые далее автоматы – сильно связанные, поэтому *для целей настоящей статьи* отмечать входы и выходы необязательно.

Отметим, что в наших построениях обычно образуется новый столбец – но не новая строка. Некоторые подробности (неформальное объяснение этого факта) будут приведены в части II.

В. Исходный автомат $K^\#$ и автоматы, связанные с ним

Для дальнейших построений необходим автомат $K^\#$ – и мы ещё раз рассмотрим [24, Таб. 4]; это – автомат, приведённый в таб. 2 настоящей статьи.

(Таб. 2 приведена в конце текста статьи.)

«Ненужные» для дальнейших построений входы и выходы мы не приводим¹⁴. И отметим специально ещё раз, что, в отличие от [24], «исходного» автомата K мы не рассматриваем; однако на основе той же статьи [24] можно утверждать, что *какой-то* автомат¹⁵ с такой таблицей $\#$ существует – и поэтому обозначение $K^\#$ корректно.

Также отметим, что приводимые в таблицах настоящей статьи обозначения букв (пометки столбцов таблиц) отличаются от применяемых как в [24] (в том числе во всех таблицах той статьи), так и в определениях статьи настоящей. Это сделано для удобочитаемости – причём соответствие «старых» и «новых» обозначений очевидно: например, соответствуют одна другой буквы, обозначенные

$$a_x^A \text{ и } X - A.$$

Далее мы продолжаем работать с автоматом $K^\#$ – теперь рассмотрим $(K^\#)^R$, см. таб. 3 ниже.

(Таб. 3 приведена в конце текста статьи.)

Начиная с этой таблицы в некоторых клетках (в том числе – в обозначениях строк-состояний) могут располагаться *несколько* букв; обычно это делается при описании

¹³ Среди прочего, в этих публикациях был описан *процесс построения* некоторого заданного регулярного языка (некоторого определяющего его конечного автомата) на основе заданной таблицы $\#$. В этом процессе применены (неэквивалентные) преобразования автомата, сохраняющие заданное бинарное отношение $\#$ и аналогичные преобразования, применяемые в настоящей статье. И можно даже сказать, что такой процесс был ранее рассмотрен не для заданного бинарного отношения $\#$, а для заданного (произвольным образом) регулярного языка.

¹⁴ Ещё один «философский вопрос» – действительно ли они нужны... Ведь в будущем – для того, чтобы *формально* удовлетворить определению лепесткового автомата – одно из состояний нужно будет объявить входом и выходом. Это – «новый столбец» таблицы $\#$, т. е. новое состояние строимого автомата $K_\#$; это состояние, возможно, совпадёт с одним из уже существующих. Немного подробнее об этом см. в части II.

¹⁵ Немного точнее – «какой-то регулярный язык». Однако нас в данном случае интересуют именно конечные автоматы.

процесса детерминизации автомата. Некоторые комментарии об этом мы многократно приводили в предыдущих публикациях, кратко повторим и здесь; такие обозначения практически всегда можно рассматривать *одновременно*:

- и как несколько состояний недетерминированного автомата;
- и как т. н. агрегатное состояние строимого эквивалентного детерминированного автомата.

Процесс детерминизации (канонизации) последнего автомата приведён в таб. 4:

(Таб. 4 приведена в конце текста статьи.)

При этом, как иногда в наших предыдущих публикациях, справа (т. е. в столбце, который «вне таблицы») показаны будущие состояния автомата

$$K_\# = \widetilde{(K^\#)^R}. \quad (3)$$

Однако: ранее в [24] мы применяли обозначение, аналогичное равенству (3), и считали два приведённых в этом равенстве автомата одним и тем же объектом; но удобнее (и такие обозначения применяются в настоящей статье, в частности, во всех таблицах) применять *немного* иные договорённости:

- $\widetilde{(K^\#)^R}$ – это автомат «в процессе детерминизации», у него в обозначениях строк и клетках – множества состояний автомата $K^\#$ (или, что то же самое, т. н. агрегатные состояния будущего детерминированного автомата);
- а $K_\#$ – автомат, полученный «после детерминизации и канонизации»¹⁶.

В связи с этим равенством (3)¹⁷ мы больше пользоваться не будем.

Результирующий канонический автомат $K_\#$ приведён в таб. 5 ниже; этот же автомат – [24, таб. 7].

(Таб. 5 приведена в конце текста статьи.)

