

Упрощённые регулярные языки и специальное отношение эквивалентности на классе регулярных языков. Часть II

Б. Ф. Мельников, В. Н. Долгов

Аннотация—Рассматриваемое в статье отношение эквивалентности S на классе регулярных языков необходимо для более полного исследования ранее определённого в наших статьях отношения R . Кроме этого, мотивация к рассмотрению отношения S заключается в необходимости его применения для исследования т.н. лепестковых автоматов – которое также было начато в одной из наших предыдущих статей. Конкретно, для получения лепесткового автомата необходим такой алгоритм неэквивалентного преобразования автомата, как объединение двух букв рассматриваемого алфавита, – и с этим преобразованием связаны оба упомянутых отношения эквивалентности на классе регулярных языков, R и S . В свою очередь лепестковые автоматы возникают при описании нескольких различных алгоритмов проверки бинарного отношения эквивалентности в бесконечности, также рассмотренного в нескольких наших предыдущих статьях, в частности – при использовании в этих алгоритмах конечных автоматов $PR1$ и $NSPRI$.

Итак, в настоящей статье мы определяем S – специальное бинарное отношение на множестве регулярных языков, показываем выполнение всех свойств отношения эквивалентности – и поэтому отношение S разделяет весь класс регулярных языков на непересекающиеся подклассы. Вследствие этого большинство рассматриваемых в теории языков задач можно решать только для одного представителя каждого такого подкласса – причём обычно желательно рассматривать так называемый упрощённый язык, он один в каждом подклассе.

Понятие упрощённого языка основано на соединении т.н. параллельных букв, и такому упрощённому языку соответствует специально вводимый нами упрощённый конечный автомат. Всё сказанное позволяет работать лишь с конечным числом рассматриваемых регулярных языков, при условии, что соответствующие им конечные автоматы имеют априори ограниченное число состояний.

Оба эти отношения эквивалентности, S и R , сохраняют отношение $\#$, определяемое для конкретного регулярного языка, – и, следовательно, позволяют использовать многие доказанные ранее свойства регулярных языков и недетерминированных конечных автоматов с любым из автоматов его класса эквивалентности и соответствующим ему языком.

В представляемой части II мы продолжаем рассмотрение свойств отношения S – существенно более сложных свойств, чем рассматривались в части I. Среди прочего мы приводим варианты применения отношения параллельности букв в нескольких задачах теории формальных языков.

Ключевые слова—регулярные языки, недетерминированные конечные автоматы, бинарные отношения, отношение эквивалентности, упрощение автоматов, алгоритмы.

Статья получена 10 октября 2022 г.

Борис Феликсович Мельников, Университет МГУ – ППИ в Шэньчжэне (bormel@smbu.edu.cn).

Василий Николаевич Долгов, Ульяновский государственный технический университет (terenga74@mail.ru).

Предлагаемая часть II настоящей статьи является продолжением части I, т.е. [1]. Мы рассматриваем более сложные свойства специального отношения эквивалентности на классе регулярных языков. Нумерация определений и утверждений продолжается, а нумерация ссылок на использованные источники – новая, начинается с 1.

V. БОЛЕЕ СЛОЖНЫЕ СВОЙСТВА ОТНОШЕНИЯ ПАРАЛЛЕЛЬНОСТИ БУКВ И ПОНЯТИЕ УПРОЩЁННОГО АВТОМАТА

В этом разделе мы продолжаем рассматривать свойства отношения параллельности букв – применительно к детерминированным и, в частности, каноническим автоматам. На основе этих свойств в разделе вводится понятие упрощённого автомата – и уже для него формулируется одно из его свойств¹.

В начале приведём два важных утверждения, 4 и 5. Первое из них мы будем использовать для последующего определения *упрощённого* автомата, а второе нужно для рассмотрения отношения $\#$, относящегося к языку такого автомата.

Утверждение 4: Пусть \tilde{K} – канонический автомат для заданного регулярного языка $L \subseteq \Sigma^*$. Тогда для каждой пары букв $a, b \in \Sigma$ выполнено следующее:

$$a \parallel^L b \iff a \parallel^{\tilde{K}} b.$$

Доказательство.

\Rightarrow По определению 1 и утверждению 1 (часть I);

\Leftarrow По определению 1. \square

Таким образом, параллельность букв в языке $a \parallel^L b$ однозначно определяется автоматом \tilde{K} .

Утверждение 5: Пусть L – некоторый регулярный язык над алфавитом Σ . Тогда для любой пары букв $a, b \in \Sigma$ выполнено следующее:

$$a \parallel^L b \iff a \parallel^{L^R} b,$$

т.е. параллельность $a \parallel^L b$ выполняется и для зеркального языка.

¹ То есть здесь мы применяем обычный, «прямой» порядок определений: сначала – для автоматов, а потом – для соответствующих языков. (Напомним, что в разделе III части I был использован «обратный порядок» – о чём там была соответствующая сноска.)

Доказательство.

