

Расширенный базисный конечный автомат. Часть II. Описание вспомогательных языков и некоторые свойства

Б. Ф. Мельников, А. А. Мельникова

Аннотация—Во второй части настоящей статьи мы продолжаем рассмотрение специального расширения класса недетерминированных конечных автоматов. Мы на примерах показываем следующий факт: существуют регулярные языки, в базисных автоматах которых у некоторых вершин нет циклов, соответствующих циклам соответствующих вершин других эквивалентных автоматов; однако именно такие циклы имеются в аналогичных вершинах расширенных базисных автоматов.

Также мы продолжаем рассмотрение расширенного базисного конечного автомата. Мы доказываем, что для такого автомата все получаемые в построениях вспомогательные языки регулярны – этот факт является основным предметом части II. Всё перечисленное позволяет упростить само определение расширенного базисного автомата.

Кроме того, мы показываем возможность применения расширенных конечных автоматов для доказательства утверждений, связанных с «обычными» недетерминированными конечными автоматами, а также обратную возможность – применение «обычных» автоматов для доказательства утверждений, связанных с расширенными базисными автоматами.

В заключении статьи мы описываем возможности применения рассмотренного нами формализма – в частности, применения расширенных автоматов в алгоритмах минимизации недетерминированных конечных автоматов.

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

VIII. ВВЕДЕНИЕ В ЧАСТЬ II

Настоящая часть II – продолжение части I, т.е. [1]. Мы продолжаем нумерацию разделов, утверждений, формул и рисунков – однако «сбрасываем» нумерацию сносок и библиографически ссылок.

В первой части статьи мы рассмотрели определение расширенных конечных автоматов, привели некоторые примеры и определили расширенный базисный конечный автомат. Мы также показали адекватность его определения – т.е. утверждение о том, что построенный на основе исходного регулярного языка расширенный базисный автомат действительно определяет этот язык. На одном из примеров (который будет далее продолжен) мы показали следующий факт: существуют регулярные языки, в базисных автоматах которых [2] у некоторых

вершин нет циклов, соответствующих циклам соответствующих вершин других эквивалентных автоматов; однако именно такие циклы имеются в аналогичных вершинах *расширенных* базисных автоматов.

По поводу материала, приведённого в [1], стоит ещё раз отметить следующее. Мы использовали два разных варианта употребления слова «соответствующие» – а именно, применительно к циклам и состояниям рассматриваемых автоматов. Однако можно сказать, что в обоих этих случаях мы рассматриваем соответствие с объектами базисного автомата, см. [3], [4], [5] и др.¹

В названии настоящей части II употреблено слово «описание»; при этом мы имеем в виду, что можно задать (т.е. описать) *регулярные* языки, определяющие

$$\zeta_{\mathcal{E}(L)}, \zeta_{\mathcal{E}(L)}^{in} \text{ и } \zeta_{\mathcal{E}(L)}^{out}.$$

(Однако, как будет сказано ниже, о *конструктивном* построении соответствующих регулярных языков речь пока не идёт. Такое построение – предмет дальнейших исследований.)

Структура второй части статьи такова. В разделе IX мы рассматриваем одно из утверждений об эквивалентном преобразовании «обычных» недетерминированных конечных автоматов. Это утверждение (утверждение 5) уже было рассмотрено нами ранее, например в [6], однако применение расширенных автоматов даёт очень краткий способ его доказательства. И наоборот, само утверждение 5 применяется в разделе X в качестве вспомогательного – для доказательства основного утверждения этого раздела, о возможности добавления в граф переходов «обычного» недетерминированного автомата некоторой петли; при этом, как обычно в наших публикациях, сформулированные при этом условия делают такое преобразование эквивалентным. (В части I уже отмечалось, что подобное добавление не всегда является «усложнением» автомата, более того, иногда его можно назвать упрощением структуры его графа переходов.)

¹ По-видимому, говоря про состояния, слово «соответствующие» стоит рассматривать в качестве *прилагательного*: мы *сразу* (без дополнительных построений) имеем определение вершин, соответствующих некоторой вершине базисного автомата. (Конечно, при этом имеется в виду, что функции разметки состояний уже построены – однако мы это неявно предполагаем во всех наших публикациях, если специально не отмечаем иного.)

И наоборот, говоря про циклы, слово «соответствующие» стоит рассматривать в качестве *причастия*: для определения цикла, соответствующего некоторому циклу базисного автомата, необходимо *выполнить некоторые вспомогательные построения*, т.е. действия.

