УДК 512.53, 510.54
DOI 10.21685/2072-3040-2017-3-8

Б. Ф. Мельников, С. Ю. Корабельщикова, Н. П. Чурикова

ОБ АЛГОРИТМАХ ПРОВЕРКИ ВЫПОЛНЕНИЯ НЕКОТОРЫХ БИНАРНЫХ ОТНОШЕНИЙ В ГЛОБАЛЬНОМ НАДМОНОИДЕ СВОБОДНОГО МОНОИДА

Аннотация.

Актуальность и цели. Предметом исследования являются свободные полугруппы и специальные бинарные отношения, заданные на глобальном надмоноиде свободного моноида (свободной полугруппы). Рассматриваются некоторые специальные бинарные отношения на языках (элементах глобального надмоноида), прежде всего так называемое отношение эквивалентности в бесконечности, рассматривавшееся нами в предыдущих работах и, как следует из названия, являющееся отношением эквивалентности. Мы описываем алгоритмы проверки выполнения этих отношений для элементов глобального надмоноида свободного моноида. Также рассматривается специальное отношение порядка на подмножествах множества слов, которое дает язык, минимальный по этому отношению среди всех языков, эквивалентных в бесконечности заданному.

Материалы и методы. Используются общие методы анализа и синтеза, а также специальные методы теории формальных языков, методы описания алгоритмов и методы работы с полугруппами, в частности, метод построения морфизма полугруппы.

Результаты. Приводятся примеры построения двух языков (двух элементов надмоноида), для которых в надмоноиде выполняется рассматриваемое нами отношение эквивалентности. Наиболее важный результат работы – описание эффективного алгоритма, отвечающего на вопрос, являются ли итерации двух заданных конечных языков эквивалентными в бесконечности. Приведенные алгоритмы и примеры актуальны для прикладных построения специальных вариантов автоматизированного преобразования регулярных грамматических структур и контекстно-свободных грамматик в системах автоматизации построения компиляторов.

Выводы. Во многих частных случаях, описанных нами в предыдущих публикациях, аналоги рассмотренных нами алгоритмов являются полиномиальными. Однако в общем случае вопрос о существовании полиномиальных алгоритмов, отвечающих на поставленные в статье вопросы, остается открытым.

Ключевые слова: формальные языки, бесконечные языки, свободный моноид, глобальный надмоноид (супермоноид), итерация языка.

B. F. Mel'nikov, S. Yu. Korabel'shchikova, N. P. Churikova

ON VERIFICATION ALGORITHMS FOR SOME BINARY RELATIONS ON THE GENERAL SUPERMONOID OF A FREE MONOID

Abstract.

Background. The subjects of the study are free semigroups and special binary relations (free semigroup) defined on the global supermonoid of a free monoid. In this paper, we consider some special binary relations in languages (elements of the glob-

al supermonid), especially the so-called equivalence relation at infinity, which we considered in previous papers and which is an equivalence relation. We describe algorithms for verifying the fulfillment of these relations for elements of the global supermonoid of a free monoid. We also consider a special order relation on subsets of a set of words that is given by a language that is minimal in this relation among all languages equivalent at infinity to the given one.

Material and methods. General methods of analysis and synthesis are used in this work. Special methods of the theory of formal languages, methods for describing algorithms and methods of working with semigroups are also used, in particular, the method of constructing a morphism of a semigroup.

Results. The article gives examples of constructing two languages (two elements of a supermonoid) for which the equivalence relation we are considering is satisfied in the supermonoid. The most important result of the paper is a description of the effective algorithm that answers the question whether the iterations of two given finite languages are equivalent at infinity. Algorithms and examples presented in the article are relevant for the application of special variants of the automated transformation of regular grammatical structures and context-free grammars in compiler building automation systems.

Conclusns. In many special cases, described by us in previous publications, the analogues of the considered algorithms are polynomial. However, in the general case, the question of the existence of polynomial algorithms that answer the questions posed in the article remains open.

Key words: formal languages, infinite languages, free monoid, supermonoid, iteration of language.

Введение

В настоящей статье мы рассматриваем один из возможных подходов к построению алгоритмов проверки равенства бесконечных итераций множеств слов. При этом мы имеем в виду алгоритмы, близкие не к рассматриваемым в [1] (они фактически включают в себя перебор подмножеств достаточно большого множества слов, поэтому их, по-видимому, сложнее довести до *реально выполняемых* алгоритмов), а к рассматриваемым в [2]. Иными словами – для элементов глобальных надмоноидов свободных моноидов [3] нами, в частности, рассматриваются задачи, близкие к проблеме равенства $A^m = B^n$.

