*Сибирский математический журнал*
*Ноябрь—декабрь, 2002. Том 43, № 6*

*УДК 512:519.4*

# О СЛОЖНОСТИ ПРОБЛЕМЫ РАВЕНСТВА ДЛЯ КОНЕЧНО ОПРЕДЕЛЕННЫХ КОММУТАТИВНЫХ ПОЛУГРУПП

**В. Ю. Попов**

**Аннотация:** Для любой конечно определенной коммутативной полугруппы проблема равенства распознаваема на машине Тьюринга реального времени.

**Ключевые слова:** коммутативная полугруппа, проблема равенства

Рассмотрим бесконечный алфавит $X = \{x_1, x_2, \dots, x_n, \dots\}$. Каждой букве $x_i$ из алфавита $X$ можно поставить в соответствие некоторое натуральное число $s(x_i)$, называемое *размером.* Обычно это длина двоичной последовательности, которой кодируется буква. Обозначим через $X^+$ свободную полугруппу над алфавитом $X$ и распространим определение функции $s$ на неоднобуквенные элементы полугруппы $X^+$ следующим образом: для любого $w \in X^+$ если $w = x_i w'$, то $s(w) = s(x_i) + s(w')$. Пусть $\mathfrak{V}$ — некоторое многообразие полугрупп с разрешимой проблемой равенства. Рассмотрим следующие два множества:

$B_D = \{(u, v, R) \mid u, v$ — слова в алфавите $X$, $R$ — множество соотношений $u_i = v_i$, где $u_i$ и $v_i$ — слова в алфавите $X\}$,

$S_D = \{(u, v, R) \in B_D \mid$ соотношение $u = v$ выводимо из множества соотношений $R$ и множества тождеств, задающих многообразие $\mathfrak{V}\}$.

Определим размер тройки $(u, v, R)$ как сумму:

$$s(u, v, R) = s(u) + s(v) + \sum_{u_i = v_i \in R} s(u_i) + \sum_{u_i = v_i \in R} s(v_i).$$

Проблему равенства для многообразия $\mathfrak{V}$ можно сформулировать как проблему $D$ вхождения элементов множества $B_D$ в множество $S_D$. Так как $\mathfrak{V}$ — многообразие полугрупп с разрешимой проблемой равенства, можно полагать, что $D$ реализуется некоторой машиной Тьюринга с одной лентой и одной головкой, способной считывать информацию с ленты и записывать новую информацию на ленту. При этом можно считать, что машина Тьюринга имеет голосовой синтезатор, способный говорить два слова: «да», «нет». Начиная работать с элемента множества $B_D$, записанного на ленте, машина Тьюринга через конечное число тактов работы останавливается и говорит «да», если записанный на ленте элемент множества $B_D$ является элементом множества $S_D$, и «нет» в противном случае.

Каждому алгоритму $A$, решающему проблему $D$ на машине Тьюринга, можно поставить в соответствие две важные функции:

$s_A(n)$ — пространственная функция сложности,

$t_A(n)$ — временная функция сложности.

Для любого $z \in B_D$ обозначим через $\mathrm{time}_A(z)$ количество тактов работы машины Тьюринга по алгоритму $A$, когда на ленте записан элемент $z$, через $\mathrm{size}_A(z)$ — количество ячеек на ленте, которые посетила головка машины Тьюринга с момента начала работы до завершения, если на ленте записан элемент $z$. Для любого натурального числа $n$ обозначим через $W_n$ множество $\{z \mid z \in B_D,\ s(z) \le n\}$. Если $W_n = \varnothing$, то $s_A(n) = t_A(n) = 0$. Если $W_n \ne \varnothing$, то $s_A(n) = \max\limits_{z \in W_n} \mathrm{size}_A(z)$, $t_A(n) = \max\limits_{z \in W_n} \mathrm{time}_A(z)$. Отметим очевидное неравенство

$$s_A(n) \le t_A(n), \quad n \in \mathbb{N}.$$

Если проблема $D$ реализуется алгоритмом $A$ на машине Тьюринга и функция $t_A(n)$ может быть ограничена сверху некоторой полиномиальной (экспоненциальной) функцией, то говорят, что проблема $D$ может быть решена за полиномиальное (экспоненциальное) время. Если проблема $D$ реализуется алгоритмом $A$ на машине Тьюринга и функция $s_A(n)$ может быть ограничена сверху некоторой полиномиальной (экспоненциальной) функцией, то говорят, что проблема $D$ может быть решена на полиномиальном (экспоненциальном) пространстве.

