Автоматы – полные инварианты для сильно связанных регулярных языков

Б. Ф. Мельников

***Аннотация*—В настоящей статье мы продолжаем рассматривать сильно связанные автоматы – как подмножество обычных недетерминированных конечных автоматов. На основе этого естественным образом определяются сильно связанные регулярные языки. Мы рассматриваем некоторые свойства понятия сильной связанности, в частности, незамкнутость этого класса относительно обычных теоретико-множественных операций.**

В названии статьи входит слово «инварианты» – таковыми (причём полными инвариантами) для регулярных языков являются не только канонический автомат, но и канонический автомат для зеркального языка, а также базисный и универсальный автоматы, являющиеся для сильно связанных языков основным предметом настоящей статьи. Одним из важных неполных инвариантов является бинарное отношение #, которое также рассматривается в настоящей статье.

Мы показываем связь понятия «сильная связанность» с базисным и универсальным конечными автоматами. Так, у сильно связанного языка базисный автомат не обязательно является сильно связанным, а универсальный автомат – таковым обязательно является. Последний факт мы считаем самым важным результатом, относящимся к сильно связанным языкам: он позволяет эквивалентно переформулировать определение сильно связанного регулярного языка – таковым можно считать язык, для которого сильно связанным является его универсальный автомат. Рассмотрение базисных и универсальных автоматов для сильно связанных языков мы иллюстрируем несколькими примерами, рассмотрение которых было начато в предыдущей статье про сильно связанные языки.

***Ключевые слова*—регулярные языки, недетерминированные конечные автоматы, подклассы классов языков, базисный автомат, универсальный автомат.**

I. ВВЕДЕНИЕ

Настоящую статью можно рассматривать как продолжение [1]. В той статье мы определили сильно связанные недетерминированные конечные автоматы и сильно связанные регулярные языки, показали на примерах, что это понятие (сильной связанности) не сопряжено с обычными алгоритмами пробразования недетерминированных конечных автоматов: например, мы показали, что для детерминированных конечных автоматов, эквивалентных либо заданным автоматам, либо их зеркальным образам (причём в обоих случаях – для автоматов, являющихся сильно связанными), возможны все 4 варианта выполнения / невыполнения условия сильной связанности.

В настоящей статье мы рассматриваем некоторые другие свойства понятия сильной связанности, в частности, незамкнутость этого класса относительно обычных теоретико-множественных операций. Однако, как мы уже отмечали в [1], слово «класс» использовать, конечно, можно: ведь это слово всегда употребляется, например, применительно к множеству контекстно-свободных языков – которое, как известно, относительно операций пересечения и дополнения незамкнуто.

В названии статьи входит слово «инварианты» – таковыми (причём *полными* инвариантами) для регулярных языков являются не только канонический автомат, но и канонический автомат для зеркального языка, а также базисный ($\mathcal{BA}$) и универсальный ($\mathcal{COM}$) автоматы, являющиеся для сильно связанных языков основным предметом настоящей статьи. А одним из важных *неполных* инвариантов является бинарное отношение #, которое также рассматривается в настоящей статье.

Мы рассматриваем, какое отношение понятие сильной связанности имеет к базисным и универсальным автоматам. Так, у сильно связанного языка базисный автомат вовсе не обязательно является сильно связанным, а универсальный автомат – таковым обязательно является. Последний факт мы считаем *самым важным* результатом, относящимся к сильно связанным языкам – он позволяет *эквивалентно переформулировать* определение сильно связанного регулярного языка: таковым можно считать язык, для которого сильно связанным является его универсальный автомат.

Кроме того, как это было отмечено ранее в [1] при изложении «мотивации», мы в следующих публикациях собираемся рассмотреть важный подкласс рассматриваемого здесь класса – а именно, итеративные конечные автоматы и соответствующие им итеративные языки – см., например, [2], [3], [4], [5], [6] [1]. Поэтому в настоящей статье мы продолжаем рассматривать более общий подкласс – более общий по сравнению с итеративными языками.

Итак, основной предмет статьи полностью описывается её названием; немного иными словами, основной предмет статьи – это базисные и универсальные автоматы для сильно связанных регулярных языков.

Структура статьи такова. В разделе II мы формулируем простейшие свойства сильно связанных языков – в первую очередь незамкнутость класса таких языков относительно обычных теоретико-множественных операций.

Раздел III фактически продолжает раздел «Предварительные сведения» статьи [1]. В нём мы рассматриваем краткую информацию о базисном автомате для зданного

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

Статья получена 26 марта 2021 г.

[1] В качестве альтернативного названия (вместо «итеративные») можно было бы использовать кальку английского термина “semi-flower automata”, он используется в указанной нами серии статей. Но, по-видимому, такое название («полуцветочные автоматы»?) было бы неудачно для русской терминологии.

регулярного языка. При этом заранее отметим, что четыре следующих раздела:

- раздел III статьи [1];
- разделы III, VI и VII настоящей статьи –

вместе можно рассматривать как *краткое введение в теорию конечных автоматов, изложенную «с точки зрения функций разметки и базисных автоматов»*. На русском языке в полном объёме автор этот материал ещё не публиковал: например, изданную в 2018 г. монографию [7] можно с этой точки зрения считать «уже немного устаревшей».

По-видимому, на основе результатов статьи [1] можно считать очевидным, что базисные автоматы сильно связанных языков не обязаны быть сильно связанными. Однако, несмотря на такую «очевидность», мы в разделе IV рассмотрим несколько конкретных примеров, а в разделе V приведём несколько утверждений, описывающих свойства базисных автоматов для сильно связанных языков. Заранее отметим, что выполнимость этих утверждений *не* следует из рассмотренных примеров, и поэтому действительно требует специальных доказательств.

Далее мы переходим к универсальным автоматам – для заданного регулярного языка L такой автомат обозначаем $\mathcal{COM}(L)$. Раздел VI, как мы уже отмечали, – продолжение предварительных све́дений, а именно – краткая информация об универсальном автомате, формальное его определение. А раздел VII – окончание этих предварительных све́дений, «краткий обзор» всех применяемых нами обозначений, связанных с конечными автоматами («резюме»).

После этого мы продолжаем рассмотрение универсальных автоматов, причём теперь – применительно к конкретным сильно связанным языкам. В разделе VIII мы приводим примеры гридов и универсальных автоматов для тех же языков, которые ранее рассматривались в разделе IV. А в разделе IX мы рассматриваем уже упомянутое выше основное свойство класса сильно связанных языков, а именно доказываем, что универсальный автомат $\mathcal{COM}(L)$ для сильно связанного языка L также является сильно связанным.

В заключении (раздел X) мы кратко перечисляем результаты статьи и формулируем связанные с ними возможные направления для дальнейшей работы.

II. ПРОСТЕЙШИЕ СВОЙСТВА КЛАССА СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ

В этом разделе мы формулируем основные свойства сильно связанных языков – в первую очередь незамкнутость класса таких языков относительно теоретико-множественных операций.

