ТЕОРЕТИЧЕСКИЕ ОСНОВЫ ИНФОРМАТИКИ

УДК 510.665, 519.765

ОБ ОПРЕДЕЛИМОСТИ В АЛГЕБРЕ КОНЕЧНЫХ ЯЗЫКОВ С КОНКАТЕНАЦИЕЙ МНОЖЕСТВА ОДНОСИМВОЛЬНЫХ ЯЗЫКОВ[1]

Дудаков С.М.
Тверской государственный университет, г. Тверь

Поступила в редакцию 08.10.2020, после переработки 20.10.2020.

Мы рассматриваем алгебру всех конечных языков над многосимвольным алфавитом с операцией конкатенации. Ранее было показано, что если взять подобную алгебру, но состоящую из всех регулярных многосимвольных языков, то в ней можно интерпретировать алгебру регулярных односимвольных языков, откуда следует, что теория обеих этих алгебр эквивалентна элементарной арифметике. В настоящей работе мы доказываем аналогичный результат для алгебры конечных языков: в ней определима подалгебра односимвольных языков, а сама она имеет теорию алгоритмически эквивалентную элементарной арифметике.

Ключевые слова: язык, конечный язык, конкатенация, элементарная арифметика.

Вестник ТвГУ. Серия: Прикладная математика. 2020. № 4. С. 5–13.
https://doi.org/10.26456/vtpmk601

Введение

Формальные языки являются одним из важных объектов дискретной математики, который имеет большое и теоретическое, и практическое значение [7]. Они широко применяются для описания свойств каких-либо объектов, правил преобразования и т. д.

Самой естественной операцией над отдельными словами является конкатенация — «склейка» двух слов в одно. Эта операция непосредственно переносится на языки: она заключается в построении конкатенаций всевозможных слов, входящих в исходные языки. Многие естественные классы языков замкнуты относительно этой операции, следовательно, такие классы с операцией конкатенации образуют алгебры. Таковыми, например, являются класс всех языков, а также классы регулярных, контекстно-свободных или конечных языков. Естественно возникает вопрос об алгоритмической разрешимости теории этой алгебры. Он имеет не только теоретическую, но и практическую значимость. Например, многие современные

[1]Работа выполнена при финансовой поддержке РФФИ, проект 20-01-00435.

системы управления базами данных (СУБД) допускают хранение конечных множеств объектов, в частности, слов, поэтому вопрос об алгоритмической разрешимости прямо связан с возможностью эффективного анализа языков запросов для таких СУБД [3, 8].

Явно в такой формулировке вопрос о разрешимости алгебры языков с некоторыми операциями был поставлен в работе [5] для класса регулярных языков. Заметим, что теория алгебры подмножеств, вообще говоря, отличается от соответствующей теории второго порядка исходной системы [2]: отсутствуют переменные первого порядка в явном виде. Это приводит к тому, что алгебра подмножеств может оказаться алгоритмически более простой, чем исходная, соответствующий пример приведён в [1].

Статья [4] содержит исследование алгоритмической разрешимости теории односимвольных языков с операцией конкатенации, то есть в случае, когда алфавит содержит один символ. В частности, там доказано, что для классов конечных языков и регулярных языков полученная теория эквивалентна элементарной арифметике [9]. Непосредственно перенести эти результаты на языки в многосимвольных алфавитах невозможно, так как при доказательстве существенно используется коммутативность и некоторые другие свойства операции конкатенации характерные только для односимвольных языков.

В дальнейшем, в работе [6], результат для класса регулярных языков был обобщён на произвольный алфавит с помощью интерпретации алгебры регулярных односимвольных языков. Однако, чтобы воспользоваться методом из [6] для определения области интерпретации, необходимо иметь возможность построения итерации произвольного языка. Поскольку результатом итерации является бесконечный язык, то такой метод не годится для алгебры конечных языков в произвольном алфавите.