⇒ Пусть $a \parallel^L b$, тогда для некоторого определяющего язык L автомата $K = (Q, \Sigma, \gamma, S, F)$ выполнен следующий факт: $a \parallel^K b$. Рассмотрим автомат

$$K^R = (Q, \Sigma, \gamma^R, S, F),$$

определяющий язык L^R . Для него

$$\begin{aligned} \gamma^R(q_1, q_2) = \gamma(q_2, q_1) \ni a &\iff \\ \gamma^R(q_1, q_2) = \gamma(q_2, q_1) \ni b. & \end{aligned}$$

⇐ Обратный факт доказывается аналогично. □

Пример 2: Рассмотрим автомат K , приведённый на рис. 3 (для удобства повторяем этот рисунок из части I),

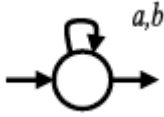


Рис. 3. Канонический автомат \tilde{K} .

также рассмотрим соответствующий язык L . Как мы уже знаем, выполнено следующее:

$$a \parallel^L b \quad \text{и} \quad a \parallel^K b.$$

То же самое выполняется для языка L^R и автомата K^R :

$$a \parallel^{L^R} b \quad \text{and} \quad a \parallel^{K^R} b$$

(в рассматриваемом нами примере они просто совпадают соответственно с L и K).

Итак, мы знаем, что каждый регулярный язык L даёт некоторое (зависящее от этого языка) разбиение заданного алфавита Σ на подмножества параллельных букв. При этом зеркальный регулярный язык L^R , согласно утверждению 5, даёт то же самое разбиение:

$$\Sigma = \bigcup_{i=1}^n \sigma_i,$$

где

$$\sigma_i \neq \emptyset, \quad \sigma_i \cap \sigma_j = \emptyset \quad \text{для} \quad i, j \in \{1, 2, \dots, n\}, \quad i \neq j;$$

будем обозначать это разбиение следующим образом:

$$\bar{\sigma} = \{\sigma_1, \sigma_2, \dots, \sigma_n\}.$$

При этом далее мы будем рассматривать регулярные языки и конечные автоматы над множеством $\bar{\sigma}$, которое можно считать *новым алфавитом*; при этом каждый элемент $\sigma_i \in \bar{\sigma}$ будем рассматривать:

- и в качестве буквы этого нового алфавита $\bar{\sigma}$,
- и, как выше, в качестве подмножества множества букв старого алфавита Σ .

Всюду далее мы предполагаем, что L – заданный регулярный язык над алфавитом Σ , \tilde{K} – канонический автомат² определяющий L , а $\bar{\sigma}$ – разбиение алфавита Σ ,

² Ещё раз отметим, что согласно предыдущим договорённостям возможно применение таких обозначений:

$$\tilde{K} = (\tilde{Q}, \Sigma, \tilde{\delta}, \tilde{S}, \tilde{F}).$$

соответствующее языку L . Приведём два эквивалентных утверждению 4 варианта – утверждения 6 и 7.

Утверждение 6: Пусть $a, b \in \Sigma$. В этом случае условие $a, b \in \sigma_i$ ($\sigma_i \in \bar{\sigma}$) выполняется тогда и только тогда, когда $a \parallel^{\tilde{K}} b$. □

Утверждение 7: Пусть $a, b \in \Sigma$. Тогда $a, b \in \sigma_i$ (где $\sigma_i \in \bar{\sigma}$) в том и только том случае, если

$$(\forall q \in \tilde{Q}) (\tilde{\delta}(q, a) = \tilde{\delta}(q, b)). \quad \square$$

Доказательства обоих утверждений очевидны: они являются следствием утверждения 4 и вариантов определения параллельных букв. При этом утверждение 7 позволяет дать определение так называемого упрощённого конечного автомата над алфавитом $\bar{\sigma}$. Этот автомат определяет регулярный язык, который мы также будем называть упрощённым.

Определение 5: Упрощённый автомат для заданного регулярного языка L определяется как

$$\bar{K} = (\tilde{Q}, \bar{\sigma}, \tilde{\delta}, \tilde{S}, \tilde{F}),$$

где:

- $\bar{\sigma}$ – определённый ранее упрощённый алфавит для языка L ;
- \tilde{Q} , \tilde{S} и \tilde{F} – элементы пятёрки \tilde{L} , который также определён ранее;
- для всех пар, состоящих из состояния $q \in \tilde{Q}$ и буквы $a \in \Sigma$ (где при этом $a \in \sigma_i$) полагаем

$$\bar{\delta}(q, \sigma_i) = \tilde{\delta}(q, a). \quad \square$$

Заметим, что это определение можно рассматривать и в качестве *алгоритма*, т.е. как процесс «объединения» параллельных букв. При этом число букв в получающемся алфавите может только уменьшаться – отсюда и название «упрощённый». Также отметим, что \bar{K} определяется только на основе рассматриваемого регулярного языка L , поскольку автомат \tilde{K} является его инвариантом. При этом возможно равенство

$$\bar{\delta}(q, \sigma_i) = \tilde{\delta}(q, a),$$

поскольку, согласно утверждению 7,

$$\tilde{\delta}(q, a) = \tilde{\delta}(q, b) \quad \text{при} \quad a, b \in \sigma_i.$$

Определение автомата \bar{K} и утверждение 7 могут быть объединены вместе следующим образом.