С рассматриваемыми нами проблемами тесно связана задача «извлечения корня» из языка: для заданного языка A требуется найти максимально возможное $n \in N$ и зависящий от n язык B такие, что $A = B^n$. До конца эта задача авторами еще не исследована, однако в частном случае получен ряд результатов, опубликованных в [4, 5]. Также очевидна связь этой проблемы с некоторыми вопросами теории формальных языков, в частности, с итерациями языков, исследовавшихся нами во всех уже упомянутых статьях [1–5], а также в других публикациях. Действительно, A и B можно рассматривать как произвольные языки над одним и тем же, причем не обязательно конечным, алфавитом Σ. При этом, в отличие, например, от [5], мы в настоящей статье *не будем* считать, что по крайней мере один из языков обладает свойством префикса.

Ниже мы рассмотрим алгоритмы проверки выполнения специальных бинарных отношений для элементов глобального надмоноида свободного моноида. Некоторые из этих алгоритмов уже были приведены нами в предыдущих публикациях, например в [2], однако при этом:

• некоторые из применяемых нами вспомогательных алгоритмов ранее публиковались только на английском языке, причем в малодоступных для отечественного читателя журналах ([1] и особенно [6]);

• сами соответствующие доказательства корректности алгоритмов (например, утверждение 8) изменены и улучшены; в них также исправлены замеченные опечатки;

• все алгоритмы сформулированы согласно новым обозначениям, согласованным с [3, 4]; эти обозначения значительно более удобны для формулировки алгебраических результатов, см. [5, 7] и др.;

• для сформулированных в настоящей статье алгоритмов проще (чем для формулировок, приведенных в [2]) формулировать оценки сложности; мы предполагаем привести некоторые из таких оценок в одной из следующих публикаций;

• приведенные здесь формулировки являются началом реализации кратко сформулированного в [3] плана *переформулировки* проблемы *P = NP в терминах итераций конечных языков*.

Некоторые *прикладные* вопросы полученных результатов связаны с публикациями одного из авторов настоящей статьи [8, 9]. В частности, приведенные в статье примеры актуальны для прикладных задач построения специальных вариантов автоматизированного преобразования регулярных грамматических структур и контекстно-свободных грамматик в системах автоматизации построения компиляторов. Кроме того, аналогично [3, 4] мы показываем связь рассматриваемого в статье материала с некоторыми вопросами теории недетерминированных конечных автоматов. В первую очередь мы имеем в виду статьи [10–13]: возникающие далее подмножества некоторого языка, обрабатываемые некоторым недетерминированным конечным автоматом, при специальной переформулировке (которую мы предполагаем привести в одной из следующих публикаций) дадут определение рассматриваемых в этих статьях универсальных автоматов.

1. Предварительные сведения

Ниже буквой ε будем обозначать единицу исходного свободного моноида, т.е. пустое слово над заданным алфавитом Σ. Множество всех слов над алфавитом Σ, включая ε, будем обозначать Σ^*. Множество всех языков над некоторым алфавитом Σ с операцией конкатенации «·» образует свободный моноид с единицей {ε}, являющийся для исходного свободного моноида $(\Sigma^*, \cdot, \varepsilon)$ *глобальным надмоноидом*, элементами которого являются не слова, а множества слов, т.е. языки.

Если $u = a_1 a_2 \ldots a_n$, а $v = a_1 a_2 \ldots a_m$, где $0 \le m \le n$, то слово v будем называть префиксом слова u и писать $v \in \mathrm{pref}(u)$ (или $v \in \mathrm{pref}_m(u)$, явно указывая длину); при этом можно считать, что $\mathrm{pref}(u)$ – множество всех префиксов слова u. Обозначение pref будем естественным образом использовать и для языка (возможно, бесконечного): это объединение всех префиксов всех его слов. $\mathrm{opref}(u)$ – множество всех собственных префиксов слова u, т.е. множество $\mathrm{pref}(u)$ без слова u.

Стандартным образом вводятся следующие понятия: ω-слово – бесконечная последовательность букв, ω-язык – множество ω-слов; см. [14] и др.

Среди всех ω-языков нас будут в наибольшей степени интересовать языки вида A^{ω} для некоторого языка $A \subseteq \Sigma^*$, т.е. языки, каждое ω-слово которых можно представить (для заданного языка A) в виде

$$\alpha = u_1 u_2 \ldots u_n \ldots \ (u_i \in A \text{ для всех } u_i).$$

Обозначение морфизма $h: \Delta^* \to \Sigma^*$ также вводится обычным образом, т.е. мы расширяем некоторое заданное отображение $h: \Delta \to \Sigma^*$ на случай всех входов, принадлежащих Δ^*, следующим образом: для каждого $u \in \Sigma^*$, $u = a_1 a_2 \ldots a_n$, мы считаем, что

$$h(a_1 a_2 \ldots a_n) = h(a_1) h(a_2) \ldots h(a_n).$$

Определение 1. Пусть U и V – бесконечные языки. Если одновременно выполнены условия

$$U \subseteq \text{pref}(V) \quad \text{и} \quad V \subseteq \text{pref}(U), \tag{1}$$

то будем писать $U \cong V$ и называть языки U и V *эквивалентными в бесконечности*.