Следуя [1], введем понятие полиномиальной сводимости одной проблемы к другой. Пусть $Q_1$ и $Q_2$ — две проблемы, каждая из которых задана как проблема вхождения элементов множества $B_{Q_i}$ в его подмножество $S_{Q_i}$. Говорят, что проблема $Q_1$ *полиномиально сводима* к проблеме $Q_2$, если существует функция $\phi$, переводящая элементы множества $B_{Q_1}$ в элементы множества $B_{Q_2}$, такая, что

• $x \in S_{Q_1}$ тогда и только тогда, когда $\phi(x) \in S_{Q_2}$ для любого $x \in B_{Q_1}$,

• по элементу $x$ элемент $\phi(x)$ может быть вычислен за полиномиальное время.

Ясно, что если проблема $Q_1$ полиномиально сводима к проблеме $Q_2$, то сложность проблемы $Q_2$ не может быть меньше сложности проблемы $Q_1$.

Для любого конечно базируемого непериодического многообразия полугрупп $\mathscr{V}$ следующие условия эквивалентны: для любой конечно определенной в $\mathscr{V}$ полугруппы проблема равенства разрешима; проблема равенства в многообразии $\mathscr{V}$ разрешима (см. [2, теорема 3.28], а также [3, 4]). В работах [5] и [6] независимо доказана разрешимость проблемы равенства в многообразии всех коммутативных полугрупп, в [7, 8] — что разрешима элементарная теория любой полугруппы, конечно определенной в многообразии всех коммутативных полугрупп. Однако с точки зрения сложности алгоритмов проблема равенства в многообразии всех коммутативных полугрупп и проблемы равенства в полугруппах, конечно определенных в этом многообразии, существенно различаются.

Будем говорить, что проблема $Q$ является EXPSPACE-полной, если она может быть решена на экспоненциальном пространстве, и всякая проблема, которая может быть решена на экспоненциальном пространстве, полиномиально сводима к проблеме $Q$. В [9] доказано, что существует алгоритм, решающий проблему равенства в многообразии всех коммутативных полугрупп на экспоненциальном пространстве, в [10] — что каждая проблема, которая может быть решена на экспоненциальном пространстве, полиномиально сводима к проблеме равенства в многообразии всех коммутативных полугрупп. Таким образом, проблема равенства в многообразии всех коммутативных полугрупп EXPSPACE-полна. С другой стороны, в [9] доказано, что для любой конечно определенной коммутативной полугруппы существует алгоритм, решающий проблему равенства в этой полугруппе за линейное время.

В [11] доказано, что для любой конечно определенной гиперболической группы существует алгоритм, решающий проблему равенства в этой группе за линейное время. В [12] доказано, что оценку сложности проблемы равенства для гиперболических групп, полученную в [11], можно понизить. Прежде чем сформулировать результат работы [12], введем ряд определений.

Пусть $A$ — конечное подмножество множества $X$. Обозначим через $A^*$ свободный моноид, порожденный множеством $A$. Произвольное подмножество $L$ множества $A^*$ называется *языком.* Проблему равенства для конечно определенной группы $G$ можно сформулировать как проблему вхождения элементов $A^*$ в язык $L = \{u \mid G \models u = 1\}$. Говорят, что *язык $L$ распознаваем в реальном времени*, если его можно распознать на $n$-ленточной машине Тьюринга, работающей в реальном времени, для некоторого $n \geq 0$. Такая машина имеет конечное множество внутренних состояний $q_0, q_1, \ldots, q_m$ и внешний алфавит $a_1, a_2, \ldots, a_p$. При этом $q_0$ — начальное состояние и среди состояний $q_1, q_2, \ldots, q_m$ выделено некоторое множество допустимых состояний. Каждая из $n$ рабочих лент бесконечна в обе стороны. Кроме того, машина имеет $(n+1)$-ю вспомогательную ленту, на которой содержится входная информация, записанная во внешнем алфавите машины. Машина начинает работать в состоянии $q_0$ с пустыми рабочими лентами и головкой, установленной на вспомогательной ленте на первый символ входной информации. За один такт работы машина передвигает вспомогательную ленту так, чтобы головка обозревала следующий символ входной информации. При этом она может изменить внутреннее состояние. На каждой из лент она может написать одну из букв внешнего алфавита, после чего может сдвинуть каждую из лент на одну ячейку вправо или влево. Прочитав последний символ на вспомогательной ленте, машина останавливается. Слово, записанное на вспомогательной ленте, считается допустимым, если машина остановилась в допустимом состоянии. Более подробную информацию о машинах Тьюринга, работающих в реальном времени, можно найти в [13, 14].