С. Выбор порядка вершин автомата $K^\#$ и его первое неэквивалентное упрощение

А далее мы приводим автоматы (также связанные с этим же примером), которые в [24] не рассматривались. Будем преобразовывать исходный автомат $K^\#$ – и теперь преобразования будут *неэквивалентными*, аналогично [24].

Во-первых, выберем «центральную» (главную) вершину предполагаемого будущего лепесткового автомата¹⁸; например пусть ею будет такая же, как стартовая вершина в примере из [24], т. е. А. В принципе это условие не является обязательным, т. е. в качестве «центральной» можно выбирать любую вершину.

¹⁶ Название «канонизация» всё-таки не является стандартным термином, но смысл, конечно, понятен: это – минимизация уже построенного эквивалентного *детерминированного* автомата.

¹⁷ Его можно было бы назвать определением автомата $K_\#$.

¹⁸ В связи с названием «лепестковый» удобно употреблять терминологию теории графов: будем чаще называть чаще «вершина» (графа переходов лепесткового автомата), а не «состояние».

Во-вторых, для остальных вершин выбираем какой-нибудь линейный порядок^{19,20}. Например, выберем такой порядок:

$$B > C > D,$$

т. е., включая в порядок выбранную ранее «центральную» (главную) вершину, получаем

$$A > B > C > D.$$

В-третьих, для получения нового исследуемого автомата мы удаляем из последней таблицы все дуги, являющиеся переходами из «бóльших» состояний в «меньшие»; заодно удаляем и все петли. Однако при этом никакие переходы, относящиеся к «центральной» вершине (в том числе её петли), мы не удаляем. В таблицах автоматов мы для наглядности сначала отмечаем все удаляемые дуги красным цветом – см. таб. 6...

(Таб. 6 приведена в конце текста статьи.)

... а потом приводим автомат без таких дуг – таб. 7.

(Таб. 7 приведена в конце текста статьи.)

Получившийся автомат обозначим K' – и заранее отметим, что он *не является* «самым интересным» автоматом, который мы можем получить на основе $K^\#$.

Теперь построим автомат $(K')^R$ – см. таб. 8;

(Таб. 8 приведена в конце текста статьи.)

в ней мы красным цветом обозначили множества состояний, которые, как мы *уже* знаем (на основе проделанного в предыдущих построениях), *не являются* элементами автомата $(K^\#)^R$.

«Красными» остаются и состояния автомата $(K')^R$. Сформулируем то же самое строго: детерминируя (и каноизируя) последний автомат, мы в процессе этих действий получаем состояния, для которых нет соответствующих (т. е. состояний, обозначенных таким же образом), возникающих при детерминизации и канонизации исходного автомата $K^\#$ ^{21,22}. Итак, в процессе канонизации получается автомат, приведённый в таб. 9.

(Таб. 9 приведена в конце текста статьи.)

¹⁹ Отметим, что если лепестковый автомат задан – чего в нашем случае не наблюдается – то для его состояний частичный порядок желательно вводить естественным образом: можно считать, что он просто определяется всеми дугами этого лепесткового автомата. Этот частичный порядок *даст варианты полного линейного порядка* – для автомата, соответствующего заданной таблице.

²⁰ Также отметим, что в следующих публикациях мы предполагаем рассмотреть и другие возможные ограничения на выбираемый согласно бинарному отношению $\#$ полный линейный порядок.

²¹ Несколько точнее, конечно, не «получаем» (состояния), а «можем получить». Кроме того важно, что при этом (т. е. в процессе детерминизации и канонизации) получается не только «центральное» состояние A , но и некоторые другие.

²² При этом может возникнуть вопрос о корректности сравнения таких действий – ведь мы не доказывали, что у *нового* автомата K' будут те же самые состояния его канонического автомата. Однако для проведения канонизации нового автомата нам не обязательно проверять это условие – например, просто потому, что мы заранее оговорились про *неэквивалентность* проводимых нами преобразований автоматов. Действительно, автомат K (он даже не рассматривается в настоящей статье – можно, например, просто считать, что он совпадает с автоматом $K^\#$), конечно же, неэквивалентен автомату K' .

Аналогично таб. 4, в ней используется «внешний» столбец – для обозначения *будущих* состояний детерминированного автомата. Однако, в отличие от предыдущего случая, мы некоторые такие будущие состояния обозначаем красным цветом²³; понятно, что это – состояния, которых не было при проведении преобразований исходного автомата $K^\#$. (А такие состояния, которые в нём были, понятны уже на основе таб. 1.)