В предыдущих работах нами использовались и другие обозначения для этого же бинарного отношения. Более того, в некоторых работах аналогичное отношение вводилось на *конечных* множествах – пусть A и B – таких, что в бинарном отношении, аналогичном введенному выше в определении 1, находятся бесконечные языки A^* и B^*. Однако все это не должно затруднить прочтение настоящей статьи, поскольку, как было отмечено выше, все необходимые определения мы повторяем.

Самым простым примером пары языков, эквивалентных в бесконечности, могут служить $U = \{a^{2i} | i \in N\}$ и $V = \{a^{2i-1} | i \in N\}$. Целый класс более сложных примеров таких пар языков будет рассмотрен в следующем разделе.

Пользуясь определением 1, можно показать, что *необходимым и достаточным* условием существования какого-либо языка V, эквивалентного в бесконечности заданному языку U, является следующее:

$$(\forall u \in U)(\exists v \in U)(u \in \text{opref } v). \tag{2}$$

Условию (2) удовлетворяют, например, упомянутые выше языки A^* при любом A (мы всюду будем считать, что $A \neq \varnothing$ и $\varepsilon \notin A$).

Легко доказывается, что на множестве языков, удовлетворяющих, условию (2), введенное отношение является отношением эквивалентности (т.е. название отношения корректно).

Немного переформулировав условие (2) (причем снова рассматривая два бесконечных языка U и V), мы вводим обозначение $\underset{\infty}{\subset}$ следующим образом: $U \underset{\infty}{\subset} V$ *тогда и только тогда, когда*

$$(\forall u \in U)(\exists v \in V)(u \in \text{pref } v).$$

К этой записи приведем два замечания. Во-первых, понятно, что в данном случае употребление pref вместо opref непринципиально. Во-вторых, объединение условий $U \underset{\infty}{\subset} V$ и $V \underset{\infty}{\subset} U$ дает в точности (1).

Еще два важных обозначения будут введены в следующем разделе после описания свойств введенного нами отношения $\cong$.

2. Сложные примеры эквивалентных пар языков

Несложно доказываемое *достаточное* условие равенства $A^* \cong B^*$ формулируется в виде *алгоритма*, состоящего из следующих трех шагов ([1, 4] и др.).

1. Рассматриваем новый конечный алфавит Δ и два некоторых максимальных префиксных кода [1] над этим алфавитом; пусть это A и B. При этом условие $A^* \cong B^*$ выполнено, поскольку $A^\omega = B^\omega = \Delta^\omega$.

В качестве примера рассмотрим алфавит $\Delta = \{0,1\}$ и два таких языка: $A = \{0,10,11\}$ и $B = \{00,01,1\}$.

2. Добавляем некоторые слова к языкам A и B (для A при этом достаточно рассматривать только такие слова u, для которых $u \notin pref(A)$, аналогично для B). При этом условие $A^* \cong B^*$ выполнено, поскольку и равенство $A^\omega = B^\omega$ тоже выполняется. Заметим, что полученные языки, вообще говоря (если мы добавляли более 1 слова), не являются префиксными; согласно [4] будем их называть *дополненными максимальными префиксными кодами*.

В качестве продолжения предыдущего примера рассмотрим языки $A = \{0,10,11,1101\}$ и $B = \{00,01,1,1110\}$.

3. Для полученных языков рассмотрим некоторый морфизм $h: \Delta^* \to \Sigma^*$; заметим, что язык $h(D) = \{h(c) \mid c \in D\}$ тоже, вообще говоря, не является префиксным. Для полученных языков (здесь их тоже удобно обозначать A и B) условие $A^* \cong B^*$ также выполняется.

Продолжим рассматривать предыдущий пример. Для морфизма $h: \Delta^* \to \Sigma^*$ такого, что $h(0) = ab$, $h(1) = aba$, мы получаем следующие языки: $\{ab, abaab, abaaba, abaabaababa\}$ и $\{abab, ababa, aba, abaabaabaab\}$.

Итак, очевидное достаточное условие таково: если языки A и B получены описываем здесь способом, то $A^* \cong B^*$.

Обозначения mp и mp^+ вводим согласно [1]; например, для приведенных выше примеров языков выполнено условие

$$\{ab, abaab, abaaba, abaabaababa\} \in mp^+(\{ab,\ aba\}).$$

3. Основные утверждения

Утверждение 1. Если каждое из множеств U и V удовлетворяет условию (2), то условие $U \cong V$ выполняется тогда и только тогда, когда

$$(\forall n \in N)(\mathrm{pref}_n(U) = \mathrm{pref}_n(V)).$$

Сформулированное утверждение 1 доказывается несложно. В качестве его следствия можно доказать, что равносильным определением отношения эквивалентности в бесконечности для удовлетворяющих условию (2) языков является следующее: $\mathrm{pref}(U) = \mathrm{pref}(V)$.

