Восемь вариантов конечных автоматов для проверки выполнения отношения покрытия итераций двух конечных языков. Часть II

Б. Ф. Мельников, Мэн Линцянь

Аннотация—**Задача, для которой рассматриваются все автоматы настоящей статьи, заключается в описании алгоритмов проверки т.н. бинарного отношение покрытия, рассматривавшегося во многих наших предыдущих статьях: оно выполняется для двух непустых конечных языков тогда и только тогда, когда любая итерация слов первого языка является префиксом некоторой итерации слов второго языка. Эту задачу можно рассматривать с нескольких разных точек зрения, что и делалось в предыдущих работах.**

В настоящей статье нас в первую очередь интересуют различные варианты реализации алгоритмов для проверки выполнения этого отношения, а более конкретно – применение для таких алгоритмов конечных автоматов. 2 варианта таких автоматов уже были ранее описаны в наших предыдущих работах, а в настоящей статье мы к ним добавляем ещё 6 вариантов.

Основная цель рассмотрения нескольких автоматов-алгоритмов – такой. Ранее нами был описан полиномиальный алгоритм построения одного из таких автоматов. Однако существование такого полиномиального алгоритма вовсе не означает, что мы можем полиномиально проверить выполнение необходимого нам условия – в связи со следующим обстоятельством. Если мы для ответа на поставленный вопрос строим недетерминированный автомат таким же образом, как мы описывали ранее, то он для положительного ответа должен принимать любое слово универсального языка – а последнее условие за полиномиальное время проверить нельзя. Однако мы также не можем утверждать, что подобного полиномиального алгоритма – необходимы дополнительные исследования. Поэтому наличие четырёх описаний недетерминированных автоматов, отвечающих на поставленный вопрос о выполнении рассматриваемого бинарного отношения, даст бóльшие возможности для поиска возможного алгоритма, проверяющего основное условие статьи за полиномиальное время.

8 вариантов рассматриваемых в статье автоматов естественным образом получаются с помощью трёх различных дихотомий: одна из них – «обычная» (автомат детерминированный или недетерминированный), а две других непосредственно связаны с рассматриваемыми в статье вопросами проверки выполнения бинарного отношения пркрытия. Конкретно, эти две дихотомии таковы: во-первых, какой именно из двух рассматриваемых конечных языков является «главным» (именно для него мы применяем основной морфизм для проверки); во-вторых, рассматриваем ли мы требуемые покрытия каждого слова-итерации непосредственно при его возникновении – либо не сразу, а после формирования некоторого вспомогательного языка.

В представляемой второй части настоящей статьи мы продолжаем описывать варианты конечных автоматов для проверки выполнения отношения покрытия.

Статья получена 8 ноября 2023 г.

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

Мэн Линцянь, Московский государственный университет им. М. В. Ломоносова (kostya45e@ mail.ru).

Ключевые слова—**формальные языки, итерации языков, морфизмы, конечные автоматы, алгоритмы.**

Во второй части статьи мы продолжаем описывать варианты конечных автоматов, предназначенных для проверки выполнения отношения покрытия итераций двух конечных языков.

Отметим, что очень кратко варианты, рассматриваемые в части II, были охарактеризованы в части I, т. е. [1]. При этом в части II мы описываем *действительно новые* варианты – поскольку два варианта, описанные в части I, можно считать переописанием версий двух конечных автоматов, описанных ранее; они в предыдущих статьях обозначались просто как PRI (детерминированный) и NSPRI (недетерминированный) – без верхнего и нижнего индексов.

Мы продолжаем нумерацию разделов и рисунков, начатую в первой части; нумерация ссылок новая.

V. (3) АВТОМАТ 010, Т. Е. $\mathrm{PRI}^{\rightarrow|}_{\downarrow}$

Как уже было отмечено в части I (раздел II), автомат 010, или, в других обозначениях,

$$\mathrm{PRI}^{\rightarrow|}_{\downarrow}, \quad -$$

это детерминированный автомат, проверяющий условие $A \trianglelefteqslant B$ для некоторых языков A и B. В таком автомате проверка осуществляется для «основного» языка B – путём описания для каждого слова $v \in B^*$ всех возможных слов языка $u \in A^*$, таких что:

- u является префиксом v, обозначим $uw = v$, допускаем возможность $w = \varepsilon$;
- для каждого такого u существует некоторое $u' \in A$, такое, что uu' не является префиксом v.

Такое описание представляет собой модификацию функции, которая в предыдущих публикациях (в том числе в части I настоящей статьи, [1]) обозначалась Ф, иногда с необходимыми индексами. Мы далее будем её рассматривать как функцию переходов для некоторого конечного автомата.

Для описания такого модифицированного варианта функции Ф все выбранные слова (w в этих обозначениях) рассматриваются *как специальный язык* (а не по отдельности) – именно поэтому получающийся далее автомат будет детерминированным.

Важно также отметить следующее. Для функции Ф мы не будем ставить индексы: такие функции Ф мы будем считать разными в разных разделах. Точнее, такие

функции будут одинаковыми для пары, состоящей из детерминированного автомата (таковые определяются в разделах V, VII, IX и ранее в разделе III части I), а также определяемого в каждом последующем разделе соответствующего недетерминированного автомата. То же самое будет верно и для «промежуточных» автоматов – которые во всех детерминированных случаях будут обозначены $\overline{\mathrm{PRI}}$, а в недетерминированных $\mathrm{NSPRI}^{\#}$.

Вернёмся к материалу, непосредственно относящемуся к паре автоматов 010 и 011. Для них дополнительно отметим, что мы приводим не алгоритм для окончательного получения ответа на вопрос о выполнении / невыполнении отношения покрытия, а *всю необходимую работу, необходимую для получения этого ответа*. Иными словами – представляемым автоматом описывается именно такая работа алгоритма.

Формальное описание варианта функции Φ

(Сразу отметим, что все описания вариантов функции Φ – приведённые как в этом разделе, так и в других «нечётных» разделах VII и IX, – можно сравнить с соответствующим материалом раздела III части I.)