Утверждение 8: Пусть задан некоторый регулярный язык L ; рассмотрим задающие его автоматы K и \bar{K} . Тогда для каждой буквы $a \in \Sigma$ и буквы-множества $\sigma_i \in \bar{\sigma}$ выполнено следующее условие: $a \in \sigma_i$ тогда и только тогда, когда

$$(\forall q \in \tilde{Q}) (\bar{\delta}(q, \sigma_i) = \tilde{\delta}(q, a)).$$

Доказательство утверждения очевидно. □

VI. ПРИМЕНЕНИЕ ПАРАЛЛЕЛЬНОСТИ БУКВ В ЗАДАЧАХ ТЕОРИИ ФОРМАЛЬНЫХ ЯЗЫКОВ

Этот раздел озаглавлен «Применение ... в задачах теории формальных языков». При этом мы имеем в виду, что доказываемые в этом разделе теоремы формулируют такие важные свойства регулярных языков, как:

- вид отношения $\#$ для рассматриваемого языка – свойства этого отношения формулируются *явно*;
- вид функций φ_K^{in} и φ_K^{out} для некоторого соответствующего рассматриваемому языку конечного автомата K – свойства этих функций формулируются *неявно*, поскольку, например, использующиеся в теоремах языки (\mathcal{L}_K^{in} , $\mathcal{L}_{K^R}^{in}$ и т. п.) применяются также при определении упомянутых функций φ_K^{in} и φ_K^{out} .

В начале раздела введём следующие новые обозначения.

- Пусть $a \in \Sigma$; обозначим σ_a подмножество множества Σ ($\sigma_a \in \bar{\sigma}$), содержащее a .
- Пусть $u = a_1 \dots a_m$, $\bar{u} = \sigma_{a_1} \dots \sigma_{a_m}$; тогда будем писать $u \in \bar{u}$. Аналогично в общем случае, при этом \bar{u} – слово языка $\bar{\sigma}^*$.
- \bar{L} обозначает язык автомата \bar{K} (*упрощённый язык*).

Теперь покажем, что в процессе объединения параллельных букв сохраняются важные свойства регулярных языков. В частности, автомат \bar{K} также является каноническим, а язык \bar{L} имеет ту же самую таблицу бинарного отношения $\#$, что и язык L .

Теорема 1: Автомат \bar{K} – детерминированный.

Доказательство. Покажем, что выполняются все необходимые свойства детерминированного автомата:

- \tilde{S} не изменяется, т. е. число стартовых состояний не превышает 1;
- согласно определению автомата \bar{K} , ε -переходы отсутствуют;
- для каждого состояния $q \in \tilde{Q}$ и буквы $a \in \sigma_a$

$$|\bar{\delta}(q, \sigma_a)| = |\tilde{\delta}(q, a)| \leq 1,$$

поскольку детерминированным является и автомат \bar{K} . \square

Теорема 2: Пусть $u \in \bar{u}$. Тогда для каждого состояния $q \in \tilde{Q}$ выполнено следующее:

$$u \in \mathcal{L}_{\bar{K}}^{in}(q) \iff \bar{u} \in \mathcal{L}_{\bar{K}}^{in}(q).$$

Доказательство.

\Rightarrow Пусть $u = a_1 \dots a_m$, причём $u \in \mathcal{L}_{\bar{K}}^{in}(q)$. Тогда в графе переходов автомата \bar{K} существует следующий путь:

$$s \in S \xrightarrow{a_1} q_1 \xrightarrow{a_2} \dots \xrightarrow{a_m} q;$$

и поэтому

$$\bar{u} = \sigma_{a_1} \dots \sigma_{a_m}, \quad a_i \in \sigma_{a_i}, \quad \bar{\delta}(q, \sigma_a) = \tilde{\delta}(q, a)$$

применяя также утверждение 7, мы получаем, что в автомате \bar{K} также имеется аналогичный путь:

$$s \in S \xrightarrow{\sigma_{a_1}} q_1 \xrightarrow{\sigma_{a_2}} \dots \xrightarrow{\sigma_{a_m}} q;$$

отсюда $u \in \mathcal{L}_{\bar{K}}^{in}(q)$.

\Leftarrow Обратный факт доказывается аналогично. \square

Примерно так же доказывается и следующая теорема.

Теорема 3: Пусть $u \in \bar{u}$. Тогда для каждого состояния $q \in \tilde{Q}$ выполнено следующее:

$$u \in \mathcal{L}_{\bar{K}}^{out}(q) \iff \bar{u} \in \mathcal{L}_{\bar{K}}^{out}(q). \quad \square$$

А следующая теорема 4, по-видимому, неочевидна – несмотря на очень простую формулировку.

Теорема 4: Автомат \bar{K} – канонический автомат для языка \bar{L} .

Доказательство состоит из следующих пунктов.