(Прилагательное – собственный признак предмета, причастие – признак предмета по действию. Прилагательное образуется от существительного, причастие – от глагола.)

Статья получена 29 февраля 2020 г.

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

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

В разделе XI мы приводим несколько новых примеров, а именно, рассматриваем автоматы, построенные на основе «обычных» базисных автоматов (т.е. не расширенных базисных автоматов) путём эквивалентных преобразований, причём возможность применения таких преобразований следует из доказанных в настоящей статье утверждений.

В разделе XII показывается то, что все вспомогательные языки, применяемые при описании расширенного базисного автомата, являются регулярными. Однако, как мы уже отмечали, мы пока не приводим конструктивного доказательства этого факта – т.е. не формулируем сами алгоритмы построения таких языков². А в разделе XIII мы рассматриваем ещё один пример, связанный с приведённым в статье материалом.

В заключении (раздел XIV) мы кратко перечисляем полученные результаты и формулируем направления дальнейшей работы, связанные с рассмотренными в настоящей статье расширенными конечными автоматами.

IX. ПРИМЕНЕНИЕ РАСШИРЕННЫХ АВТОМАТОВ ПРИ ДОКАЗАТЕЛЬСТВЕ УТВЕРЖДЕНИЙ

Немного уточняя название раздела можно сказать, что мы показываем пример применения расширенных автоматов при доказательстве утверждений для обычных недетерминированных конечных автоматов.

В начале раздела приведём доказательство утверждения, которое кажется очевидным; однако в наших предыдущих публикациях были приведены только либо схема доказательства, либо неполное доказательство (например, см. [6], причём на английском языке). Более того, известное авторам полное доказательство очень громоздко. Здесь мы приводим полное доказательство утверждения. Оно выполненное с применением изложенной в настоящей статье теории расширенных конечных автоматов – и, по-видимому, его нельзя назвать громоздким. После этого мы показываем возможность его применения – причём также для расширенных автоматов, но для других связанных с ними вопросов.

Утверждение 5: Пусть для автомата (1) и некоторых двух его состояний $q, q' \in Q$ выполняются следующие условия:

$$\varphi_K^{in}(q') \subseteq \varphi_K^{in}(q) \quad \text{и} \quad \varphi_K^{out}(q') \subseteq \varphi_K^{out}(q); \quad (11)$$

кроме того,

$$\varphi_K^{in}(q') \neq \emptyset \quad \text{и} \quad \varphi_K^{out}(q') \neq \emptyset. \quad (12)$$

Тогда автомат $K' = \mathcal{J}^{qq'}(K)$ эквивалентен K .³

Кроме того,

$$\varphi_K^{in}(q) = \varphi_{K'}^{in}(q) \quad \text{и} \quad \varphi_K^{out}(q) = \varphi_{K'}^{out}(q).$$

Доказательство. Рассмотрим некоторое слово

$$v \in \mathcal{L}_K^{io}(q, q').$$

² Такие алгоритмы могли бы быть сформулированы, например, путём описания соответствующих регулярных выражений, либо с помощью «обычных» недетерминированных автоматов

³ Заметим, что если не выполняется хотя бы одно из условий (12), то и сформулированное утверждение не выполняется. Соответствующий пример может быть описан достаточно просто (подробный пример мы приводить не будем) – нужно рассмотреть автомат, имеющий состояния, являющиеся одновременно бесполезными и недостижимыми. При этом каждое из таких состояний может быть рассмотрено как состояние q' в (11) и (12).

Согласно свойствам канонического автомата \tilde{L} и определению его функции φ_K^{in} , мы получаем, что

$$(\forall A \in \varphi_K^{in}(q)) (\exists A' \in \varphi_K^{in}(q')) (A \xrightarrow[\tilde{L}]{v} A').$$

(Последнее – просто следствие того, что автомат \tilde{L} является детерминированным.)

Теперь рассмотрим расширенный автомат из определения 4, сконструированный на основе (1), где:

- для рассматриваемого состояния q полагаем $\zeta(q) = \{v, \varepsilon\}$; а для любого другого состояния $r \in Q$ полагаем $\zeta(r) = \{\varepsilon\}$;
- для каждого состояния $s \in S$ полагаем $\zeta^{in}(s) = \{\varepsilon\}$; а для любого другого состояния $r \in Q$ полагаем $\zeta^{in}(r) = \emptyset$;
- для каждого состояния $f \in F$ полагаем $\zeta^{out}(f) = \{\varepsilon\}$; а для любого другого состояния $r \in Q$ полагаем $\zeta^{out}(r) = \emptyset$.