Итак, *во-первых*, для пары непустых конечных языков $A, B \subseteq \Sigma^*$, таких что $A \not\ni \varepsilon$ и $B \not\ni \varepsilon$, язык

$$\underset{A-B}{\Phi}(u) \subseteq \Sigma^*$$

строится для некоторого слова $u \in \Sigma^*$ по следующему алгоритму – в нём используется вспомогательный язык D' и формируется язык-ответ D.

1) Для начала работы алгоритма полагаем
$$D' = \{u\} \cdot B, \quad D = \varnothing.$$
2) Если $D' = \varnothing$, то выход из алгоритма с ответом D.
3) Выбираем некоторое слово $uv \in D'$, исключая его из D'.
4) Рассматриваем каждое представление слова $uv \in D'$ в виде
$$uv = ww',$$
где одновременно:
   - $w \in A$;
   - $(\exists w'' \in A)(ww'' \notin \mathrm{pref}(uv))$, т. е. для выбранного таким образом w'' выполнено условие $w'' \notin \mathrm{pref}(w')$.
5) Каждое такое найденное w' включаем в D, т. е.
$$D +: w'.$$
6) Переходим на п. 2.

Во-вторых, аналогично сделанному ранее в части I, для языка $C \subseteq \Sigma^*$ определяем

$$\underset{A-B}{\Phi}(C) = \bigcup_{u \in C} \underset{A-B}{\Phi}(u).$$

Некоторые дополнительные пояснения приведены на рис. 5 (очень кратко) и на рис. 6 (значительно более подробно). По поводу рис. 6 надо повторить (причём почти без изменений) некоторые комментарии к рис. 1 из части I:

- во-первых, как надеются авторы, пояснения на английском языке, приведённые на рисунке, понятны без специального перевода;

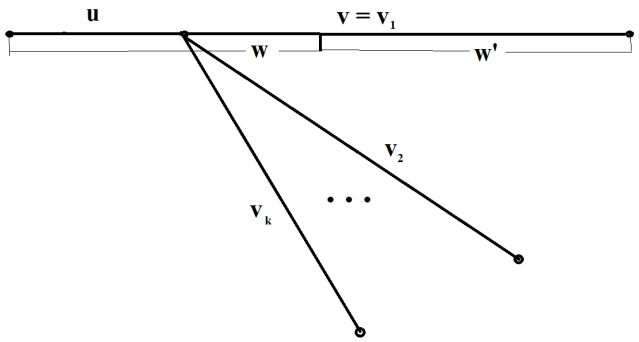

Рис. 5. Краткая иллюстрация алгортма, соответствующего автоматам, обозначенным в статье 010 и 011.

- во-вторых, рис. 6 относится как к рассматриваемому здесь детерминированному автомату, так и к приведённому в следующем разделе автомату недетерминированному; важно, что оба автомата реализуют один и тот же алгоритм.

Формальное описание автомата

Детерминированный конечный автомат $\overline{\mathrm{PRI}}(A, B)$ для заданных конечных языков $A, B \subseteq \Sigma^*$ определяется следующим образом [1]:

$$\overline{\mathrm{PRI}}(A, B) = \big(\overline{Q}_A, \Delta_B, \delta_{B-A}, \{\{\varepsilon\}\}, \{\varnothing\}\big),$$

где:

- $\Delta_B = \{0, 1, \ldots, n-1\}$, здесь n – число слов языка B [2];
- $\overline{Q}_A = \mathcal{P}(\mathrm{opref}(A))$;
- для некоторых $q_1, q_2 \in \overline{Q}_A$ и $a \in \Delta_B$ полагаем
$$q_1 \xrightarrow[\delta_{B-A}]{a} q_2$$
тогда и только тогда, когда
$$\underset{h_A(a)-B}{\Phi}(q_1) = q_2\,.$$

Автомат, получающийся после удаления всех недостижимых состояний из автомата $\overline{\mathrm{PRI}}(A, B)$, будем обозначать

$$\mathrm{PRI}^{\rightarrow|}_{\downarrow} = \big(Q_A, \Delta_B, \delta_{B-A}, \{\{\varepsilon\}\}, \{\varnothing\}\big),$$

где $Q_A \subseteq \mathcal{P}(\mathrm{opref}(B))$.

[1] Мы, начиная с раздела III (часть I), на основе *совершенно разных* определений функций Φ *практически одним и тем же способом* последовательно определяем следующие автоматы: $\overline{\mathrm{PRI}}$, PRI, $\mathrm{NSPRI}^{\#}$, NSPRI (как мы уже отмечали, они в разных разделах обозначены одинаково). Разница в определениях разных разделов совсем небольшая: иногда «меняются местами» языки A и B – но при желании можно было бы почти везде (кроме, конечно, алфавитов Δ_A и Δ_B) сделать совершенно одинаковые обозначения. Однако, повторим, определяемые ранее варианты функций Φ различаются принципиально.

Здесь возникает аналогия с применением виртуальных функций (можно сказать «наследования» и «полиморфизма») в объектно-ориентированных языках программирования.

[2] Напомним, что мы при этом одновременно естественным образом определяем и морфизм

$$h_B : \Delta_B^* \to \Sigma^*.$$

Упорядочивание слов в языке B при таком определении выбирается произвольным образом – а получающиеся функции переходов автоматов (и сами автоматы) оказываются одинаковыми с точностью до переобозначения (перестановки) букв алфавита Δ_B.

Аналогичное замечание относится ко всем определяемым в статье детерминированным автоматам (их 4).

