УДК 519.681

А.С. ШУКУРЯН

ЭКВИВАЛЕНТНОСТЬ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В ЧАСТИЧНО КОММУТАТИВНОМ АЛФАВИТЕ

Ключевые слова: *регулярные выражения, частично коммутативный алфавит, автоматы, эквивалентность.*

Алгоритмическая разрешимость проблемы эквивалентности регулярных выражений в коммутативном алфавите доказана В.Н. Редько в 1964 г. Из результата, полученного В.А. Тузовым в 1971 г., следует, что в общем случае проблема эквивалентности регулярных выражений в частично коммутативном алфавите алгоритмически неразрешима. Под частично коммутативным алфавитом понимается объединение непересекающихся подмножеств, в котором любые два символа из одного и того же подмножества не перестановочны, а любые два символа из разных подмножеств перестановочны. В определенном смысле этот алфавит является обобщением коммутативного алфавита, который соответствует случаю, когда мощность каждого подмножества равна единице. В.А. Тузов показал, что если существует хотя бы два подмножества мощности, большей единицы, то проблема эквивалентности алгоритмически неразрешима. Таким образом, только для одного специального случая не было известно, разрешима ли проблема эквивалентности: когда мощность одного из подмножеств в частично коммутативном алфавите больше единицы, а мощность всех остальных подмножеств равна единице. В данной статье доказано, что и в этом случае проблема эквивалентности регулярных выражений алгоритмически разрешима.

Пусть Y— конечный алфавит, $Y_1, \dots, Y_n$ — разбиение алфавита Y по отношению Θ на непересекающиеся подмножества. Если Y— множество, то мощность этого множества обозначим $|Y|$.

Пусть R_1 и R_2— регулярные выражения [1] в алфавите Y. Если для каждого $w \in R_1$существует слово $w' \in R_2$, совпадающее с w с точностью до перестановки смежных символов, относящихся к разным классам разбиения алфавита Y и, наоборот, для каждого слова $w' \in R_2$ существует слово $w \in R_1$, совпадающее с w' с точностью до перестановки смежных символов, относящихся к классам разбиения алфавита Y, то регулярные выражения R_1 и R_2 назовем Θ-эквивалентными и обозначим $R_1 \sim R_2 (\Theta)$.

С одной стороны, В.Н. Редько показал [2], что если все классы разбиения алфавита одноэлементны, то проблема Θ-эквивалентности алгоритмически разрешима. С другой стороны, как вытекает из результата В.А. Тузова [3], если разбиение алфавита содержит хотя бы два класса с числом символов, не меньшим двух, то проблема Θ-эквивалентности сводится к проблеме эквивалентности недетерминированных многоленточных автоматов [4] и, следовательно, алгоритмически неразрешима.

В настоящей работе рассматривается «промежуточный случай», когда все классы разбиения алфавита Y, кроме одного, одноэлементные. Исследование основано на использовании некоторых результатов теории частично коммутативных полугрупп [5] и автоматов с многомерными лентами [6, 7].

В разд. 1 описывается специальное представление элементов частично коммутативной полугруппы, на котором основаны дальнейшие построения. В разд 2 показано, что проблема Θ-эквивалентности, когда мощности подмножеств равны единице, алгоритмически разрешима. Доказательство отличается от доказательства в [2], и демонстрирует идею, применяемую для доказательства рассмотренного в статье «промежуточного случая». Специальная многомерная лента [6] используется для записи следов выполнения данного автомата. Монотонное движение вперед по ленте выполняется относительно диагоналей ленты: от текущей диагонали к смежной. В разд. 3 доказана разрешимость проблемы эквивалентности регулярных выражений в некоммутативном алфавите. Доказательство также отлично от известного доказательства в [4] и использует многомерную ленту, но каждый шаг i движения по ленте выполняется не к смежной диагонали, а к диагонали с номером $\ell + 2^i$, где ℓ — номер текущей диагонали. В разд. 4 построена специальная комбинация случаев, рассмотренных в разд. 2 и 3, которая соответствует поставленной задаче. Доказано, что проблема Θ-эквивалентности здесь алгоритмически разрешима. Таким образом, получена классификация по алгоритмической разрешимости проблемы Θ-эквивалентности.

1. ЧАСТИЧНО КОММУТАТИВНЫЕ ПОЛУГРУППЫ

Если X — алфавит, то полугруппу всех слов в алфавите X, включая и пустое слово, обозначим F_X, а полугруппу всех n-компонентных векторов-слов — F_X^n.

Пусть G — полугруппа с единицей, порожденная множеством образующих $Y = \{y_1, \ldots, y_n\}$. Она называется свободной частично коммутативной полугруппой, если задается конечным числом определяющих соотношений вида $y_i y_j = y_j y_i$ [5].

Пусть K: $F_Y \to F_{\{0,1\}}^n$ — гомоморфизм, определенный над полугруппой F_Y и сопоставляющий словам из F_Y n-ки слов в бинарном алфавите $\{0, 1\}$ [6]. На образующих F_Y гомоморфизм K задается равенством $K(y_i) = (\alpha_{1i}, \ldots, \alpha_{ni})$, где

$$\alpha_{ij} = \begin{cases} 1, \text{ если } i = j, \\ e, \text{ если } y_i y_j = y_j y, \\ 0, \text{ если } y_i y_j \neq y_j y_i. \end{cases}$$

При этом $K(e) = (e, \ldots, e)$.

Лемма 1 [6]. Пусть y_i, y_j — образующие G, $y_i \neq y_j$, $g_1 = y_i y_j$, $g_2 = y_j y_i$ — элементы G, полученные в результате применения операции полугруппы G к образующим y_i, y_j. Тогда $g_1 = g_2 \Leftrightarrow K(y_i y_j) = K(y_j y_i)$.

Это утверждение позволяет рассматривать гомоморфизм K как отображение, действующее не только на полугруппе F_Y, но и на частично коммутативной полугруппе G.