При этом одно из новых «красных» состояний мы специально обозначили необычным образом – русской буквой Ъ. Об этом подробно далее, а пока очень кратко скажем так: при будущем *удалении* некоторых элементов из рассматриваемого автомата – это состояние Ъ должно остаться, более того, оно и станет «главным» состоянием построенного лепесткового автомата.

А после обычного переобозначения состояний (а перед этим – рассмотрения «агрегатных» состояний в качестве обычных состояний детерминированного автомата) получается автомат, который мы обозначим $K'_\#$ ²⁴ – см. таб. 10. Этот автомат по построению является детерминированным – и на основе анализа последней таблицы ([10] и мн. др.) можно показать, что он *для своего языка* является каноническим.

(Таб. 10 приведена в конце текста статьи.)

Однако цель, сформулированную в начале настоящего раздела, пока ещё вряд ли можно считать достигнутой: таблица отношения $\#$ полученного автомата действительно «включает исходную таблицу как собственное подмножество»²⁵ – однако в ней добавляется много новых состояний (а мы хотели, чтобы добавилось не более одного). Поэтому мы продолжим описание начатых преобразований в разделе V.

Итак, как мы уже отмечали во введении, продолжение рассмотрения примера (а именно, всё связанное с непосредственным получением лепесткового автомата), а также формальное описание самого алгоритма построения такого объекта на основе полного автомата и доказательство корректности этого алгоритма, будут приведены в завершающей части II настоящей статьи.

Список литературы

- [1] Singh Sh., Krishna K. *On syntactic complexity of circular semi-flower automata* // Implementation and Application of Automata, CIAA 2018, Lecture Notes in Computer Science, 10977, Springer International Publishing Ag. – 2018. – P. 312–323.
- [2] Баумгертнер С., Мельников Б. *Обобщенные недетерминированные конечные автоматы* // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2013. – № 2 (26). – С. 64–74.
- [3] Melnikov B., Melnikova A. *Pseudo-automata for generalized regular expressions* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 1. – P. 1–8.
- [4] Мельников Б. *Подклассы класса контекстно-свободных языков (монография)*. – М., Изд-во Московского университета. – 1995. – 174 с. – ISBN 5-211-03448-1.

²³ Таким образом, мы используем красный цвет для двух разных целей – но это, по-видимому, не должно вызывать недоразумений.

²⁴ Это обозначение *немного* противоречит нашим предыдущим договорённостям – поскольку этот автомат не является полным (complete) ни для какого регулярного языка; однако, по-видимому, такое противоречие большой проблемы не представляет.

²⁵ Понятно, что взятую в кавычки фразу можно переформулировать строго – однако мы не будем этого делать.