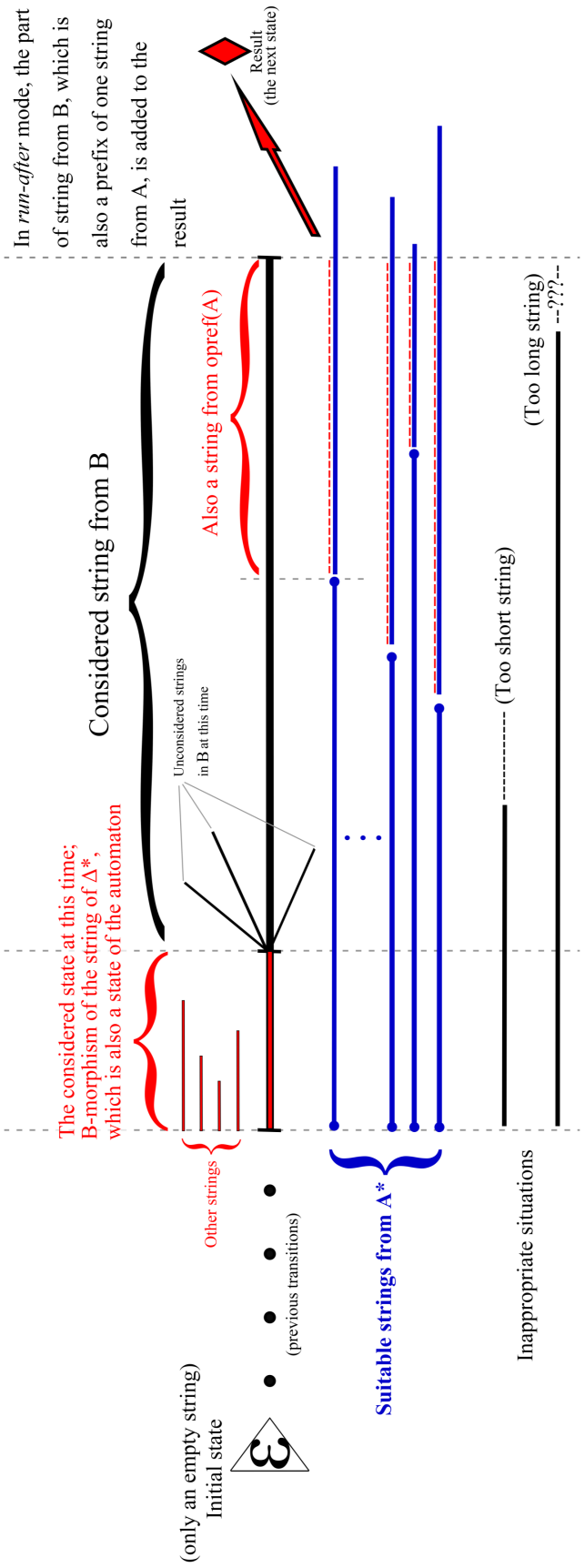

Рис. 6. Иллюстрация алгортма, соответствующего автоматам, обозначенным в статье 010 и 011. (Run-after mode, основной язык B.)

	automaton 0100	aaaa	abb	abba	bbb
0	Ø	0	0	0	0
1	{ε}	2	3	1	4
2	{a}	5	6	1	0
3	{abb}	1	1	2	0
4	{b}	0	0	0	1
5	{aa}	1	1	2	0
6	{aabb}	1	1	2	0

Рис. 7. Автомат, обозначенный в статье 0100.

	automaton 0101	aba	abaab	abab	ababa	abababababa
0	Ø	0	0	0	0	0
1	{ε}	2	3	4	2	2
2	{a,aba}	5	4	4	2	2
3	{ε,abaab}	2	3	4	2	2
4	{ε,ab}	2	3	4	2	2
5	{ε,a,abaaba}	2	3	4	2	2

Рис. 8. Автомат, обозначенный в статье 0101.

	automaton 1000	aaa	aabba	abba	bb
0	Ø	0	0	0	0
1	{ε}	2	0	3	4
2	{a}	5	0	0	6
3	{aaa,aaaa,abb,abba,bb,bba,bbb}	7	3	3	8
4	{b}	0	0	0	5
5	{aaaa,abb,abba,bbb}	7	3	3	9
6	{ε,a,aaaa,abb,abba,bbb}	7	3	3	9
7	{a,aa,aaaa,abb,abba,bbb}	10	3	3	9
8	{ε,a,aaaa,abb,abba,b,bb,bbb}	7	3	3	8
9	{ε,a,aaaa,abb,abba,b,bbb}	7	3	3	9
10	{a,aa,aaa,aaaa,abb,abba,bb,bba,bbb}	10	3	3	8

Рис. 9. Автомат, обозначенный в статье 1000.

	automaton 1001	ab	abaab	abaaba	abaabab
0	Ø	0	0	0	0
1	{ε}	2	3	4	5
2	{a,aab,ab,aba,abababaaba}	5	3	4	5
3	{a,aab,ab,aba,abaab,abab, ababa,abababaaba, abababababa}	5	3	4	5
4	{ε,ab,aba,abaab,abab,ababa, abababababa,b,ba,baab,bab, baba,babababa,bababababa}	5	3	4	5
5	{ε,a,aab,ab,aba,abaab, abab,ababa,abababa, abababababa,abababababa}	5	3	4	5

Рис. 10. Автомат, обозначенный в статье 1001.

Примеры автоматов, дающих разные ответы

Такие примеры автоматов приведены на рис. 7 (отрицательный ответ) и на рис. 8 (положительный ответ). Про соответствующие языки см. информацию в части I. Повторим, что в рассматриваемом здесь случае, т. е. для автоматов 0100 и 0101, для получения окончательного ответа на вопрос о выполнении отношения покрытия нужны несложные дополнительные алгоритмы.

VI. (4) АВТОМАТ 011, Т. Е. $\mathrm{NSPRI}^{\rightarrow|}_{\downarrow}$

Работа со словами в автомате 011, т. е. $\mathrm{NSPRI}^{\rightarrow|}_{\downarrow}$, производится совершенно таким же образом, как и в разделе V: функцию переходов Φ мы там определяли сначала для слов (только потом для языков) – и определение для слов подходит в для недетерминированного варианта. Все выбранные слова (w в применяемых в предыдущем разделе обозначениях) рассматриваются *по отдельности* (а не как специальный язык) – именно поэтому получающийся автомат будет недетерминированным. Повторим, что автомат 0101 недетерминированно моделирует процесс получения ответа на основной вопрос статьи, причём его (автомата) работа этот процесс описывает; но для получения окончательного ответа нужны несложные дополнительные алгоритмы.

