

Об одном подклассе класса регулярных языков («сильно связанные» языки): определения и соответствующие канонические автоматы

Б. Ф. Мельников

Аннотация—В наших предыдущих публикациях был рассмотрен упрощённый по сравнению с классикой вариант определения омега-автоматов, задающих омега-языки. Однако, несмотря на эту «упрощённость», такое определение даёт возможность задавать нужные нам в самых разных задачах подклассы класса омега-языков, описывать соответствующие алгоритмы эквивалентных преобразований определённых нами омега-автоматов и т. п.

При этом ранее мы уже рассматривали группу вопросов, относящихся к омега-автоматам и соответствующим омега-вычислениям, но которые, при некоторой их переформулировке, могут быть отнесены и к обычным недетерминированным конечным автоматам. Мы определили итеративные (iterating) и сильно связанные (strongly connected) омега-автоматы – однако на основе рассмотренных в предыдущих статьях определений и свойств подобных автоматов можно заключить, что практически те же самые понятия можно применять и к обычным НКА. В связи с этим мы в настоящей статье определяем сильно связанные автоматы – как подмножество обычных НКА.

На основе этого понятия мы определяем сильно связанные языки, после чего показываем их связь с имеющимися в классе регулярных языков инвариантами – каноническим, базисным и универсальным конечными автоматами. В частности, мы показываем, что для сильно связанного языка и его зеркального образа (mirror image) возможны все 4 варианта выполнения условия сильной связности соответствующих канонических автоматов.

В настоящей статье, а также в предполагаемой статье-продолжении, мы рассмотрим подклассы класса сильно связанных языков, приведём алгоритмы для эквивалентных преобразований соответствующих им сильно связанных автоматов, опишем некоторые специальные подклассы класса сильно связанных языков и некоторые свойства этих подклассов, что даст возможность описания алгоритмов эквивалентного преобразования для упомянутого выше варианта омега-автоматов.

Ключевые слова—регулярные языки, недетерминированные конечные автоматы, подклассы классов языков, алгоритмы эквивалентного преобразования автоматов.

I. ВВЕДЕНИЕ

В статье автора [1] был рассмотрен упрощённый (по сравнению с классикой, [2], [3] и др.) вариант определения ω -автоматов, задающих ω -языки. Однако, несмотря на отмеченную «упрощённость», такое определение даёт возможность задавать нужные нам в самых разных задачах подклассы класса ω -языков, описывать соответствующие алгоритмы эквивалентных преобразований

определённых нами ω -автоматов и т. п. – см. подробности там же, в [1]. Мы покажем эти возможности в предполагаемых статьях – её продолжениях; в них мы рассмотрим описания специальных подклассов класса ω -языков, некоторые их свойства, а также новые алгоритмы эквивалентного преобразования определённого нами варианта ω -автоматов.

При этом в нескольких последних разделах статьи [1] (а именно, в разделах V–VII) мы фактически уже рассматривали группу вопросов, которые, конечно, относятся к ω -автоматам и соответствующим ω -вычислениям... но которые (при некоторой их переформулировке) могут быть отнесены и к *обычным* недетерминированным конечным автоматам (НКА). Мы определили итеративные (iterating) и сильно связанные (strongly connected) ω -автоматы – однако на основе рассмотренных там определений и свойств подобных автоматов можно заключить, что практически те же самые понятия можно применять и к обычным НКА. В связи с этим мы в настоящей статье определяем сильно связанные автоматы – как подмножество обычных НКА. В дальнейших публикациях мы также предполагаем рассмотреть итеративные автоматы¹ и только потом – опубликовать продолжение статьи [1], т. е. её часть II.

Таким образом, мы будем рассматривать сильно связанные языки и соответствующие им недетерминированные конечные автоматы – т. е. такие автоматы, графы переходов которых являются сильно связанными. (Все термины теории графов, применяемые в статье, согласованы с терминологией классических книг Харари – [4] и др.) При этом, как нам кажется, *мы рассматриваем достаточно простые определения и результаты – однако в литературе мы ничего соответствующего не нашли.* И ещё раз отметим, что материал настоящей статьи будет необходим для продолжения статьи [1]: мы в настоящей статье и в статье-продолжении покажем связь понятия «сильно связанные языки» с имеющимися в классе регулярных языков инвариантами – каноническим, базисным и универсальным конечными автоматами.

Структура настоящей статьи такова. В разделе II перечислены несколько причин нашего интереса к сильно связанным языкам – и стоит отметить, что этими причинами наш интерес к ним далеко не ограничивается. В разделе III приведены употребляемые нами основные обозначения (в том числе нестандартные), связанные с

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

Статья получена 27 декабря 2020 г.

¹ Сначала – тоже без «выхода» в ω -вычисления.

недетерминированными конечными автоматами, применяемые в работах автора.

В разделе IV мы определяем сильно связанные конечные автоматы и сильно связанные регулярные языки, а также приводим некоторые соответствующие примеры. В разделе V мы показываем, что свойство сильной связности не сохраняется при переходе к каноническому автомату, эквивалентному некоторому заданному – более того, для сильно связанного языка и его зеркального образа (mirror image) возможны все 4 варианта выполнения условия сильной связности соответствующих канонических автоматов.

В разделе VI мы формулируем одно очень важное свойство класса сильно связанных языков – *несуществование полиномиального алгоритма* проверки того факта, что некоторый регулярный язык является сильно связанным; мы также приводим схему доказательства этого свойства. (Подробный вариант доказательства требует большой детализации и поэтому является очень громоздким.) А в заключении (раздел VII) мы описываем возможные направления дальнейшей работы, связанные с рассмотренной в настоящей статье тематикой.

II. «МОТИВАЦИЯ»

Основная причина нашего интереса к сильно связанным языкам фактически уже была сформулирована во введении – однако имеются и другие «варианты мотивации».