На множестве образующих Y задается линейный порядок [6], обозначаемый $<$, который совпадает с порядком перечисления образующих (лексикографическим порядком). Отношение $<$ используется далее для упорядочения представлений элементов G и определения их канонического представления. Пусть $h \in F_Y$. Маской вхождений образующей $y \in Y$ в слово h называется слово из нулей и единиц, получающееся из h в результате замены всех вхождений образующей y на 1, а всех вхождений остальных образующих — на 0. Если рассматривать маски не только как слова в алфавите $\{0, 1\}$, но и как целые двоичные числа, то появляется возможность сравнивать их. Будем считать, что лексикографический порядок задает порядок сравнения двоичных представлений для образующих. Пусть g — элемент полугруппы G; $h, q \in F_Y$ — его представления. Считается, что представление h меньше представления q, $h < q$, если существует такая образующая y, что: 1) если $y' < y$, то маски вхождений образующей y' в слова h и q совпадают; 2) маска вхождений образующей y в слово h меньше маски вхождений этой образующей в слово q.

Лемма 2 [6]. Любые два разные представления одного и того же элемента полугруппы G сравнимы по отношению $<$.

Из леммы 2 следует, что для каждого элемента из G существует наименьшее представление, называемое канонической формой элемента, индуцированной порядком $<$ на множестве образующих [6]. Отметим следующее свойство каноническихкого представления элементов: любые два символа, стоящие рядом в каноническом представлении, либо не перестановочны, либо левый меньше (в смысле порядка $<$) правого.

Определяется отношение эквивалентности ρ на полугруппе F_Y следующим образом: если w_1 и w_2 — слова из F_Y, то $w_1 \rho w_2$ тогда и только тогда, когда w_1 совпадает с w_2 с точностью до определяющих соотношений.

Отношение ρ делит F_Y на непересекающиеся классы — классы коммутативности. Очевидно, что каждому элементу g из G соответствует некоторый класс коммутативности C_g, а элемент из C_g, являющийся канонической формой элемента g, назовем представителем класса коммутативности C_g.

Лемма 3 [6]. Всякая свободная частично коммутативная полугруппа с n образующими изоморфна некоторой подполугруппе прямого произведения n свободных полугрупп с двумя образующими.

Доказательство. Доказательство этой леммы в [6] опущено как очевидное. Здесь оно приводится, поскольку содержит некоторые рассуждения, используемые в дальнейшем.

Из леммы 1 следует, что гомоморфизм K можно рассмотреть на полугруппе G. Лемма 3 показывает, что полугруппа G изоморфна своему гомоморфному образу $K(G)$. Доказательство проводится индукцией по длине канонической формы.

Пусть $f, g \in G$. Из определения гомоморфизма следует, что $K(f) = K(g) \Rightarrow$ $\Rightarrow f = g$ для образующих полугруппы G.

Предположим, что лемма верна для произвольных $f, g \in G$, для которых длины канонических форм равны ℓ. Покажем, что лемма верна и в том случае, когда длина канонических форм $f, g \in G$ равна $\ell + 1$.

Пусть, наоборот, $K(f) = K(g)$, но $f \neq g$. Рассмотрим компоненту α_i вектора $K(f) = \alpha = (\alpha_1, \ldots, \alpha_n)$, которая не является пустой и начинается с символа «1». Допустим, что эта компонента соответствует образующей y_i. Предположим, что «1» и самые левые символы «0» компонент, соответствующих образующим, не коммутативным с y_i, удалены, а результат обозначим $\alpha' = (\alpha'_1, \ldots, \alpha'_n)$.

Согласно построению гомоморфизма K существуют f' и $g' \in G$, длины канонических форм которых равны ℓ и $f = f' y_i$, $g = g' y_i$ и $K(f') = K(g') = \alpha'$. Но тогда согласно предположению индукции $f' = g'$ и, следовательно, $f = g$. Последнее противоречит предположению, что $f \neq g$.

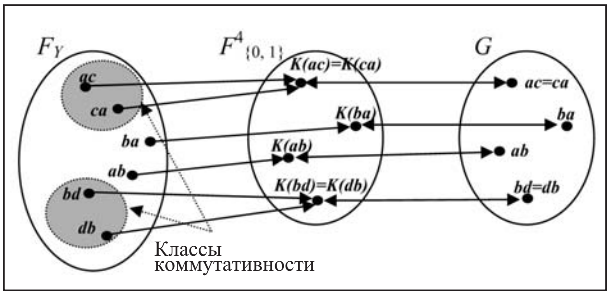


Рис. 1

Рассмотрим пример. Пусть задано множество образующих $Y = \{a, b, c, d\}$ с соотношениями $ac = ca$, $ad = da$, $bc = cb$, $bd = db$, $cd = dc$. Построим гомоморфизм K: $F_Y \to F^4_{\{0,1\}}$: $K(a) = (1, 0, e, e)$, $K(b) = (0, 1, e, e)$, $K(c) = (e, e, 1, e)$, $K(d) = (e, e, e, 1) =$ $= (e, e, e, 1)$. Легко заметить, что $K(ac) = K(a)K(c) = (1, 0, 1, e) = K(c)K(a) =$ $= K(ca)$, тогда как $K(ab) = (10, 01, e, e)$ и $K(ba) = (01, 10, e, e)$ (рис. 1).

Ниже приведен алгоритм CFBuilder (Canonical Form Builder), который по произвольному вектору — образу гомоморфизма $K: F_Y \to F^n_{\{0,1\}}$, строит каноническую форму элемента $g \in G$.