- Покажем, что в графе переходов автомата \bar{K} *отсутствуют недостижимые состояния*. Пусть $q \in \tilde{Q}$, тогда, поскольку автомат \bar{K} является каноническим, выполнено следующее

$$\mathcal{L}_{\bar{K}}^{in}(q) \neq \emptyset;$$

т. е. существует некоторое слово $u \in \mathcal{L}_{\bar{K}}^{in}(q)$. Отсюда согласно теореме 2 мы получаем, что

$$u \in \mathcal{L}_{\bar{K}}^{in}(q), \quad \text{и поэтому} \quad \mathcal{L}_{\bar{K}}^{in}(q) \neq \emptyset,$$

т. е. q – достижимое состояние автомата \bar{K} .

- *Отсутствие* в графе переходов автомата \bar{K} *бесполезных состояний*. Доказывается аналогично (согласно теореме 3).

- Покажем, что в автомате \bar{K} *отсутствуют эквивалентные состояния* (т. е. такие, у которых совпадают выходные языки).

Предположим противное, т. е. пусть в автомате \bar{K} $q_1 \sim q_2$, или,

$$\mathcal{L}_{\bar{K}}^{out}(q_1) = \mathcal{L}_{\bar{K}}^{out}(q_2).$$

Тогда по теореме 3 мы получаем следующее:

$$\mathcal{L}_{\bar{K}}^{out}(q_1) = \left\{ u \in \Sigma^* \mid u \in \bar{u}, \bar{u} \in \mathcal{L}_{\bar{K}}^{out}(q_1) \right\},$$

$$\mathcal{L}_{\bar{K}}^{out}(q_2) = \left\{ u \in \Sigma^* \mid u \in \bar{u}, \bar{u} \in \mathcal{L}_{\bar{K}}^{out}(q_2) \right\},$$

т. е. $\mathcal{L}_{\bar{K}}^{out}(q_1) = \mathcal{L}_{\bar{K}}^{out}(q_2)$. Отсюда получаем, что в автомате \bar{K} также выполняется условие эквивалентности $q_1 \sim q_2$, что приводит к противоречию со сделанным предположением. \square

Итак, упрощённый автомат \bar{K} действительно является каноническим, причём он определяет упрощённый язык. Теперь докажем, что упрощённому языку \bar{L} соответствует то же самое отношение эквивалентности $\#$, что и исходному языку. В первую очередь покажем соответствие между языками L и \bar{L} .

Теорема 5: $u \in L \iff u \in \bar{u}^3$, где $\bar{u} \in \bar{L}$.

Доказательство. $L = \mathcal{L}_{\bar{K}}^{out}(s)$, где s – стартовое состояние автомата \bar{K} . Согласно теореме 3,

$$u \in \mathcal{L}_{\bar{K}}^{out}(s) \iff \bar{u} \in \mathcal{L}_{\bar{K}}^{out}(s),$$

где $\mathcal{L}_{\bar{K}}^{out}(s) = \bar{L}$. \square

Утверждение 9: Если $v \in \bar{v}$ и $w \in \bar{w}$, то $\bar{v} \cdot \bar{w} = \overline{v \cdot w}$.

Доказательство. Применяем отображение $a \rightarrow \sigma_a$ (где $a \in \Sigma$, $\sigma_a \in \bar{\sigma}$). \square

³ Напомним, что необходимые обозначения были введены в начале раздела.

Теорема 6: Отношение # у языков L и \bar{L} одно и то же.

Доказательство. Согласно очевидным следствиям, получаемым на основе утверждения 5, у языков L и L^R разбиение $\bar{\sigma}$ одно и то же. Поэтому все приведённые выше утверждения и теоремы выполняются для этих двух языков одновременно (при этом для языка L^R соответствующие автоматы естественно обозначить K^R и \bar{K}^R).

Перейдём к непосредственному доказательству того, что отношение # не изменяется при преобразовании автомата \tilde{K} в \bar{K} и обратно. Для этого мы ниже записываем определения отношения # для двух конкретных состояний (а именно, q_1 и q_2), и, используя теоремы 2 и 5, показываем возможность менять во всех формулах \tilde{K} на \bar{K} и наоборот.

\Rightarrow (т.е. $\tilde{K} \Rightarrow \bar{K}$).

Для автомата \tilde{K} отношение $B\#Y$ выполняется, когда

$$(\exists u \in L) (u = vw, v \in \mathcal{L}_{\tilde{K}}^{in}(B), w^R \in \mathcal{L}_{\tilde{K}^R}^{in}(Y)),$$

отсюда $\bar{u} \in \bar{L}$, а поэтому

$$\bar{u} = \bar{v} \cdot \bar{w} = \bar{v} \cdot \bar{w},$$

где $\bar{v} \in \mathcal{L}_{\bar{K}}^{in}(B)$ и $\bar{w}^R \in \mathcal{L}_{\bar{K}^R}^{in}(Y)$. Таким образом, для языка \bar{L} и автомата \bar{K} выполнено условие $B\#Y$.

\Leftarrow (т.е. $\bar{K} \Rightarrow \tilde{K}$).