Говорят, что машина Тьюринга, работающая в реальном времени, *распознает язык $L$*, если для любого $w \in A^*$ машина, начиная работать с $w$ в качестве входной информации, останавливается в допустимом состоянии тогда и только тогда, когда $w \in L$. В [12] доказано, что по любой конечно определенной гиперболической группе $G$ можно определить натуральное число $\delta$ такое, что язык, соответствующий проблеме равенства в группе $G$, распознаваем на 4-ленточной машине Тьюринга, работающей в реальном времени и позволяющей осуществлять до $96\delta - 2$ операций сдвига и перезаписи на каждой ленте за один такт работы. Отсюда вытекает, что по любой конечно определенной гиперболической группе $G$ можно определить натуральное число $\delta$ такое, что язык, соответствующий проблеме равенства в группе $G$, распознаваем на $(384\delta - 8)$-ленточной машине Тьюринга, работающей в реальном времени. Отметим, что если для некоторой проблемы $D$ существует машина Тьюринга, работающая в реальном времени и решающая проблему $D$, то для проблемы $D$ существует алгоритм, решающий эту проблему за линейное время. Однако существует язык $L$, распознаваемый за линейное время, но не распознаваемый на машине Тьюринга, работающей в реальном времени.

Для конечно определенной полугруппы $S$ с множеством образующих элементов $a_1, a_2, \dots, a_p$ рассмотрим свободную полугруппу

$$\{a_1, a_2, \dots, a_p, [=]\}^+, \tag{1}$$

где $[=]$ — новая буква. Очевидно, что проблему равенства для полугруппы $S$ можно рассматривать как проблему вхождения элементов множества (1) в подмножество $L = \{u[=]v \mid S \models u = v\}$ и по аналогии с случаем групп ставить вопрос о распознаваемости множества $L$ машиной Тьюринга, работающей в реальном времени. В свете результатов [9, 12] становится естественным вопрос о распознаваемости проблемы равенства для коммутативных полугрупп на машинах Тьюринга, работающих в реальном времени. Ответ на него дает следующая

**Теорема.** *Для любой конечно определенной коммутативной полугруппы проблема равенства распознаваема на машине Тьюринга, работающей в реальном времени.*

Доказательство теоремы. Каждая конечно определенная коммутативная полугруппа $S$ с $p$ образующими является фактор-полугруппой конечно порожденной свободной коммутативной полугруппы $A_p$ с $p$ образующими по некоторой конечно порожденной конгруэнции $\sigma$. В [15] доказано, что для каждой конечно порожденной конгруэнции $\sigma$ свободной коммутативной полугруппы $A_p$ можно эффективно построить формулу логики первого порядка $\theta(x, y)$ с двумя свободными переменными $x$ и $y$ такую, что

$$A_{2p} \models \theta(x, y) \Leftrightarrow (x, y) \in \sigma.$$

В [16] доказано, что множество $T \subseteq A_n$ выделяется формулой логики первого порядка тогда и только тогда, когда является эффективно строящимся конечно порожденным полулинейным множеством, т. е. объединением конечного числа множеств вида

$$L(x, P) = \{x + n_1p_1 + n_2p_2 + \dots n_kp_k \mid n_1, n_2, \dots,\ n_k \in \mathbb{N},\ p_1, p_2, \dots,\ p_k \in P\}, \tag{2}$$

где $x$ — фиксированный элемент полугруппы $A_n$, $P$ — некоторое конечное подмножество полугруппы $A_n$ (о представимости конечно определенных коммутативных полугрупп полулинейными множествами см. также [17]). Таким образом, для доказательства теоремы нам достаточно показать, что для каждого конечно порожденного полулинейного множества существует машина Тьюринга, работающая в реальном времени и распознающая это множество.