Также отметим, что поскольку мы моделируем процесс работы автомата, нам для этого достаточно рассматривать не классические конечные автоматы, а какой-либо их упрощённый вариант, например – какой-либо вариант ω-автоматов. При этом можно выбрать самый простой вариант ([2] и др.), т. е. автоматы, не имеющие:

- ни выходов,
- ни их аналога (для «бесконечной работы») – т. е. обозначаемого таким же образом множества специальных состояний F.

Однако для дальнйших построений мы выбрали классический вариант определения автоматов. Некторые подробности – как об ω-автоматах, так и о возможном дальнейшем использовании автомата $\mathrm{NSPRI}^{\rightarrow|}_{\downarrow}$ – см. в [2], [3] и в заключении настоящей статьи. См. также работы одного из авторов на эту и связанную тематику [4], [5], [10], [7].

Формальное описание автомата

Сначала для заданных конечных языков $A, B \subseteq \Sigma^*$ определим (недетерминированный) конечный автомат $\mathrm{NSPRI}^{\#}(A, B)$ следующим образом:

$$\mathrm{NSPRI}^{\#}(A, B) = \left(Q^{\#}_A, \Delta_B, \delta^{\#}_{B-A}, \{\varepsilon\}, \varnothing \right),$$

где:

- $Q^{\#}_A = \mathrm{opref}(A) \cup \{\varnothing\}$;
- для некоторых $q_1, q_2 \in Q^{\#}_A$ и $d \in \Delta_B$ наличие перехода
$$q_1 \xrightarrow[\delta^{\#}_{B-A}]{d} q_2$$
мы полагаем тогда и только тогда, когда
$$\Phi_{\{h_B(d)\}-A}(q_1) \ni q_2 .$$

(При этом функцию Φ мы специально не определяем, она совпадает с аналогичной функцией из предыдущего раздела.)

Для заданных конечных языков $A, B \subseteq \Sigma^*$ (недетерминированный) конечный автомат $\mathrm{NSPRI}^{\rightarrow|}_{\downarrow}$ определяется следующим образом:

$$\mathrm{NSPRI}^{\rightarrow|}_{\uparrow}(A, B) = \left(\mathrm{opref}(A), \Delta_B, \delta^{\#}_{B-A}, \{\varepsilon\}, \mathrm{opref}(A) \right),$$

где функция переходов $\delta^{\#}_{B-A}$ совпадает с введённой в этом разделе выше.

Пример автомата

Пример автомата, дающего положительный ответ, почти совпадает с приведённым описанием автомата 0100: в нём нет состояний, соответствующих 2 или более словам языка. Разница состоит только в том, что состояния (1-элементные множества слов), обозначенные числами от 1 до 6, заменяются на соответствующие слова: например, состояние 3 в новом автомате соответствует слову abb (а не языку $\{abb\}$). Поэтому мы не будем отдельно приводить изменённый вариант автомата – практически не отличающийся от рис. 7.

VII. (5) АВТОМАТ 100, Т. Е. $\mathrm{PRI}^{|\rightarrow}_{\uparrow}$

Этот вариант автомата и соответствующего ему алгоритма является, по-видимому, наиболее простым, и при этом наиболее близким к алгоритму из [8].

Формальное описание автомата

Для пары непустых конечных языков $A, B \subseteq \Sigma^*$, таких что $A \not\ni \varepsilon$ и $B \not\ni \varepsilon$, язык

$$\Phi_{A-B}(u) \subseteq \Sigma^*$$

определяется следующим образом. Рассматриваем все варианты $x \in B$, такие что $u \in \mathrm{pref}(x)$. Для каждого такого x проделываем следующее:

- обозначаем $x = uv$;
- рассматриваем все варианты $y \in A$, такие что $v \in \mathrm{pref}(y)$;
- обозначаем $y = vw$.

Все выбранные таким образом слова w в совокупности образуют искомый язык $\Phi_{A-B}(u)$.

Некоторые дополнительные пояснения приведены:

- на рис. 11 (кратко; для него считаем, что имеется m подходящих слов языка B, а для первого из них – n подходящих слов языка A);
- а также на рис. 12 (подробнее).

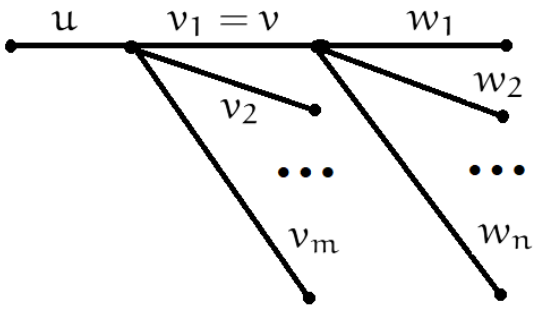

Рис. 11. Краткая иллюстрация алгортма, соответствующего автоматам, обозначенным в статье 100 и 101.