Начнём со следующего. В настоящей статье мы рассматриваем *фундаментальный* объект, определение которого можно сформулировать сразу после определения НКА и который мы предполагаем использовать в дальнейших публикациях – поэтому этот объект стоит определять без использования многочисленных дополнительных определений и утверждений, приведённых в опубликованной ранее литературе. А возможность такого определения есть – и для полного понимания материала настоящей статьи можно обойтись без (или почти без) ссылок на наши предыдущие работы.

«Вторым пунктом мотивации» являются так называемые 2ω -язык – которые мы предполагаем рассмотреть уже после второй части статьи про ω -языки. Мы рассматривали 2ω -язык в статье [5], опубликованной очень давно, – но автору до сих пор неизвестны другие работы, где рассматриваются такие объекты.

Каждое 2ω -слово можно рассматривать как функцию вида

$$\sigma : \mathbb{Z} \rightarrow \Sigma.$$

Например, для простейшего 2ω -слова ($a^{2\omega}$), каждая буква которого – a , функция σ имеет вид $\sigma(i) = a$ для всех $i \in \mathbb{Z}$. Мы не будем различать два 2ω -слова, если одно из них (пусть σ_1) может быть получено из другого (σ_2) с помощью следующего условия:

$$(\exists n \in \mathbb{Z}) (\forall i \in \mathbb{Z}) (\sigma_1(i) = \sigma_2(i + n)).$$

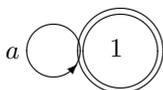


Рис. 1

Некоторые из 2ω -языков могут быть определены с помощью 2ω -автоматов – фактически являющихся НКА без входов и выходов². Соответственно, графы переходов 2ω -автоматов могут быть изображены как обычные графы переходов конечных автоматов без входных и выходных стрелок. Простейший ω -КА (определяющий 2ω -язык $\{a^{2\omega}\}$, содержащий только одно рассмотренное выше 2ω -слово) приведён на рисунке 1; отметим, что в рассматриваемом графе переходов автомата (transition graph) нет не только выходных стрелок (аналогично [1]), но и входных.

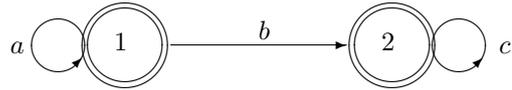


Рис. 2

А автомат на рисунке 2 определяет 2ω -язык, содержащий такие 2ω -слова:

- $a^{2\omega}$;
- $c^{2\omega}$;
- а также 2ω -слово вида³

$$\dots aaa \dots aaa b ccc \dots ccc \dots$$

Отметим, что последний автомат (точнее, соответствующий ему граф переходов) не является сильно связанным – однако в некоторых задачах нам достаточно рассматривать только все компоненты связности по отдельности.

Мы достаточно подробно описали 2ω -слова и соответствующие автоматы – хотя они и не являются предметом настоящей статьи. Однако они – один из возможных объектов приложения рассматриваемых нами сильно связанных языков и автоматов. По-видимому, связь между этими моделями видна из вышеприведённого текста.

У рассматриваемых нами 2ω -автоматов имеется также некоторая аналогия с автоматами, определяемыми в [6, Гл. 4]. В этой монографии в качестве единственного варианта конечных автоматов⁴ фактически используется такое же определение. Однако на этом «аналогия кончается»: оно используется для определения специальной подгруппы (и именно для неё эти автоматы и рассматриваются). Понятно, что «для наших целей».

Перейдём к ещё одному «пункту мотивации». В наших предыдущих статьях, посвящённых ω - и 2ω -автоматам (и соответствующим им ω - и 2ω -языкам), мы в первую очередь рассматривали вопросы, связанные с так называемыми «обструкциями» (т.е. словами, которые не

² Аналогия с первой частью настоящей статьи – в том, что там мы тоже «убирали» элемент (один) из пятёрки, определяющей обычный НКА.

³ Иногда удобнее записывать это 2ω -слово как

$$(a^\omega)^R b (c^\omega)$$

(где буква R в верхнем индексе использована так же, как и для обычных регулярных языков). Отметим, что ранее в литературе публиковались некоторые варианты соглашений о записи ω -слов и ω -языков «в виде, аналогичном регулярным выражениям», эти же соглашения обобщаются на случай 2ω -слов и 2ω -языков – однако, по-видимому, приведённая запись понятна без формулировки этих соглашений.

⁴ По-видимому, единственному: в предметном указателе обоих томов других вариантов нет.

могут входить в качестве подслово в ω - и 2ω -языки, определяемые соответствующими автоматами). Однако вопросы, связанные с ω - и 2ω -языками и соответствующими автоматами, конечно же, далеко не ограничиваются подобной тематикой; некоторые из таких вопросов и рассматриваются в настоящей статье.

При описании «обычных» регулярных языков ω -автоматы (и тем более 2ω -автоматы) тоже могут представлять определённый интерес – поскольку они описывают самые важные свойства этих языков⁵. При этом почти всегда достаточно рассматривать компоненты связности графа переходов 2ω -автомата (а также «конструировать» эквивалентные автоматы и языки из компонентов связности графа переходов автомата и соответствующих им регулярных языков) – и, следовательно, только соответствующие сильно связанные автоматы⁶. Яркий пример подобной задачи – проблема звёздной высоты регулярного языка, см. [7] и некоторые ссылки, приведённые в той статье.

А в качестве «последнего⁷ пункта мотивации» назовём такой. Мы уже упомянули определённые нами ранее итеративные (iterating) ω -автоматы – но совершенно аналогично их можно определить и для обычных НКА. При этом возможны такие два варианта:

- множество выходных состояний совпадает со всем множеством состояний НКА;
- множество выходных состояний состоит из единственного состояния, совпадающего со стартовым.

Отметим, что оба варианта должны найти применение, например, в задачах, связанных с нашими предыдущими публикациями [8], [9].