Для автомата \tilde{K} отношение $B\#Y$ выполняется, когда

$$(\exists \bar{u} \in \bar{L}) (u = \bar{v}\bar{w}, \bar{v} \in \mathcal{L}_{\tilde{K}}^{in}(B), \bar{w}^R \in \mathcal{L}_{\tilde{K}^R}^{in}(Y)).$$

Выберем некоторые $v \in \bar{v}$ и $w \in \bar{w}$; для них

$$\bar{u} = \bar{v} \cdot \bar{w} = \bar{v} \cdot \bar{w} \quad \text{и} \quad u = vw \in L,$$

где $v \in \mathcal{L}_{\tilde{K}}^{in}(B)$ и $w^R \in \mathcal{L}_{\tilde{K}^R}^{in}(Y)$. Это означает выполнение условия $B\#Y$ для заданного языка L и автомата \tilde{K} . \square

VII. ЭКВИВАЛЕНТНЫЕ РЕГУЛЯРНЫЕ ЯЗЫКИ

Итак, в предыдущем разделе мы начали рассматривать, среди прочего, связь отношения параллельности букв в частности (и понятие упрощённого автомата вообще) – с отношением # для рассматриваемого языка, т.е. фактически эквивалентность в «более широком» смысле – эквивалентность по отношению #, собственно отношение R на регулярных языках. Здесь же, в этом и следующем разделах, мы переходим непосредственно к эквивалентности в «более узком» смысле – к отношению S на регулярных языках.

Сначала определим эквивалентность путём рассмотрения упрощённых регулярных языков. Конечно же, существуют несовпадающие регулярные языки⁴, которым соответствует один и тот же упрощённый язык (с точностью до переобозначения букв рассматриваемого алфавита $\bar{\sigma}$). Более формально об этом говорится в следующем определении.

Определение 6: Языки L_1 и L_2 называются эквивалентными по упрощению, если языку L_i соответствует

⁴ Приведём такую аналогию – для более сложного случая. Всем регулярным языкам, имеющим одно и то же отношение #, соответствует один и тот же регулярный язык, определяемый автоматом, который мы в предыдущих публикациях обозначали $K^\#$.

упрощённый язык \bar{L}_i над алфавитом $\bar{\sigma}_i$ ($i = 1, 2$), и существует биекция

$$\bar{\sigma}_1 \longleftrightarrow \bar{\sigma}_2,$$

такая что она может быть расширена до \bar{L}_1 и \bar{L}_2 (т.е. упрощённые автоматы \bar{K}_1 и \bar{K}_2 совпадают с точностью до переобозначения букв и состояний). \square

Отметим, что последнее определение действительно является определением отношения эквивалентности, поскольку упрощённый язык – как и упрощённый автомат – являются инвариантами регулярного языка⁵; следовательно, все регулярные языки разбиты на подмножества языков, в каждом из которых упрощённый язык один и тот же.

Пример 3: Языки, определяемые регулярными выражениями

$$L_1 = a^* + b^* + (a + b)^* \quad \text{и} \quad L_2 = c^*$$

являются эквивалентными по упрощению, поскольку обоим языкам соответствует один и тот же упрощённый язык $\bar{L}_i = \bar{\sigma}_i^*$.

Стоит отметить, что упрощение языков можно рассматривать как первый небольшой шаг в подходе к *классификации регулярных языков*: любой регулярный язык может быть получен из некоторого упрощённого языка с помощью операции, обратной операции упрощения:

$$\sigma_i \rightarrow \{a_{i_1}, a_{i_2}, \dots, a_{i_n}\}, \quad \text{где} \quad \sigma_i \cap \sigma_j = \emptyset,$$

и так далее. При этом возможности применения нескольких вариантов такой «обратной операции» были показаны нами в [2] (см. также [4], [5]).

Ещё отметим, что можно также определить и более сложный объект, который можно условно назвать «буквой, параллельной языку»; проще всего такое определение ввести для некоторого класса «простых» языков (например, конечных языков, которые могут быть записаны с помощью регулярных выражений, не использующих звезду Клини). После подобного определения можно расширить и рассмотренное выше отношение эквивалентности языков.

VIII. АВТОМАТЫ БЕЗ ПАРАЛЛЕЛЬНЫХ БУКВ

Как следует из названия раздела, мы будем рассматривать автоматы, не имеющие параллельных букв. Фактически алгоритм построения такого автомата, эквивалентного некоторому заданному, можно получить на основе предыдущего материала статьи (прежде всего – на основе раздела VII); но в настоящем разделе мы показываем, что приведённое ранее определение 5 можно считать *конструктивным* определением автомата без параллельных букв.

Поскольку пока не существует законченной классификации упрощённых языков, было бы полезно уже сейчас найти какое-то другое применение этому понятию. Одним из возможных применений описывается тем фактом, что при рассмотрении упрощённых языков мы можем не принимать во внимание автоматы с имеющимися в них параллельными буквами. Это, в частности, позволяет нам

⁵ Неполными.

исследовать только конечное число автоматов – в тех случаях, когда мы ограничиваем сверху количество их состояний.

Следующее определение можно рассматривать как операцию «предварительного соединения» параллельных (с точки зрения рассматриваемого автомата) букв. А согласно дальнейшей теореме 7, такое предварительное объединение не изменяет не только получаемый упрощённый язык, но и получаемый упрощённый автомат.

Определение 7: Пусть $K = (Q, \Sigma, \delta, S, F)$, $a, b \in \Sigma$ и $a \parallel b$. Также предполагаем, что $\Sigma' = \Sigma \setminus \{b\}$, а δ' – соответствующее ограничение функции переходов δ для нового алфавита Σ' ⁶.