Поэтому соответствующий расширенный автомат, содержащий для состояния q цикл с пометкой v ,⁴ эквивалентен исходному автомату.

Аналогичный факт для слова

$$v \in \mathcal{L}_K^{io}(q', q)$$

получается тем же самым способом, но вместо функции φ_K^{in} используется φ_K^{out} . А оставшаяся часть доказательства получается стандартным образом, т.е. путём рассмотрения произвольного допускающего пути автомата K' и пометок его циклов (это может быть выполнено полностью аналогично индукции, приведённой в доказательстве теоремы 1). \square

X. АЛГОРИТМЫ ДОБАВЛЕНИЯ ПЕТЛИ

Теперь покажем возможность применения последнего утверждения, причём также для расширенных автоматов. Однако мы рассматриваем не объединение вершин (как в утверждении 5) – а добавление элементов в определение автомата; при таком добавлении мы, аналогично предыдущим публикациям, рассматриваем каждый из элементов пятёрки (1) в качестве множества.

Как мы уже отмечали в части I, если мы добавим слово u в язык $\zeta_{\mathcal{E}(L)}$ (т.е. если мы таким образом дополним определение 4, то очевидно, что мы получим автомат, эквивалентный заданному; этот факт можно рассматривать, например, как следствие определений 1 и 4. При этом выполняется и более сложное утверждение – следующее.

Следующее утверждение можно рассматривать как следствие утверждения 5.

Утверждение 6: Пусть для автомата (1) и двух его состояний $q, q' \in Q$ выполнены условия (11) и (12). Пусть также

$$v \in \mathcal{L}_K^{io}(q, q') \cup \mathcal{L}_K^{io}(q', q).$$

Тогда

$$v \in \zeta_{\mathcal{E}(L)}(q). \quad \square$$

Последнее утверждение может быть использовано для добавления цикла к состоянию q заданного автомата (1) –

⁴ То есть автомат из определения 4 с «добавленным» циклом v . Отметим, что в наших построениях этот цикл можно называть петлёй.

а именно, цикла, имеющего пометку v . (Автомат, полученный объединением q и q' , содержит такой цикл.)

Также очень важно отметить, что утверждение 6 можно использовать в качестве своеобразного шаблона для формулировки возможностей добавления цикла. На основе него могут быть получены различные утверждения-следствия, описывающие различные варианты выполнения условий (11) и (12). Мы в настоящей работе не будем формулировать такие утверждения-следствия – ограничившись лишь двумя конкретными примерами, приведёнными в следующем разделе.

XI. НОВЫЕ ПРИМЕРЫ (АВТОМАТЫ, ПОСТРОЕННЫЕ НА ОСНОВЕ БАЗИСНЫХ)

Продолжим рассмотрение примера, приведённого в части I (раздел V). При этом, поскольку мы в нём рассматриваем базисный автомат, то, согласно классификации, приведённой нами в [7], [8], мы для всех «естественных» значений букв, состояний, множеств состояний и реально существующих пометок циклов (т.е., например, $a \in \Sigma$, $q \in Q$, $\alpha \subseteq Q_\pi$, $\beta \subseteq Q_\rho$, $\psi \subseteq Q_\pi \times Q_\rho$ и т.п.) можем употреблять названия «соответствующие циклы», « A_X -cyclic states», « ψ -proper loops» и т.п.⁵ Напомним, что на этом примере мы показали следующий важный факт: существуют регулярные языки, в базисных автоматах которых у некоторых вершин нет циклов, соответствующих циклам некоторых соответствующих вершин некоторых других эквивалентных автоматов⁶.

Итак, продолжим рассматривать пример регулярного выражения (6). На основе показанного выше факта

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

и теории из этой главы, мы можем к состоянию (B, Y) автомата $\mathcal{B}(L)$ добавить цикл (или петлю) с пометкой ab . Для этого, например, мы можем добавить к автомату $\mathcal{B}(L)$ новое состояние, пусть q , а также дуги

$$\delta((B, Y), a) \ni q \text{ и } \delta(q, b) \ni (B, Y).$$

Согласно вышеизложенному, получающийся автомат (рис. 4) определяет тот же самый регулярный язык⁷.

Важно отметить, что эквивалентность автоматов (последнего и приведённого в части I на рис. 1) мы показали без построения эквивалентных канонических автоматов.