- [5] 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. 495–505.
- [6] Алексеева А., Мельников Б. *Итерации конечных и бесконечных языков и недетерминированные конечные автоматы* // Вестник Северного (Арктического) федерального университета. Серия: Естественные науки. – 2011. – № 3 (17). – С. 30–33.
- [7] Корабельщикова С., Мельников Б. *Максимальные префиксные коды и подклассы класса контекстно-свободных языков* // Вестник Северного (Арктического) федерального университета. Серия: Естественные науки. – 2015. – № 1. – С. 121–129.
- [8] Корабельщикова С., Мельников Б. *Итерации языков и максимальные префиксные коды* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2015. – № 2. – С. 106–120.
- [9] Мельников Б. *Полурешётки подмножеств потенциальных корней в задачах теории формальных языков. Часть I. Извлечение корня из языка* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 4. – P. 1–9.
- [10] Мельников Б. *Регулярные языки и недетерминированные конечные автоматы (монография)*. – М., Изд-во Российского государственного социального ун-та. – 2018. – 179 с. – ISBN 978-5-7139-1355-7.
- [11] Brosalina A., Melnikov B. *Commutation in global supermonoid of free monoids* // Informatica (Lithuanian Academy of Sciences Ed.). – 2000. – Vol. 11. No. 4. – P. 353–370.
- [12] Melnikov B., Melnikova A. *Some more on ω -finite automata and ω -regular languages. Part I: The main definitions and properties* // International Journal of Open Information Technologies. – 2020. – Vol. 8. No. 8. – P. 1–7.
- [13] Мельников Б. *Об одном подклассе класса регулярных языков («сильно связанные» языки): определения и соответствующие канонические автоматы* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 3. – P. 1–10.
- [14] Мельников Б. *Автоматы – полные инварианты для сильно связанных регулярных языков* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 6. – P. 1–10.
- [15] Мельников Б. *Полурешётки подмножеств потенциальных корней в задачах теории формальных языков. Часть II. Построение инверсного морфизма* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 5. – P. 1–8.
- [16] Мельников Б. *Полурешётки подмножеств потенциальных корней в задачах теории формальных языков. Часть III. Условие существования решётки* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 7. – P. 1–9.
- [17] Melnikov B., Tsyganov A. *The state minimization problem for nondeterministic finite automata: the parallel implementation of the truncated branch and bound method* // В сб.: Proceedings – 5th International Symposium on Parallel Architectures, Algorithms and Programming, PAAP-2012. – 2012. – P. 194–201.
- [18] Melnikov B., Melnikova A. *Edge-minimization of non-deterministic finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2001. – Vol. 8. No. 3. – P. 469–479.
- [19] Долгов В., Мельников Б. *Об алгоритмах автоматического построения Ватерлоо-подобных конечных автоматов на основе полных автоматов* // Эвристические алгоритмы и распределенные вычисления. – 2014. – Т. 1. № 4. – С. 24–45.
- [20] Долгов В., Мельников Б. *Построение универсального конечного автомата. II. Примеры работы алгоритмов* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2014. – № 4. – С. 78–85.
- [21] Melnikov B., Melnikova A. *Some properties of the basis finite automaton* // The Korean Journal of Computational and Applied Mathematics (Journal of Computational and Applied Mathematics). – 2002. – Vol. 9. No. 1. – P. 135–150.
- [22] Melnikov B., Melnikova A. *A new algorithm of constructing the basis finite automaton* // Informatica (Lithuanian Academy of Sciences Ed.). – 2002. – Vol. 13. No. 3. – P. 299–310.
- [23] Долгов В., Мельников Б., Мельникова А. *Циклы графа переходов базисного конечного автомата и связанные вопросы* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2016. – № 4. – С. 95–111.
- [24] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [25] Мельников Б. *Варианты конечных автоматов, соответствующих бесконечным итерационным деревьям морфизмов. Часть I* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 7. – P. 5–13.
- [26] Kirsten D. *Distance desert automata and the star height problem* // Informatique Théorique et Applications. – 2005. – Vol. 39. No. 3. – P. 455–509.
- [27] Melnikov B. *The star-height of a finite automaton and some related questions* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 7. – P. 1–5.
- [28] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta universitatis sapientiae informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.

Борис Феликсович МЕЛЬНИКОВ,
 профессор Университета МГУ – ППИ в Шэньчжэне
 (<http://szmsubit.ru/>),
 email: 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).

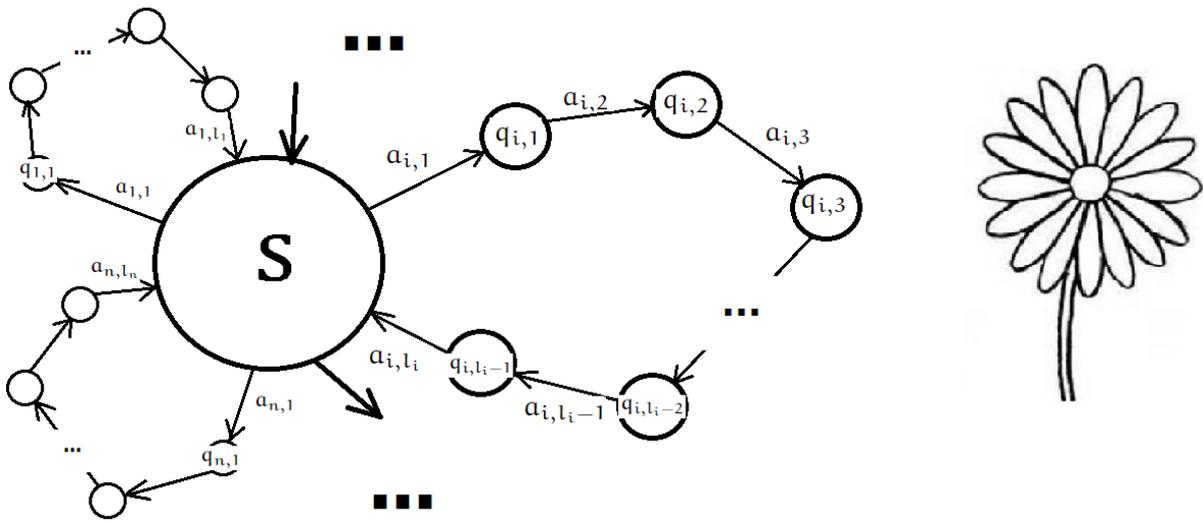


Рис. 2. Подробная общая схема графа переходов лепесткового автомата и (для аналогии) схематичный рисунок ромашки.