Алгоритм CFBuilder. Если K — вектор, то его компоненту i обозначим $K[i]$, а число компонент вектора — $\ell(K)$. Вектор, используемый для хранения значения гомоморфного образа $K(w)$ слова w, $w \in F_Y$, обозначим K.

```
begin;
// 1. Вход: конечный входной алфавит Y, множество C всех определяющих
// соотношений вида y_i y_j = y_j y_i, y_i, y_j ∈ Y, i, j ∈ {1, …, |Y|}, i ≠ j, вектор K.
// Предполагается, что вхоное значение вектора K корректно, т.е. совпадает с
// гомоморфным образом некоторого слова из F_Y .
input K;
input Y;
input C;
// Начальное значение результата — пустое слово.
assign ResultWord = e;
// Итерация алгоритма.
while (K ≠ (e,…,e))
{
        // 2. Выбираем минимальный индекс таких компонент вектора K,
        // которые начинаются с «1». Удаляем «1» из кода компоненты,
        // соответствующей этому индексу.
        for (i = 1, …, |Y|)
        {
            // i — индекс компоненты, начинающейся с «1».
            if ((K[i])[1] = 1)
            {
                    erase (K[i])[1] from K[i];
                    break;
                    }
}
        // 3. Выбираем символ в алфавите, соответствующий полученному индексу.
        // Добавляем ее справа к результату.
        assign s = Y[i];
        add s to ResultWord;
        // 4. Определяем множество Letters символов, не коммутативных с
        // выбранным символом алфавита.
        assign Letters = {y | y ∈ Y; {sy = ys} ∉ C};
        // 5. Определяем множество индексов для всех символов в Letters.
        for (∀y ∈ Letters)
        {
                add j, that Y[j] = y to Indices;
        }
        // 6. Для каждого индекса j ∈ Indices удаляем старший разряд двоичного
        // кода компоненты j вектора K, если он начинается с «0».
        for (∀j ∈ Indices)
        {
                erase (K[j])[1] from K[j];
            }
        }
    // 7. Выводим результат.
    output ResultWord;
    end;
```

Рассмотрим работу этого алгоритма на примере. Возьмем слово $w = daba$, $K(w) = (101, 010, e, 1)$. Класс коммутативности слова w — $\{adba, abda, abad, daba\}$. Из $K = (101, 010, e, 1)$ алгоритм сначала выбирает первую компоненту $K[1] = 101$, соответствующую символу a в алфавите Y. Символ, не коммутативный с a, — это b, следовательно, $K = (01, 10, e, 1)$. Затем алгоритм выбирает вторую компоненту, соответствующую b. В этом случае $K = (1, 0, e, 1)$. Алгоритм выбирает a: $K = (e, e, e, 1)$, затем d: $K = (e, e, e, e)$. Итак, получаем слово $abad$, являющееся представителем класса коммутативности $\{adba, abda, abad, daba\}$.

Лемма 4. Алгоритм CFBuilder корректен. Для произвольного вектора $K_w = K(w)$ он получает каноническую форму элемента $g \in G$, представляющую слово w в полугруппе F_Y.

Доказательство. Заметим, что если вектор $K = (e, \ldots, e)$, то алгоритм выводит пустое слово, не входя в главный цикл.

Сформулируем некоторые очевидные предусловия для главного цикла алгоритма, вытекающие из построения гомоморфизма K:

1) $K \neq (e, \ldots, e)$, $\ell(K) = n$;

2) $\forall i, j \in \{1, \ldots, n\}$, $i \neq j$, $y_i y_j \neq y_j y_i$:

 2.1) $K[i] \neq K[j]$,

 2.2) $(K[i])[1] = 1 \Rightarrow (K[j])[1] = 0$,

 $(K[i])[1] = 0 \Rightarrow \exists!\, j'$, $j' \in \{1, \ldots, n\}$, $i \neq j'$,

 $y_i y_j{}' \neq y_j{}' y_i$, $(K[j'])[1] = 1$,

 2.3) $\ell(K[i]) = \ell(K[j])$;

3) $\exists i \in \{1, \ldots, n\} (K[i])[1] = 1 \ \forall j \in \{1, \ldots, n\}$ $y_i y_j \neq y_j y_i$ $(K[j])[1] = 0$;

4) $\ell(ResultWord) = 0$

(теперь, чтобы доказать корректность алгоритма CFBuilder, покажем, что для произвольной итерации имеют место следующие инварианты цикла):

5) $ResultWord[it+1] = ResultWord[it] y_i \Rightarrow (K_{it}[i])[1] = 1$ и $\forall j \in \{1, \ldots, n\}$

$$(K_{it}[j])[1] = 1 \Rightarrow i < j,\ K_{it+1}[i] = (K_{it}[i])[2] \ldots (K_{it}[i])[n],\ \ell(K_{it+1}[i]) = \ell(K_{it}[i]) - 1;$$

6) $ResultWord[it+1] = ResultWord[it] y_i\, (K_{it}[i])[1] = 1$ и $\forall j \in \{1, \ldots, n\}$

$$(K_{it}[j])[1] = 0,\ y_i y_j \neq y_j y_i \Rightarrow K_{it+1}[j] = (K_{it}[j])[2] \ldots (K_{it}[j])[n],$$

$$\ell_{it+1}(K[j]) = \ell_{it}(K[j]) - 1.$$

Инвариант 5) означает, что всегда выбирается минимальный индекс компонент вектора K, которые начинаются с «1». Затем «1» удаляется и символ алфавита, соответствующий выбранному индексу, выбирается и добавляется справа к *ResultWord* в каждой итерации. Истинность инварианта очевидным образом следует из семантики шагов 2 и 3 алгоритма с учетом предусловия 3).

Инвариант 6), в свою очередь, означает, что первый символ каждой компоненты, соответствующей символу в алфавите, не коммутативному с последним символом ResultWord, удаляется. Согласно предусловию 2) удаляемый символ — это «0». Истинность же инварианта 6) вытекает из семантики шагов 5 и 6 алгоритма.

Из истинности инвариантов 5) и 6) следует, что начальный вектор K будет преобразован в $K = (e, \ldots, e)$ через конечное число итераций.

Согласно инварианту 5) порядок выбора символов соответствует порядку символов в алфавите. Следовательно, любые два символа, стоящие рядом, либо не перестановочны, либо левый меньше (в смысле порядка $<$) правого. Таким образом, *ResultWord* является канонической формой.