В настоящей работе мы решаем сформулированную задачу для конечных языков в произвольном алфавите. На самом деле мы доказываем более сильное утверждение: в алгебре конечных языков определимы подалгебры односимвольных языков. Это автоматически даёт возможность интерпретации алгебры односимвольных языков и, следовательно, нижние оценки алгоритмической неразрешимости, следующие из работы [4]. Верхняя оценка оказывается такой же. Следовательно, независимо от алфавита теория конечных языков с операцией конкатенации имеет степень неразрешимости эквивалентную элементарной арифметике.

1. Определения

Язык — множество слов, составленных из символов какого-либо конечного алфавита $\Sigma \neq \varnothing$. Если алфавит Σ содержит только один символ, то язык называется *односимвольным.*

Конкатенацией языков x и y называется язык

$$x \,\&\, y = \{ab : a \in x, b \in y\}.$$

Класс языков *замкнут* относительно конкатенации, если вместе с любыми языками x и y он содержит язык $x \,\&\, y$.

Многие естественные, встречающиеся на практике классы языков обладают этим свойством: их примерами могут служить класс всех языков, класс регулярных языков или класс конечных языков.

Если класс A языков замкнут относительно конкатенации, то вместе с операцией & он образует *алгебру* $(A, \&)$, которая является полугруппой, так как операция &, очевидно, ассоциативна. Если A содержит язык e, состоящий из одного пустого слова ϵ, то эта алгебра будет моноидом, потому что e является нейтральным элементом: $x \,\&\, e = e \,\&\, x = x$ для любого языка x.

В дальнейшем, как это обычно бывает при рассмотрении полугрупп и моноидов, мы будет использовать запись xy вместо $x \,\&\, y$ и x^n вместо n-кратной конкатенации языка x с собой.

В настоящей работе мы рассматриваем моноид $(F, \&, e)$, который состоит из конечных языков над некоторым заранее фиксированным алфавитом. Отметим некоторые тривиальные свойства этого моноида, которые будем применять в дальнейшем. В этом моноиде можно определить пустой язык $\varnothing$:

$$x = \varnothing \equiv (\forall y) xy = x.$$

Очевидно, что конкатенация непустых языков снова даст непустой язык, поэтому непустые языки образуют определимый подмоноид $(F^*, \&, e)$, который далее и будет изучаться. Таким образом, в дальнейшем мы всегда предполагаем, что рассматриваемые языки непусты.

2. Определимость в моноиде непустых языков

Основной нашей целью будет выделить в моноиде $(F^*, \&, e)$ какой-либо подмоноид односимвольных языков, что откроет путь для использования результата из [4] об интерпретации элементарной арифметики. Чтобы это сделать, определим следующие шесть отношений:

$$\begin{aligned}
C(x) &\equiv (\exists u)(x^3 = xu \wedge u \neq x^2);\\
P(x, y) &\equiv (\forall u, v)(y = uv \wedge v \neq e \to (\exists w) y = uxw);\\
E(x) &\equiv (\forall y)(xy = yx \wedge C(y) \to (\exists u)(P(x, u) \wedge P(x, uy)));\\
R_2(x) &\equiv (\exists u)(x^3 = xu \wedge u \neq x^2 \wedge (\forall v)(x^3 = xv \to v = x^2 \vee v = u));\\
L_1(x) &\equiv E(x) \wedge R_2(x) \wedge (\forall u, v)(uvx = xuv \to uv = vu);\\
L(x, y) &\equiv L_1(x) \wedge xy = yx.
\end{aligned}$$

Далее мы докажем ряд утверждений о свойствах этих предикатов в моноиде $(F^*, \&, e)$.

Предложение 1. *Если язык x содержит пустое слово и хотя бы одно непустое слово, то есть $x = \{\epsilon, a, \dots\}$, где $a \neq \epsilon$, то условие $C(x)$ выполнено.*

Доказательство. Очевидно, x^2 содержит a. В качестве языка u возьмём $x^2 \setminus \{a\}$, тогда будет выполнено $xu \subseteq x^3$.