К последнему замечанию можно «придаться», т.е. возразить, сказав, что базисный автомат нельзя построить без построения функций разметки и бинарного отношения $\#$, т.е. фактически построения эквивалентного канонического автомата – что, вообще говоря, требует экспоненциального времени. Однако:

⁵ В предыдущих публикациях мы не для всех подобных терминов приводили возможные русские названия.

⁶ Мы здесь уточнили формулировку, приведённую в части I.

⁷ При этом, конечно, получаемый автомат не будет являться $\mathcal{B}(L)$. Более того, для дальнейшего изложения важно отметить и такой факт: после подобного добавление цикла мы не можем получить базисный конечный автомат ни для какого регулярного языка. (Простейшее объяснение этого факта следующее: подобное добавление цикла не изменяет язык рассматриваемого автомата, т.е. его базисный автомат – тот же самый.)

Также отметим, что подобное преобразование может изменять функции φ_K^m и φ_K^{out} – в том числе для уже существовавших в автомате состояний. Однако подробные примеры на эту тему рассматривать не будем.

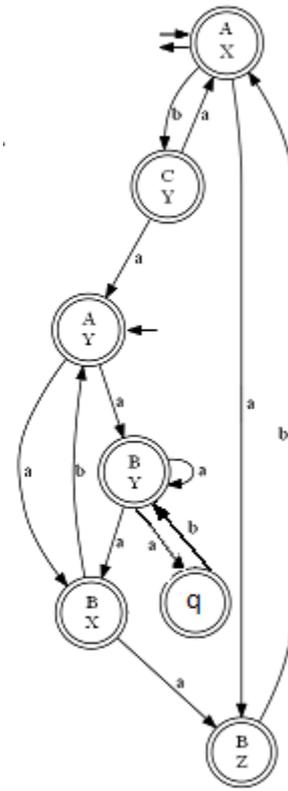


Рис. 4. Базисный автомат для языка, определяемого РВ (6), с добавленными элементами (состоянием q и двумя дугами)

- во-первых, мы строим указанные объекты (функции разметки и др.) во всех наших построениях – это общие предварительные действия во всех наших статьях;
- во-вторых, при применении теории, приведённой в настоящей статье эквивалентные канонический и, при необходимости, базисный автоматы строятся только один раз – а именно, для исходного автомата; в то же время стандартная проверка эквивалентности автоматов (равенства соответствующих регулярных языков) требует построения двух соответствующих канонических автоматов.

Рассмотрим ещё один пример – в котором мы даже не будем увеличивать число состояний. А именно, в рассмотренный выше базисный автомат добавим дугу (петлю)

$$\begin{matrix} A \\ Y \end{matrix} \xrightarrow{a} \begin{matrix} A \\ Y \end{matrix}.$$

При этом тот факт, что

$$\zeta_{\mathcal{B}(L)}((A, Y)) \ni a$$

также следует из материала части I. Получаем следующий автомат (рис. 5, ниже), также эквивалентный исходному.

Итак, в этом разделе мы описали некоторые возможные примеры эквивалентного преобразования недетерминированных конечных автоматов – примеры, реализуемые на основе приведённой в настоящей статье теории.

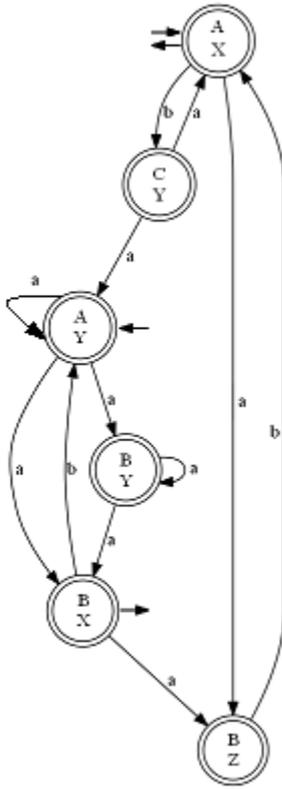


Рис. 5. Базисный автомат для языка, определяемого РВ (6), с добавленной A_Y -петлей

XII. РЕГУЛЯРНОСТЬ ВСПОМОГАТЕЛЬНЫХ ЯЗЫКОВ

«С первого взгляда» может показаться, что регулярность языков

$$\zeta_{\mathcal{B}\mathcal{E}(L)}(A_X), \zeta_{\mathcal{B}\mathcal{E}(L)}^{in}(A_X) \text{ и } \zeta_{\mathcal{B}\mathcal{E}(L)}^{out}(A_X)$$

(для любого возможного состояния A_X базисного автомата $\mathcal{B}\mathcal{A}(L)$) очевидна. Однако это далеко не так – и возможные трудности можно увидеть из рассмотренных нами выше примеров. Постараемся прокомментировать эти возможные трудности доказательства более подробно.