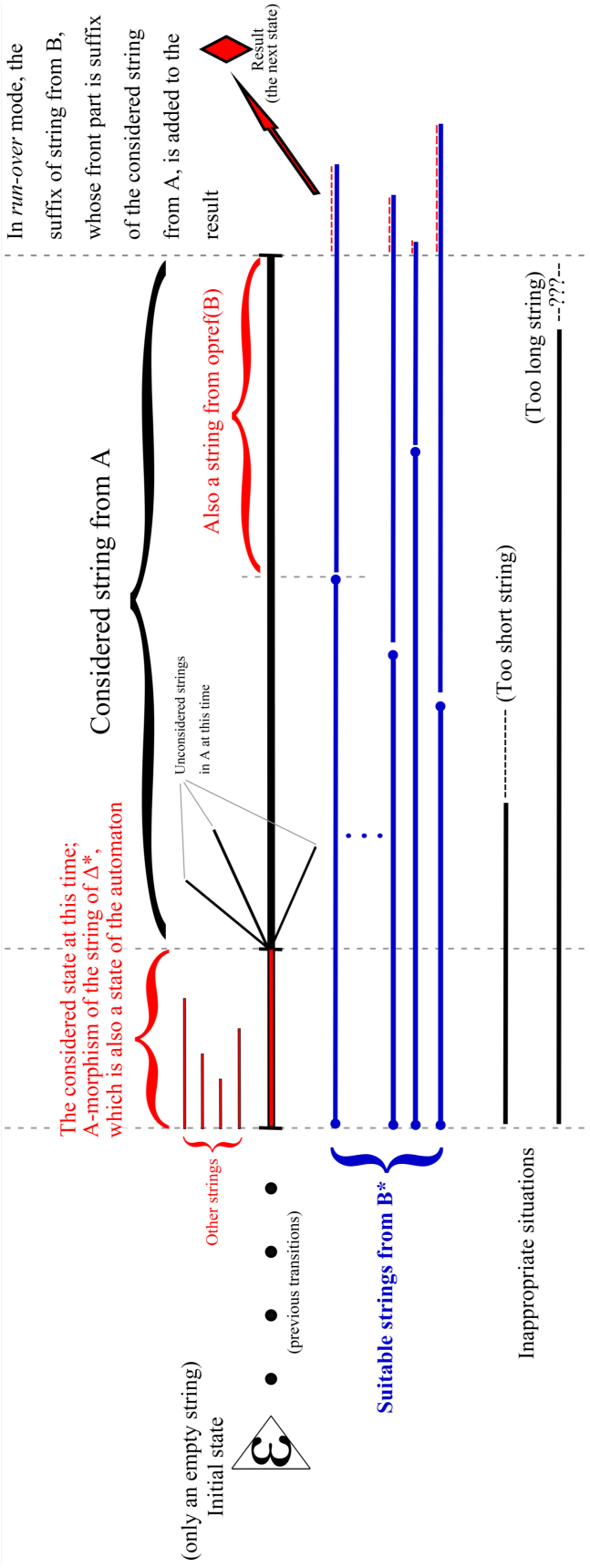

Рис. 12. Иллюстрация алгортма, соответствующего автоматам, обозначенным в статье 100 и 101. (Run-over mode, основной язык A.)

	a.1010	aaa	aabba	abba	bb
0	Ø	0	0	0	0
1	ε	2	0	11 12 13 14 15 16 17	4
2	a	12 13 14 17	0	0	1 2 12 13 14 17
4	b	0	0	0	12 13 14 17
11	aaa	18	11 12 13 14 15 16 17	0	0
12	aaaa	2	0	11 12 13 14 15 16 17	4
13	abb	2 18	11 12 13 14 15 16 17	11 12 13 14 15 16 17	4
14	abba	2 12 13 14 17	0	11 12 13 14 15 16 17	1 2 4 12 13 14 17
15	bb	0	0	0	16
16	bba	0	0	0	0
17	bbb	2	0	11 12 13 14 15 16 17	4
18	aa	11 15 16	0	0	0

Рис. 13. Автомат, обозначенный в статье 1010.

	automaton 1100	aaaa	abb	abba	bbb
0	Ø	0	0	0	0
1	{ε}	2	3	4	5
2	{aa,abba,bba}	6	7	8	5
3	{a}	9	3	4	0
4	{aaa,aabba,abba,bb}	2	3	4	5
5	{b}	0	0	0	4
6	{a,aa,aaa,aabba,abba,bb,bba}	6	7	8	5
7	{a,aaa,aabba,abba,bb}	10	3	4	5
8	{aa,aaa,aabba,abba,bb,bba}	6	7	8	5
9	{a,bba}	9	3	4	0
10	{a,aa,abba,bba}	6	7	8	5

Рис. 14. Автомат, обозначенный в статье 1100.

	automaton 1101	aba	abaab	abab	ababa	abababababa
0	Ø	0	0	0	0	0
1	{ε}	2	3	4	2	2
2	{ab,aba,abab,b,baab,baaba, baabab}	5	6	4	2	2
3	{ε,a,ab,abaab,abaaba, abaabab}	2	3	4	2	2
4	{ε,aab,aaba,aabab,ab, abaab,abaaba,abaabab}	2	3	4	2	2
5	{ε,ab,aba,abaab,abaaba, abaabab,abab,b,baab, baaba,baabab}	5	6	4	2	2
6	{ε,a,aab,aaba,aabab,ab, abaab,abaaba, abaabab}	2	3	4	2	2

Рис. 15. Автомат, обозначенный в статье 1101.

	automaton 1110	aaaa	abb	abba	bbb
0	Ø	0	0	0	0
1	ε	11 12 13	3	12 14 15 16	5
3	a	3 13	3	12 14 15 16	0
5	b	0	0	0	12 14 15 16
11	aa	12 14 15 16	12 14 15 16	11 12 13	0
12	abba	11 12 13	3	12 14 15 16	5
13	bba	3 13	3	12 14 15 16	0
14	aaa	11 12 13	3	12 14 15 16	5
15	aabba	11 12 13	3	12 14 15 16	5
16	bb	11 12 13	3	12 14 15 16	5

Рис. 16. Автомат, обозначенный в статье 1110.

Также аналогично сделанному ранее, для языка $C \subseteq \Sigma^*$ определяем

$$\underset{A-B}{\Phi}(C) = \bigcup_{u \in C} \underset{A-B}{\Phi}(u).$$

Дальнейшие определения (собственно автоматов $\overline{\text{PRI}}$ и далее PRI) полностью совпадают с приведёнными в разделе III (часть I) – конечно же, с заменой языка $\underset{A-B}{\Phi}(u)$ на новый его вариант; поэтому мы эти определения опускаем. Получающийся автомат будем обозначать $\text{PRI}^{|\rightarrow}_{\uparrow}$.

Примеры автоматов, дающих разные ответы

Такие примеры автоматов приведены выше на рис. 9 (отрицательный ответ) и на рис. 10 (положительный ответ).

VIII. (6) АВТОМАТ 101, Т. Е. $\text{NSPRI}^{|\rightarrow}_{\downarrow}$

Как и в предыдущих «чётных» разделах (раздел IV части I и раздел VI), определяемый автомат является недетерминированным аналогом автомата детерминированного, определяемого в каждом из соответствующих «нечётных» разделах.