III. ПРЕДВАРИТЕЛЬНЫЕ СВЕДЕНИЯ: ОСНОВНЫЕ ОБОЗНАЧЕНИЯ, СВЯЗАННЫЕ С АВТОМАТАМИ

Как мы уже отмечали, для полного понимания материала настоящей статьи можно обойтись без (или почти без) ссылок на наши предыдущие работы. Однако необходимы основные обозначения (в том числе нестандартные), связанные с недетерминированными конечными автоматами, применяемые в работах автора. Эти обозначения уже были ранее приведены в нескольких публикациях, в частности в [10], [11], [12], [13], – однако мы дадим их ещё раз: в отличие от упомянутых статей, здесь мы их приводим на русском языке, причём именно в том объёме, который нужен для настоящей статьи.

Пусть

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

недетерминированный конечный автомат Медведева (Рабина–Скотта), задающий регулярный язык

$$L = \mathcal{L}(K).$$

Здесь Q – множество состояний, $S \subseteq Q$ и $F \subseteq Q$ – соответственно множества входных и выходных состояний. Специально отметим, что функция переходов δ автомата (1) будет далее рассматриваться как

$$\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q),$$

⁵ Конечно, строго определить понятие «самые важные свойства» вряд ли возможно.

⁶ В этом абзаце мы сделали «переход» от обычных автоматов «через» 2ω -автоматы «назад» к обычным автоматам.

⁷ Последнего, приведённого в настоящей статье.

(где запись $\mathcal{P}(Q)$ означает множество подмножеств множества Q), а не в виде

$$\delta : Q \times (\Sigma \cup \{\varepsilon\}) \rightarrow \mathcal{P}(Q);$$

таким образом, мы будем рассматривать автоматы без ε -переходов. Некоторую дугу $\delta(q, a) \ni r$ мы иногда будем записывать в виде

$$q \xrightarrow[\delta]{a} r$$

или, если это не вызовет неоднозначности, в виде

$$q \xrightarrow{a} r.$$

Зеркальный автомат для автомата, заданного с помощью (1), определяется как

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

где

$$q' \xrightarrow[\delta^R]{a} q'' \quad \text{если и только если} \quad q'' \xrightarrow[\delta]{a} q'.$$

Мы будем обозначать его записью K^R ; заметим, что K^R определяет (регулярный) язык L^R .

Далее в этом разделе будем считать, что некоторый регулярный язык L задан; для него будем использовать следующие обозначения. Его канонический автомат⁸ будем обозначать записью \tilde{L} ; специально подчеркнём, что это – автомат, а не язык. Элементы «пятёрки», задающих автоматы \tilde{L} и \tilde{L}^R , таковы:

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

и

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

Более того, мы не будем рассматривать язык $L = \emptyset$, поэтому оба канонических автомата должны иметь (по одному) стартовому состоянию.

Определим бинарное отношение $\#$ и функции разметки состояний φ^{in} и φ^{out} ; некоторые подробности см. в. [10], [12]. Отношение

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

определяется для пар состояний автомата \tilde{L} и автомата \tilde{L}^R следующим образом: условие $A \# X$ выполняется тогда и только тогда, когда

$$(\exists uv \in L) (u \in \mathcal{L}_{\tilde{L}}^{in}(A), v \in \mathcal{L}_{\tilde{L}^R}^{in}(X)).$$

Отметим, что это определение неконструктивное⁹; его эквивалентный конструктивный аналог (т.е. определение-алгоритм) приведён в [14], [15].

(Прямая) функция разметки состояний

$$\varphi_K^{in} : Q \rightarrow \mathcal{P}(Q_\pi)$$

определяется следующим образом:

$$\varphi_K^{in}(q) \ni \tilde{q} \quad \text{если и только если} \quad \mathcal{L}_K^{in}(q) \cap \mathcal{L}_{\tilde{L}}^{in}(\tilde{q}) \neq \emptyset.$$

А обратная функция разметки состояний

$$\varphi_K^{out} : Q \rightarrow \mathcal{P}(Q_\rho)$$

⁸ В предыдущих публикациях мы отмечали, что канонические автоматы всегда рассматриваются без возможного «дохлого» состояния (“dead state”). Отметим ещё, что вследствие этого мы, конечно, не требуем, чтобы канонический автомат был всюду определённым (“total”).

⁹ Но более полезное для теоретических построений.

определяется таким же образом, но для другой пары автоматов, K^R и \tilde{L}^R вместо K и \tilde{L} .

На основе определённого выше бинарного отношения $\#$ формируется множество *псевдогридов* (см. [12] и др.)¹⁰: а именно, каждый из них 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 $p\#r$.

В произвольном НКА для заданного регулярного языка L каждое состояние образует некоторый псевдогрид. Более того, *необходимым* условием того, что автомат действительно определяет заданный язык L , является «покрытие» всеми блоками¹¹ всех элементов бинарного отношения $\#$, см. подробнее [14], [15]¹².

А 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 мы будем называть *гридом*.

(Алгоритм построения множества (псевдо)гридов кажется простым – однако даже у этого алгоритма сложность более чем полиномиальна. Это следует хотя бы из (достижимой) верхней оценки числа гридов – которую можно получить, например, на основе [16]: эта оценка для бинарного отношения, в котором мощности обоих множеств Q_π и Q_ρ совпадают и равны n , равна $M(n)$ – n -му числу Дедекинда, [17], [18] и др.)