Естественное предположение таково. Для некоторого $u \in \Sigma^*$ условие

$$\zeta_{\mathcal{B}\mathcal{E}(L)}(A_X) \ni u$$

выполнено тогда и только тогда, когда

$$(\forall v \in \mathcal{L}_{\mathcal{B}\mathcal{A}(L)}^{in}(A_X)) (\forall w \in \mathcal{L}_{\mathcal{B}\mathcal{A}(L)}^{out}(A_X)) (\forall i \in \mathbb{N}_0) (vu^i w \in L)$$

(где $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$). То же самое предположение можно записать и без использования базисного автомата, т.е. на основе двух канонических автоматов:

$$(\forall v \in \mathcal{L}_L^{in}(A)) (\forall w' \in \mathcal{L}_L^{in}(X)) (\forall i \in \mathbb{N}_0) (vu^i (w')^R \in L).$$

И, конечно, аналогичные предположения могут быть сделаны и для языков

$$\zeta_{\mathcal{B}\mathcal{E}(L)}^{in}(A_X) \text{ и } \zeta_{\mathcal{B}\mathcal{E}(L)}^{out}(A_X).$$

Понятно, что подобные условия определяют *регулярные* языки – что можно несложно показать, построив соответствующие регулярные выражения либо вспомогательные конечные автоматы.

Однако у авторов нет ни доказательств (того, что эти условия действительно формируют нужные языки), ни контрпримеров к этому утверждению. Сложность возможного доказательства связана с тем, что сформулированные предположения выполнены для автомата $\mathcal{B}\mathcal{A}(L)$ (либо пары соответствующих канонических автоматов \tilde{L} и \tilde{L}^R) – в то время как их *надо было бы* сформулировать для расширенного автомата $\mathcal{B}\mathcal{E}(L)$; мы же такой автомат ещё не построили. Иными словами, некоторое слово, входящее в язык $\zeta_{\mathcal{B}\mathcal{E}(L)}(A_X)$ для некоторого состояния A_X расширенного базисного автомата, *может влиять*⁸ на язык $\zeta_{\mathcal{B}\mathcal{E}(L)}(B_Y)$ любого другого состояния B_Y того же самого расширенного базисного автомата (т.е. может входить в некоторые слова этого языка в качестве под-слова).

Возможен и другой подход к доказательству регулярности этих языков; например, для языка

$$\zeta_{\mathcal{B}\mathcal{E}(L)}(A_X)$$

он заключается в следующем. Для всех возможных недетерминированных конечных автоматов (пусть некоторый такой автомат – K), определяющих заданный регулярный язык L , нужно рассмотреть все вершины (пусть q), такие что

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

после этого нужно рассмотреть язык, являющийся объединением пометок всех возможных циклов:

$$\bigcup_{K,q} \mathcal{L}_K^{io}(q). \tag{13}$$

Однако применение такой формулы не приводит к успеху⁹: имеются тривиальные примеры объединения бесконечного множества регулярных языков, дающего нерегулярный язык. Например, над алфавитом $\{a, b\}$ таким языком является

$$\bigcup_{i \geq 0} \{a^i b^i\}.$$

Но, несмотря на эти проблемы, *небольшая* модификация второго подхода к успеху приводит, т.е., как мы уже отмечали, можно доказать, что языки

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

регулярны. Простой вариант доказательства этого факта получается следующим образом¹⁰.

⁸ Должно влиять!

⁹ Точнее, её запись нельзя считать полным доказательством регулярности рассматриваемого языка.

¹⁰ Ниже приведена *схема* доказательства. Строго мы доказательство оформлять не будем – по следующим двум причинам:

- во-первых, схема доказательства практически совпадает с самим доказательством: эквивалентность преобразований, заключающихся в добавлении дуг, очевидна – но при этом строгое рассмотрение всех возможных допускающих путей рассматриваемых автоматов, громоздко и, по-видимому, малоинтересно;
- во-вторых, как мы уже отмечали, такое доказательство *неконструктивно* – и мы собираемся в следующих публикациях рассмотреть конструктивный вариант (т.е. доказательство-алгоритм).