Тогда будем говорить, что автомат $K' = (Q, \Sigma', \delta', S, F)$ получен из автомата K объединением параллельных букв a и b ⁷. □

Теорема 7: Пусть $K = (Q, \Sigma, \delta, S, F)$, $a, b \in \Sigma$, $a \parallel b$, а автомат K' получен из K объединением параллельных букв a и b . Тогда $\bar{K} = \bar{K}'$.

Доказательство. Мы уже знаем, что операции

$$K \rightarrow \hat{K} \quad \text{и} \quad \hat{K} \rightarrow \tilde{K}$$

сохраняют свойство параллельности букв $a \parallel b$ (см. утверждение 1). При применении упрощающих операций к автоматам K и K' получаемый автомат \bar{K}' будет отличаться от автомата \bar{K} только тем, что множество σ_i (где $\sigma_i \ni a, b$) изменится на $\sigma_i \setminus \{b\}$ – а это для алфавита $\bar{\Sigma}$ несущественно. □

Итак, любой автомат может быть упрощён с помощью последовательности (возможно, пустой) «предварительного объединения» параллельных букв – с получением автомата, параллельных букв не имеющего. Из предыдущего материала статьи также следует, что подобные объединения могут быть проведены в произвольном порядке. Поэтому, как уже было сказано в начале раздела, приведённое ранее определение 5 можно считать *конструктивным* определением автомата без параллельных букв⁸. А поскольку для решения большинства задач, связанных с регулярными языками, важны только языки (автоматы) без параллельных букв, то наличие конструктивных алгоритмов построения таких автоматов даёт возможность в таких задачах рассматривать *только конечное число автоматов* – например, в тех случаях,

⁶ Фактически это – новая функция переходов, получающаяся в результате переименования имеющейся буквы b в также имеющуюся букву a .

⁷ В первой части статьи несколько раз употреблялось обозначение $\mathcal{J}^{q, q'}(K)$ – для объединения двух состояний, q и q' . Похожее обозначение можно было бы ввести и в рассматриваемом случае, т. е. для объединения двух букв. Более того, при этом, по-видимому, упростятся некоторые формулы, связанные с преобразованиями полного автомата, описание таких преобразований должно стать более наглядным; наши статьи с этими описаниями процитированы выше, в том числе в части I. Мы предполагаем ввести подобные обозначения в следующих публикациях.

⁸ В принципе, таковым его можно было считать и без последующего (т. е. приведённого после этого определения 5) материала. Однако здесь возникает примерно такая же ситуация, как и с разными типами языков в иерархии Хомского; например, для особенно интересного нам типа 3: можно ли считать *конструктивным* определением регулярного языка его запись в виде праволинейной грамматики (либо регулярного выражения) – или для этого необходим некоторый конечный автомат? По-видимому, это – «вопросы без ответа».

когда мы можем заранее ограничить сверху число их состояний.

Поэтому представляют интерес возможные оценки сверху для числа автоматов, обладающих рассматриваемыми в статье свойствами. Одна из таких возможных оценок приведена в следующей теореме.

Теорема 8: Пусть n – заданное заранее максимально возможное число состояний упрощённого автомата. Тогда число различных упрощённых недетерминированных конечных автоматов⁹ не превышает значения

$$2^{2^{n^2-1}} \cdot (2^n - 1)^2. \quad (1)$$

Доказательство. Согласно теореме 7, число упрощённых автоматов не превышает числа автоматов, имеющих n состояний, и при этом *не имеющих параллельных букв*. Приведём одну из возможных оценок этого числа.

Пусть состояния рассматриваемого автомата, образующие множество

$$Q = \{q_1, q_2, \dots, q_n\}, \quad (2)$$

некоторым образом упорядочены (например, согласно индексации (2)), а сам этот автомат обозначен пятёркой

$$K = (Q, \Sigma, \gamma, S, F) \quad (3)$$

Сопоставим каждой букве $a \in \Sigma$ (число $|\Sigma|$ будет определено позже) двоичную матрицу размера $n \times n$ (см. рис. 4), где каждый элемент b_{ij} равен 0 или 1; при этом мы считаем, что $b_{ij} = 1$ тогда и только тогда, когда выполнено условие

$$a \in \gamma(q_i, q_j).$$

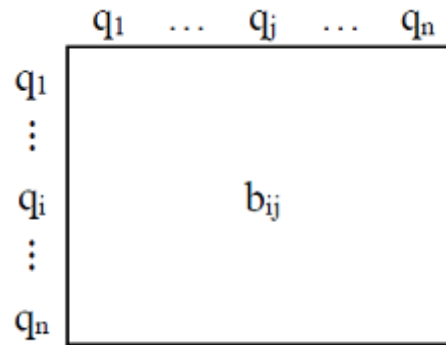


Рис. 4. Булева матрица, определяющая для заданной буквы $a \in \Sigma$ соответствие состояний автомата K .