Таб. 2. Заданный полный автомат (complete automaton $K^\#$).

$K^\#$	X-A	X-B	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-B	U-C	U-D
A	—	—	—	—	A	B	C	D	A	B	C	D	—	—	—	—
B	A	B	C	D	—	—	—	—	A	B	C	D	—	—	—	—
C	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
D	A	B	C	D	A	B	C	D	A	B	C	D	—	—	—	—

Таб. 3. Автомат, зеркальный к полному.

$(K^\#)^R$	X-A	X-B	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-B	U-C	U-D
A	BCD	—	—	—	ACD	—	—	—	ABCD	—	—	—	C	—	—	—
B	—	BCD	—	—	—	ACD	—	—	—	ABCD	—	—	—	C	—	—
C	—	—	BCD	—	—	—	ACD	—	—	—	ABCD	—	—	—	C	—
D	—	—	—	BCD	—	—	—	ACD	—	—	—	ABCD	—	—	—	C

Таб. 4. Процесс детерминизации (канонизации) последнего автомата.

$\wedge R$ -кан.	X-A	X-B	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-B	U-C	U-D
BCD	—	BCD	BCD	BCD	—	ACD	ACD	ACD	—	ABCD	ABCD	ABCD	—	C	C	C
ACD	BCD	—	BCD	BCD	ACD	—	ACD	ACD	ABCD	—	ABCD	ABCD	C	—	C	C
ABCD	BCD	BCD	BCD	BCD	ACD	ACD	ACD	ACD	ABCD	ABCD	ABCD	ABCD	C	C	C	C
C	—	—	BCD	—	—	—	ACD	—	—	—	ABCD	—	—	—	C	—

Таб. 5. Результирующий канонический автомат $K_\#$.

$K_\#$	X-A	X-B	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-B	U-C	U-D
X	—	X	X	X	—	Y	Y	Y	—	Z	Z	Z	—	U	U	U
Y	X	—	X	X	Y	—	Y	Y	Z	—	Z	Z	U	—	U	U
Z	X	X	X	X	Y	Y	Y	Y	Z	Z	Z	Z	U	U	U	U
U	—	—	X	—	—	—	Y	—	—	—	Z	—	—	—	U	—

Таб. 6. Удаляемые из $K^\#$ дуги (согласно порядку $A > B > C > D$).

$K^\#$	X-A	X-B	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-B	U-C	U-D
A	—	—	—	—	A	B	C	D	A	B	C	D	—	—	—	—
B	A	B	C	D	—	—	—	—	A	B	C	D	—	—	—	—
C	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
D	A	B	C	D	A	B	C	D	A	B	C	D	—	—	—	—

Таб. 7. Автомат K' .

K'	X-A	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-D
A	—	—	—	A	B	C	D	A	B	C	D	—	—
B	A	C	D	—	—	—	—	A	—	C	D	—	—
C	A	—	D	A	—	—	D	A	—	—	D	A	D
D	A	—	—	A	—	—	—	A	—	—	—	—	—

Таб. 8. Автомат $(K')^R$.

$(K')^R$	X-A	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-D
A	BCD	—	—	ACD	—	—	—	ABCD	—	—	—	C	—
B	—	—	—	—	A	—	—	—	A	—	—	—	—
C	—	B	—	—	—	A	—	—	—	AB	—	—	—
D	—	—	BC	—	—	—	AC	—	—	—	ABC	—	C

Таб. 9. Автомат $\widehat{(K')^R}$ – процесс детерминизации.