Утверждение 2. $A^* \cong \Sigma^*$ тогда и только тогда, когда $A \in mp^+(\Sigma)$.

Теорема 1. Пусть $A^* \cong B^*$. Тогда существует язык D такой, что $A^* \cong D^*$ и $A, B \in mp^+(D)$.

Упрощенный (префиксный вариант) этой теоремы формируется следующим образом.

Теорема 2. Пусть $A^* \cong B^*$, при этом язык A обладает свойством префикса. Тогда существует префиксный язык D такой, что $A^* \cong D^*$, причем $A \in mp(D)$ и $B \in mp^+(D)$.

Из теорем 1 и 2 несложно выводится, что для любого языка A существует единственный язык D_A, такой что $A^* \cong D_A^*$ (более того, $A \in mp^+(D_A)$), и при этом ни для какого D, отличного от D_A, не выполняется условие $D_A \in mp^+(D)$. Соответственно в префиксном случае: теорема 2, отношение mp (вместо mp^+), условия $A \in mp(D_A)$ и $D_A \notin mp(D)$.

Как в префиксном, так и в общем случае можно специальным образом [1] определить некоторое отношение порядка на подмножествах Σ^* так, что язык D_A будет минимальным по этому отношению среди всех языков, эквивалентных в бесконечности заданному A. На основе этого факта можно получить следующий *алгоритм* проверки условия $A_1^* \cong A_2^*$ в конечном случае: для $i = 1,2$ нужно построить множества D_{A_i} и сравнить их; при этом условие $A_1^* \cong A_2^*$ будет равносильно равенству построенных языков $D_{A_1} = D_{A_2}$.

Обсуждение *эффективного* алгоритма решения этой задачи (построения языка D_A по заданному языку A) не входит в круг вопросов настоящей статьи. Очевидно, что простейший переборный алгоритм (а именно такой: рассмотреть все $D \in \Sigma^*$ такие, что $D_{\max} \le A_{\max}$, и среди удовлетворяющих условию $A \in mp^+(D)$ выбрать нужное D_A) можно значительно улучшить, например следующим образом. Обозначим

$$\tilde{D} = D\left\{u \mid (\exists n \in N, v \in A)(v = u^n)\right\}.$$

Тогда для выбора D_A можно рассматривать только такие D, что $D \subseteq (\tilde{D})^*$ и $D_{\max} \le A_{\max}$.

Применение этих алгоритмов (или, иными словами, алгоритмов сравнения ω-языков такого вида) *в различных задачах теории формальных языков* описано, например, в [3–6, 8, 9, 15] и др. А дальнейшая часть настоящей статьи посвящена описанию альтернативного алгоритма, *не использующего* результаты рассмотренных выше теорем 1 и 2.

[1] Важно отметить, что это можно сделать многими способами. В конкретных задачах теории формальных языков способ определения этого отношения порядка может быть важен. Наш случай прост: для наших целей достаточно выбрать норму, равную длине максимального слова рассматриваемого конечного языка.

4. Вспомогательный алгоритм

Сначала опишем вспомогательный алгоритм проверки условия $A^* \underset{\infty}{\subset} B^*$. При описании алгоритма будем использовать естественные обозначения $:=$ («положить равным») и $:+$ («добавить»). Эти обозначения употребляются для всех используемых в алгоритме переменных: переменных-слов, переменных-множеств, переменных-функций.

Алгоритм 1. *Вход:* $A=\{u_1,\ldots,u_n\}$, B.

Выход: 1 – если $A^* \underset{\infty}{\subset} B^*$;

0 – в противном случае.

Некоторые замечания к описанию алгоритма. Будем рассматривать алфавит $\Delta_A=\{d_1,\ldots,d_n\}$. Слова в этом алфавите и языки над ним будут обычно иметь нижний индекс Δ. Функция $F: P(\Sigma^*)\to P(\Sigma^*)$ определяется следующим образом:

$$F(C)=\left\{v\in \operatorname{opref}(B)\mid \left(\exists u\in B^*\right)(uv\in C)\right\}.$$

Будут использоваться переменные-функции p, s (не всюду определенные) и переменные-множества H, L следующего вида:

$$L\subseteq\Delta^*,\ H\subseteq P(\Sigma^*),\ p\subseteq\Delta'\times P(\Sigma^*),\ s\subseteq\Delta'\times\Delta^*,$$

а также переменная-слово $w\in\Delta^*$.

Если алгоритм даст ответ 0, то будем при этом фиксировать элемент $w\in\Delta^*$ такой, что $h_A(w_A)\notin \operatorname{pref}(B^*)$.

Описание.

Шаг 0. $L:=\varnothing, H:=\{\{e\}\}$, $p(e):=\{e\}$, функция $p(u_\Delta)$ при $u_\Delta\neq e$ и функция $s(u_\Delta)$ при любом u_Δ не определены.