Чтобы доказать противоположное включение $x^3 \subseteq xu$, рассмотрим $ca \in x^3$, где $c \in x$. Если $c = \epsilon$, то $ca = a\epsilon$, где $a \in x$ и $\epsilon \in u$. Если $c \neq \epsilon$, то $ca = \epsilon(ca)$, где $\epsilon \in x$ и $ca \in u$, потому что $ca \neq a$. В любом случае $ca \in xu$.

Для слова $w \in x^3$ любого вида отличного от рассмотренного ca принадлежность $w \in xu$ очевидна, так как любые слова из языка x^2 отличные от a содержатся и в языке u. □

Предложение 2. *Если $x \neq e$ и условие $P(x, y)$ выполнено, то $y = x^n$ для некоторого натурального n.*

Доказательство. Если $y = e$, то $y = x^0$.

Если $y \neq e$, то из разложения $y = ey$ мы получаем $y = exw_1 = xw_1$ для некоторого языка w_1. Аналогично из $y = xw_1$ получаем $y = xxw_2 = x^2w_2$ для некоторого языка w_2. Этот процесс получения разложений $y = x^m w_m$ для каких-то w_m можно продолжать до тех пор, пока $w_m \neq e$. Покажем, что это рано или поздно должно произойти. Так как $x \neq e$, то длины слов в языках x^k с ростом k тоже возрастают. Следовательно, с ростом m неограниченно возрастают длины слов и в языках $x^m w_m$. Но в языке y длины слов заранее ограничены в силу его конечности. Значит, сколь угодно большим m быть не может и для некоторого n мы получим $y = x^n e = x^n$. □

Предложение 3. *Если условие $E(x)$ выполнено для языка x, то x содержит пустое слово ϵ.*

Доказательство. Рассмотрим любой язык x, для которого выполнено $E(x)$. Если $x = e$, то утверждение очевидно.

Пусть теперь $x \neq e$. Возьмём в качестве y язык $e \cup x$, полученный добавлением к x пустого слова ϵ. Очевидно, выполнено равенство $yx = xy$, а также условие $C(y)$ по предложению 1. С другой стороны, из предложения 2 вытекает, что $u = x^m$ и $uy = x^n$ для некоторых m и n. Следовательно,

$$uy = x^m(e \cup x) = x^m \cup x^{m+1} = x^n.$$

Как мы уже отмечали, при $x \neq e$ длины слов в языках x^k с ростом k постоянно растут, поэтому равенство $x^m \cup x^{m+1} = x^n$ возможно только при $m+1 = n$. Значит, $x^m \cup x^{m+1} = x^{m+1}$, что означает $x^m \subseteq x^{m+1}$. Если длина самого короткого слова в языке x равна l, то самое короткое слово в языке x^m имеет длину ml, а в языке x^{m+1} — длину $ml + l$. Но из включения $x^m \subseteq x^{m+1}$ тогда вытекает, что $ml \geq ml + l$, то есть $l \leq 0$. Ввиду неотрицательности l получаем $l = 0$, иначе говоря, язык x содержит пустое слово. □

Предложение 4. *Если язык x содержит пустое слово ϵ, то условие $R_2(x)$ выполнено в том и только том случае, когда x содержит кроме пустого слова ϵ ещё ровно одно слово, то есть x имеет вид $\{\epsilon, a\}$, $a \neq \epsilon$.*

Доказательство. Если $x = \{\epsilon, a\}$, $a \neq \epsilon$, то $x^3 = xx^2 = x\{\epsilon, a^2\}$. При этом, очевидно, $\{\epsilon, a^2\} \neq \{\epsilon, a, a^2\} = x^2$. Других разложений вида $x^3 = xv$ нет, так как $xv = v \cup av = x^3$, откуда получаем $v \subseteq x^3 = \{\epsilon, a, a^2, a^3\}$. Далее простым перебором всех вариантов легко убедиться, что никакие v, кроме указанных выше, равенству $x^3 = xv$ не удовлетворяют.