Далее рассмотрим пример ко всем определённым нами объектам, связанным с заданным регулярным языком L ; см. рис. 3, этот автомат уже рассматривался в наших предыдущих публикациях:

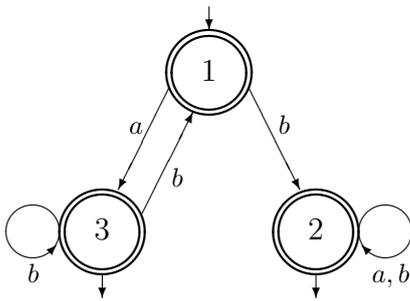


Рис. 3

Процедуры построения объектов описывать не будем (это также сделано в опубликованных ранее статьях) –

¹⁰ Ранее в литературе на русском языке автор употреблял термин «псевдоблоки» (и «блоки» для объектов, определяемых далее). Однако, по-видимому, термины «псевдогриды» и «гриды» всё-таки более удачны – хотя и «подчёркиваются красным цветом» Word-подобными программами.

¹¹ Здесь псевдогриды рассматриваются как множества своих элементов; аналогично – для всего бинарного отношения $\#$.

¹² При этом с *формальной точки зрения* возникает следующее уточнение – которое, однако, вовсе не противоречит приведённой в процитированных статьях теории.

Неэквивалентность автомата формально не даёт возможности применять *определённые в настоящей статье* функции разметки φ^n и φ^{out} – и, следовательно, говорить о бинарном отношении $\#$, определённом таким образом: ведь эти отношения строятся на основе двух канонических автоматов, которые не совпадают с рассматриваемыми.

Однако, во-первых, всё это «перестает быть противоречием» при рассмотрении более сложных определений функций φ^n и φ^{out} – определений, выполненных на основе произвольных детерминированных автоматов (а не обязательно канонических автоматов для рассматриваемых языков L и L^R); но такие «более сложные» определения вряд ли интересны для настоящей статьи. И, во-вторых, несложные доказательства неэквивалентности автоматов (в случае «непокрытия» всех элементов множества $\#$) остаются правильными при рассмотрении определений, приведённых в настоящей статье.

приведём только результаты работы этих процедур.

Зеркальный автомат (K^R) , он определяет язык L^R , рис. 4:

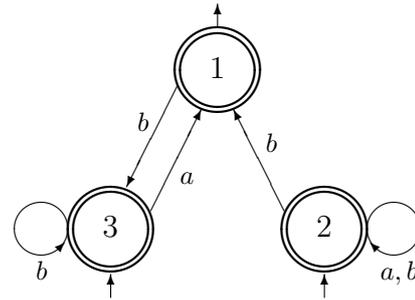


Рис. 4

Соответствующие канонические автоматы \tilde{L} и \tilde{L}^R приведены на рисунке 5 и 6:

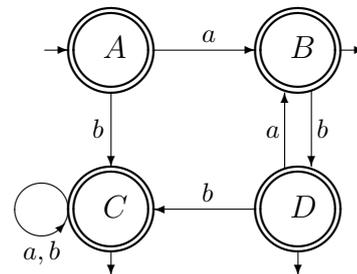


Рис. 5

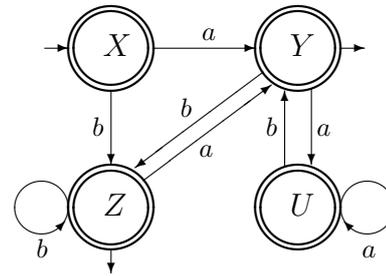


Рис. 6

На основе этих автоматов мы получаем бинарное отношение $\#$ (таб. 1):

Таб. 1

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

И уже на основе последней таблицы мы получаем такие 5 гридов (все возможные для этого бинарного отношения):

- (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) $\{C, D\} \times \{X, Y, Z\}$.

Мы будем использовать *те же самые номера гридов* и в дальнейших примерах, в частности, при определении состояний *новых* автоматов и в статьях-продолжениях.

(Определённые здесь гриды применяются, среди прочего, в алгоритмах минимизации НКА – однако эти вопросы не являются предметом настоящей статьи.)

В заключение раздела отметим, что для всех приведённых здесь объектов имеются *конструктивные алгоритмы* их построения. При этом надо дополнительно отметить следующее:

- мы не всегда приводили эти конструктивные варианты определений: иногда удобнее другие варианты; то же верно и для дальнейшего изложения;
- некоторые из этих определений-алгоритмов – экспоненциальные, причём сложность не может быть понижена (например, потому, что число состояний канонического автомата, эквивалентного некоторому заданному, может экспоненциально зависеть от числа состояний заданного НКА).

IV. СИЛЬНО СВЯЗАННЫЕ РЕГУЛЯРНЫЕ ЯЗЫКИ: ОСНОВНЫЕ ОПРЕДЕЛЕНИЯ И ПРИМЕРЫ

В этом разделе мы определяем сильно связанные конечные автоматы и сильно связанные регулярные языки, а также приводим некоторые соответствующие примеры. Как мы уже отмечали, все термины теории графов, применяемые в статье, согласованы с терминологией классических книг Харари, [4] и др.

Определение 1: Конечный автомат называется *сильно связанным*, если сильно связанным является граф переходов этого автомата. □

Ещё раз отметим очевидную связь этого определения с определением сильно связанного ω -автомата, приведённого в [1].

Среди автоматов, *часто рассматривавшихся в предыдущих статьях автора*, приведём, во-первых, сильно связанный автомат (рис. 7):

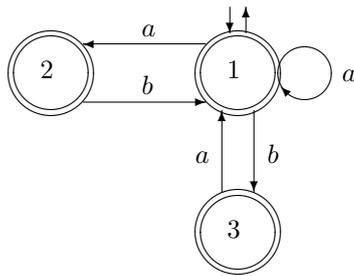


Рис. 7

и, во-вторых, автомат, не являющийся таковым, рис. 3¹³.

Существенно более важным является следующее определение.