Шаг 1. Выбрать какое-либо из таких $u_\Delta\in\Delta^*$, что определено $p(u_\Delta)$ и не определены $p(u_\Delta d_i)$ при всех $i\in\overline{1,n}$.

Шаг 2. Для всех $i\in\overline{1,n}$ выполнить следующие действия (будем здесь обозначать $p_i=p(u_\Delta d_i)$.

2.1. $p_i=F(p(u_\Delta u_i))$.

2.2. Если $(\exists C\in H)(C\subset p_i)$,

то $L:+\{u_\Delta d_i\}$ и $9(u_\Delta):=p^{-1}(C)$,

иначе $H:+p_i$ и $s(u_\Delta d_i):=u_\Delta d_i$.

2.3. Если $p_i=\varnothing$,

то выход с ответом 0 и зафиксированным элементом $u_\Delta d_i$.

Шаг 3. $w := u_\Delta$.

Шаг 4. Если $(\forall i \in \overline{1,n})\left(wd_i \in L\right)$,

то $L :+ \{w\}$,

иначе переход на шаг 1.

Шаг 5. Если $w = e$, то выход из алгоритма с ответом 1.

Шаг 6. $w := \mathrm{pref}_{|w|-1}(w)$ и переход на шаг 4.

(Конец описания алгоритма.)

Предположим теперь, что к некоторым конечным языкам $A, B \subset \Sigma^*$ применен алгоритм 1. Будем употреблять все встречающиеся в его описании обозначения и в последующем тексте. Рассмотрим также функции $S \in \Delta^* \times \Delta^*$ и $P \subseteq \Delta^* \times P\left(\Sigma^*\right)$, заданные следующим образом:

$$S(e) = e, \left(\forall u_\Delta \in \Delta^*\right)(\forall d \in \Delta)\left(S\left(u_\Delta d\right) = s\left(S\left(u_\Delta\right)d\right)\right);$$

$$\left(\forall u_\Delta \in \Delta^*\right)\left(P\left(u_\Delta\right) = p\left(S(u_\Delta)\right)\right).$$

Будем ниже говорить, что некоторое слово $u \in \Sigma^*$ покрыто, если оно является префиксом какого-либо слова из B^*. Само слово $w = v_1 \ldots v_m$ (где $v_i \in B$ для $i \in \overline{1,m}$) при этом назовем покрытием слова u, если $u \in \mathrm{pref}(w)$, но $u \notin \mathrm{pref}(v_1 \ldots v_{m-1})$. Для некоторого u все слова v из множества $\mathrm{opref}(B)$ такие, что слова uv являются его покрытиями, обозначим $F\left(\{u\}\right)$. Для произвольного множества C положим $F\left(C\right) = \bigcup_{u \in C} F\left(\{u\}\right)$.

Мы показываем, что условие $A^* \subseteq B^*$ выполняется тогда и только тогда, когда все слова из A^* покрыты. Однако проверить, покрыто ли все бесконечное множество слов над алфавитом Δ, мы не можем, поэтому в алгоритме неявно употребляется специальное отношение эквивалентности, заданное на Σ^*: два слова $u, v \in \Sigma^*$ эквивалентны, если $F\left(\{u\}\right) = F\left(\{v\}\right)$. Из предыдущего следует, что для каждого класса эквивалентности по этому отношению достаточно проверить, покрыто ли какое-нибудь одно слово из этого класса.

Пусть L – множество тех слов из Δ^*, для которых уже проверено, покрыты ли слова языка $h_A(L)$; H – совокупность множеств $F\left(\{u\}\right)$ для всех $u \in h_A(L)$. Если $s\left(u_\Delta\right) = v_\Delta$, то слова $h_A(u_\Delta)$ и $h_A(v_\Delta)$ принадлежат одному классу эквивалентности по описанному выше отношению. В $p(u_\Delta)$ мы засылаем некоторое множество, содержащее $F(h_A(u_\Delta))$, причем такое, что оно само является значением функции $F(h_A(v_\Delta))$ для некоторого $v_\Delta \in \Delta^*$. Заглавные буквы S и P обозначают функции, аналогичные обозначаемым соответствующими строчными, но определенные всюду, где это возможно, а не только для слов u_Δ, обрабатываемых алгоритмом 1.

В процессе работы алгоритма для всех возможных $u_\Delta \in \Delta^*$ функции S и P не строятся, но совершенно несложными являются доказательства их существования, а также возможности построить их значении для любого конкретного $u_\Delta \in \Delta^*$, такого что слово $h(u_\Delta)$ покрыто.

Утверждение 3. При каждом выполнении шага 1 выбор требуемого u_Δ осуществим, причем любое выбираемое слово отлично от всех выбиравшихся ранее.

Утверждение 4. Во время работы и в момент окончания алгоритма 1 переменная-функция p всегда является сюръекцией, причем при выполнении подшага 2.2 значение $p^{-1}(C)$ всегда существует.