Начнём с самых простых свойств. Обычно в теории формальных языков для некоторого формализма рассматриваются понятия бесполезности и недостижимости – для объектов, входящих в этот формализм. Эти понятия, вообще говоря, можно рассматривать и для состояний некоторого сильно связанного автомата – но, конечно, вследствие непустоты рассматриваемого языка и сильной связанности автомата среди состояний этого автомата бесполезных и недостижимых быть не может.

Также обычно в теории формальных языков рассматриваются соотношения между «произвольностью» – од-

Рис. 1

нозначностью – детерминированностью (рис. 1). Например, такие картинки в студенческих курсах – либо по теории формальных языков, либо по теории трансляции, либо по теории вычислимости и теории сложности – рассматриваются как для классов языков (регулярных и контекстно-свободных), так и для определяющих языки этих классов формализмов (конечных и магазинных автоматов); понятно, что в одном из получающихся при этом 4 случаев включения – несобственные, однако рисунок всё равно можно считать правильным.

Перейдём к связи рисунка с нашей тематикой. На самом деле, в [1] и ниже в настоящей статье мы уже рассматривали (либо будем рассматривать) похожие задачи: могут ли существовать такие сильно связанные языки, для которых соответствующие 4 автомата (2 канонических, базисный и универсальный) принимают некоторый набор значений, где каждое из этих значений таково: сильно связанный / не сильно связанный. Однако для «задач на картинке» интересны и другие постановки; например, *одна из многочисленных* задач, относящихся к рис. 1, такова: существует ли сильно связанный конечный автомат, являющийся однозначным, но не являющийся детерминированным? Мы уже фактически знаем, что такой автомат существует (например, таковым является зеркальный автомат к приведённому в [1, таб. 12]) – но, повторим, на основе рис. 1 можно сформулировать целую группу похожих задач, причём (как обычно в подобных случаях) и для автоматов, и для языков.

Теперь перейдём непосредственно к теоретико-множественным операциям – и к незамкнутости класса сильно связанных языков [2]. Во-первых – рассмотрим операцию *объединения*, здесь всё очевидно: язык, определяемый регулярным выражением $a + b$, не является сильно связанным, поэтому для него не существует сильно связанного автомата.

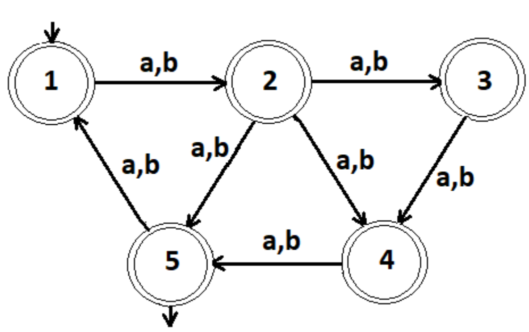

Рис. 2

[2] Заранее отметим, что некоторые из примеров ниже можно было бы не приводить, существование требуемых языков можно было бы вывести из рассмотренных до того операций – однако, конечно, конструктивные доказательства более интересны.

Далее рассмотрим автомат, приведённый на рис. 2. Понятно, что он – сильно связанный, и поэтому может быть использован для задач про незамкнутость рассматриваемого нами класса. Мы используем этот автомат для доказательства незамкнутости относительно операции *дополнения*.

Язык этого автомата может быть описан регулярным выражением

$$(a+b)\cdot(a+b)\cdot(a+b)^*,$$

поэтому его дополнение (алфавит $\{a,b\}$) описывается регулярным выражением

$$\varepsilon+a+b$$

т. е. тоже представляет собой язык, не являющийся сильно связанным.

Перейдём к *исключающему «или»* (XOR, $\oplus$). Немного изменим рис. 2 и рассмотрим такой автомат:

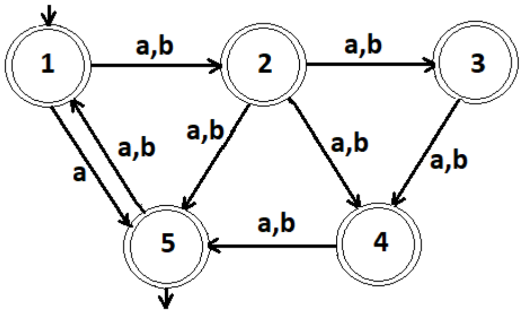

Рис. 3

Он сильно связанный.

Рассмотрим также автомат, получающийся из последнего заменой дуги

$$1 \xrightarrow{a} 5$$

на дугу

$$1 \xrightarrow{b} 5$$

(рисунок приводить не будем). Понятно, что операция исключающего «или», применённая к языкам двух последних автоматов даёт язык, описываемый регулярным выражением $a+b$, который не является сильно связанным.

Последней рассмотрим операцию *пересечения*. Для неё сначала рассмотрим такой сильно связанный автомат:

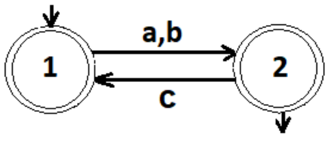

Рис. 4

Далее рассмотрим также автомат, получающийся из последнего заменой дуги

$$2 \xrightarrow{c} 1$$

на дугу

$$2 \xrightarrow{d} 1$$

(рисунок снова приводить не будем). Пересечение этих двух языков снова даёт язык, описываемый регулярным выражением $a+b$, который не является сильно связанным.

III. ПРЕДВАРИТЕЛЬНЫЕ СВЕ́ДЕНИЯ – 1: КРАТКО О БАЗИСНОМ АВТОМАТЕ

Здесь мы фактически продолжаем раздел «Предварительные сведения» из статьи [1] и рассмотрим *базисные автоматы*.

Неформально такой автомат для заданного регулярного языка L – это «декартово произведение» двух определённых в [1] канонических автоматов[3] для этого языка, т. е.

$$\widetilde{L} = (Q_\pi, \Sigma, \delta_\pi, \{s_\pi\}, F_\pi)$$

и

$$\widetilde{L^R} = (Q_\rho, \Sigma, \delta_\rho, \{s_\rho\}, F_\rho).$$

Сто́ит ещё отметить, что почти всегда в наших формулах мы обозначаем

$$s_\pi = A \quad \text{и} \quad s_\rho = X\,.$$

А *формально* мы базисный автомат определяем так[4]:

$$\mathcal{BA}(L) = \big(Q_{\mathcal{BA}}, \Sigma, \delta_{\mathcal{BA}}, S_{\mathcal{BA}}, F_{\mathcal{BA}}\big),$$

где элементы пятёрки $\mathcal{BA}(L)$ таковы.