Определение 2: Регулярный язык называется *сильно связанным*, если для него существует (хоть один) сильно связанный конечный автомат. □

Среди примеров рассмотрим сначала тривиальный: в 2ω -автомат, рассматривавшийся выше на рисунке 1,

¹³ Отметим, что мы выше также рассмотрели эквивалентные канонический, базисный и универсальный НКА. При этом канонический и базисный автоматы не являются сильно связанными, а универсальный – является.

добавим вход и выход (для получения обычного НКА, причём невырожденного) ...

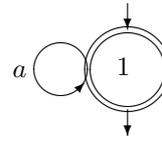


Рис. 8

... после чего «продублируем» получившийся автомат:

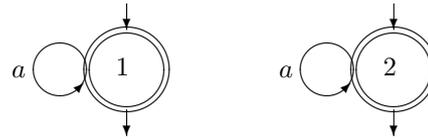


Рис. 9

Очевидно, что эти автоматы определяют один и тот же язык (регулярное выражение для него – a^*), но сильно связанным из этих автоматов является только первый. Также согласно приведённым определениям, язык этих автоматов является сильно связанным.

В заключение раздела приведём пример, часто рассматривающийся в связи с задачами о звёздной высоте ([19], [7] и др.) – причём звёздной высоте как автомата, так и регулярного языка. Рассмотрим такой автомат:

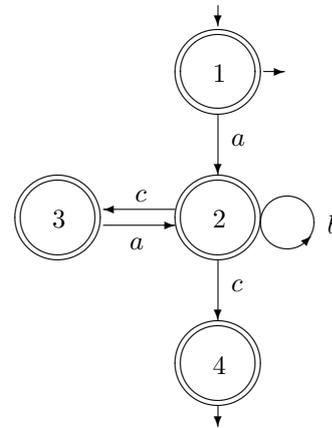


Рис. 10

Конечно же, он не является сильно связанным, и «с ходу» непонятно, существует ли некоторый сильно связанный автомат, ему эквивалентный ...

Такой автомат существует:

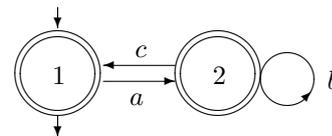


Рис. 11

И действительно, соответствующее регулярное выражение наиболее естественно записывать в виде $(ab^*c)^*$.

Интересно отметить, что этот «естественный» переход – от рис. 10 к рис. 11:

- «с точки зрения» рассматриваемой в настоящей статье сильной связности является *упрощением*,
- а «с точки зрения» звёздной высоты он является *усложнением* (и поэтому в процитированных выше работах, связанных с звёздной высотой регулярно языка, в примерах рассматривается «обратный» переход – от рис. 11 к рис. 10).

V. КАНОНИЧЕСКИЕ АВТОМАТЫ ДЛЯ СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ

Канонический автомат является (полным) инвариантом регулярного языка ([20], [21]) – поэтому возникает такой естественный вопрос: сохраняется ли свойство сильной связности при переходе к каноническому автомату, эквивалентному некоторому заданному?¹⁴ Однако, вообще говоря, ответ на этот вопрос отрицателен – что мы продемонстрируем на следующем примере.

Рассмотрим такой автомат (рис. 12), будем считать его исходным:

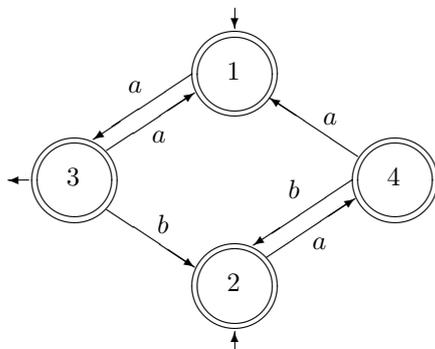


Рис. 12

Очевидно, что он является сильно связанным.

Перепишем его следующим образом (таблица 2)¹⁵:

Таб. 2

	delta	a	b
→	1	3	
→	2	4	
←	3	1	2
	4	1	2

Процесс детерминизации и канонизации опишем в виде следующей таблицы 3 (см. далее). Приведём к ней следующие комментарии:

- во всех таблицах функции переходов будем называть “delta”, вызвать недоразумений это не должно;

¹⁴ (Полными) инвариантами регулярного языка являются также базисный и универсальный автоматы, поэтому возникают ещё и такие вопросы: сохраняется ли свойство сильной связности при построении базисного и универсального автоматов, эквивалентных некоторому заданному. Мы рассмотрим эти вопросы в статье-продолжении, но заранее отметим, что про базисный автомат ответ отрицателен, а про универсальный – положителен, и последний факт даёт возможность рассматривать альтернативное определение сильно связанного языка – определение, формулируемое на основе универсального автомата для этого языка.

Ещё отметим, что (неполным) инвариантом регулярного языка является бинарное отношение #.

¹⁵ Для удобства «сложные» автоматы иногда будем *определять* тоже в виде таблиц. В них входные состояния помечены стрелками слева направо, а выходные – стрелками справа налево.

Таб. 3

	delta	a	b	
→	12	34		A
←	34	1	2	B
	1	3		C
	2	4		D
←	3	1	2	B
	4	1	2	E

- как обычно, запись 34 означает {3, 4} (и т.п.); недоразумений это также не вызовет, т.к. для обозначений состояний в наших примерах достаточно использовать цифры от 1 до 9;
- множества состояний, обозначающие агрегатные состояния, выписываем как обозначения строк «в порядке их появления» – сверху вниз, слева направо;
- после окончания собственно детерминизации переходим к агрегатным состояниям, обозначаемым так же;
- получаем, что состояния {3} и {3, 4} эквивалентны (и больше пар эквивалентных состояний нет);
- поэтому для обозначения попарно неэквивалентных состояний нужно 5 символов – они приведены в последнем столбце;
- как обычно, «дохлого» состояния не используем: вместо перехода в него оставляем пустую клетку.

Полученный эквивалентный канонический автомат «в обычных обозначениях» приведён в таблице 4:

Таб. 4

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

Очевидно, что этот автомат не является сильно связанным.