2. ПРОБЛЕМА ЭКВИВАЛЕНТНОСТИ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В КОММУТАТИВНОМ АЛФАВИТЕ

Пусть $A_R = \langle Y, S, \delta, F, s_0 \rangle$ — детерминированный автомат, распознающий в точности регулярное выражение R во входном алфавите Y, $|Y| = n$, с множеством состояний S, функцией переходов δ, множеством заключительных состояний F и начальным состоянием s_0. Предположим также, что задано разбиение алфавита Y на непересекающиеся классы коммутативности, в каждом из которых содержится по одному символу: $Y = Y_i \cup \ldots \cup Y_n$, $|Y_i| = 1$, $i = 1, \ldots, n$. Покажем, что в этом случае проблема Θ-эквивалентности регулярных выражений алгоритмически разрешима.

Введем понятие следов выполнения заданного автомата A на многомерной ленте. Пусть $N=\{0,\ 1,\ \ldots\}$. Множество N^n называется n-мерной лентой [6, 7], элементы множества N^n— n-ки вида $(\alpha_1,\ldots,\alpha_n)$ — ячейками ленты, а числа $\alpha_1,\ldots,\alpha_n$— координатами соответствующей ячейки. Ячейка (0, …, 0) называется начальной. Ячейку $a_1=(\alpha_{11},\ldots,\alpha_{1n})$ назовем предшественницей ячейки $a_2=(\alpha_{21},\ldots,\alpha_{2n})$, если одна из координат a_1 меньше соответствующей координаты a_2 на единицу, а все остальные координаты обеих ячеек совпадают. Введем специальный предикат для представления отношения предшествования и обозначим его π, $\pi(a_1,a_2)$ истинно тогда и только тогда, когда a_1 является предшественницей a_2. Введем также другой предикат π_j, $\pi_j(a_1,a_2)$ истинно тогда и только тогда, когда $\pi(a_1,a_2)$ истинно и ячейки a_1, a_2 отличаются значениями координаты j.

Множество всех подмножеств множества состояний S автомата A обозначим 2^S, а множество всех предшественниц ячейки a — $P(a)$.

Основываясь на функции переходов δ автомата A, на множестве 2^S определим функцию переходов Δ следующим образом: $\Delta(\{s_1,\ldots,s_m\},y_1)=\{\delta(s_1,y_i),\ \ldots$ $\ldots,\delta(s_m,y_i)\}$, $y_i\in Y$, $s_j\in S$, $j=1,\ldots,m$, $\{s_1,\ldots,s_m\}\in 2^S$.

Отображение $\varphi_A: N^n\to 2^S$ будем называть множеством следов выполнения автомата A тогда и только тогда, когда:

1) $\varphi_A(0,\ \ldots,0)=\{s_0\}$, $P(0,\ldots,0)=\varnothing$;

2) $\forall a\in N^n\setminus\{0,\ \ldots,0\}$, $P(a)=\{a_{\text{pred}}^{(j)}\mid \pi_j(a_{\text{pred}}^{(j)},a)$ истинно, $j=n_1,\ \ldots,\ n_k,\ k\le n\}$,

$\varphi_A(a)=\Delta(\varphi_A(a_{\text{pred}}^{(n_1)}),y_{n_1})\cup\ldots\cup\Delta(\varphi_A(a_{\text{pred}}^{(n_k)}),y_{n_k})$.

Конечное подмножество следов выполнения, содержащее все ячейки, у которых сумма координат каждой ячейки меньше или равна $n-1$, назовем словом следов (выполнения) длины k. Множество всех слов следов обозначим Ω_A, множество всех ячеек, использованных в данном слове следов ω, — U_ω.

Часть слова следов ω, сумма координат ячеек которой равна k, назовем диагональю k слова следов ω и обозначим $d_k(\omega)$. Длина $d_k(\omega)$ равна $k+1$, длина $\ell(\omega)$ слова следов ω — количеству содержимых диагоналей.

Для данного слова следов ω определим путь $p=a_{p_1}\ldots a_{p_m}$, $m\ge 1$, $a_{p_j}\in U_\omega$, $j\in\{1,\ \ldots,\ m\}$, как последовательность ячеек, для которых $\pi(a_{p_v}\ldots a_{p_{v+1}})$, $v=1,\ \ldots,\ m-1$.

Естественно связать с путем $p=a_{p_1}\ldots a_{p_m}$ его характеристику $\chi_p=$ $=y_{p_1}\ldots y_{p_{m-1}}$, где $y_{p_i}\in Y_i$, $i\in\{1,\ ...,\ n\}$, $j\in\{1,\ ...,\ m-1\}$, тогда и только тогда, когда координата i увеличивается на 1 в ячейке $a_{p_{j+1}}$ по сравнению с ячейкой a_{p_j}.

Путь $p=a_{p_1}\ldots a_{p_m}$ назовем полным, если a_{p_1} = (0, …, 0). Скажем, что полный путь $p=a_{p_1}\ldots a_{p_m}$ принимается автоматом A, если $\varphi_A(a_{p_m})$ содержит заключительное состояние автомата A. Координаты ячейки a_{p_m} назовем канонической формой полного пути $p=a_{p_1}\ldots a_{p_m}$.

Для данного автомата A множество всех путей, принимаемых автоматом A, в слове следов ω обозначим $AP_A(\omega)$, а множество их канонических форм — $CF_{AP_A(\omega)}$.

Графическое представление слова следов и его диагоналей приведено на рис. 2 .

Пусть автоматы A_1 и A_2 распознают в точности регулярные выражения R_1 и R_2 соответственно, а S_1, S_2 — их множества состояний. Предположим также, что $k=2^s-1$, где $S=\max\{|S_1|,|S_2|\}$, ω_1 и ω_2 — слова следов, $\ell(\omega_1)=\ell(\omega_2)=k$.