- Множество состояний $Q_{\mathcal{BA}}$ совпадает с множеством пар отношения #. При этом каждая такая пара, принадлежащая этому бинарному отношению, рассматривается *как состояние определяемого автомата*.
- Функция $\delta_{\mathcal{BA}}$ такова: дуга
$$\begin{smallmatrix}A\\X\end{smallmatrix} \xrightarrow[\delta_{\mathcal{BA}}]{a} \begin{smallmatrix}B\\Y\end{smallmatrix}$$
существует тогда и только тогда, когда одновременно
$$A \xrightarrow[\delta_\pi]{a} B \quad \text{и} \quad Y \xrightarrow[\delta_\rho]{a} X.$$
- Входы таковы:
$$\begin{smallmatrix}A\\X\end{smallmatrix} \in S_{\mathcal{BA}} \quad \text{тогда и только тогда, когда} \quad A \in Q_\pi\,.$$
- Аналогично, выходы таковы:
$$\begin{smallmatrix}A\\X\end{smallmatrix} \in F_{\mathcal{BA}} \quad \text{тогда и только тогда, когда} \quad X \in Q_\rho\,.$$

При этом можно только на основе определения доказать следующие два факта. Во-первых, каждый выход первого автомата, пусть это $f_\pi \in F_\pi$, находится в отношении # с единственным входом второго, т. е.

$$(\forall f_\pi \in F_\pi)\,(f_\pi \# s_\rho)\,;$$

и наоборот,

$$(\forall f_\rho \in F_\rho)\,(s_\pi \# f_\rho)\,.$$

[3] Здесь очень важно, что канонические автоматы мы рассматриваем *без* их единственного возможного бесполезного состояния («дохлого состояния», “dead state”) – в отличие от большинства других подходов.

Однако практически всюду в наших предыдущих публикациях можно было *бы* переформулировать все результаты (определения, утверждения, теоремы и т. п.) так, чтобы такие «дохлые состояния» всё-таки рассматривать; конечно, изложение теории немного усложнилось бы – но, повторим, *принципиальная возможность была*. Например, для функций разметки φ^{in} и φ^{out}: можно обозначить (возможные) «дохлые состояния» автоматов $\widetilde{L}$ и $\widetilde{L^R}$ через M и N (в одной из наших публикаций так и было сделано); при этом *согласно таким же определениям* ни M, ни N не войдут ни в одно из значений функций φ^{in} и φ^{out} (где каждое значение этих функций рассматривается как множество).

А здесь, при рассмотрении сильной связанности, такое допущение «дохлых состояний» канонических автоматов невозможно.

[4] Отметим, что нижние индексы правильнее обозначать не $\mathcal{BA}$, а $\mathcal{BA}(L)$; аналогично ниже, для автомата $\mathcal{COM}(L)$. Однако, по-видимому, приведённые здесь обозначения не вызовут проблем – именно потому, что мы считаем рассматриваемый язык L заданным.

Во-вторых, очевидное достаточное условие существования пустого слова в рассматриваемом языке L

$$s_\pi \# s_\rho \;\Rightarrow\; \varepsilon \in L$$

является необходимым и достаточным (что не столь очевидно), т. е.

$$s_\pi \# s_\rho \iff \varepsilon \in L\,.$$

Подробности, связанные с определениями компонентов базисного автомата, и доказательства приведённых утверждений см., например, в [8], [9].

IV. БАЗИСНЫЕ АВТОМАТЫ ДЛЯ СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ: ПРИМЕРЫ

По-видимому, на основе результатов [1] можно считать очевидным, что базисные автоматы для сильно связанных языков не обязаны быть сильно связанными. Однако, несмотря на такую «очевидность», мы в этом разделе рассмотрим конкретные примеры, а в следующем – несколько утверждений, связанных со свойством сильной связанности у базисных автоматов.

Сначала продолжим рассмотрение примера автомата, заданного с помощью [1, рис. 3] – в той статье мы уже рассмотрели автоматы $\widetilde{L}$ и $\widetilde{L^R}$ для соответствующего языка. Повторим исходный рисунок:

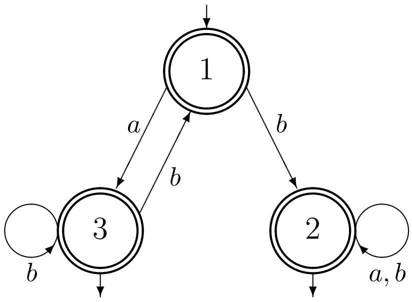

Рис. 5

Назовём язык этого автомата *языком Ъ*.

Напомним, что таблицу бинарного отношения # языка Ъ мы тоже рассмотрели в [1, таб. 1]. Поэтому, согласно определениям, рассмотренным в настоящей статье, мы для эквивалентного автомата $\mathcal{BA}(L)$ получаем [5] следующую таблицу $1\,,1'$:

Таб. 1

	$\mathcal{BA}(L)$	a	b
→	$A\#Y$	$B\#Z$, $B\#X$	$C\#U$
→	$A\#Z$	–	$C\#Y$, $C\#Z$, $C\#X$
←	$B\#X$	–	–
	$B\#Z$	–	$D\#Y$, $D\#Z$, $D\#X$
	$D\#Y$	$B\#Z$, $B\#X$	$C\#U$
←	$C\#X$	–	–
	$C\#Y$	$C\#Z$, $C\#X$	$C\#U$
	$C\#Z$	–	$C\#Y$, $C\#Z$, $C\#X$

[5] Для удобства чтения мы обозначаем в таблицах состояние $\frac{A}{X}$ либо записью $A\#X$, либо даже записью AX (и т. п.). Подробнее см. далее.

Кроме того, мы в таблицах, как и ранее, не применяем обозначения множеств, т. е., например, вместо $\left\{\frac{A}{X}, \frac{A}{Y}\right\}$ пишем либо $A\#X$, $A\#Y$, либо AX, AY.

Таб. $1'$

	$\mathcal{BA}(L)$	a	b
	$C\#U$	$C\#Y$, $C\#U$	–
←	$D\#X$	–	–
	$D\#Z$	–	$C\#Y$, $C\#Z$, $C\#X$

Теперь продолжим рассматривать автомат, приведённый в [1, рис. 13]; повторим его, это следующий рис. 6:

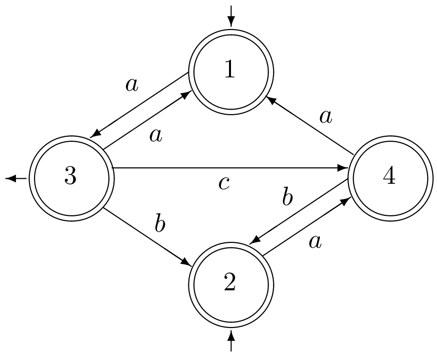

Рис. 6

Назовём язык этого автомата *языком Ы*. А в формулах (в настоящем разделе до окончания рассмотрения этого языка) будем называть его L.

Последний автомат – сильно связанный (а, следовательно, сильно связанным является и его язык); при этом, как мы уже знаем, таковым является и автомат $\widetilde{L^R}$, но не является $\widetilde{L}$.