Продолжим рассмотрение этого же примера – для того, чтобы показать, что для некоторого сильно связанного языка *оба* соответствующих ему канонических автомата (т.е. как \tilde{L} , так и \tilde{L}^R) не являются сильно связанными.

Сначала приведём таблицу для автомата, зеркального к изображённому на рисунке 12 (или в таблице 2); это – таблица 5:

Таб. 5

	delta	a	b
→	3	1	
←	1	34	
	4	2	
←	2		34

Процесс детерминизации и канонизации отражён в следующей таблице 6; комментарии к ней – такие же, как у приведённой выше таблицы 3.

Таб. 6

	delta	a	b	
→	3	1		X
←	1	34		Y
	34	12		Z
←	12	34	34	U

Таб. 9

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

Или, «в обычных обозначениях», получаем таблицу 7:

Таб. 7

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

Как видно из неё, полученный канонический автомат не является сильно связанным – т.е. в приведённом нами примере сильно связанными не являются оба канонических автомата.

Теперь немного модифицируем исходный автомат, добавим одну дугу, см. рис. 13:

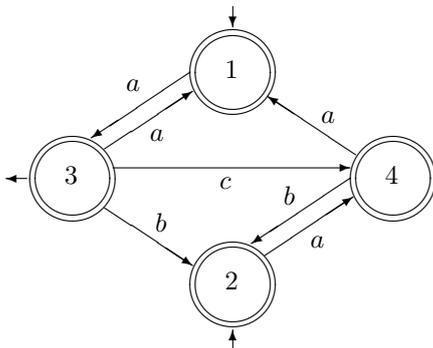


Рис. 13

Конечно, автомат (и, следовательно, его язык) – сильно связанный; рассмотрим два соответствующих канонических автомата, комментарии совпадают с приведёнными выше¹⁶.

Во-первых, его автомат \tilde{L} :

Таб. 8

	delta	a	b	c	
→	12	34			A
←	34	1	2	4	B
	1	3			C
	2	4			D
	4	1	2		E
←	3	1	2	4	B

или, после переобозначения состояний:

¹⁶ Как и всюду ранее, K – рассматриваемый в данный момент автомат, а L – его язык.

Как мы видим, автомат – аналогично более простому случаю, т.е. «без дуги c » – не является сильно связанным.

Но рассмотрим теперь зеркальный автомат K^R :

Таб. 10

	delta	a	b	c
→	3	1		
←	1	34		
	4	2		3
←	2		34	

Процесс детерминизации и канонизации отражён в следующей таблице 11:

Таб. 11

	delta	a	b	c	
→	3	1			X
←	1	34			Y
	34	12		3	Z
←	12	34	34		U

Или, «в обычных обозначениях», получаем для \tilde{L}^R таблицу 12:

Таб. 12

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

Как мы видим, полученный автомат \tilde{L}^R является сильно связанным.

Таким образом, все варианты сильной связности для автоматов \tilde{L} и \tilde{L}^R , не противоречащие определению 1, возможны. Среди них – возможны все 4 варианта, относящихся к сильно связанным языкам. Всё сказанное здесь может быть проиллюстрировано приведённой ниже таблицей 13, вот некоторые комментарии к ней:

- согласно нашим стандартным обозначениям, первый столбец описывает языки, а второй и третий столбцы – автоматы;
- «+» означает сильную связность, а «-» означает её отсутствие;

- ранее не рассмотренная пара «+/-» для автоматов \tilde{L} и L^R получается очевидным образом («без картинок и вычислений») для автомата, зеркального к приведённому на рис. 13.

Таб. 13

L	\tilde{L}	\tilde{L}^R
-	-	-
+	-	-
+	-	+
+	+	-
+	+	+

В статье – продолжении настоящей мы рассмотрим, как понятие сильной связанности отражается в базисных и универсальных автоматах.

VI. ВАЖНОЕ СВОЙСТВО КЛАССА СИЛЬНО СВЯЗАННЫХ ЯЗЫКОВ

В этом разделе мы формулируем одно очень важное свойство класса сильно связанных языков – *несуществование полиномиального алгоритма* проверки того факта, что некоторый регулярный язык является сильно связанным; мы также приводим схему доказательства этого свойства. (Как мы отмечаем во введении, подробный вариант доказательства требует подробной детализации – и поэтому является очень громоздким.) Формулировка основного утверждения этого раздела практически совпадает с частью текста предыдущего абзаца.

Теорема 1: Не существует полиномиального алгоритма проверки того факта, что некоторый регулярный язык является сильно связанным.

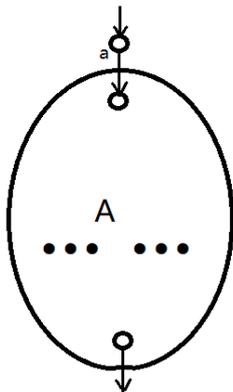


Рис. 14

Схема доказательства. Будем пользоваться тем известным фактом, что в общем случае не существует полиномиального алгоритма проверки эквивалентности двух недетерминированных конечных автоматов¹⁷. Будем пользоваться также «некоторым усложнением» этого

¹⁷ Это – “a folklore result”: для такой проверки мы должны построить (и потом сравнить) эквивалентные им канонические автоматы – а такое построение требует экспоненциального времени, поскольку число состояний эквивалентного канонического автомата, вообще говоря, экспоненциально зависит от числа состояний автомата исходного.

факта – а именно, что такого алгоритма не существует и в той ситуации, когда про один из автоматов известно, что он является сильно связанным¹⁸.

Рассмотрим некоторый сильно связанный язык A и задающий его сильно связанный автомат (также обозначим его A , рис. 14). Не ограничивая общности считаем, что у A ровно 1 вход. На рисунке из этого входа выходит только 1 дуга, помеченная a , – но это непринципиально, при увеличении числа таких дуг рисунки всего лишь станут более громоздкими, но суть не изменится. Также на рисунке у A ровно 1 выход, это также непринципиально.