Очевидно, что такая матрица полностью определяет вхождение рассматриваемой буквы $a \in \Sigma$ в функцию переходов (недетерминированного) автомата – а всё множество таких матриц определяет автомат и, следовательно, рассматриваемый регулярный язык. Также очевидно, что буквы, имеющие совпадающие матрицы, являются параллельными над автоматом K , и наоборот, если такие матрицы не совпадают, то соответствующие буквы параллельными над K не являются. Следовательно, число попарно непараллельных букв не превышает 2^{n^2} .

⁹ А следовательно, и соответствующих им упрощённых регулярных языков.

¹⁰ Напомним, что иногда применяемое в наших статьях определение функции γ – как возможной альтернативы функции переходов δ – было приведено в части I.

При этом матрицу, состоящую из одних нулей, можно не рассматривать (поскольку в этом случае буква, для которой предназначена эта матрица, языку автомата не принадлежит), то есть

$$|\Sigma| \leq 2^{n^2} - 1.$$

Мы также должны учитывать, что для любой упорядоченной пары состояний (иными словами – для любой клетки матрицы) имеется 2 варианта для каждой буквы алфавита: она либо *присутствует* в автомате, либо *отсутствует* (при этом мы не рассматриваем переименование букв). Следовательно, количество возможных способов определения функции γ равно

$$2^{2^{n^2}} - 1.$$

Нужно также рассмотреть варианты стартовых и финальных состояний. Как для S , так и для F имеется $2^n - 1$ варианта выбора из полного множества (в обоих случаях мы также не рассматриваем множества, состоящие из одних нулей). Перемножив полученные значения, мы получаем ограничение (1). \square

IX. ЗАКЛЮЧЕНИЕ

Итак, в этой статье мы определили упрощённый конечный автомат. Мы доказали, что для соответствующего языка этот автомат является каноническим; поэтому он однозначно определяет упрощённый язык, и вместо рассмотрения произвольных регулярных языков мы можем рассматривать только упрощённые. Более того, исходный регулярный язык и соответствующий упрощённый имеют одинаковые бинарные отношения $\#$, что позволяет нам использовать всю предыдущую теорию, просто рассматривая упрощённые автоматы вместо произвольных конечных автоматов.

Также мы изучали автоматы без параллельных букв, для которых получили верхнюю границу числа упрощённых языков с заранее зафиксированным числом состояний определяющего некоторый такой язык недетерминированного конечного автомата.

Как мы несколько раз отмечали в тексте статьи, некоторые связанные с упрощёнными автоматами и языками вопросы, в том числе – некоторые дополнительные подходы к классификация регулярных языков, были ранее рассмотрены в [2]. Также стоит отметить, что можно увидеть связь упомянутых в настоящей статье букв, параллельных некоторому языку, – с вопросами, рассмотренными в [6]. Всё это может дать ещё один подход к классификация регулярных языков – и в будущих публикациях мы предполагаем эти классификации объединить.

Далее мы приводим небольшую часть текста одной из наших предыдущих статей: развитие тем, описанных в этом тексте, можно считать формулировкой дальнейших работ и по темам, связанным с упрощёнными конечными автоматами и упрощёнными регулярными языками. Итак, на основе теории, рассмотренной в предыдущих публикациях, мы, среди прочего, получили доказательство утверждений в следующих темах:

- построение любого автомата для заданного регулярного языка на основе базисного автомата для этого языка, [7];
- алгоритмы построения автомата \mathcal{COM} , [3], [8], [9].

Такая тематика напрямую связана с текстом, приведённым в следующем абзаце.

В будущих публикациях мы планируем привести некоторые *результаты вычислений*, в которых получено количество упрощённых языков – для тех ситуаций, в которых автомат для рассматриваемого языка имеет ровно 2 состояния; эти результаты подтверждают приведённое выше значение верхней границы. Также стоит отметить, что мы можем применить «упрощение» *практически к любому понятию* теории регулярных языков. Например, мы можем изучить «упрощённый универсальный автомат», см. [3], [8], [9] и др. Как мы уже отмечали выше, мы также собираемся объединить две классификации обычных языков – т.е. классификацию, рассмотренную в настоящей статье, а также ту, которая может быть получена на основе [2].

Список литературы

- [1] Мельников Б., Долгов В. *Упрощённые регулярные языки и специальное отношение эквивалентности на классе регулярных языков. Часть I* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 9. – P. 12–20.
- [2] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [3] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta Universitatis Sapientiae, Informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.
- [4] Melnikov B. *Лесточковые конечные автоматы: основные определения, примеры и их связь с полными автоматами. Часть I* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 9. – P. 1–11.
- [5] Melnikov B. *Лесточковые конечные автоматы: основные определения, примеры и их связь с полными автоматами. Часть II* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 10. – P. 1–10.
- [6] Melnikov B., Vakhitova A. *Some more on the finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Computational and Applied Mathematics). – 1998. – Vol. 5, No. 3. – P. 495–505.
- [7] Мельников Б., Сайфуллина М. *О некоторых алгоритмах эквивалентного преобразования недетерминированных конечных автоматов* // Известия высших учебных заведений. Математика. – 2009. – № 4. – С. 67–72.
- [8] 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.
- [9] Долгов В., Мельников Б. *Построение универсального конечного автомата. II. Примеры работы алгоритмов* // Вестник Воронежского государственного университета. Серия: Физика. Математика. – 2014. – № 1. – С. 78–85.