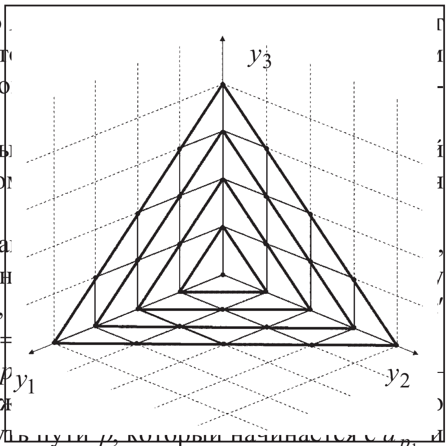
Лемма 5. $CF_{AP_{A_1}(\omega_1)}=CF_{AP_{A_2}(\omega_2)}\Leftrightarrow R_1\sim R_2(\Theta)$.

Доказательство. Сначала покажем, что $CF_{AP_{A_1}(\omega_1)}=CF_{AP_{A_2}(\omega_2)}\Rightarrow R_1\sim R_2(\Theta)$.

Предположим, что R_1 не Θ-эквивалентно … пути длины k, который принимается автомат… A_2, или, наоборот, который принимается авто… том A_1.

В силу предположения существует полны… k, $m > k$, который принимается одним из авто… автоматом A_2.

Это означает, что $\varphi_{A_1}(a_{p_m})$ содержит за… тогда как $\varphi_{A_2}(a_{p_m})$ не содержит заключительн… $m > k$, существует две ячейки a и a' в пути p, … больше суммы координат ячейки a, $\varphi_{A_1}(a) = $…



Пусть $\chi_{p'}$ — характеристика подпути p… путь от ячейки a до ячейки a'', имеющий ту ж… такой путь существует. Обозначим p''' подпуть пути p, который начинается с a_{p_1} и оканчивается предшественницей ячейки a. Конкатенацию двух путей p''' и p'' обозначим $p_{\text{new}} = p'''p''$. Его длина меньше длины начального пути p. Очевидно, что $\varphi_{A_1}(a'')$ содержит заключительное состояние автомата A_1, а $\varphi_{A_2}(a'')$ не содержит заключительного состояния автомата A_2. Если длина пути p_{new} больше k, то выполняем процедуру, описанную выше, до тех пор, пока не получим путь длины, не превышающей k. Но это противоречит начальному предположению о том, что не существует такого пути в словах следов ω_1 и ω_2, который принимался бы одним из автоматов, но не принимался бы другим.

Поскольку утверждение $R_1 \sim R_2(\Theta) \Rightarrow CF_{AP_{A_1}(\omega_1)} = CF_{AP_{A_2}(\omega_2)}$ очевидно, лемма доказана.

3. ПРОБЛЕМА ЭКВИВАЛЕНТНОСТИ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В НЕКОММУТАТИВНОМ АЛФАВИТЕ

Как отмечалось в начале статьи, при доказательстве также будем использовать понятие многомерной ленты, введенное в предыдущем разделе. Определения понятий предшественницы и пути будут изменены. Сформулируем и докажем лемму о разрешимости проблемы эквивалентности регулярных выражений в некоммутативном алфавите. Это также известный результат [4], доказанный другим способом.

Проанализируем случай, когда $|Y| = 2$.

Пусть $Y = \{y_1, y_2\}$, $y_1 y_2 \neq y_2 y_1$. Рассмотрим двумерную ленту N^2. Используя бинарную кодировку, приведенную в разд. 1: (1, 0) для y_1 и (0, 1) для y_2 и т.д., получим следующее соответствие между словами F_Y и ячейками N^2: $K(e) = (0, 0)$, $K(y_1) = (1, 0)$, $K(y_2) = (0, 1)$, $K(y_1 y_1) = (11, 00) = (3, 0)$, $K(y_1 y_2) = (10, 01) = (2, 1)$, $K(y_2 y_1) = (01, 10) = (1, 2)$, $K(y_2 y_2) = (00, 11) = (0, 3)$, ... (рис. 3).

Таким образом, следы выполнения определяются с помощью диагоналей ленты и следующего алгоритма движения: шаг 0 — диагональ с номером 0, шаг $i+1$ — диагональ с номером $d_i + 2^i$. В этом случае номер диагонали не совпадает с номером шага.

Соответственно, следы выполнения связаны только с ячейками отмеченных диагоналей. В то же время в этом случае множество состояний для данной ячейки ленты состоит только из одного состояния и каждая ячейка имеет лишь одну предшественницу. Координаты предшественницы получаются удалением самой левой единицы и самого левого нуля из координат данной ячейки.

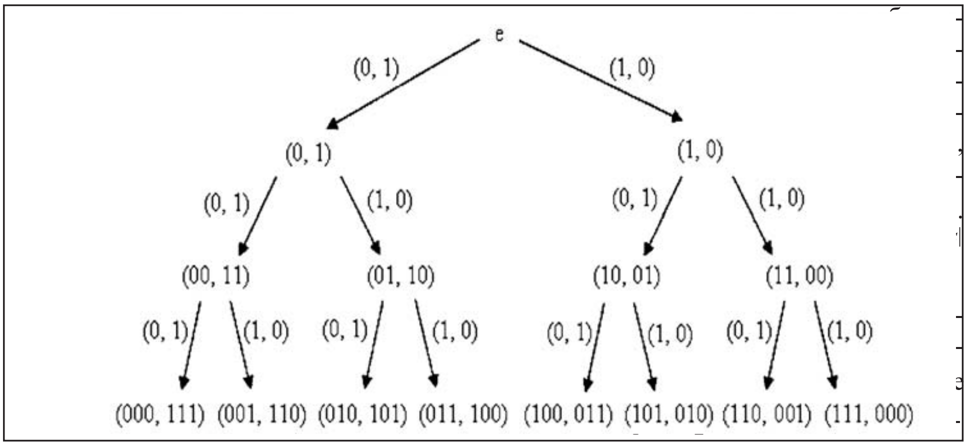


Рис. 3

ельства аналогична идее доказательства леммы 5.

Лемма 6. $CF_{AP_{A_1}}(\omega_1) = CF_{AP_{A_2}}(\omega_2) \Leftrightarrow R_1 \sim R_2(\Theta)$.