Теперь докажем обратное. Предположим, что язык x не является двухэлементным. Если он содержит одно слово, то есть в случае $x = e = \{\epsilon\}$, условие $x^3 = xv$ будет выполнено только для $v = e = x^2$. Если x содержит более двух слов, то есть когда $x = \{\epsilon, a, b, \dots\}$, где $\epsilon \neq a \neq b \neq \epsilon$, условие $x^3 = xv$ будет выполнено как минимум для трёх языков $v_0 = x^2$, $v_1 = x^2 \setminus \{a\}$ и $v_2 = x^2 \setminus \{b\}$. Последний факт доказывается таким же способом как предложение 1.

Таким образом, языки x, содержащие не два элемента, условию $R_2(x)$ не удовлетворяют. □

Предложение 5. *Условие $L_1(x)$ выполнено тогда и только тогда, когда x имеет вид $\{\epsilon, c\}$, где слово c состоит в точности из одного символа.*

Доказательство. Из истинности $E(x)$ следует, что язык x содержит пустое слово согласно предложению 3. Из истинности $R_2(x)$ тогда вытекает, что язык x должен иметь вид $\{\epsilon, a\}$, $a \neq \epsilon$, по предложению 4.

Если слово $a = a_1 \dots a_{k-1} a_k$ содержит хотя бы два различных символа, то $a \neq a_k a_1 \dots a_{k-1}$. Взяв тогда $u = a_1 \dots a_{k-1}$ и $v = a_k$, мы получим противоречие с определением L_1. Следовательно, слово a имеет вид c^n для какого-то символа алфавита c и натурального $n > 0$.

Если слово a состоит из более чем одного символа c: $x = \{\epsilon, c^n\}$, $n > 1$, то, взяв в качестве $y = \{\epsilon, c\}$, мы получим противоречие с $E(x)$, поскольку самые длинные слова в языках u и uy отличаются по длине в точности на единицу, поэтому сами языки не могут быть одновременно степенями x.

Обратно, пусть $x = \{\epsilon, c\}$. Условие $R_2(x)$ вытекает из предложения 4. Отметим, что условие $yx = xy$ может выполняться только для языков, состоящих из слов вида c^k, то есть таких, что $y \subseteq x^n$ для некоторого n. Тогда $E(x)$ выполнено, так как в качестве u можно взять x^n. Условие $L_1(x)$ выполнено, по той же причине: $u, v \subseteq x^n$ для некоторого n. □

Предложение 6. *Условие $L(x, y)$ выполнено в том и только том случае, когда язык y содержит слова, содержащие только символ c, где $x = \{\epsilon, c\}$ согласно предложению 5.*

Доказательство. Если $xy = yx$, то y не может содержать других слов. Обратное тоже тривиально. □

Следствие 1. *В моноиде $(F^*, \&, e)$ определим подмоноид односимвольных языков $(F_1^*, \&, e)$.*

Теорема 1. *Теория моноида $(F^*, \&, e)$ алгоритмически эквивалентна элементарной арифметике.*

Доказательство. Так как подмоноид $(F_1^*, \&, e)$ определим в $(F^*, \&, e)$ с помощью формул L_1 и L, то теория $(F_1^*, \&, e)$ сводится к теории $(F^*, \&, e)$ ограничением всех кванторов на множество F_1^*. Точнее, интерпретация ϕ' каждой атомной формулы ϕ совпадает с исходной. Интерпретации формул, построенных с помощью булевых связок, строятся с помощью тех же связок. Каждая формула с квантором вида $(\forall y)\phi$ или $(\exists y)\phi$ интерпретируется соответственно с помощью $(\forall y)(L(x, y) \to \phi')$ или $(\exists y)(L(x, y) \wedge \phi')$. Вся формула Φ интерпретируется как $(\exists x)(L_1(x) \wedge \Phi')$. Поскольку теория моноида $(F_1^*, \&, e)$ эквивалентна элементарной арифметике (см. [4]), то элементарная арифметика сводится к теории $(F^*, \&, e)$.