Для построения автомата $\mathcal{BA}(L)$ получим $\widetilde{L^R}$ иным способом (отличным от использованного в [1]) – а именно, на основе автомата $(\widetilde{L})^R$. Последний автомат описывается следующей таблицей [6]:

Таб. 2

	delta	a	b	c
→	B	AC		
←	A			
	C	BE		
	D		BE	
	E	D		B

А процесс детерминизации / канонизации последнего автомата описывается следующей таблицей 3:

[6] Ещё раз обратим внимание на отличие в записях вида PR, где P и R – состояния некоторых автоматов:

- для обычных недетерминированных автоматов – это множества из 2 элементов, представляющих собой состояний одного и того же автомата; поэтому в таких таблицах возможны также тройки состояний, четвёрки и т. д.;
- для записи процесса детерминизации – это сначала множества, а потом (в тех же самых таблицах) – т. н. агрегатные состояния;
- для базисных автоматов – это пары состояний двух разных канонических автоматов, или, что обычно в наших примерах то же самое, состояния базисного автомата.

Впрочем, в каждом из этих 3 (или даже 4) случаев смысл может быть понят из контекста, а такие «похожие» варианты записи мы используем для удобства восприятия каждой таблицы в отдельности.

Таб. 3

	delta	a	b	c	
→	B	AC			X
←	AC	BE			Y
	BE	ACD		B	Z
←	ACD	BE	BE		U

Конечно же, мы при этом получаем уже известный нам автомат (совпадающий с приведённым в [1, таб. 12]) – но при таком способе мы заодно получаем для рассматриваемого языка необходимое нам бинарное отношение #:

Таб. 4

	X	Y	Z	U
A		#		#
B	#		#	
C		#		#
D				#
E			#	

На основе этого бинарного отношения, а также автомата $\widetilde{L}$ (он рассматривался в [1])

Таб. 5

	delta	a	b	c
→	A	B		
←	B	C	D	E
	C	B		
	D	E		
	E	C	D	

следующего автомата $(\widetilde{L^R})^R$

Таб. 6

	delta	a	b	c
←	X			Z
→	Y	X		
	Z	YU	U	
→	U	Z		

и приведённых в прошлом разделе определений, получаем базисный автомат $\mathcal{BA}(L)$ – см. таб. 7. Отметим, кстати, что в нём имеются два полностью идентичных состояния – $\frac{B}{Z}$ и $\frac{E}{Z}$, и это не противоречит определению базисного автомата[7].

Таб. 7

	delta	a	b	c
→	AY	BX		
→	AU	BZ		
←	BX			EZ
	BZ	CY,CU	DU	
	EZ	CY,CU	DU	
	CY	BX		
	CU	BZ		
	DU	EZ		

Приведём также последний автомат в виде графа:

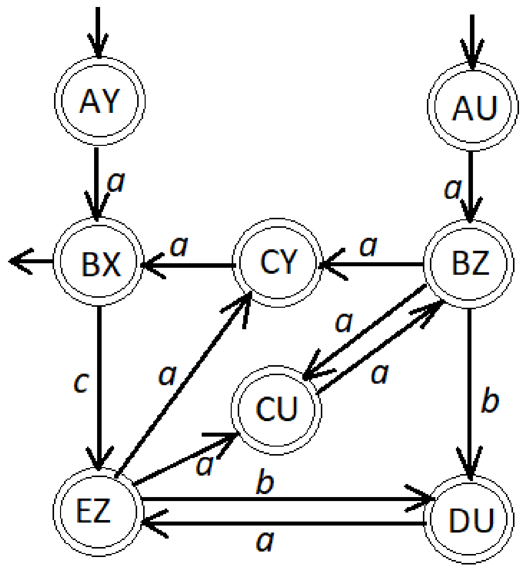

Рис. 7

Как мы видим, базисный автомат для языка Ы не является сильно связанным.

В качестве последнего примера рассмотрим *язык Ь*, задаваемый с помощью автомата, полученного удалением единственной дуги c у автомата, приведённого выше на рис. 6; этот язык мы тоже начинали рассматривать в [1], там это был рис. 12. Можно несложно показать, что для получаемого языка «всё» может быть получено таким же образом – удалением дуги c и всего связанного с ней «ненужного»[8]. А именно, переходя от языка Ы к языку Ь, мы просто удаляем:

- дугу $3 \xrightarrow{c} 4$ на рис. 6;
- далее дугу $E \xrightarrow{c} B$ в таб. 2;
- далее дугу $\{B, E\} \xrightarrow{c} \{B\}$ в таб. 3.

После этого – таблицу 4 оставляем без изменений[9] и продолжаем удалять:

- дугу $B \xrightarrow{c} E$ в таб. 5;
- далее дугу $X \xrightarrow{c} Z$ в таб. 6;

[7] Такое фактическое «совпадение» состояний базисного автомата, конечно же, даёт возможность их объединить – см., например, описание операции $\mathcal{J}$ в некоторых предыдущих публикациях автора. При этом получающийся автомат («упрощённый базисный») обладает многими нужными для практических задач свойствами базисного автомата – но при этом имеет меньшее число состояний. Важнейшим из этих свойств, по-видимому, является следующее: при реализации алгоритмов вершинной минимизации автомата можно проверять наличие всех *соответствующих* циклов (corresponding loops) не для базисного автомата, а для «упрощённого базисного». Публикаций на эту тему у автора настоящей статьи пока не было.

[8] Конечно же, подобный «алгоритм» относится только к рассматриваемому здесь примеру. Однако *при рассмотрении множества циклов* автомата $\mathcal{BA}$ (точнее, двух базисных автоматов: для исходного языка и языка, получаемого удалением состояния) применение подобных алгоритмов оказывается возможным. Автор собирается вернуться к этой тематике в одной из следующих публикаций.

[9] Это – тоже *хорошее* свойство рассматриваемого нами примера.

- далее дугу $B\#X \xrightarrow{c} E\#Z$ в таб. 7 и на рис. 7; при этом важно отметить, что ни состояние $B\#X$, ни состояние $E\#Z$ из автомата $\mathcal{BA}$ не удаляются [10] – поскольку в оба эти состояния остаются переходы из других состояний.

Получаемый базисный автомат для языка В также не является сильно связанным.

Названия этих языков – Ъ, Ы и Ь – будем употреблять и в оставшейся части статьи.

V. БАЗИСНЫЕ АВТОМАТЫ ДЛЯ СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ: ОСНОВНЫЕ СВОЙСТВА

В этом разделе мы приведём несколько утверждений, описывающих свойства базисных автоматов для сильно связанных языков. Отметим, что выполнимость утверждений *не* следует из примеров, рассмотренных выше (в том числе ранее в [1]) – и поэтому требует специальных доказательств.

Теорема 1: Если для рассматриваемого языка L базисный автомат $\mathcal{BA}(L)$ является сильно связанным, то и оба канонических автомата – $\widetilde{L}$ и $\widetilde{L^R}$ – являются сильно связанными.

Доказательство. При доказательстве мы будем пользоваться обычными соглашениями об обозначениях вспомогательных объектов, связанных с исходным языком L; эти обозначения были приведены в [1, раздел III].