Доказательство. Длина k слов следов достаточна, чтобы провести доказательство аналогично доказательству леммы 5, так как в этом случае множество со-

стояние, а число рассматривае-

чению размерности ленты, т.е.

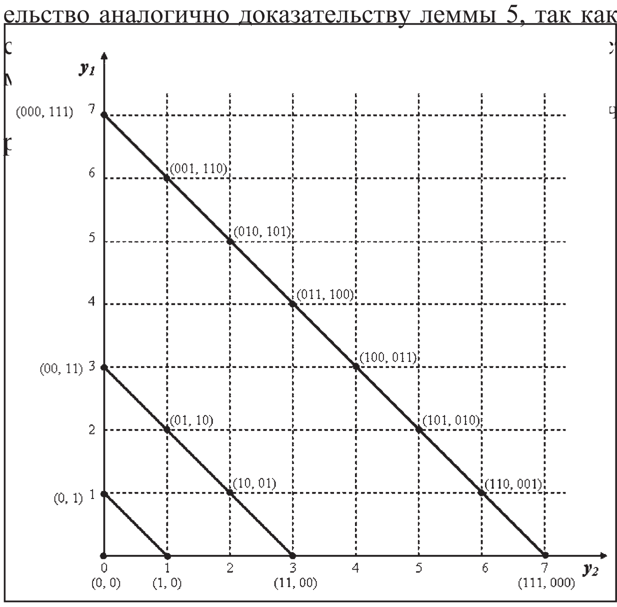


Рис. 4

Аналогично определению в разд. 2 путь $p = a_{p_1} \ldots a_{p_m}$, $m \geq 1$, $a_{p_j} \in U_\omega$, $j \in \{1, \ldots, m\}$, рассматривается как последовательность ячеек перечисленных диагоналей, где $\pi(a_{p_v}, a_{p_{v+1}})$, $v = 1, \ldots, m-1$.

Понятия полного пути, принимаемого пути и канонической формы определяются так же, как в разд. 2. Понятие характеристики для данного пути p отличается от понятия характеристики, приведенного в разд. 2, следующим: разница координат не равна единице, а является переменной и равна 2^i для шага i. Поскольку для

Предположим, что R_1 не Θ-эквивалентно R_2 и в словах ω_1 и ω_2 не существует пути длины k, который принимается автоматом A_1, но не принимается автоматом A_2, или, наоборот, который принимается автоматом A_2, но не принимается автоматом A_1.

В силу предположения существует полный путь $p = a_{p_1} \ldots a_{p_m}$ длины, большей k, $m > k$, который принимается одним из автоматов, скажем A_1, но не принимается автоматом A_2.

Это означает, что $\varphi_{A_1}(a_{p_m})$ содержит заключительное состояние автомата A_1, тогда как $\varphi_{A_2}(a_{p_m})$ не содержит заключительного состояния автомата A_2. Поскольку $m > k$, существует две ячейки a и a' в пути p, такие, что сумма координат ячейки a' больше суммы координат ячейки a, $\varphi_{A_1}(a) = \varphi_{A_2}(a')$ и $p = a_{p_1} \ldots a \ldots a' \ldots a_{p_m}$.

Пусть $\chi_{p'}$ — характеристика подпути $p' = a' \ldots a_{p_m}$ пути p и $p'' = a \ldots a''$ — путь от ячейки a до ячейки a'', имеющий ту же характеристику $\chi_{p'}$. Очевидно, что такой путь существует. Обозначим p''' подпуть пути p, который начинается с a_{p_1} и оканчивается предшественницей ячейки a. Конкатенацию двух путей p''' и p'' обозначим $p_{\text{new}} = p''' p''$. Его длина меньше длины начального пути p. Очевидно, что $\varphi_{A_1}(a'')$ содержит заключительное состояние автомата A_1, а $\varphi_{A_2}(a'')$ не содержит заключительного состояния автомата A_2. Если длина пути p_{new} больше k, то выполняем процедуру, описанную выше, до тех пор, пока не получим путь длины, не превышающей k. Но это противоречит начальному предположению о том, что не существует такого пути в словах следов ω_1 и ω_2, который принимался бы одним из автоматов, но не принимался бы другим.

Поскольку утверждение $R_1 \sim R_2(\Theta) \Rightarrow CF_{AP_{A_1}(\omega_1)} = CF_{AP_{A_2}(\omega_2)}$ очевидно, лемма доказана.

3. ПРОБЛЕМА ЭКВИВАЛЕНТНОСТИ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В НЕКОММУТАТИВНОМ АЛФАВИТЕ

Как отмечалось в начале статьи, при доказательстве также будем использовать понятие многомерной ленты, введенное в предыдущем разделе. Определения понятий предшественницы и пути будут изменены. Сформулируем и докажем лемму о разрешимости проблемы эквивалентности регулярных выражений в некоммутативном алфавите. Это также известный результат [4], доказанный другим способом.

Проанализируем случай, когда $|Y| = 2$.

Пусть $Y = \{y_1, y_2\}$, $y_1 y_2 \neq y_2 y_1$. Рассмотрим двумерную ленту N^2. Используя бинарную кодировку, приведенную в разд. 1: (1, 0) для y_1 и (0, 1) для y_2 и т.д., получим следующее соответствие между словами F_Y и ячейками N^2: $K(e) = (0,0)$, $K(y_1) = (1,0)$, $K(y_2) = (0,1)$, $K(y_1 y_1) = (11,00) = (3,0)$, $K(y_1 y_2) = (10,01) = (2,1)$, $K(y_2 y_1) = (01,10) = (1,2)$, $K(y_2 y_2) = (00,11) = (0,3)$, ... (рис. 3).

Таким образом, следы выполнения определяются с помощью диагоналей ленты и следующего алгоритма движения: шаг 0 — диагональ с номером 0, шаг $i+1$ — диагональ с номером $d_i + 2^i$. В этом случае номер диагонали не совпадает с номером шага.