Чтобы показать обратное и свести теорию $(F^*, \&, e)$ к элементарной арифметике, допустим, что алфавит содержит k символов: $a_1, \dots, a_k$. Тогда каждое слово

w можно рассматривать как запись некоторого однозначно определённого числа $n(w)$ цифрами $1, \ldots, k$ соответственно в $(k+1)$-ичной системе счисления, причём эта запись не содержит нулей, кроме как незначащих слева. Обратное тоже верно: каждое такое число соответствует некоторому слову. Напомним, что в элементарной арифметике (см. [2]) определимы функция возведения в степень x^y, а также функция $d(x, y, z)$, значение которой равно y-й цифре в z-ичной записи числа x (начиная с младших разрядов). Тогда множество чисел, которые соответствуют словам, определяется формулой

$$W(x) \equiv (\forall y)(d(x, y, k+1) = 0 \to (\forall y' > y) d(x, y', k+1) = 0).$$

При указанном представлении конкатенация слов $x \mathbin{\&} y = z$ соответствует истинности формулы $C_w(n(x), n(y), n(z))$, где

$$C_w(x, y, z) \equiv (\exists u)((k+1)^u > x \wedge (k+1)^{u-1} \leq x \wedge x \cdot (k+1)^u + y = z).$$

Конечные языки L тоже можно закодировать натуральными числами, если использовать, например, каноническую нумерацию конечных множеств натуральных чисел D (см. [9]):

$$D(L) = \sum_{w \in L} 2^{n(w)}.$$

Тогда условие $w \in L$ соответствует истинности формулы $d(D(L), n(w), 2) = 1$.

Теперь мы в состоянии описать интерпретацию моноида $(F, \&, e)$ в элементарной арифметике. Область интерпретации определяется формулой

$$I(x) \equiv (\forall y)(d(x, y, 2) = 1 \to W(y)),$$

а конкатенация языков $x \mathbin{\&} y = z$ — формулой

$$\begin{aligned}&C_L(x, y, z) \equiv \\ &\quad (\forall w)(d(z, w, 2) = 1 \leftrightarrow (\exists u, v)(d(x, u, 2) = 1 \wedge d(y, v, 2) = 1 \wedge C_w(u, v, w))). \quad \square\end{aligned}$$

Заключение

Мы показали неразрешимость алгебры конечных языков для произвольных алфавитов. Естественные вопросы, которые возникают далее, связаны с рассмотрением алгебры конечных языков как моноида конечных подмножеств над моноидом слов, а также с рассмотрением иных классов языков:

- Что можно сказать об алгоритмической разрешимости алгебры конечных подмножеств произвольного моноида?

- В частности, в доказательствах наших теорем используются свойства сократимости и бесконечности порядка элементов исходного моноида (для любого непустого слова a все слова вида a^n попарно различны). Можно ли эти свойства ослабить или совсем отбросить?

– Можно ли найти какой-либо класс языков, замкнутый относительно конкатенации, для которого доказанное в работе утверждение не выполняется: в алгебре многосимвольных языков нельзя интерпретировать алгебру односимвольных.

Список литературы

[1] Дудаков С.М. Об алгоритмических свойствах алгебры конечных подмножеств некоторых уноидов // Вестник ТвГУ. Серия: Прикладная математика. 2019. № 4. С. 108–116.

[2] Boolos G.S., Burgess J.P., Jeffrey R.C. Computability and Logic. 5th edition. New York: Cambridge University Press, 2007. 364 p.

[3] Codd E.F. Relational completeness of data base sublanguages // Database Systems. Ed. by R. Rustin. Prentice-Hall, 1972. Pp. 33–64.

[4] Dudakov S.M. On undecidability of concatenation theory for one-symbol languages // Lobachevskii Journal of Mathematics. 2020. Vol. 40, № 2. Pp. 168–175.

[5] Dudakov S.M., Karlov B.N. On decidability of regular languages theories // Proc. of 14th International Computer Science Symposium in Russia, CSR 2019. Series: LNCS. Vol. 11532. 2019. Pp. 119–130.