Достаточно доказать, не ограничивая общности, что сильно связанным является автомат $\widetilde{L}$; а для этого, также не ограничивая общности, достаточно доказать, что для любых $A, B \in Q_\pi$ выполняется условие

$$A \underset{\widetilde{L}}{\longrightarrow} B\,.$$

А оно является следствием того, что любому пути базисного автомата

$$\begin{smallmatrix} A \\ X \end{smallmatrix} \xrightarrow[\mathcal{BA}(L)]{u} \begin{smallmatrix} B \\ Y \end{smallmatrix} \tag{1}$$

($X, Y \in Q_\rho$ выбираем произвольно) соответствует путь с той же пометкой u канонического автомата

$$\begin{smallmatrix} A \\ X \end{smallmatrix} \xrightarrow[\widetilde{L}]{u} \begin{smallmatrix} B \\ Y \end{smallmatrix}\,;$$

при этом какой-либо путь вида (1) существует ввиду сильной связанности базисного автомата. □

Эквивалентная формулировка приведённой теоремы такова.

Теорема 1′: Если для рассматриваемого языка L какой-либо из двух канонический автоматов – $\widetilde{L}$ либо $\widetilde{L^R}$ – не является сильно связанным, то и базисный автомат $\mathcal{BA}(L)$ не является сильно связанным. □

Выполняется и обратная теорема – однако известное автору её доказательство существенно сложнее.

Теорема 2: Если для рассматриваемого языка L оба канонических автомата – $\widetilde{L}$ и $\widetilde{L^R}$ – являются сильно связанными, то и базисный автомат $\mathcal{BA}(L)$ является сильно связанным.

[10] А подобное удаление – вместе с выходящими из удаляемых состояний дугами – возможно в каких-либо иных примерах.

Схема доказательства. Рассмотрим некоторые два состояния автомата $\mathcal{BA}(L)$ – пусть это $\begin{smallmatrix} A \\ X \end{smallmatrix}$ и $\begin{smallmatrix} B \\ Y \end{smallmatrix}$; отметим, что возможно одно из двух равенств – $A = B$ либо $X = Y$ (а допускать возможность выполнения обоих равенств одноверменно незачем). Достаточно доказать, что в автомате $\mathcal{BA}(L)$ существует путь из $\begin{smallmatrix} A \\ X \end{smallmatrix}$ в $\begin{smallmatrix} B \\ Y \end{smallmatrix}$, т. е.

$$\begin{smallmatrix} A \\ X \end{smallmatrix} \xrightarrow[\mathcal{BA}(L)]{u} \begin{smallmatrix} B \\ Y \end{smallmatrix} \tag{2}$$

для некоторого $u \in \Sigma^*$.

Рассмотрим язык L', получаемый с помощью следующих канонических автоматов $\widetilde{L'}$ и $\widetilde{(L')^R}$:

$$\widetilde{L'} = (Q_\pi, \Sigma, \delta_\pi, \{A\}, F'_\pi)$$

и

$$\widetilde{(L')^R} = (Q_\rho, \Sigma, \delta_\rho, \{Y\}, F'_\rho).$$

(Специально отметим несколько необычное для наших построений обозначение входа второго канонического автомата.)

Важно также отметить, что именно из-за рассмотрения *нового* языка мы должны писать не «доказательство», а «схема доказательства»; вот небольшие комментарии. Чисто формально – мы не можем рассматривать для нового языка L' и его канонических автоматов вспомогательные объекты (функции разметки, бинарное отношение #, базисный автомат и т. п.) – без их предварительного «вычисления». Более того, даже состояния автоматов

$$\widetilde{L'} \quad \text{и} \quad \widetilde{(L')^R}$$

желательно обозначить по-другому.

Однако ниже станет очевидно, что все эти объекты для языка L' будут совпадать с соответствующими объектами для языка L (при одинаковых обозначениях состояний их канонических автоматов), а подробное строгое доказательство вряд ли представляет интерес.

Приведём также следующие замечания:

- как и в доказательстве предыдущей теоремы, здесь мы пользуемся обычными соглашениями об обозначениях вспомогательных объектов, связанных с исходным языком L;
- состояния A и Y – те, что выбраны нами в начале доказательства;
- также на основе приведённого в [1] однозначно определяются и множества F'_π и F'_ρ в двух последних формулах: F'_π содержит те и только те состояния $Z \in Q_\rho$, для которых $A \in Z$; аналогично – для F'_ρ.

Очевидно, что автоматы $\widetilde{L'}$ и $\widetilde{(L')^R}$ действительно можно считать каноническими (причём, конечно, для языков L' и $(L')^R$); кроме того, они отличаются от автоматов $\widetilde{L}$ и $\widetilde{L^R}$ соответственно только стартовыми и финальными состояниями – и поэтому также являются сильно связанными.

По построению, $B \in F'_\pi$ (просто потому, что выполняется условие $B\#Y$), все рассматриваемые автоматы не имеют бесполезных и недостижимых состояний, в частности,

$$\mathcal{L}^{in}_{\widetilde{L'}}(B) \neq \varnothing\,;$$

поэтому существует путь

$$\begin{smallmatrix} A \\ X \end{smallmatrix} \xrightarrow[\mathcal{BA}(L')]{u} \begin{smallmatrix} B \\ Y \end{smallmatrix}\,,$$

соответствующий для исходного языка и его автомата $\mathcal{BA}(L)$ пути (2). $\square$

VI. ПРЕДВАРИТЕЛЬНЫЕ СВЕ́ДЕНИЯ – 2: КРАТКО ОБ УНИВЕРСАЛЬНОМ АВТОМАТЕ

В дальнейшем для различных ситуаций мы *без подробных формальных определений* будем использовать обозначения (операции) α и β – со следующими неформальными пояснениями: α – первый элемент декартова произведения (например, элемента бинарного отношения), а β – второй; сам объект, от которого берутся элементы, при этом пишется в скобках. Например, для состояния базисного автомата ${}^{A}_{X}$ имеем

$$\alpha({}^{A}_{X}) = A\,, \quad \beta({}^{A}_{X}) = X$$

(здесь результат обеих операций – состояния), а для грида [11]

$$\mathcal{B} = \{A, B\} \times \{X, Y\}$$

получаем

$$\alpha(\mathcal{B}) = \{A, B\}\,, \quad \beta(\mathcal{B}) = \{X, Y\}$$

(здесь результат обеих операций – множества состояний).

Теперь переходим к основному предмету раздела – *универсальным автоматам*. Такой автомат для заданного языка L определяется на множестве гридов этого языка – используемых в качестве его состояний [12]; будем при рассмотрении универсального автомата обозначать гриды $\mathcal{B}$, обычно с нижними индексами. Отметим, что в публикациях автора универсальный автомат чаще обозначается $\mathcal{COM}(L)$, а эквивалентность этих двух определений была показана, например, в [10], [11] [13].

Итак, автомат $\mathcal{COM}(L)$ – это пятёрка

$$\mathcal{COM}(L) = \big(Q_{\mathcal{COM}}, \Sigma, \delta_{\mathcal{COM}}, S_{\mathcal{COM}}, F_{\mathcal{COM}}\big),$$

где:

- $Q_{\mathcal{COM}}$ – множество гридов бинарного отношения # для рассматриваемого языка;
- $S_{\mathcal{COM}} = \big\{ \mathcal{B} \in Q_{\mathcal{COM}} \,\big|\, \alpha(\mathcal{B}) \ni s_{\pi} \big\}$; [14]
- $F_{\mathcal{COM}} = \big\{ \mathcal{B} \in Q_{\mathcal{COM}} \,\big|\, \beta(\mathcal{B}) \ni s_{\rho} \big\}$.