Соответственно, следы выполнения связаны только с ячейками отмеченных диагоналей. В то же время в этом случае множество состояний для данной ячейки ленты состоит только из одного состояния и каждая ячейка имеет лишь одну предшественницу. Координаты предшественницы получаются удалением самой левой единицы и самого левого нуля из координат данной ячейки.

Аналогично определению в разд. 2 путь $p = a_{p_1} \ldots a_{p_m}$, $m \geq 1$, $a_{p_j} \in U_\omega$, $j \in \{1, \ldots, m\}$, рассматривается как последовательность ячеек перечисленных диагоналей, где $\pi(a_{p_v}, a_{p_{v+1}})$, $v = 1, \ldots, m-1$.

Понятия полного пути, принимаемого пути и канонической формы определяются так же, как в разд. 2. Понятие характеристики для данного пути p отличается от понятия характеристики, приведенного в разд. 2, следующим: разница координат не равна единице, а является переменной и равна 2^i для шага i. Поскольку для

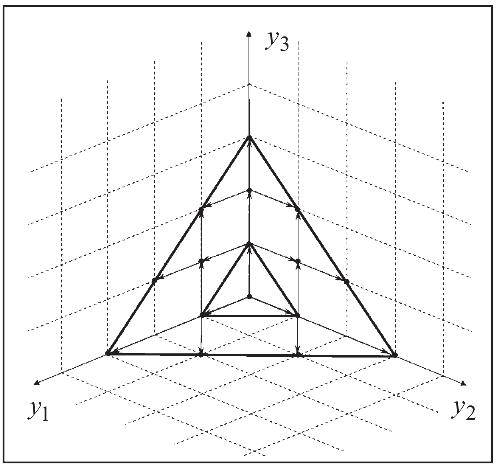


Рис. 5

данного случая рассматривается бинарное представление координат, то соответствующим образом меняется и представление канонической формы. Рассматриваются слова следов, определяемые ограничением на их длину: 2^0, 2^0+2^1, ..., $2^0+2^1+\ldots+2^{k-1}=2^k-1$, где $k=2^{|s|}$ (рис. 4).

Аналогично лемме 5 формулируется и доказывается лемма 6 для слов следов ω_1 и ω_2 длины $k=2^s-1$, где $S=\max\{|S_1|, |S_2|\}$. Идея доказательства аналогична идее доказательства леммы 5.

Лемма 6. $CF_{AP_{A_1}(\omega_1)}=CF_{AP_{A_2}(\omega_2)} \Leftrightarrow R_1 \sim R_2(\Theta)$.

Доказательство. Длина k слов следов достаточна, чтобы провести доказательство аналогично доказательству леммы 5, так как в этом случае множество состояний для каждой ячейки содержит лишь одно состояние, а число рассматриваемых диагоналей равно k.

Случай, когда $|Y|>2$, приводит только к увеличению размерности ленты, т.е. размерность ленты совпадает с $|Y|$.

4. ПРОБЛЕМА ЭКВИВАЛЕНТНОСТИ РЕГУЛЯРНЫХ ВЫРАЖЕНИЙ В ЧАСТИЧНО КОММУТАТИВНОМ АЛФАВИТЕ

Как отмечалось ранее, общий случай проблемы эквивалентности регулярных выражений в частично коммутативном алфавите алгоритмически неразрешим.

Пусть алфавит $Y=\bigcup_{i=1}^{n} Y_i$, где $|Y_1|\geq 2$, $|Y_i|=1$, $i=2,\ldots,n$. Обозначим $|Y_1|$ символом q. Рассмотрим $(n+q-1)$-мерную ленту N^{n+q-1}; q размерностей ленты используется для моделирования движения по символам из Y_1, а остальные $n-1$ — для моделирования движения по символам из Y_i, $i=2,\ldots,n$. Ниже приведены некоторые обобщения рассмотренных понятий для данного случая.

Используем бинарное кодирование координат ячеек аналогично рассмотренным в разд. 1. Если $a=(\alpha_1,\ldots,\alpha_n)$ — ячейка, то бинарное представление координаты α_i обозначим $b(\alpha_i)$, а длину бинарного представления, т.е. количество битов — $|b(\alpha_i)|$. Пусть позиции бинарного представления пронумерованы справа налево, начиная с 0. Тогда номер самого левого значимого бита $b(\alpha_i)$, т.е. самого левого бита бинарного представления, который равен единице, обозначим $\mathrm{NMSB}(\alpha_i)$. Если в $b(\alpha_i)$ нет единиц, то $\mathrm{NMSB}(\alpha_i)=0$.

Ячейка $a_1=(\alpha_{11},\ldots,\alpha_{1n+q-1})$ называется предшественницей ячейки $a_2=(\alpha_{21},\ldots,\alpha_{2n+q-1})$ и соответствующий предикат, обозначаемый $\pi(a_1,a_2)$, имеет значение истина тогда и только тогда, когда $\exists j\in\{1,\ldots,n\}$, $k=1,\ldots,j-1,j+1,\ldots,n+q-1$, что $\alpha_{1k}=\alpha_{2k}$, и

1) $\alpha_{2j}=\alpha_{1j}+0*2^{|b(\alpha_{1j})|+1}$, $|b(\alpha_{2j})|=|b(\alpha_{1j})|+1$, если $j>k$;

2) $\alpha_{2j}=\alpha_{1j}+0*2^{|b(\alpha_{1j})|+1}$, $|b(\alpha_{2j})|=|b(\alpha_{1j})|+1$, если $j\leq k$ и $\mathrm{NMSB}(\alpha_{2j})<|b(\alpha_{2j})|$;

3) $\alpha_{2j}=\alpha_{1j}+1*2^{|b(\alpha_{1j})|+1}$, если $j\leq k$ и $\mathrm{NMSB}(\alpha_{2j})=|b(\alpha_{2j})|$.