$\wedge R$ -кан.	X-A	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-D	
BCD	—	B	BC	—	A	A	AC	—	A	AB	ABC	—	C	X
B	—	—	—	—	A	—	—	—	A	—	—	—	—	P
BC	—	B	—	—	A	A	—	—	A	AB	—	—	—	Q
A	BCD	—	—	ACD	—	—	—	ABCD	—	—	—	C	—	Ь
AC	BCD	B	—	ACD	—	A	—	ABCD	—	AB	—	C	—	R
AB	BCD	—	—	ACD	A	—	—	ABCD	A	—	—	C	—	S
ABC	BCD	B	—	ACD	A	A	—	ABCD	A	AB	—	C	—	T
C	—	B	—	—	—	A	—	—	—	AB	—	—	—	U
ACD	BCD	B	BC	ACD	—	A	AC	ABCD	—	AB	ABC	C	C	Y
ABCD	BCD	B	BC	ACD	A	A	AC	ABCD	A	AB	ABC	C	C	Z

Таб. 10. Автомат $K'_\#$.

$K'_\#$	X-A	X-C	X-D	Y-A	Y-B	Y-C	Y-D	Z-A	Z-B	Z-C	Z-D	U-A	U-D
X	—	P	Q	—	Ь	Ь	R	—	Ь	S	T	—	U
P	—	—	—	—	Ь	—	—	—	Ь	—	—	—	—
Q	—	P	—	—	Ь	Ь	—	—	Ь	S	—	—	—
Ь	X	—	—	Y	—	—	—	Z	—	—	—	U	—
R	X	P	—	Y	—	Ь	—	Z	—	S	—	U	—
S	X	—	—	Y	Ь	—	—	Z	Ь	—	—	U	—
T	X	P	—	Y	Ь	Ь	—	Z	Ь	S	—	U	—
U	—	P	—	—	—	Ь	—	—	—	S	—	—	—
Y	X	P	Q	Y	—	Ь	R	Z	—	S	T	U	U
Z	X	P	Q	Y	Ь	Ь	R	Z	Ь	S	T	U	U

Petal (semi-flower) finite automata: basic definitions, examples and their relation to complete automata. Part I

Boris Melnikov

Abstract—This paper discusses (non-deterministic) finite automata of a special kind. We have not found any publications in the literature in Russian that define the title for such automata, so we propose a new name for them, i.e. “petal automata”. (In the only publication found in the English-language literature, the term “semi-flower automata” is used, apparently it is not very successful.)

The study of such automata is very important for several topics discussed in our previous publications. First of all, it is a connection with algorithms for solving problems, related to various variants of the study of the equality of infinite iterations of two finite languages: in particular, to the description of algorithms for solving these problems in the form of a special type of nondeterministic finite automata (i.e., PRI and NSPRI automata). Other problems are also close to this topic, which, in principle, can also be solved by considering all subsets of the set of potential word roots of a given language: for example, it is the problem of extracting the root of some power for the given language. But a very important auxiliary purpose is to solve these problems using algorithms, less complex than exponential ones.

Two more possible topics of the applications of petal automata are as follows. It is possible to apply the obtained results in auxiliary algorithms necessary for more general algorithms of state- and edge-minimization of nondeterministic finite automata. In addition, the study of petal automata can be associated with the study of the binary relation $\#$, since variants of such tables divide the whole set of regular languages into special equivalence classes.

The possibility of using petal automata in the last topic follows from the main result of this paper, namely, from the considered description of the algorithm for constructing such a petal automaton, the table of binary relation $\#$ of which is the same as the given table (perhaps, with the only specially added right column).

In the proposed first part of this paper, the main definitions are given and the initial phase of the required algorithm for constructing a petal automaton is considered (using the example only). This phase is associated with special transformations of the complete automaton for a given table of the binary relation $\#$; these transformations, generally speaking, are nonequivalent, but at the same time preserving this binary relation.

Keywords—formal languages, iterations of languages, morphisms, binary relations, semi-flower automata, complete automata.