Последние два утверждения, справедливость которых следует из указанных шагом 1 требований к u_Δ и способа построения p, показывают, что требуемые алгоритмом действия всегда возможны.

Утверждение 5. Для любых конечных языков $A, B \subset \Sigma^*$ алгоритм 1 завершает работу.

Утверждение 6. $F(A \cdot B) = F(F(A) \cdot B)$.

Утверждение 7. Если алгоритм 1 закончил работу с ответом 1, то при любом $u_\Delta \in \Delta^*$ определены значения $S(v_\Delta)$ и $P(v_\Delta)$, причем $P(v_\Delta) \neq \varnothing$.

Справедливость последних двух утверждений непосредственно следует из приведенных ранее определений функций F, S и P.

Утверждение 8. Если значение $p(\omega_\Delta)$ определено, то

$$p(\omega_\Delta) = F(h_A(\omega_\Delta)). \quad (3)$$

Доказательство проведем индукцией по значению $|\omega_\Delta|$. По определению, $p(e) = F(h_A(e))$, поэтому база индукции выполняется; покажем выполнение шага.

Пусть равенство (3) выполняется для некоторого (известного заранее) слова $\omega_\Delta \in \Delta^*$, а для некоторой (произвольной) буквы $d \in \Delta$ имеем $v_\Delta = \omega_\Delta d$ (т.е. $|v_\Delta| = |\omega_\Delta| + 1$); пусть, кроме того, определено $p(v_\Delta)$. Достаточно показать, что доказываемое нами утверждение выполнено для данного v_Δ (поскольку, по определению функции p, если существует $p(v_\Delta)$, то при любом $u_\Delta \in \mathrm{pref}(u_\Delta)$ определено и значение $p(u_\Delta)$.

Применяя выражение из описания подшага 2.1, а также предположение индукции (3) и утверждение 6, получаем следующую цепочку равенств:

$$p(v_\Delta) = F(p(\omega_\Delta) \cdot h_A(d)) = F(F(h_A(\omega_\Delta)) \cdot h_A(d)) =$$

$$= F(h_A(\omega_\Delta) \cdot_A (d)) = F(h_A(v_\Delta),$$

т.е. для указанного слова v_Δ выполняется условие, аналогичное (3) (полученное из (3) заменой ω_Δ на v_Δ).

Утверждение 9. Если алгоритм 1 закончил работу с ответом 1, то для произвольного $\omega_\Delta \in \Delta^*$ выполнено условие $P(\omega_\Delta) \subseteq F\left(h_A(\omega_\Delta)\right)$.

Доказательство этого утверждения аналогично доказательству предыдущего. Оно также проводится индукцией по значению $|\omega_\Delta|$.

Теорема 3. Следующие три утверждения равносильны:

$$A^\omega \subseteq B^\omega; \tag{4}$$

$$A^* \overset{\subset}{\infty} B^*; \tag{5}$$

алгоритм 1 заканчивает работу с ответом 1. (6)

Доказательство. Пусть выполнено условие (6). Возьмем любое $\alpha \in A^\omega$, $\alpha = \prod\limits_{i \in N} w_i$, где все $w_i \in A$. Для каждого $n \in N$ обозначим

$$u_{\Delta,n} = \prod_{1 \le i \le n} h_A^{-1}(w_i).$$

Вследствие утверждений 7 и 10, $(\forall n \in N)\left(F\left(h_A\left(u_{\Delta,n}\right)\right) \neq \varnothing\right)$, и по определениям F и h_A мы получаем следующее:

$$(\forall n \in N)\left(\exists u_1 \in B^*, u_2 \in \operatorname{opref}(B)\right)\left(\prod_{1 \le i \le 1} \omega_i = u_1 v_2\right).$$

Поэтому существует функция $f_\alpha : N \to B^*$ такая, что

$$(\forall n \in N)\left(\prod_{1 \le i \le n} f_\alpha(i) \in \operatorname{pref}\left(\prod_{1 \le i \le 1} \omega_i\right)\right) \& \left(\prod_{1 \le i \le 1} \omega_i \in \operatorname{pref}\left(\prod_{1 \le i \le n} f_\alpha(i)\right)\right). \tag{7}$$

Рассмотрим ω-слово $\beta = \prod\limits_{i \in N} f_\alpha(i)$. По построению, $\beta \in B^\omega$. Предположим, что $\beta \neq \alpha$, тогда для некоторого $n \in N$ имеем $\operatorname{pref}_n(\alpha) \neq \operatorname{pref}_n(\beta)$, откуда выводится противоречие с (7). Таким образом, построенное нами β совпадает с выбранным α. Поскольку выбиралось любое $\alpha \in A^\omega$, заключаем, что $A^\omega \subseteq B^\omega$, таким образом, из (6) следует (4); (5) следует из (4), согласно утверждению 2.