Формальное описание автомата

Сначала для заданных конечных языков $A, B \subseteq \Sigma^*$ следующим образом определим (недетерминированный) конечный автомат $\text{NSPRI}^{\#}(A, B)$:

$$\text{NSPRI}^{\#}(A, B) = (\, Q^{\#}_B, \Delta_A, \delta^{\#}_{A-B}, \{\varepsilon\}, \varnothing\,),$$

где:

- $Q^{\#}_B = \text{opref}(B) \cup \{\varnothing\}$;
- для некоторых $q_1, q_2 \in Q^{\#}_B$ и $d \in \Delta_A$ мы полагаем наличие перехода

$$q_1 \xrightarrow[\delta^{\#}_{A-B}]{d} q_2$$

тогда и только тогда, когда

$$\underset{\{h_A(d)\}-B}{\Phi}(q_1) \ni q_2\,.$$

(Функция Φ строится на основе материала предыдущего раздела.)

Аналогично проделанному в предыдущих раздлах можно показать, что язык построенного автомата $\text{NSPRI}^{\#}(A, B)$ совпадает с языком соответствующего детерминированного автомата – в рассматриваемом нами случае автомата $\text{PRI}^{|\rightarrow}_{\uparrow}$.

Для заданных конечных языков $A, B \subseteq \Sigma^*$ (недетерминированный) конечный автомат $\text{NSPRI}^{\rightarrow|}_{\uparrow}(A, B)$ определяется следующим образом:

$$\text{NSPRI}^{\rightarrow|}_{\uparrow}(A, B) = (\, \text{opref}(B), \Delta_A, \delta^{\#}_{A-B}, \{\varepsilon\}, \text{opref}(B)\,),$$

где функция переходов $\delta^{\#}_{A-B}$ совпадает с введённой в этом разделе.

Пример автомата, дающего отрицательный ответ

Такой пример приведён выше на рис. 13.

IX. (7) АВТОМАТ 110, Т. Е. $\text{PRI}^{|\rightarrow}_{\downarrow}$

Автоматы, кратко рассматриваемые в этом и следующем разделах, (т. е. 110 – $\text{PRI}^{|\rightarrow}_{\downarrow}$ и 111 – $\text{NSPRI}^{|\rightarrow}_{\downarrow}$) получаются из автоматов 100 ($\text{PRI}^{|\rightarrow}_{\uparrow}$) и 101 ($\text{NSPRI}^{|\rightarrow}_{\uparrow}$) соответственно – путём замены всюду в их описаниях языка A на язык B и наоборот[3].

Полностью писывает действия алгоритма-автомата и рассмотренный ранее рис. 11 – тоже, конечно, с одновременной взаимной заменой A на B в комментариях к нему. Как обычно, приводим и рисунок с более подробным описанием действий алгоритма: рис. 18.

В связи со сказынным выше мы в этом и следующем разделах опускаем обычный подраздел «Формальное описание автомата».

Примеры автоматов, дающих разные ответы

Такие примеры автоматов приведены на рис. 14 (отрицательный ответ) и на рис. 15 (положительный ответ). Повторим для них комментарии из раздела IX: во-первых, про соответствующие языки см. информацию в части I; во-вторых, в рассматриваемом случае для получения окончательного ответа на вопрос о выполнении отношения покрытия нужны несложные дополнительные алгоритмы.

X. (8) АВТОМАТ 111, Т. Е. $\text{NSPRI}^{|\rightarrow}_{\downarrow}$

Пример автомата, дающего отрицательный ответ

Такой пример автомата приведён на рис. 16. Как видно из таблицы автомата, в нём можно объединить эквивалентные состояния. После объединения состояний получается автомат, приведённый на рис. 17.

(По поводу возможности такого объединения отметим следующее. Рассматриваемый автомат является недетерминированным – поэтому здесь неприменимы «обычные студенческие» алгоритмы объединения состояний детерминированного автомата. Подробные результаты о достаточных условиях такого объединения для недетерминированного варианта – а в частных случаях о необходимых и достаточных условиях – приведены в работе одного из авторов [9], а также кратко в более ранней статье [10]. Однако при рассмотрении автомата, приведённого на рис. 16, такие варианты объединения состояний очевидны: достаточно последовательно объявлять эквивалентными те состояния, для которых соответствующие им строки совпадают.)

	automaton 1110~	{aaaa}	{abb}	{abba}	{bbb}
0	Ø	0	0	0	0
1	ε/abba/aaa/aabba/bb	1 3 11	3	1	5
3	a/bba	3	3	1	0
5	b	0	0	0	1
11	aa	1	1	1 3 11	0

Рис. 17. Автомат, обозначенный в статье 1110, после применения очевидных эквивалентных преобразований.

[3] Сто́ит отметить, что здесь, по видимому, невозможно провести аналогию с «заменой» языков в паре автоматов 010 и 011 – для «получения» пары 000 и 001. Однако в том случае подобная «замена» не даёт приемлемого результата – а подробности можно понять на основе приведённого выше описания указанных автоматов.