Можно считать, что для каждого из «объединяемых» автоматов (точнее – автоматов K , рассматриваемых в формуле (13)) выполнены следующие два условия.

- Во-первых, в нём отсутствуют пары состояний (пусть q и q'), для которых выполнялись бы «входные условия» утверждения 5 (т.е. (11) и (12)). Действительно, в противном случае можно каждый рассматриваемый автомат K заменить на эквивалентный ему автомат

$$K' = \mathcal{J}^{qq'}(K).$$

При этом, как показано в том же утверждении 5, мы «не уменьшаем» значения функций разметки – т.е. выполняются условия

$$\varphi_{K'}^{in}(q) \supseteq \varphi_K^{in}(q) \quad \text{и} \quad \varphi_{K'}^{out}(q) \supseteq \varphi_K^{out}(q).$$

- Во-вторых, согласно материалу [4], [9], [10], мы можем рассматривать *все* возможные дуги автомата $\mathcal{COM}(L)$ (фактически совпадающего с универсальным автоматом), соединяющие любые две вершины автомата K' . При этом очевидно, что для любого из «объединяемых» согласно (13) автоматов мы рассматриваем *все* необходимое множество циклов $\mathcal{L}_K^{io}(q)$.

Таким способом мы ограничиваем число рассматриваемых автоматов (их заведомо конечное число) – что даёт возможность применения формулы (13).

ХIII. ЕЩЁ ОДИН ПРИМЕР

Рассмотрим ещё один пример. Сначала может показаться, что он мало связан с приведённой в настоящей статье теорией – однако это не так: пример показывает возможность *априорного* рассмотрения двух (или более) букв рассматриваемого алфавита, которые могут быть отнесены в один класс эквивалентности¹¹. Такое объединение в один класс позволяет уменьшать количество простых циклов базисного автомата, необходимых для формирования языков

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

фактически представляющих собой *основной предмет настоящей статьи*.

Рассматриваемый пример является «развитием» одного из приведённых выше – т.е. (6); неформально он может быть получен из базисного автомата для (6) путём «удвоения» состояния q_3 , затем «удвоения» буквы a , а затем удаления некоторых дуг. (Очень кратко подобные алгоритмы описывались в [11], а похожие алгоритмы, но более подробно, – в [9]. Но эта тема – далеко не «закрота», мы предполагаем вернуться к описанию подобных алгоритмов.)

Итак, рассмотрим такой автомат – рис. 6 ниже. Алфавит – $\{a_1, a_2, b\}$. Важно отметить, что $\delta(q'_3, a_2) \not\equiv q_1$.

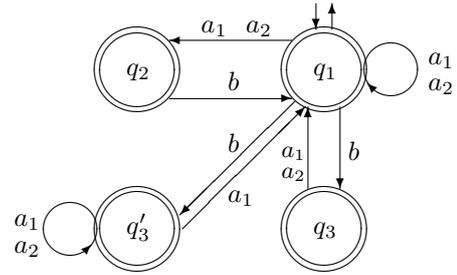


Рис. 6. Новый пример

Этот автомат определяет язык, соответствующий регулярному выражению

$$((a_1 + a_2) + (a_1 + a_2)b + b(a_1 + a_2))^*.$$

Эквивалентный канонический автомат приведён на рисунке 7:

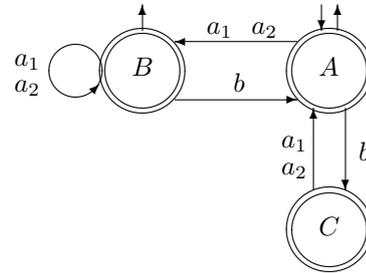


Рис. 7. Канонический автомат для рассматриваемого языка

Аналогично языку (6), для этого языка (пусть L) выполнено равенство $L^R = L$; поэтому можно сказать, что автомат L^R также изображён на рис. 7, где состояния A , B и C заменены соответственно на X , Y и Z . Функция φ_K^{in} получается такая:

$$\begin{aligned} \varphi_K^{in}(q_1) &= \{A, B\}, & \varphi_K^{in}(q_2) &= \{B\}, \\ \varphi_K^{in}(q_3) &= \{A, C\}, & \varphi_K^{in}(q'_3) &= \{A, B, C\}; \end{aligned}$$