Отображение $\varphi_A : N^n \to 2^S$ назовем множеством следов выполнения автомата A тогда и только тогда, когда:

1) $\varphi_A(0,\dots,0)=\{s_0\}$, $P(0,\dots,0)=\varnothing$;

2) $\forall a \in N^n \setminus \{0, \dots, 0\}$, $P(a)=\{a^{(j)}_{\text{pred}},\ \pi_j(a^{(j)}_{\text{pred}}, a)$ истинно, $j=n_1, \dots, n_k$, $k \le n+q-1\}$, $\varphi_A(a)=\Delta(\varphi_A(a^{(n_1)}_{\text{pred}}), y_{n_1}) \cup \dots \cup \Delta(\varphi_A(a^{(n_k)}_{\text{pred}}), y_{n_k})$.

Конечное подмножество следов выполнения, содержащее все ячейки, у которых сумма координат каждой ячейки меньше или равна $n-1$, назовем словом следов (выполнения) длины k. Множество всех слов следов обозначим Ω_A, множество всех ячеек, использованных в данном слове следов ω, — U_ω.

Часть слова следов ω, сумма координат ячеек которой равна k, назовем диагональю k слова следов ω и обозначим $d_k(\omega)$. Длина $d_k(\omega)$ равна $k+1$, длина $\ell(\omega)$ слова следов ω — количеству содержимых диагоналей.

Определим путь $p=a_{p_1}\dots a_{p_m}$, $m \ge 1$, $a_{p_j} \in U_\omega$, $j \in \{1,\dots,m\}$, для данного слова следов ω как последовательность ячеек, для которых $\pi(a_{p_v}, a_{p_{v+1}})$, $v=1,\dots,m-1$.

Для данного пути $p=a_{p_1}\dots a_{p_m}$ слово $\chi_p = y_{p_1}\dots y_{p_{m-1}}$ в алфавите Y назовем характеристикой пути p тогда и только тогда, когда $\forall j \in \{1,\dots,m\}$, $i \in \{1,\dots,n+q-1\}$ (рис. 5):

$$1)\ y_{p_j} \in Y_i,\ i \ne 1 \Rightarrow \alpha_{p_{j+1}}[i]=\alpha_{p_j}[i]+1;$$

$$2)\ y_{p_j} \in Y_1 \Rightarrow \mathrm{NMSB}(\alpha_{p_{j+1}}[i])=|b(\alpha_{p_j}[i])|+1.$$

Путь $p=a_{p_1}\dots a_{p_m}$ назовем полным, если $a_{p_1}=(0, \dots, 0)$. Скажем, что полный путь $p=a_{p_1}\dots a_{p_m}$ принимается автоматом A, если $\varphi_A(a_{p_m})$ содержит заключительное состояние автомата A. Координаты ячейки a_{p_m} назовем канонической формой полного пути $p=a_{p_1}\dots a_{p_m}$.

Для каждого автомата A множество всех принимающихся путей в слове следов ω обозначим $AP_A(\omega)$, а множество их канонических форм — $CF_{AP_A(\omega)}$.

Пусть автоматы A_1 и A_2 распознают в точности регулярные выражения R_1 и R_2 соответственно, а S_1, S_2 — их множества состояний. Предположим также, что $k=2^s-1$, где $S=\max\{|S_1|,|S_2|\}$, ω_1 и ω_2 — слова следов, автоматов A_1 и A_2 длины k.

Лемма 7. $CF_{AP_{A_1}(\omega_1)}=CF_{AP_{A_2}(\omega_2)} \Leftrightarrow R_1 \sim R_2\,(\Theta)$.

Доказательство проводится аналогично доказательству леммы 5. Согласно рассуждениям, проведенным в предыдущем разделе, слово следов длины $2^{|s_i|}-1$ для данного автомата A_i с множеством состояний S_i содержит все полные пути как для «коммутативных», так и для «некоммутативных» размерностей. В то же время рассмотрение полных путей с повторяющимися множествами состояний можно свести к рассмотрению полных путей с неповторяющимися множествами состояний, как в лемме 5.

Теорема. Пусть Y — частично коммутативный алфавит, разбитый на непересекающиеся подмножества символов со следующими свойствами:

1) любые два символа из разных подмножеств перестановочны, а любые два символа из одного и того же подмножества не перестановочны;

2) в разбиении существует лишь одно подмножество с мощностью, большей единицы.

Проблема эквивалентности регулярных выражений над частично коммутативном алфавитом Y алгоритмически разрешима.

Автор выражает глубокую благодарность научному руководителю д-ру физ.-мат. наук, академику А.А. Летичевскому и д-ру физ.-мат. наук А.Б. Годлевскому за ценные советы и замечания в процессе написания и подготовки рукописи к печати.

СПИСОК ЛИТЕРАТУРЫ

1. А х о А . , У л ь м а н Д ж . Теория синтаксического анализа, перевода и компиляции. Т. 1. — М.: Мир, 1978. — 612 с.
2. Р е д ь к о В . Н . Об алгебре коммутативных событий /// Укр. мат. жур. — 1964. — **16**, № 2. — С. 185–195.
3. Т у з о в В . А . Проблемы разрешения для граф-схем с перестановочными операторами. II // Кибернетика. — 1971. — № 5. — С. 23–32.
4. R a b i n M . O . , S c o t t D . Finite automata and their decision problems // IBM Journ. of Res. and Development. — 1959. — **3** (2). — P. 114–125.
5. P r e s t o n C . The Algebraic Theory of Semigroups // Amer. Math. Soc. — 1961. — N 7. — P. 148–159.
6. Г о д л е в с к и й А . Б . , Л е т и ч е в с к и й А . А . , Ш у к у р я н С . К . О сводимости проблемы функциональной эквивалентности схем программ над невырожденным базисом ранга единица к эквивалентности автоматов с многомерными лентами // Кибернетика. — 1980. — № 6. — С. 1–7.
7. G r i g o r i a n H . A . , S h o u k o u r i a n S . K . The equivalence problem of multidimensional multitape automata // Journ. of Comput. and System. Sci. — 2008. — **74**, N 7. — P. 1131–1138.

Поступила 06.11.2008