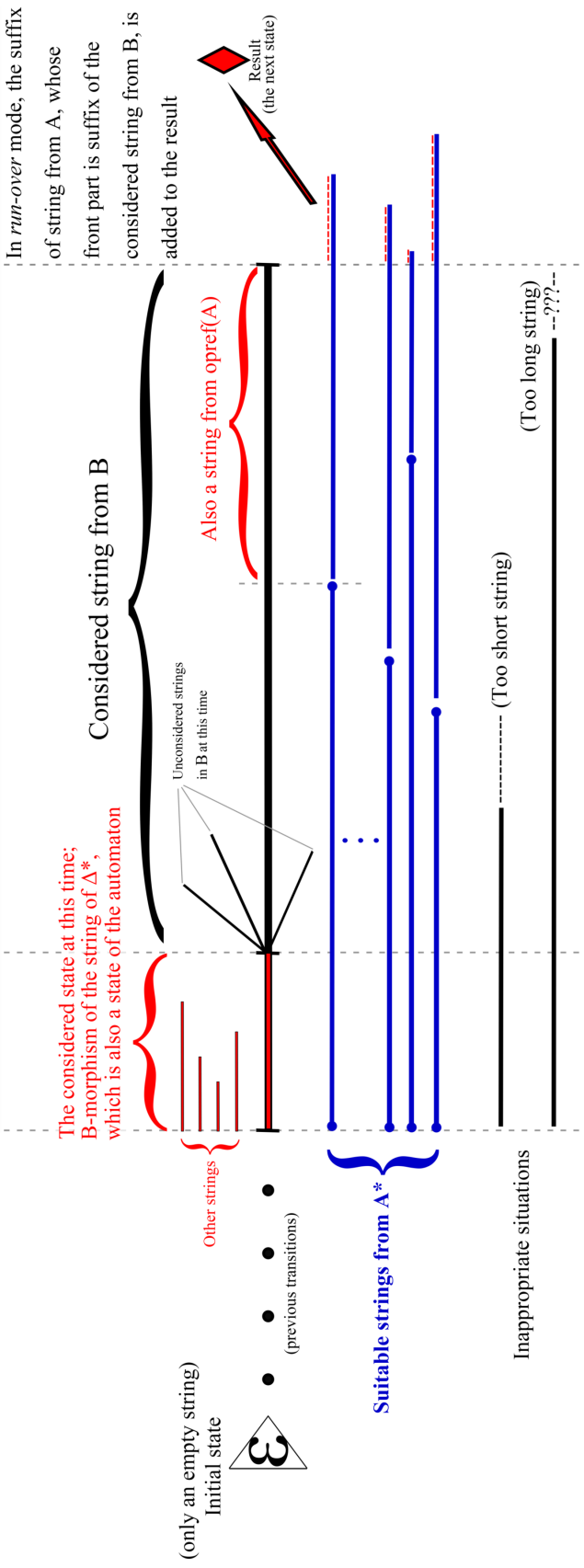

Рис. 18. Иллюстрация алгортма, соответствующего автоматам, обозначенным в статье 110 и 111. (Run-over mode, основной язык B.)

На рис. 17 в клетке таблицы автомата, соответствующей словам для состояния 1, записи «/» обозначают любое из разделённых этими знаками слов. Но при этом, конечно, в исходном автомате переход всегда был только в одно из них – а не во множество слов, как в детерминированном случае.

XI. ЗАКЛЮЧЕНИЕ

Данная статья предполагает несколько возможных вариантов продолжения.

Во-первых, мы выше несколько раз говорили, что из приведённых в статье автоматов два *формулируют работу* соответствующих алгоритмов, причём полную работу, – но *не возвращают* окончательный ответ. В связи с этим хотя бы для одного случая желательно построить соответствующий автомат, который будет возвращать окончательный ответ.

Кратко подобный автомат можно описать следующим образом. Мы – одновременно с рассмотрением языка A для «главного» языка B – «пропускаем через этот же автомат» сам исходный язык B (а не только язык A). При этом узнаём, какие именно *другие* слова из множества $\mathrm{pref}(B)$ могут быть префиксами некоторого рассматриваемого автоматом слова языка B^*. Например, в алгоритме для автомата 010 мы для некоторого слова $\nu \in B^*$ при этом проверяем, какие префиксы нужны для того, чтобы получить это слово ν – поэтому всё множество таких префиксов можно условно назвать «недоходами» (нам «не хватает» какого-либо из этих префиксов). Если при этом *не все* слова из языка A могут быть продолжены – то мы ставим вместо рассматриваемого перехода «выход», причём выход «плохой». То есть у итогового автомата состояния могут представлять собой *пары* следующего вида:

$$\{\text{пометки состояний уже описанного автомата}\} \times \{\text{соответствующие «недоходы»}\},$$

по таким «недоходам» формируются следующие состояния, а работа всего автомата заключается в том, что у него не должно быть ни одного пути ни в один из вышеупомянутых «плохих выходов».

Второе направление состоит в том, чтобы по произвольному недетермимнированному конечному автомату описать пару языков (в наших обозначениях – всегда A и B), которая может ему соответствовать – то есть, иыми словами, решить обратную задачу. Можно видеть, что это удобно делать не для «самого первого» определённого нами недетерминированного автомата NSPRI ([11] и др.), а для одного из алгоритмов-автоматов, где «главным» языков является не A, а B.

Третье направление связано с возможной полиномиальностью описываемых алгоритмов. Как несложно убедиться, все описанные автоматы соответствуют алгоритмам, полиномиальными не являющимся: мы либо работаем со множествами подмножеств слов (в случае детерминированных вариантов), либо не имеем возможности за полиномиальное время проверить результат работы (в случае недетерминированных вариантов). Но такие проблемы имеются в случае использования простых, *самых очевидных* алгоримов. Однако, может быть, как-то удастся воспользоваться тем, что при построении автоматов возникают *не любые* (а только некоторые специальные) подмножества множества суффиксов?

А в простом виде полиномиальный вариант алгоритма желательно описать для какого-либо такого построения, которое выше было названо вторым направлением.

БЛАГОДАРНОСТИ

Настоящая работа была частично поддержана грантом научной программы китайских университетов “Higher Education Stability Support Program” (раздел “Shenzhen 2022 – Science, Technology and Innovation Commission of Shenzhen Municipality”) – 深圳市 2022 年高等院校稳定支持计划资助项目.

Список литературы

[1] Мельников Б.Ф., Мэн Линцянь. Восемь вариантов конечных автоматов для проверки выполнения отношения покрытия итераций двух конечных языков. Часть I. // International Journal of Open Information Technologies. 2023. Vol. 11, No. 11. P. 1–9.

[2] Khoussainov B., Nerode A. Automata Theory and its Applications // Progress in Computer Science and Applied Logic, Vol. 21. – Springer, Berlin, 2001.

[3] 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.

[4] Мельников Б.Ф. Об ω-языках специальных биллиардов // Дискретная математика (РАН). – 2002. – №3. – С. 95–108.

[5] Melnikov B.F., Vakhitova A.A. Some more on the finite automata // Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). 1998. – No. 3. – P. 495–505.

[6] Melnikov B.F., Sciarini-Guryanova N.V. Possible edges of a finite automaton defining a given regular language // Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). 2002. – Vol. 9. No. 2. – P. 475–485.

[7] Melnikov B.F., Melnikova A.A. Some more on ω-finite automata and ω-regular languages. Part I: The main definitions and properties // International Journal of Open Information Technologies. 2020. Vol. 8, No. 8. P. 1–7.