Борис Феликсович МЕЛЬНИКОВ,
профессор Совместного университета МГУ – ППИ
в Шэньчжэне, Китай

(<http://szmsubit.ru/>),
email: bormel@smbu.edu.cn,
email: bf-melnikov@yandex.ru,
mathnet.ru: personid=27967,
elibrary.ru: authorid=15715,
scopus.com: authorId=55954040300,
ORCID: orcidID=0000-0002-6765-6800.

Василий Николаевич ДОЛГОВ,
заместитель директора Тереньгульского лицея
при Ульяновском государственном
техническом университете (<https://ulstu.ru/>),
email: terenga74@mail.ru,
elibrary.ru: authorid=700000,
ORCID: orcidID=0000-0001-5925-541X.

Simplified regular languages and a special equivalence relation on the class of regular languages. Part II

Boris Melnikov, Vasily Dolgov

Abstract—The equivalence relation S on the class of regular languages, considered in this paper, is necessary for a more complete study of the relation R previously defined in our articles. In addition, the motivation for considering the relation S is the need to apply it to the study of so-called petal (semi-flower) automata, which was also started in one of our previous papers. Specifically, in order to obtain a petal automaton, there is necessary to consider such an algorithm of the nonequivalent transformation of the automaton as the union of two letters of the considered alphabet, and both the mentioned equivalence relations on the class of regular languages, i.e. relations R and S , are associated with this transformation. In turn, petal automata arise when describing several different algorithms for checking the binary relation “equivalence at infinity”, also discussed in some our previous papers, in particular, when using PRI and NSPRI finite automata in these algorithms.

Thus, in this paper we define S , a special binary relation on a set of regular languages, show the fulfillment of all the properties of equivalence relations; therefore the relation S divides the whole class of regular languages into some disjoint subclasses. As a result, for most of the problems of the theory of formal languages, we can take only one representative of each such class; and it is usually desirable to consider the so-called simplified language, it is the only in each such subclass.

The concept of a simplified language is based on the combination of so-called parallel letters, and the simplified finite automaton specially introduced by us corresponds to such a simplified language. All this makes it possible to limit the number of regular languages under consideration to work with a finite number of corresponding finite automata; moreover, such automata have a pre-fixed number of states.

Both these equivalence relations, S and R , preserve the relation $\#$ defined for a particular regular language, and, therefore, allow to use many previously proven properties of regular languages and nondeterministic finite automata for arbitrary automaton of its equivalence class and the corresponding language.

In the presented Part II, we continue to consider the properties of the relation S , which are significantly more complicated than ones considered in Part I. Among other things, we present variants of the application of the parallel letter in some problems of the theory of formal languages.

Keywords—regular languages, nondeterministic finite automata, binary relations, equivalence relation, simplification of automata, algorithms.

References

- [1] Melnikov B., Dolgov V. *Simplified regular languages and a special equivalence relation on the class of regular languages. Part I* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 9. – P. 12–20 (in Russian).
- [2] Melnikov B. *The complete finite automaton* // International Journal of Open Information Technologies. – 2017. – Vol. 5. No. 10. – P. 9–17.
- [3] Melnikov B., Dolgov V. *Some more algorithms for Conway's universal automaton* // Acta Universitatis Sapientiae, Informatica. – 2014. – Vol. 6. No. 1. – P. 5–20.
- [4] Melnikov B. *Petal (semi-flower) finite automata: basic definitions, examples and their relation to complete automata. Part I* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 9. – P. 1–11 (in Russian).
- [5] Melnikov B. *Petal (semi-flower) finite automata: basic definitions, examples and their relation to complete automata. Part II* // International Journal of Open Information Technologies. – 2022. – Vol. 10. No. 10. – P. 1–10 (in Russian).
- [6] Melnikov B., Vakhitova A. *Some more on the finite automata* // The Korean Journal of Computational and Applied Mathematics (Journal of Computational and Applied Mathematics). – 1998. – Vol. 5, No. 3. – P. 495–505.
- [7] Melnikov B., Sayfullina M. *On some algorithms for equivalent transformation of nondeterministic finite automata* // News of higher educational institutions. Mathematics. – 2009. – No. 4. – P. 67–72 (in Russian).
- [8] Dolgov V., Melnikov B. *Building 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).
- [9] Dolgov V., Melnikov B. *Building 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).

Boris MELNIKOV,
Professor of Shenzhen MSU–BIT University, China
(<https://szmsubit.ru/en/>),
email₁: bormel@smbu.edu.cn,
email₂: bf-melnikov@yandex.ru,
mathnet.ru: personid=27967,
elibrary.ru: authorid=15715,
scopus.com: authorId=55954040300,
ORCID: orcidID=0000-0002-6765-6800.

Vasily DOLGOV,
Vice-director of Terenga Lyceum
at Ulyanovsk State Technical University
(<https://ulstu.ru/>),
email: terenga74@mail.ru,
elibrary.ru: authorid=700000,
ORCID: orcidID=0000-0001-5925-541X.