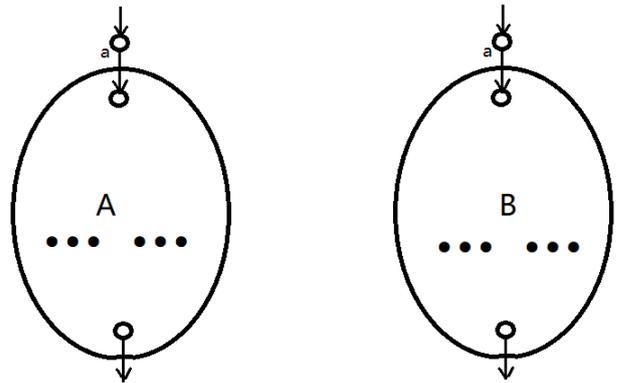


Рис. 15

Теперь будем рассматривать *произвольный* автомат B , рис. 15; единственное ограничение на него – в нём тоже ровно 1 вход, из которого выходят те же самые дуги (поэтому на приведённом рисунке из входа тоже выходит ровно 1 дуга, помеченная a).

Далее рассмотрим автомат, показанный на рис. 16.

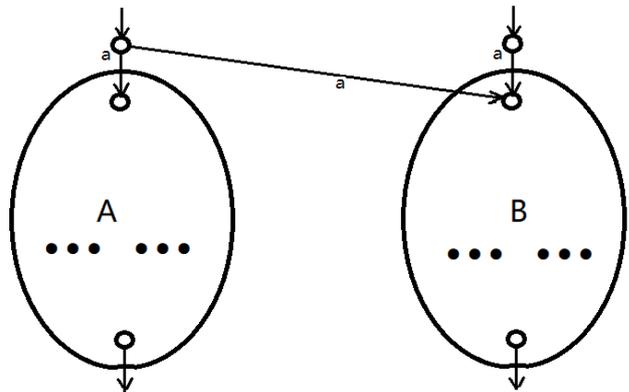


Рис. 16

Предположим, что утверждение теоремы неверно, т. е. пусть алгоритм, про который сказано в условии теоремы, всё-таки существует. Тогда наш «существующий» алгоритм за полиномиальное время проверяет сильную связанность автомата, показанного на рис. 16. А в связи с произвольностью автомата B этот факт равносильен эквивалентности автоматов B и A ¹⁹. □

¹⁸ Это «усложнение» кажется очевидным – однако короткого и при этом *строгого* доказательства возможности такого «усложнения», по-видимому, не существует. Это – *первая* причина, из-за которой мы пишем не «доказательство», а «схема доказательства».

¹⁹ Последнее более-менее очевидно – но, конечно, также требует строгого доказательства; т. е. это *вторая* причина, из-за которой мы пишем «схема доказательства».

VII. ЗАКЛЮЧЕНИЕ

Опишем возможные направления дальнейшей работы, связанные с рассмотренной в настоящей статье тематикой.

В ближайшей статье – продолжении настоящей мы предполагаем рассмотреть некоторые другие свойства понятия сильной связанности, в частности, незамкнутость этого класса относительно теоретико-множественных операций²⁰.

Мы также рассмотрим, как понятие сильной связанности «отражается» в базисных и универсальных автоматах. Так, у сильно связанного языка базисный автомат вовсе не обязательно является сильно связанным, а универсальный автомат – обязательно является. Последний факт²¹ позволит эквивалентно переформулировать определение сильно связанного регулярного языка: таковым можно считать такой язык, у которого сильно связанным является универсальный автомат.

Кроме того, как это было отмечено ранее в «мотивации», мы собираемся рассмотреть подкласс класса сильно связанных языков – а именно, итеративные НКА и соответствующие им итеративные языки.

Список литературы

- [1] Melnikov B., Melnikova A. *Some more on omega-finite automata and omega-regular languages. Part I: The main definitions and properties* // International Journal of Open Information Technologies. – 2020. – Vol. 8. No. 8. – P. 1–7.
- [2] Büchi J. *On a decision method in restricted second order arithmetic* // In: Proceedings of International Congress on Logic, Method, and Philosophy of Science. – Stanford University Press, Stanford, 1962. – P. 425–435.
- [3] Khoussainov B., Nerode A. *Automata Theory and its Applications* // Progress in Computer Science and Applied Logic, Vol. 21. – Springer, Berlin, 2001.
- [4] Харари Ф. *Теория графов* // М.: Мир, 1973.
- [5] Melnikov B. *2ω -finite automata and sets of obstructions of their languages* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 1999. – Vol. 6. No. 3. – P. 565–576.
- [6] *Общая алгебра. Том 2* / Под общ. ред. Л. Скорнякова // М.: Наука, 1991.
- [7] 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.
- [8] Melnikov B. *The equality condition for infinite catenations of two sets of finite words* // International Journal of Foundations of Computer Science. – 1993. – Vol. 4. No. 3. – P. 267–272.
- [9] Melnikov B., Sciarini-Guryanova N. *Possible edges of a finite automaton defining a given regular language* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2002. – Vol. 9. No. 2. – P. 475–485.
- [10] 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.
- [11] Melnikov B., Tsyganov A. *The state minimization problem for nondeterministic finite automata: the parallel implementation of the truncated branch and bound method* // Proceedings – 5th International Symposium on Parallel Architectures, Algorithms and Programming, PAAP-2012. – 2012. – P. 194–201.
- [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] 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.
- [15] Долгов В., Мельников Б. *Построение универсального конечного автомата. II. Примеры работы алгоритмов* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2014. – №. 1. – С. 78–85.
- [16] Lombardy S., Sakarovitch J. *The universal automaton* // Logic and Automata, Amsterdam University Press. – 2008. – P. 457–504.
- [17] Dedekind R. *Über Zerlegungen von Zahlen durch ihre größten gemeinsamen Teiler* // Gesammelte Werke 2, Braunschweig. – 1897. – S. 103–148.
- [18] Коршунов А. *О числе монотонных булевых функций* // Проблемы кибернетики. – 1981. – Т. 38. – С. 5–108.
- [19] Саломая А. *Жемчужины теории формальных языков* // М.: Мир, 1986.
- [20] *Математическая энциклопедия. Том 2* / Под общ. ред. И. Виноградова // М.: Советская энциклопедия, 1979.
- [21] *Математический энциклопедический словарь* / Под общ. ред. Ю. Прохорова // М.: Советская энциклопедия, 1988.