Функция переходов $\delta_{\mathcal{COM}}$ определяется следующим образом: дуга

$$\mathcal{B}_1 \xrightarrow[\delta_{\mathcal{COM}}]{a} \mathcal{B}_2$$

существует тогда и только тогда, когда одновременно выполнены следующие два условия [15]:

$$\big(\forall A \in \alpha(\mathcal{B}_1)\big)\ \big((\delta_{\pi}(A, a) \neq \emptyset)\ \&\ (\delta_{\pi}(A, a) \subseteq \alpha(\mathcal{B}_2))\big)\,;$$
$$\big(\forall Y \in \beta(\mathcal{B}_2)\big)\ \big((\delta_{\rho}(Y, a) \neq \emptyset)\ \&\ (\delta_{\rho}(Y, a) \subseteq \beta(\mathcal{B}_1))\big)\,.$$

VII. ПРЕДВАРИТЕЛЬНЫЕ СВЕ́ДЕНИЯ – 3: РЕЗЮМЕ

…Итак, можно полагать, что вместе с заданным регулярным языком L мы одновременно задаём и следующие связанные с ним объекты [16]:

- два канонических автомата (т. е. $\widetilde{L}$ и $\widetilde{L^R}$), а также их состояния, функци переходов и т. п.;
- бинарное отношение #, определённое на парах состояний автоматов $\widetilde{L}$ и $\widetilde{L^R}$;
- множество гридов, определённое на основе бинарного отношения #;
- функции разметки состояний φ^{in} и φ^{out};
- эквивалентый базисный автомат $\mathcal{BA}(L)$;
- эквивалентый автомат $\mathcal{COM}(L)$.

При этом отметим, что если исходный язык L задан с помощью некоторого конечного автомата (пусть это автомат K, имеющий n состояний), то все эти объекты мы получаем, выполняя, вообще говоря, $\sim 2^n$ операций; и можно добавить, что:

- последнюю оценку нельзя уменьшить (ведь эквивалентный канонический автомат имеет, вообще говоря, $\sim 2^n$ состояний);
- но такая оценка и не больше – это следует из приведённых выше конкретных вариантов описания необходимых нам алгоритмов [17].

Таким образом, все перечисленные объекты можно построить за время $\sim 2^n$. При этом очень часто в практических задачах нужны и другие объекты – которые

[11] Кратко повторим основные определения из [1]. Каждый *псевдогрид* представляет собой пару (P, R) (где $P \subseteq Q_{\pi}$ и $R \subseteq Q_{\rho}$), такую что для каждой пары состояний $p \in P$ и $r \in R$ выполнено условие $p\#r$. Если для некоторого псевдогрида (P, R) мы не можем расширить ни множество P, ни множество R – без того, чтобы не нарушилось определение псевдогрида, то такой псевдогрид называется *гридом*.

[12] Аналогично тому, как базисный автомат определялся на множестве пар состояний, удовлетворяющих бинарному отношению #.

[13] На самом деле, ранее нами были опубликованы *два разных* варианта доказательства совпадения (с точностью до переобозначения состояний) универсального автомата и автомата $\mathcal{COM}$: неконструктивное «алгебраическое» доказательство и конструктивное доказательство «путём описания дуг». Подробности приводить не будем, они для настоящей статьи не очень интересны – хотя полученные результаты используются ниже в одном из доказательств.

[14] Ещё раз отметим, что первый элемент грида *является множеством* – это объясняет корректность приведённого обозначения; аналогично для $F_{\mathcal{COM}}$.

[15] При этом специально отметим необходимость проверки условия неравенства пустому множеству: мы уже отмечали, что рассматриваем определение канонического автомата без возможного «дохлого» состояния – поэтому автоматы $\widetilde{L}$ и $\widetilde{L^R}$ не являются, вообще говоря, всюду определёнными (total). Поэтому равенство пустому множеству фактически означало бы возможность перехода в это «дохлое» состояние (в случае его рассмотрения), и говорить об эквивалентности автоматов (исходного и строимого) мы бы не могли.

[16] Выше мы почти не говорили о сложности алгоритмов для построения этих объектов – и к этому можно добавить как «отрицательный», так и «положительный» комментарий.

«Отрицательным» является тот факт (мы его уже отмечали), что эта сложность не может быть менее экспоненциальной – поскольку канонический автомат, эквивалентный заданному, имеющему n состояний, может содержать до $2^n - 1$ состояний.

А «положительный комментарий» таков: в процессе построения этих объектов мы избегаем дублирования многих действий: например, процедуру канонизации недетерминированного автомата мы применяем только 2 раза – хотя формально она является частью определения каждого из этих объектов.

[17] При этом *для практики* важны следующие два замечания.

Во-первых, повторим ещё раз: описанные нами алгоритмы экономят реальное время работы соответствующих программ (хотя оценку $\sim 2^n$, конечно же, понизить не могут) – поскольку построения вспомогательных объектов в них осуществляется одновременно.

Во-вторых, мы строим автомат $\widetilde{L^R}$ (или даже $(\widetilde{L^R})^R$, последнее преобразование практически не требует дополнительных временны́х затрат). Формально это можно делать, эквивалентно преобразуя любой автомат для рассматриваемого языка L^R (производя канонизацию этого автомата) – и необходимая оценка $\sim 2^n$ получается, например, при рассмотрении автомата K^R (где K – автомат исходный). Однако на практике обычно удобнее рассматривать автомат $(\widetilde{L})^R$ – хотя *чисто формально* сложность алгоритмов при этом получается существенно большей, $\sim 2^{2^n}$.

за такое время построить невозможно [18]. Одним из таких объектов является множество покрывающих автоматов – см. [10], [12], [13] и др.; мы собираемся вернуться к этим автоматам (конечно, в первую очередь к их связи с сильной связанностью регулярного языка) в одной из ближайших публикаций. А здесь отметим лишь следующее. Во-первых, перебор множества покрывающих автоматов – это одна из возможностей решения задачи вершинной минимизации недетерминированных автоматов, *такой* способ её решения даёт время $\sim 2^{2^n}$. Во-вторых, лишь в 1993 г. (относительно недавно) в [14] была доказана NP-трудность этой проблемы.

VIII. ГРИДЫ И УНИВЕРСАЛЬНЫЕ АВТОМАТЫ ДЛЯ СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ: ПРИМЕРЫ

Приведём примеры гридов и универсальных автоматов для языков, рассмотренных ранее в разделе IV. Заранее отметим, что все три универсальных автомата получаются сильно связанными.