а у функции φ_K^{out} имеются совпадающие значения:

$$\begin{aligned} \varphi_K^{out}(q_1) &= \{X, Y\}, & \varphi_K^{out}(q_2) &= \{X, Z\}, \\ \varphi_K^{out}(q_3) &= \varphi_K^{out}(q'_3) &= \{Z\}. \end{aligned}$$

Поскольку

$$\delta(q'_3, a_2) \not\equiv q_1 \quad \text{и} \quad \delta(q_3, a_2) \not\equiv q_3,$$

мы не можем записать включения для языков

$$\mathcal{L}_K^{out}(q_3) \quad \text{и} \quad \mathcal{L}_K^{out}(q'_3) \quad \text{—}$$

в то время как для языков

$$\mathcal{L}_K^{in}(q_3) \quad \text{и} \quad \mathcal{L}_K^{in}(q'_3)$$

аналогичный факт очевиден. Однако, поскольку выполнены условия

$$\varphi_K^{in}(q_3) \subset \varphi_K^{in}(q'_3) \quad \text{и} \quad \varphi_K^{out}(q_3) \subseteq \varphi_K^{out}(q'_3)$$

(специально отметим, что в первом случае включение получается собственным), мы можем использовать утверждение 5 и объединить состояния q_3 и q'_3 .

¹¹ В настоящий момент принята в печать статья одного из авторов, в которой рассматриваются подобные классы эквивалентности, рассматриваемые на множестве букв.

XIV. ЗАКЛЮЧЕНИЕ

Итак, легко видеть, что настоящая статья является *существенным расширением* публикации 2010 года [12]. Мы рассмотрели определение расширенных конечных автоматов, определили расширенный базисный конечный автомат и рассмотрели некоторые его свойства, привели различные примеры. Мы также описали подробную схему доказательства того, что для расширенного базисного автомата все получаемые в построениях вспомогательные языки являются регулярными.

Кроме того, мы доказали адекватность определения расширенного базисного конечного автомата — т.е. утверждение о том, что построенный на основе исходного регулярного языка расширенный базисный автомат действительно определяет этот язык. На примерах языков и соответствующих автоматов мы доказали следующий факт: существуют регулярные языки, в базисных автоматах которых у некоторых вершин нет циклов, соответствующих циклам соответствующих вершин других эквивалентных автоматов; при этом именно такие циклы имеются в аналогичных вершинах расширенных базисных автоматов.

Основные направления дальнейшей работы, связанные с рассмотренными в настоящей статье расширенными конечными автоматами, таковы. Мы предполагаем в следующих публикациях рассмотреть *конструктивный* вариант описания регулярных языков

$$\zeta_{\mathcal{E}(L)}, \zeta_{\mathcal{E}(L)}^{\text{in}} \text{ и } \zeta_{\mathcal{E}(L)}^{\text{out}}$$

(т.е. привести соответствующий алгоритм). При этом, *возможно*, удастся доказать и правильность сделанного предположения для первого из этих языков (и аналогичных предположений для двух остальных языков) — в разделе XII мы называли такое возможное доказательство первым подходом.

В первой части мы уже отмечали, что алгоритмы, которые могут быть созданы на основе добавления циклов, на первый взгляд вряд ли могут быть названы «алгоритмами упрощения» автомата. Однако во многих задачах преобразования конечных автоматов эти алгоритмы могут быть *вспомогательными*: мы «временно ухудшаем» структуру автомата чтобы впопыхах её существенно «улучшить»; такое «улучшение» заключается в удалении некоторых состояний автомата и/или его дуг — причём этого нельзя было сделать без описываемого нами добавления цикла. Подобное удаление состояний автомата является одним из этапов алгоритмов вершинной минимизации.

Кроме того, с расширенными конечными автоматами связаны алгоритмы, в которых некоторое подмножество алфавита рассматривается «как одна буква»; подобный пример был рассмотрен в разделе XIII. В будущем мы предполагаем описать достаточные условия того, что с некоторым подмножеством алфавита можно так поступать — и понятно, что описываемые на основе этих условий алгоритмы преобразований автомата также можно считать алгоритмами упрощения структуры рассматриваемого автомата¹².

¹² Отметим, что здесь, в отличие от большинства наших работ, рассматриваются *неэквивалентные* преобразования.