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.

²⁰ Однако, по-видимому, слово «класс» употреблять можно: ведь это слово всегда употребляется, например, применительно к множеству контекстно-свободных языков, которое, как известно, незамкнуто относительно операций пересечения и дополнения.

²¹ Возможно, это самый важный результат, относящийся к сильно связанным языкам.

On a subclass of the regular language class (“strongly connected” languages): definitions and corresponding automata

Boris Melnikov

Abstract—In our previous publications, we considered a simplified version of the definition of omega-automata that define omega-languages compared to the classics. However, despite this “simplification”, such a definition makes it possible to specify the subclasses of the class of omega-languages that we need in a lot of problems, to describe the corresponding algorithms for equivalent transformations of the omega-automata we have defined, etc.

At the same time, we have already considered a group of questions related to omega automata and the corresponding omega computations, but which, with some reformulation, can also be attributed to ordinary nondeterministic finite automata. We have defined iterating and strongly connected omega automata; however, based on the definitions and properties of such automata discussed in previous papers, we can conclude that almost the same concepts can be applied to ordinary NFAs. In this regard, in this paper we define strongly connected automata as a subset of ordinary NFAs.

Based on this concept, we define strongly connected languages, and then show their relation to the invariants available in the class of regular languages: canonical, basis, and universal finite automata. In particular, we show that for a strongly connected language and its mirror image, all 4 variants of the strong connectivity condition of the corresponding canonical automata are possible.

In this paper, as well as in the proposed continuations, we shall consider subclasses of strongly connected languages, present algorithms for equivalent transformation of corresponding strongly connected automata, describe some special subclasses of the class of strongly connected languages and some properties of these subclasses. They will give the possibility to describe algorithms of equivalent transformations for the above-mentioned variant of omega-automata.

Keywords—regular languages, nondeterministic finite automata, subclasses of the regular language class, algorithms for equivalent transformation of automata.

References

- [1] Melnikov B., Melnikova A. *Some more on omega-finite automata and omega-regular languages. Part I: The main definitions and properties* // International Journal of Open Information Technologies. – 2020. – Vol. 8. No. 8. – P. 1–7.
- [2] Büchi J. *On a decision method in restricted second order arithmetic* // In: Proceedings of International Congress on Logic, Method, and Philosophy of Science. – Stanford, Stanford University Press, 1962. – P. 425–435.
- [3] Khoussainov B., Nerode A. *Automata Theory and its Applications* // Progress in Computer Science and Applied Logic, Vol. 21. – Springer, Berlin, 2001.
- [4] Harary F. *Теория графов* // М.: Мир, 1973. *Graph Theory*. Addison Wesley (Boston), 1969, 274 p.
- [5] Melnikov B. *2ω -finite automata and sets of obstructions of their languages* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 1999. – Vol. 6. No. 3. – P. 565–576.
- [6] Skorniyakov L. (Ed.) *General Algebra. Vol. 2* // Moscow: Nauka, 1991. (in Russian)
- [7] 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.
- [8] Melnikov B. *The equality condition for infinite concatenations of two sets of finite words* // International Journal of Foundations of Computer Science. – 1993. – Vol. 4. No. 3. – P. 267–272.
- [9] Melnikov B., Sciarini-Guryanova N. *Possible edges of a finite automaton defining a given regular language* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2002. – Vol. 9. No. 2. – P. 475–485.
- [10] 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.
- [11] Melnikov B., Tsyganov A. *The state minimization problem for nondeterministic finite automata: the parallel implementation of the truncated branch and bound method* // Proceedings – 5th International Symposium on Parallel Architectures, Algorithms and Programming, PAAP-2012. – 2012. – P. 194–201.
- [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] Dolgov V., Melnikov B. *The 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)
- [15] Dolgov V., Melnikov B. *The construction of a universal finite automaton. II. Examples of the work of the algorithms* // Bulletin of the Voronezh state University. Series: Physics. Mathematics. – 2014. – No. 1. – P. 78–85. (in Russian)
- [16] Lombardy S., Sakarovitch J. *The universal automaton* // Logic and Automata, Amsterdam University Press. – 2008. – P. 457–504.
- [17] Dedekind R. *Über Zerlegungen von Zahlen durch ihre größten gemeinsamen Teiler* // Gesammelte Werke 2, Braunschweig. – 1897. – S. 103–148. (in German)
- [18] Korshunov A. *On the number of monotone Boolean functions* // Problems of Cybernetics. – 1981. – Vol. 38. – P. 5–108. (in Russian)
- [19] Salomaa A. *Jewels of formal language theory* // Maryland, USA: Computer Science Press, Inc., 1981.
- [20] Vinogradov I. (Ed.) *Mathematical Encyclopedia. Vol. 2* // Moscow: Sovetskaya Encyclopedia, 1979. (in Russian)
- [21] Prohorov Yu. (Ed.) *Mathematical Encyclopedic Thesaurus* // Moscow: Sovetskaya Encyclopedia, 1988. (in Russian)

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