Во-первых, продолжим рассмотрение языка Ъ, заданного автоматом, приведённым на рис. 5. Повторим все 5 гридов этого языка согласно [1]:

$$\begin{array}{ll} (1) & \{A,C,D\}\times\{Y,Z\}, \\ (2) & \{A,B,C,D\}\times\{Z\}, \\ (3) & \{B,C,D\}\times\{X,Z\}, \\ (4) & \{C\}\times\{X,Y,Z,U\}, \\ (5) & \{C,D\}\times\{X,Y,Z\}. \end{array}$$

Используя определения раздела VI, мы для этого языка получаем следующий автомат $\mathcal{COM}$ (таб. 8):

Таб. 8

$\mathcal{COM}(L)$	a	b
$\rightarrow$ 1	$2,3$	$1,2,3,4,5$
$\rightarrow$ 2	–	$1,2,3,5$
$\leftarrow$ 3	–	$1,2,3,5$
$\leftarrow$ 4	$1,2,3,4,5$	$1,2,3,4,5$
$\leftarrow$ 5	$2,3$	$1,2,3,4,5$

Теперь продолжим рассмотрение языка Ы (исходный автомат был приведён на рис. 6). На основе таб. 4 мы получаем такое множество гридов:

$$\begin{array}{ll} (1) & \{A,C\}\times\{Y,U\}, \\ (2) & \{A,C,D\}\times\{U\}, \\ (3) & \{B\}\times\{X,Z\}, \\ (4) & \{B,E\}\times\{Z\}. \end{array}$$

[18] По-видимому, точнее говорить «по-видимому, невозможно».

Аналогично языку Ъ, мы будем использовать *те же самые номера гридов* [19] и в дальнейшем рассмотрении этого примера – при определении состояний универсального автомата.

Согласно приведённым определениям, мы для языка Ы получаем следующий автомат $\mathcal{COM}$ (таб. 9):

Таб. 9

	delta	α x β	a	b	c
→	1	AC x YU	3		
→	2	ACD x U	4		
←	3	B x XZ	1	2	4
	4	BE x Z	1	2	

Несложно убедиться, что канонизация этого автомата даёт в точности автомат, приведённый в таб. 5.

Далее снова рассмотрим «переход» от языка Ы к нашему последнему примеру, языку Ь. При этом мы можем сказать то же самое, что и ранее в разделе IV: *здесь* достаточно просто «удалить» всё связанное с единственной дугой, помеченной c. А именно:

- множество гридов не изменяется;
- в таб. 9 просто удаляется последний столбец [20].

Итак, во всех трёх примерах языков мы получаем сильно связанные универсальные автоматы. Возможно, «самым интересным» из них является последний пример, язык Ь: кроме исходного автомата [1, рис. 12], все остальные связанные с ним автоматы, т. е.

$$\widetilde{L},\ \widetilde{L^R},\ (\widetilde{L})^R,\ (\widetilde{L^R})^R,\ \mathcal{BA}(L),$$

не являются сильно связанными – а универсальный автомат $\mathcal{COM}(L)$ таковым является.

IX. УНИВЕРСАЛЬНЫЕ АВТОМАТЫ ДЛЯ СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ: ОСНОВНОЕ СВОЙСТВО

В этом разделе мы рассматриваем основное свойство класса сильно связанных языков: доказываем тот факт, что универсальный автомат $\mathcal{COM}(L)$ для сильно связанного языка L также является сильно связанным.

Теорема 3: Если для рассматриваемого языка L существует определяющий его автомат, являющийся сильно связанным, то и универсальный автомат $\mathcal{COM}(L)$ является сильно связанным.

Доказательство. Рассмотрим некоторые два разных состояния автомата $\mathcal{COM}(L)$ – пусть это $\mathcal{B}_1$ и $\mathcal{B}_2$; достаточно доказать, что в автомате $\mathcal{COM}(L)$ существует путь из $\mathcal{B}_1$ в $\mathcal{B}_2$, т. е.

$$\mathcal{B}_1 \xrightarrow[\mathcal{COM}(L)]{x} \mathcal{B}_2 \tag{3}$$

для некоторого $x \in \Sigma^*$.

Пусть существующий сильно связанный автомат – это

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

[19] Мы эти номера «подгоняем под ответ» – чтобы в процессе построений сразу получалось нечто похожее на исходный автомат. Но, конечно же, при использовании любой перестановки этих номеров мы в конце концов получим то же самое.

[20] В очередной раз повторим, что подобный «алгоритм» применим далеко не для всех языков.

и, конечно, $L = \mathcal{L}(K)$. Для этого автомата рассмотрим пару состояний $q_1, q_2 \in Q$, такую что [21]

$$\alpha(\mathcal{B}_1) \ni \varphi_K^{in}(q_1), \quad \beta(\mathcal{B}_1) \ni \varphi_K^{out}(q_1),$$

$$\alpha(\mathcal{B}_2) \ni \varphi_K^{in}(q_2), \quad \beta(\mathcal{B}_2) \ni \varphi_K^{out}(q_2);$$

согласно «алгебраическим определениям» универсального автомата ([10], [12] и др.), такой выбор возможен.

Для выбранной пары состояний рассмотрим некоторое слово $y \in \mathcal{L}(K)$, такое что:

- $y = uvw$;
- $u \in \mathcal{L}_K^{in}(q_1)$;
- $v \in \mathcal{L}_K^{io}(q_1, q_2)$;
- $w \in \mathcal{L}_K^{out}(q_2)$.

Из изложенного в [15] описания множества возможных дуг произвольного автомата для заданного языка L с помощью дуг автомата $\mathcal{COM}(L)$ мы получаем, что

- $u \in \mathcal{L}_{\mathcal{COM}}^{in}(\mathcal{B}_1)$;
- $v \in \mathcal{L}_{\mathcal{COM}}^{io}(\mathcal{B}_1, \mathcal{B}_2)$;
- $w \in \mathcal{L}_{\mathcal{COM}}^{out}(\mathcal{B}_2)$,

т. е. мы можем считать выбранное слово v требуемым в условии (3) словом x. $\square$

Таким образом, на основе доказанной теоремы мы имеем возможность следующим образом *эквивалентно переформулировать определение* сильной связанности для регулярного языка: таковым является язык L, для которого универсальный автомат $\mathcal{COM}(L)$ является сильно связанным.

X. ЗАКЛЮЧЕНИЕ

Итак, в настоящей статье мы подробно рассмотрели вопросы, относящиеся к базисному и универсальному автоматам для сильно связанных регулярных языков. Отметим ещё раз, что все соответствующие алгоритмы могут быть выполнены за время порядка $\sim 2^n$, где n – число состояний некоторого «исходного» автомата, с помощью которого рассматриваемый регулярный язык задаётся. Важно отметить, что мы обычно (в частности, в оценках времени алгоритмов этой статьи) не рассматриваем зависимость от числа букв рассматриваемого алфавита – впрочем, это обычный подход.