References

- [1] Singh Sh., Krishna K. *On syntactic complexity of circular semi-flower automata* // Implementation and Application of Automata, CIAA 2018, Lecture Notes in Computer Science, 10977, Springer International Publishing Ag. – 2018. – P. 312–323.
- [2] Baumgärtner S., Melnikov B. *Generalized nondeterministic finite automata* // News of higher educational institutions. Volga region. Physical and mathematical sciences. – 2013. – No. 2 (26). – P. 64–74 (in Russian).
- [3] Melnikov B., Melnikova A. *Pseudo-automata for generalized regular expressions* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 1. – P. 1–8.
- [4] Melnikov B. *Subclasses of the context-free languages class (monograph)*. – Moscow, Moscow State University Ed. – 1995. – 174 p. – ISBN 5-211-03448-1. (in Russian).
- [5] 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. 495–505.
- [6] Alekseeva A., Melnikov B. *Iterations of finite and infinite languages and nondeterministic finite automata* // Vector of Science of Togliatti State University. – 2011. – No. 3 (17). – P. 30–33 (in Russian).
- [7] Korabelshchikova S., Melnikov B. *Maximum prefix codes and subclasses of the context-free languages class* // Bulletin of the Northern (Arctic) Federal University. Series: Physics. Mathematics. – Серия: Natural sciences. – 2015. – No. 1. – P. 121–129 (in Russian).
- [8] Korabelshchikova S., Melnikov B. *Iterations of languages and maximum prefix codes* // Bulletin of the Voronezh State University. Series: Physics. Math. – 2015. – No. 2. – P. 106–120 (in Russian).
- [9] Melnikov B. *Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part I. Extracting the root from the language* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 4. – P. 1–9 (in Russian).
- [10] Melnikov B. *Regular languages and nondeterministic finite automata (monograph)*. – Moscow, Russian State Social University Ed. – 2018. – 179 p. – ISBN 978-5-7139-1355-7. (in Russian).
- [11] Brosalina A., Melnikov B. *Commutation in global supermonoid of free monoids* // Informatica (Lithuanian Academy of Sciences Ed.). – 2000. – Vol. 11. No. 4. – P. 353–370.
- [12] Melnikov B., Melnikova A. *Some more on ω -finite automata and ω -regular languages. Part I: The main definitions and properties* // International Journal of Open Information Technologies. – 2020. – Vol. 8. No. 8. – P. 1–7.
- [13] Melnikov B. *On the subclass of the regular languages class (“strongly connected” languages): определения и соответствующую канонические автоматы* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 3. – P. 1–10 (in Russian).
- [14] Melnikov B. *Automata – complete invariants for strongly connected regular languages* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 6. – P. 1–10 (in Russian).
- [15] Melnikov B. *Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part II. Constructing an inverse morphism* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 5. – P. 1–8 (in Russian).
- [16] Melnikov B. *Semi-lattices of the subsets of potential roots in the problems of the formal languages theory. Part III. The condition for the existence of a lattice* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 7. – P. 1–9 (in Russian).
- [17] Melnikov B., Tsyganov A. *The state minimization problem for nondeterministic finite automata: the parallel implementation of the truncated branch and bound method* // В сб.: Proceedings – 5th International Symposium on Parallel Architectures, Algorithms and Programming, PAAP–2012. – 2012. – P. 194–201.
- [18] Melnikov B., Melnikova A. *Edge-minimization of non-deterministic finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2001. – Vol. 8. No. 3. – P. 469–479.
- [19] Dolgov V., Melnikov B. *About algorithms of automatic construction of Waterloo-like finite automata based on complete automata* // Heuristic algorithms and distributed computing. – 2014. – Vol. 1. No. 4. – P. 24–45 (in Russian).
- [20] Dolgov V., Melnikov B. *Constructing universal finite automaton. II. Examples of algorithms* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2014. – No. 4. – P. 78–85 (in Russian).
- [21] Melnikov B., Melnikova A. *Some properties of the basis finite*

- automaton* // The Korean Journal of Computational and Applied Mathematics (Journal of Computational and Applied Mathematics). – 2002. – Vol. 9. No. 1. – P. 135–150.
- [22] Melnikov B., Melnikova A. *A new algorithm of constructing the basis finite automaton* // Informatica (Lithuanian Academy of Sciences Ed.). – 2002. – Vol. 13. No. 3. – P. 299–310.
- [23] Dolgov V., Melnikov B., Melnikova A. *Loops of the transition graph of a basic finite automaton and related questions* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2016. – No 4. – P. 95–111 (in Russian).
- [24] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [25] Melnikov B. *Variants of finite automata, corresponding to infinite iterative morphism trees. Part I* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 7. – P. 5–13 (in Russian).
- [26] Kirsten D. *Distance desert automata and the star height problem* // Informatique Théorique et Applications. – 2005. – Vol. 39. No. 3. – P. 455–509.
- [27] Melnikov B. *The star-height of a finite automaton and some related questions* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 7. – P. 1–5.
- [28] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta universitatis sapientiae informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.

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