Список литературы

- [1] Melnikov B., Melnikova A. *Extended basis finite automaton. Part I. The basic definitions* // International Journal of Open Information Technologies. — 2018. — Vol. 7. No. 10. — P. 1–8.
- [2] 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.
- [3] Melnikov B., Vakhitova A. *Some more on the finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). — 1998. — Vol. 5. No. 3. — P. 495–506.
- [4] 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.
- [5] Мельников Б. *Регулярные языки и недетерминированные конечные автоматы* // М.: Изд-во Российского государственного социального университета, 2018. — 180 с.
- [6] Melnikov B. *Once more on the edge-minimization of nondeterministic finite automata and the connected problems* // Fundamenta Informaticae. — 2010. — Vol. 104. No. 3. — P. 267–283.
- [7] Melnikov B., Melnikova A. *An approach to the classification of the loops of finite automata. Part I: Long corresponding loops* // International Journal of Open Information Technologies. — 2018. — Vol. 6. No. 9. — P. 9–14.
- [8] Melnikov B., Melnikova A. *An approach to the classification of the loops of finite automata. Part II: The classification of the states based on the loops* // International Journal of Open Information Technologies. — 2018. — Vol. 6. No. 11. — P. 1–6.
- [9] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. — 2017. — Vol. 5. No. 10. — P. 9–17.
- [10] Melnikov B., Melnikova A. *Edge-minimization of non-deterministic finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). — 2001. — Vol. 8. No. 3. — P. 469–479.
- [11] Мельников Б., Сайфуллина М. *О некоторых алгоритмах эквивалентного преобразования недетерминированных конечных автоматов* // Известия высших учебных заведений. Математика. — 2009. — № 4. — С. 67–72.
- [12] Melnikov B. *Extended nondeterministic finite automata* // Fundamenta Informaticae. — 2010. — Vol. 104. No. 3. — P. 255–265.

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.

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

Extended basis finite automaton. Part II. Description of auxiliary languages and some properties

Boris Melnikov, Aleksandra Melnikova

Abstract—In the second part of this paper, we continue to consider a special extension of the class of nondeterministic finite automata. We show by examples the following fact: there are regular languages, in the basis automata of which, some vertices do not have loops corresponding to the loops of the corresponding vertices of other equivalent automata; however, precisely such loops do exist at similar vertices of extended basic automata.

We also continue to consider the extended basic finite automaton. We prove that for such an automaton, all auxiliary languages obtained in the constructions are regular; this fact is the main subject of Part II. All of the above allows us to simplify the definition of the extended basic automaton.

In addition, we show the possibility of using extended finite automata for proving statements related to “ordinary” nondeterministic finite automata, as well as the inverse possibility of using “ordinary” automata to prove statements related to extended basic automata.

At the end of the paper, we describe the possibilities of applying this formalism, in particular, the use of extended automata in algorithms for minimizing non-deterministic finite automata.

Keywords—Regular languages, non-deterministic finite automata, extension of automata, algorithms of equivalent transformation of automata.

References

- [1] Melnikov B., Melnikova A. *Extended basis finite automaton. Part I. The basic definitions* // International Journal of Open Information Technologies. – 2018. – Vol. 7. No. 10. – P. 1–8.
- [2] 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.
- [3] Melnikov B., Vakhitova A. *Some more on the finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 1998. – Vol. 5. No. 3. – P. 495–506.
- [4] 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.
- [5] Melnikov B. *Regular languages and nondeterministic finite automata* // Moscow: Russian State Social University Ed., 2018. – 180 p. (in Russian)
- [6] Melnikov B. *Once more on the edge-minimization of nondeterministic finite automata and the connected problems* // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 267–283.
- [7] Melnikov B., Melnikova A. *An approach to the classification of the loops of finite automata. Part I: Long corresponding loops* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 9. – P. 9–14.
- [8] Melnikov B., Melnikova A. *An approach to the classification of the loops of finite automata. Part II: The classification of the states based on the loops* // International Journal of Open Information Technologies. – 2018. – Vol. 6. No. 11. – P. 1–6.
- [9] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [10] Melnikov B., Melnikova A. *Edge-minimization of non-deterministic finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). – 2001. – Vol. 8. No. 3. – P. 469–479.
- [11] Melnikov B., Sayfullina M. *On some equivalent transformation algorithms for nondeterministic finite automata* // Higher Education News. Mathematics. – 2009. – No. 4. – P. 67–72. (in Russian)
- [12] Melnikov B. *Extended nondeterministic finite automata* // Fundamenta Informaticae. – 2010. – Vol. 104. No. 3. – P. 255–265.

Boris Feliksovich 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).

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