Теперь докажем, что из (5) следует (6). Предположим противное: (5) выполнено, а (6) – нет. Тогда в силу утверждения 5 алгоритм 1 заканчивает работу с ответом 0. Для зафиксированного при выходе из алгоритма элемента (пусть это v_Δ) выполнено условие $p(v_\Delta) \neq \varnothing$ (это ***следует*** из описания алгоритма 1) и (согласно утверждению 8) $F\left(h_A(v_\Delta)\right) = \varnothing$. По определению функции F, $\left(\forall u \in B^*\right)\left(h_A(v_\Delta) \notin \operatorname{pref}(u)\right)$, а это противоречит условию (5).

Заключение

Аналогично [2] (но, вообще говоря, с иными накладываемыми условиями), мы, дважды применяя алгоритм 1, получаем алгоритм проверки условия $A^{\omega} = B^{\omega}$. Кроме того, мы получаем такое следствие: равенство $A^{\omega} \subseteq B^{\omega}$ и условие $A \cong B^{*}$ равносильны.

Библиографический список

1. **Melnikov, B. F.** The equality condition for infinite catenations of two sets of finite words / B. F. Melnikov // International Journal of Foundation of Computer Sciences. – 1993. – Vol. 4, № 3. – P. 267–273.
2. **Мельников, Б. Ф.** Алгоритм проверки равенства бесконечных итераций конечных языков / Б. Ф. Мельников // Вестник Московского университета. Сер. 15, Вычислительная математика и кибернетика. – 1996. – № 4. – С. 49–55.
3. **Мельников, Б. Ф.** Описание специальных подмоноидов глобального надмоноида свободного моноида / Б. Ф. Мельников // Известия высших учебных заведений. Математика. – 2004. – № 3. – С. 46–56.
4. **Алексеева, А. Г.** Итерации конечных и бесконечных языков и недетерминированные конечные автоматы / А. Г. Алексеева, Б. Ф. Мельников // Вектор науки Тольяттинского государственного университета. – 2011. – № 3. – С. 30–33.
5. **Корабельщикова, С. Ю.** Об общем корне элементов глобального надмоноида / С. Ю. Корабельщикова, Б. Ф. Мельников // Вестник Северного (Арктического) федерального университета. Сер.: Естественные науки. – 2016. – № 3. – С. 91–96.
6. **Melnikov, B.** Some equivalence problems for free monoids and for subclasses of the CF-grammars class / B. Melnikov // Number theoretic and algebraic methods in computer science. – 1995. – P. 125–137.
7. **Данг, В. В.** О задаче нахождения минимальной полугруппы аппроксимации / В. В. Данг, С. Ю. Корабельщикова, Б. Ф. Мельников // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2015. – № 3 (35). – С. 88–99.
8. **Дубасова, О. А.** Об одном расширении класса контекстно-свободных языков / О. А. Дубасова, Б. Ф. Мельников // Программирование. – 1995. – № 6. – С. 46–57.
9. **Melnikov, B.** Some grammatical structures of programming languages as simple bracketed languages / B. Melnikov, E. Kashlakova // Informatica (Lithuanian Acad. of Sciences). – 2000. – Т. 11, № 4. – С. 441–446.
10. **Баумгертнер, С. В.** Обобщенные недетерминированные конечные автоматы / С. В. Баумгертнер, Б. Ф. Мельников // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2013. – № 2 (26). – С. 64–74.
11. **Долгов, В. Н.** Построение универсального конечного автомата. I. От теории к практическим алгоритмам / В. Н. Долгов, Б. Ф. Мельников // Вестник Воронежского государственного университета. Сер.: Физика. Математика. – 2013. – № 2. – С. 173–181.
12. **Доглов, В. Н.** Построение универсального конечного автомата. II. Примеры работы алгоритмов / В. Н. Долгов, Б. Ф. Мельников // Вестник Воронежского государственного университета. Сер.: Физика. Математика. – 2014. – № 1. – С. 78–85.
13. **Зубова, М. А.** Вспомогательные алгоритмы для проблемы вершинной минимизации недетерминированных конечных автоматов / М. А. Зубова, Б. Ф. Мельников // Стохастическая оптимизация в информатике. – 2015. – Т. 11, № 2. – С. 17–29.
14. **Cohen, R. S.** Theoryof ω-languages. I. Caractirizations of ω-context-free lahguages / R. S. Cohen, Y. Gold // J. Comp and System Sci. – 1977. – № 15. – P. 169–184.

15. **Мельников, Б. Ф.** Некоторые следствия условия эквивалентности однозначных скобочных грамматик / Б. Ф. Мельников // Вестник Московского университета. Сер. 15, Вычислительная математика и кибернетика. – 1991. – № 3. – С. 51–53.