[8] Мельников Б.Ф. Алгоритм проверки равенства бесконечных итераций конечных языков // Вестник Московского университета. Серия 15: Вычислительная математика и кибернетика. 1996. – №4. – С. 49–55.

[9] Melnikov B.F. Once more on the edge-minimization of nondeterministic finite automata and the connected problems // Fundamenta Informaticae. 2010. – Vol. 104. No. 3. – P. 267–283.

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

[11] Мельников Б.Ф. Варианты конечных автоматов, соответствующих бесконечным итерационным деревьям морфизмов. Часть I. International Journal of Open Information Technologies. 2021. Vol. 9, No. 7. P. 5–13.

Борис Феликсович МЕЛЬНИКОВ,
профессор Университета МГУ – ППИ в Шэньчжэне
(http://szmsubit.ru/),
email$_1$: bormel@smbu.edu.cn,
email$_2$: bf-melnikov@yandex.ru,
mathnet.ru: personid=27967,
elibrary.ru: authorid=15715,
scopus.com: authorId=55954040300,
ORCID: orcidID=0000-0002-6765-6800.

МЭН Линцянь,
магистрант Московского государственного университета им. М. В. Ломоносова
(https://cs.msu.ru/),
email$_1$: 1120190010@smbu.edu.cn,
email$_2$: kostya45e@mail.ru,
ORCID: orcidID=0009-0009-2933-5684.

Eight variants of finite automata for checking the fulfillment of the coverage relation of iterations of two finite languages. Part II

Boris Melnikov, Meng Lingqian

***Abstract*—The task for which all automata of this paper are considered is to describe algorithms for checking the so-called binary coverage relation, which was considered in many of our previous papers: it is performed for two nonempty finite languages if and only if any iteration of words of the first language is a prefix of some iteration of words of the second language. This task can be viewed from several different points of view, which was done in previous works.**

In this paper, we are primarily interested in various variants for implementing algorithms to verify the fulfillment of this relationship, and more specifically, the use of finite automata for such algorithms. 2 variants of such automata have already been previously described in our previous works, and in this paper we add 6 more variants to them.

The main purpose of considering several automata-algorithms is as follows. Earlier, we described a polynomial algorithm for constructing one of these automata. However, the existence of such a polynomial algorithm does not mean that we can check the fulfillment of the condition we need in a polynomial way, due to the following circumstance. If we build a nondeterministic automaton to answer the question in the same way as we described earlier, then for a positive answer. it must accept any word of the universal language; the last condition cannot be checked in polynomial time. However, we also cannot say that such a polynomial algorithm does not exist: additional research is needed. Therefore, the presence of four descriptions of nondeterministic automata that answer the question about the fulfillment of the considered binary relation will give some new opportunities to search for a possible algorithm that checks the main condition of the paper in polynomial time.

8 variants of the automata considered in the paper are naturally obtained using three different dichotomies: one of them is the "usual" one (whether the automaton is deterministic), and the other two dichotomies are directly related to the issues of checking the implementation of the binary coverage relation considered in the paper. Specifically, these two dichotomies are as follows: first, which of the two finite languages under consideration is the "main" one (it is that for which we apply the basic morphism to check); secondly, do we consider the required covers of each iteration word directly when it occurs, or not immediately, but after the formation of some auxiliary language.

In the presented second part of the paper, we continue to describe variants of finite automata for checking the fulfillment of the coverage relation.

***Keywords*—formal languages, iterations of languages, morphisms, finite automata, algorithms.**

References

[1] Melnikov B.F., Meng Lingqian. Eight variants of finite automata for checking the fulfillment of the coverage relation of iterations of two finite languages. Part I. // International Journal of Open Information Technologies. 2023. Vol. 11, No. 11. P. 1–9. (in Russian)

[2] Khoussainov B., Nerode A. Automata Theory and its Applications // Progress in Computer Science and Applied Logic, Vol. 21. – Springer, Berlin, 2001.

[3] 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.

[4] Melnikov B.F. On ω-languages of special billiards // Discrete Mathematics (Russian Academy of Sciences). – 2002. – No. 3. – P. 95–108. (in Russian)

[5] Melnikov B.F., Vakhitova A.A. Some more on the finite automata // Journal of Applied Mathematics and Computing. 1998. – No. 3. – P. 495–505.

[6] Melnikov B.F., Sciarini-Guryanova N.V. Possible edges of a finite automaton defining a given regular language // Korean Journal of Computational and Applied Mathematics (Journal of Applied Mathematics and Computing). 2002. – Vol. 9. No. 2. – P. 475–485.

[7] Melnikov B.F., Melnikova A.A. Some more on ω-finite automata and ω-regular languages. Part I: The main definitions and properties // International Journal of Open Information Technologies. 2020. Vol. 8, No. 8. P. 1–7.

[8] Melnikov B.F. Algorithm for checking the equality of infinite iterations of finite languages // Bulletin of the Moscow University. Series 15: Computational Mathematics and Cybernetics. 1996. – No. 4. – P. 49–55. (in Russian)

[9] Melnikov B.F. Once more on the edge-minimization of nondeterministic finite automata and the connected problems // Fundamenta Informaticae. 2010. – Vol. 104. No. 3. – P. 267–283.

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

[11] Melnikov B.F. Variants of finite automata corresponding to infinite iterative morphism trees. Part I. International Journal of Open Information Technologies. 2021. Vol. 9, No. 7. P. 5–13. (in Russian)

Boris MELNIKOV,
Professor of Shenzhen MSU – BIT University, China
(http://szmsubit.ru/),
email$_1$: bormel@smbu.edu.cn,
email$_2$: bf-melnikov@yandex.ru,
mathnet.ru: personid=27967,
elibrary.ru: authorid=15715,
scopus.com: authorId=55954040300,
ORCID: orcidID=0000-0002-6765-6800.

MENG Lingqian,
Master student of Moscow State Lomonosov University,
Russia (https://cs.msu.ru/),
email$_1$: 1120190010@smbu.edu.cn,
email$_2$: kostya45e@mail.ru,
ORCID: orcidID=0009-0009-2933-5684.