[6] Dudakov S., Karlov B. On decidability of theories of regular languages // Theory of Computing Systems. 2020. http://doi.org/10.1007/s00224-020-09995-4

[7] Hopcroft J.E., Motwani R., Ullman J.D. Introduction to Automata Theory, Languages, and Computation. Harlow: Pearson, 2013. 560 p.

[8] Kanellakis P., Kuper G., Revesz P. Constraint query languages // Journal of Computer and System Sciences. 1995. Vol. 51. Pp. 26–52.

[9] Rogers H. Theory of Recursive Functions and Effective Computability. Cambridge: MIT Press, 1987. 506 p.

Образец цитирования

Дудаков С.М. Об определимости в алгебре конечных языков с конкатенацией множества односимвольных языков // Вестник ТвГУ. Серия: Прикладная математика. 2020. № 4. С. 5–13. https://doi.org/10.26456/vtpmk601

Сведения об авторах

1. **Дудаков Сергей Михайлович**
декан факультета прикладной математики и кибернетики Тверского госуниверситета.

Россия, 170100, г. Тверь, ул. Желябова, д. 33, ТвГУ.
E-mail: *sergeydudakov@yandex.ru*

ON DEFINABILITY OF ONE-SYMBOL LANGUAGES IN THE MONOID OF FINITE LANGUAGES WITH CONCATENATION

Dudakov Sergey Mikhailovich
Dean of Applied Mathematics and Cybernetics faculty, Tver State University
Russia, 170100, Tver, 33, Zhelyabova str., TverSU.
E-mail: sergeydudakov@yandex.ru

Received 08.10.2020, revised 20.10.2020.

We consider an algebra of all finite languages with the concatenation operation. For one-symbol languages it is known that its theory is equivalent to the first-order arithmetic. Earlier it was proved that for regular languages a one-symbol algebra can be interpreted in multi-symbol algebras. Here we show how to define a one-symbol subalgebra in multi-symbol algebras for finite languages.

Keywords: language, finite language, concatenation, first-order arithmetic.

Citation

Dudakov S.M., "On definability of one-symbol languages in the monoid of finite languages with concatenation", *Vestnik TvGU. Seriya: Prikladnaya Matematika [Herald of Tver State University. Series: Applied Mathematics]*, 2020, № 4, 5–13(in Russian). https://doi.org/10.26456/vtpmk601

References

[1] Dudakov S.M., "On algorithmic properties of finite subset algebra for some unoids", *Vestnik TvGU. Seriya: Prikladnaya Matematika [Herald of Tver State University. Series: Applied Mathematics]*, 2019, № 4, 108–116 (in Russian).

[2] Boolos G.S., Burgess J.P., Jeffrey R.C., *Computability and Logic*, 5th edition, Cambridge University Press, New York, 2007, 364 pp.

[3] Codd E.F., "Relational completeness of data base sublanguages", *Database Systems*, ed. R. Rustin, Prentice-Hall, 1972, 33–64.

[4] Dudakov S.M., "On undecidability of concatenation theory for one-symbol languages", *Lobachevskii Journal of Mathematics*, **40**:2 (2020), 168–175.

[5] Dudakov S.M., Karlov B.N., "On decidability of regular languages theories", *Proc. of 14th International Computer Science Symposium in Russia.* V. 11532, CSR 2019, LNCS, 2019, 119–130.

[6] Dudakov S., Karlov B., "On decidability of theories of regular languages", *Theory of Computing Systems*, 2020, http://doi.org/10.1007/s00224-020-09995-4.

[7] Hopcroft J.E., Motwani R., Ullman J.D., *Introduction to Automata Theory, Languages, and Computation*, Pearson, Harlow, 2013, 560 pp.

[8] Kanellakis P., Kuper G., Revesz P., "Constraint query languages", *Journal of Computer and System Sciences*, **51** (1995), 26–52.

[9] Rogers H., *Theory of Recursive Functions and Effective Computability*, MIT Press, Cambridge, 1987, 506 pp.