В дальнейшем – если говорить о сложности алгоритмов – мы предполагаем рассмотреть сохранение / несохранение сильной связанности при применении алгоритмов минимизации автоматов (что без применения эвристик требует времени порядка $\sim 2^{2^n}$). В частности, мы предполагаем опубликовать возможность автоматической проверки сохранения / несохранения сильной связанности в процессе построения покрывающих автоматов – а такая проверка может осуществляться за время, вычисляемое в зависимости от числа букв рассматриваемого алфавита, поскольку зависит от множества циклов, имеющихся в базисном автомате [22].

А ещё одно направление продолжения работ, кратко упомянутое в настоящей статье – это исследование эквивалентных состояний базисного автомата и автоматов, получающихся после объединения таких состояний (их условное название – «упрощённые базисные»). Как мы уже отмечали выше, они обладают многими нужными для практических задач свойствами базисных автоматов.

Список литературы

[1] 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.

[2] Thierrin G. *Permutation automata* // Mathematical Systems Theory. – 1968. – No. 2. – P. 83–90.

[3] Krawetz B., Lawrence J., Shallit J. *State complexity and the monoid of transformations of a finite set* // International Journal of Foundations of Computer Science. – 2005. – Vol. 16. No. 3. – P. 547–563.

[4] Singh S. N. *Semi-Flower Automata. PhD thesis* // Indian Institute of Technology Guwahati, 2012.

[5] Singh S. N., Krishna K. V. *The rank and Hanna Neumann property of some submonoids of a free monoid* // Annals Math. Inform. – 2012. – Vol. 40. – P. 113–123 (arXiv:1112.4250).

[6] Singh S. N., Krishna K. V. *A sufficient condition for the Hanna Neumann property of submonoids of a free monoid* // Semigroup Forums. – 2013. – Vol. 86. No. 3. – P. 537–554.

[7] Мельников Б. Ф. *Регулярные языки и недетерминированные конечные автоматы: монография.* – М.: Изд-во Российского государственного социального университета, 2018. – 179 с.

[8] Melnikov B., Melnikova A. *Some properties of the basis finite automaton* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2002. – Vol. 9. No. 1. – P. 135–150.

[9] Мельникова А. *Некоторые свойства базисного автомата* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2012. – № 2. – С. 184–189.

[10] Долгов В., Мельников Б. *Построение универсального конечного автомата. I. От теории к практическим алгоритмам* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2013. – № 2. – С. 173–181.

[11] Долгов В., Мельников Б. *Построение универсального конечного автомата. II. Примеры работы алгоритмов* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2014. – № 1. – С. 78–85.

[12] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta Universitatis Sapientiae, Informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.

[13] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.

[14] Jiang T., Ravikumar B. *Minimal NFA problems are hard* // SIAM Journal on Computing (SICOMP). – 1993. – Vol. 22. No. 6. – P. 1117–1141.

[15] 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. 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`.

[21] Мы используем функции разметки – поэтому, аналогично теореме 2, чисто формально мы должны сначала определить два канонических автомата для рассматриваемого языка, указать, что значения α для состояний автомата $\mathcal{COM}(L)$ определены на том же множестве, что и значения функции φ_K^{in} (то же самое – для β и φ_K^{out}), и т. д…

Мы для простоты этого не делаем – однако, в отличие от упомянутой теоремы 2, здесь мы по понятным причинам («недоделано» очень мало) всё-таки пишем «доказательство», а не «схема доказательства».

[22] Но, конечно, мы должны потратить время порядка $\sim 2^n$ на построение самого базисного автомата. Здесь имеется в виду, что для таких задач можно обойтись без алгоритмов, имеющих временну́ю сложность $\sim 2^{2^n}$).

Automata – complete invariants for strongly connected regular languages

Boris Melnikov

Abstract—**In this paper, we continue to consider strongly coupled automata, as a subset of ordinary nondeterministic finite automata. Based on this, strongly related regular languages are naturally defined. We consider some properties of the concept of the strong connectedness, in particular, the non-closure of this class with respect to ordinary set-theoretic operations.**

The title of the paper includes the word "invariants"; such ones for the regular languages are not only the canonical automata (besides, they are complete invariants), but also the canonical automaton for the mirror language, as well as the basis and universal automata, which ones (for strongly connected languages) are the main subject of this paper. One of the important incomplete invariants is the binary relation #, which is also discussed in this paper.

We show the connection of the concept of "strong connectedness" with the basic and universal finite automata. Thus, for a strongly connected language, the basic automaton is not necessarily strongly connected, and the universal automaton is necessarily so. We consider the last fact to be the most important result related to strongly connected languages: it allows us to equivalently reformulate the definition of a strongly connected regular language, such can be considered a language for which its universal automaton is strongly connected. We illustrate the consideration of basis and universal automata for strongly connected languages with some examples, the consideration of which was started in the previous paper on strongly connected languages.

Keywords—**regular languages, nondeterministic finite automata, subclasses of the language classes, basis automaton, universal automaton.**

References

[1] Melnikov B. *On a subclass of the regular language class ("strongly connected" languages): definitions and corresponding canonical automata* // International Journal of Open Information Technologies. – 2021. – Vol. 9. No. 3. – P. 1–10 (in Russian).

[2] Thierrin G. *Permutation automata* // Mathematical Systems Theory. – 1968. – No. 2. – P. 83–90.

[3] Krawetz B., Lawrence J., Shallit J. *State complexity and the monoid of transformations of a finite set* // International Journal of Foundations of Computer Science. – 2005. – Vol. 16. No. 3. – P. 547–563.

[4] Singh S. N. *Semi-Flower Automata. PhD thesis* // Indian Institute of Technology Guwahati, 2012.

[5] Singh S. N., Krishna K. V. *The rank and Hanna Neumann property of some submonoids of a free monoid* // Annals Math. Inform. – 2012. – Vol. 40. – P. 113–123 (arXiv:1112.4250).

[6] Singh S. N., Krishna K. V. *A sufficient condition for the Hanna Neumann property of submonoids of a free monoid* // Semigroup Forums. – 2013. – Vol. 86. No. 3. – P. 537–554.

[7] Melnikov B. F. *Regular languages and nondeterministic finite automata: monograph*. – Moscow: Russian State Social University Ed., 2018. – 179 p. (in Russian).

[8] Melnikov B., Melnikova A. *Some properties of the basis finite automaton* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2002. – Vol. 9. No. 1. – P. 135–150.

[9] Melnikova A. *Some properties of the basis automaton* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2012. – No. 2. – P. 184–189 (in Russian).

[10] Dolgov V., Melnikov B. *Construction of a universal finite automaton. I. From the theory to the practical algorithms* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2013. – No. 2. – P. 173–181 (in Russian).

[11] Dolgov V., Melnikov B. *Construction of a universal finite automaton. II. Examples of how algorithms work* // Bulletin of the Voronezh State University. Series: Physics. Mathematics. – 2014. – No. 1. – P. 78–85 (in Russian).

[12] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta Universitatis Sapientiae, Informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.

[13] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.

[14] Jiang T., Ravikumar B. *Minimal NFA problems are hard* // SIAM Journal on Computing (SICOMP). – 1993. – Vol. 22. No. 6. – P. 1117–1141.

[15] 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. 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`.