References

1. Melnikov B. F. *International Journal of Foundation of Computer Sciences*. 1993, vol. 4, no. 3, pp. 267–273.
2. Mel'nikov B. F. *Vestnik Moskovskogo universiteta. Ser. 15, Vychislitel'naya matematika i kibernetika* [Bulletin of Moscow University. Series 15. Calculus mathematics and cybernetics]. 1996, no. 4, pp. 49–55.
3. Mel'nikov B. F. *Izvestiya vysshikh uchebnykh zavedeniy. Matematika* [University proceedings. Mathematics]. 2004, no. 3, pp. 46–56.
4. Alekseeva A. G., Mel'nikov B. F. *Vektor nauki Tol'yattinskogo gosudarstvennogo universiteta* [The vector of science of Togliatty State University]. 2011, no. 3, pp. 30–33.
5. Korabel'shchikova S. Yu., Mel'nikov B. F. *Vestnik Severnogo (Arkticheskogo) federal'nogo universiteta. Ser.: Estestvennye nauki* [Bulletin of Northern (Arctic) Federal University. Series: Natural Sciences]. 2016, no. 3, pp. 91–96.
6. Melnikov B. *Number theoretic and algebraic methods in computer science*. 1995, pp. 125–137.
7. Dang V. V., Korabel'shchikova S. Yu., Mel'nikov B. F. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki* [University proceedings. Volga region. Physical and mathematical sciences]. 2015, no. 3 (35), pp. 88–99.
8. Dubasova O. A., Mel'nikov B. F. *Programmirovanie* [Programming]. 1995, no. 6, pp. 46–57.
9. Melnikov B., Kashlakova E. *Informatica (Lithuanian Acad. of Sciences)*. 2000, vol. 11, no. 4, pp. 441–446.
10. Baumgertner S. V., Mel'nikov B. F. *Izvestiya vysshikh uchebnykh zavedeniy. Povolzhskiy region. Fiziko-matematicheskie nauki* [University proceedings. Volga region. Physical and mathematical sciences]. 2013, no. 2 (26). – S. 64–74.
11. Dolgov V. N., Mel'nikov B. F. *Vestnik Voronezhskogo gosudarstvennogo universiteta. Ser.: Fizika. Matematika* [Bulletin of Voronezh State University. Series: Physics. Mathematics]. 2013, no. 2, pp. 173–181.
12. Doglov V. N., Mel'nikov B. F. *Vestnik Voronezhskogo gosudarstvennogo universiteta. Ser.: Fizika. Matematika* [Bulletin of Voronezh State University. Series: Physics. Mathematics]. 2014, no. 1, pp. 78–85.
13. Zubova M. A., Mel'nikov B. F. *Stokhasticheskaya optimizatsiya v informatike* [Stochastic optimization in informatics]. 2015, vol. 11, no. 2, pp. 17–29.
14. Cohen R. S., Gold Y. *J. Comp and System Sci*. 1977, no. 15, pp. 169–184.
15. Mel'nikov B. F. *Vestnik Moskovskogo universiteta. Ser. 15, Vychislitel'naya matematika i kibernetika* [Bulletin of Moscow University. Series: Calculus mathematics and cybernetics]. 1991, no. 3, pp. 51–53.

Мельников Борис Феликсович
доктор физико-математических наук, профессор, кафедра информационных систем и сетей, Российский государственный социальный университет (Россия, г. Москва, ул. Вильгельма Пика, 4)

E-mail: bf-melnikov@yandex.ru

Mel'nikov Boris Feliksovich
Doctor of physical and mathematical sciences, professor, sub-department of information systems and networks, Russian State Social University (4 Wilgelma Pika street, Moscow, Russia)

Корабельщикова Светлана Юрьевна
кандидат физико-математических наук, доцент, кафедра информатики и информационной безопасности, Северный (Арктический) федеральный университет имени М. В. Ломоносова (Россия, г. Архангельск, набережная Северной Двины, 17)

E-mail: s.korabelsschikova@narfu.ru

Korabel'shhikova Svetlana Jur'evna
Candidate of physical and mathematical sciences, associate professor, sub-department of informatics and information security, Northern (Arctic) Federal University named after M. V. Lomonosov (17 Severnoy Dviny embankment, Arkhangelsk, Russia)

Чурикова Надежда Петровна
аспирант, Самарский национальный исследовательский университет имени академика С. П. Королева (Россия, г. Самара, Московское шоссе, 34)

E-mail: claorisel@gmail.com

Churikova Nadezhda Petrovna
Postgraduate student, Samara National Research University named after academician S. P. Korolyov (34 Moskovskoe highway, Samara, Russia)

УДК 512.53, 510.54
Мельников, Б. Ф.

Об алгоритмах проверки выполнения некоторых бинарных отношений в глобальном надмоноиде свободного моноида / Б. Ф. Мельников, С. Ю. Корабельщикова, Н. П. Чурикова // Известия высших учебных заведений. Поволжский регион. Физико-математические науки. – 2017. – № 3 (43). – С. 87–99. DOI 10.21685/2072-3040-2017-3-8