Рассмотрим некоторое конечно порожденное полулинейное множество $L$. По определению $L$ можно представить в виде конечного объединения множеств вида (2). Допустим, что $L = \bigcup\limits_{r=1}^{l} L_r(x_r, P_r)$, где

$$L_r(x_r, P_r) = \{x_r + n_1p_{1,r} + n_2p_{2,r} + \dots n_{k_r}p_{k_r,r} \mid \\ n_1, n_2, \dots, n_{k_r} \in \mathbb{N},\ p_{1,r}, p_{2,r}, \dots, p_{k_r,r} \in P_r\}$$

и $x_r \in A_n$, $P_r \subseteq A_n$. Предположим, что для любого $r$ существует $n_r$-ленточная машина Тьюринга $\mathscr{T}_r$, работающая в реальном времени и распознающая множество $L_r(x_r, P_r)$. Для любого $r \in \{1, 2, \dots, l\}$ введем следующие обозначения:

$m_r + 1$ — количество внутренних состояний машины $\mathscr{T}_r$,

$k_r$ — количество допустимых состояний машины $\mathscr{T}_r$,

$p_r$ — количество символов внешнего алфавита машины $\mathscr{T}_r$,

$q_0^r, q_1^r, \ldots, q_{m_r}^r$ — множество всех внутренних состояний машины $\mathscr{T}_r$,
$q_0^r$ — начальное состояние машины $\mathscr{T}_r$,
$q_1^r, q_2^r, \ldots, q_{k_r}^r$ — допустимые состояния машины $\mathscr{T}_r$,
$a_1, a_2, \ldots, a_{p_r}$ — внешний алфавит машины $\mathscr{T}_r$,
$\mathfrak{T}_r$ — программа машины $\mathscr{T}_r$.

Множество $\mathfrak{T}_r$ состоит из команд вида

$$q_i^r \alpha \beta_1 \beta_2 \ldots \beta_{n_r} \to q_j^r T_1 \gamma_1 T_2 \gamma_2 \ldots T_{n_r} \gamma_{n_r},$$

где $q_i^r$ — внутреннее состояние машины $\mathscr{T}_r$ до выполнения команды, $q_j^r$ — внутреннее состояние машины $\mathscr{T}_r$ после выполнения команды, $\alpha$ — символ на вспомогательной ленте, обозреваемый головкой машины $\mathscr{T}_r$ до выполнения команды, $\alpha \in \{a_1, a_2, \ldots, a_{p_r}\}$, $\beta_k$ — символ на $k$-й рабочей ленте, обозреваемый головкой машины $\mathscr{T}_r$ до выполнения команды, $\beta_k \in \{a_1, a_2, \ldots, a_{p_r}, 1\}$, где 1 — пустой символ, $T_k \in \{-1, 0, 1\}$, если $T_k = -1$, то $k$-я рабочая лента в результате выполнения команды перемещается на одну ячейку вправо, если $T_k = 1$, то $k$-я рабочая лента в результате выполнения команды перемещается на одну ячейку влево, если $T_k = 0$, то $k$-я рабочая лента в результате выполнения команды остается на месте, $\gamma_k \in \{a_1, a_2, \ldots, a_{p_r}, 1\}$, если $\gamma_k \neq 1$, то в обозреваемую ячейку $k$-й рабочей ленты записывается символ $\gamma_k$, если $\gamma_k = 1$, то на $k$-ю рабочую ленту ничего не записывается.

Пусть $\mathscr{T}_{1,2,\ldots,l}$ — машина Тьюринга, работающая в реальном времени, с $n_1 + n_2 + \cdots + n_l$ рабочими лентами, заданная следующим образом.

• Для любых $r \in \{1, 2, \ldots, l\}$ и $k \in \{1, 2, \ldots, n_r\}$ ленте с номером $n_1 + n_2 + \cdots + n_{r-1} + k$ машины $\mathscr{T}_{1,2,\ldots,l}$ соответствует лента с номером $k$ машины $\mathscr{T}_r$.

• Количество символов внешнего алфавита машины $\mathscr{T}_{1,2,\ldots,l}$ равно $p = \max\{p_1, p_2, \ldots, p_r\}$.

• Внешний алфавит машины $\mathscr{T}_{1,2,\ldots,l}$ состоит из символов $a_1, a_2, \ldots, a_p$.

• Число внутренних состояний машины $\mathscr{T}_{1,2,\ldots,l}$ равно

$$m = (m_1 + 1) \cdot (m_2 + 1) \cdot \cdots \cdot (m_l + 1).$$

При этом существует взаимно однозначное отображение $\mathfrak{m}$ множества

$$\{(i_1, i_2, \ldots, i_l) \mid 0 \leq i_j \leq m_j, j \in \{1, 2, \ldots, l\}\}$$

на множество $\{1, 2, \ldots, m\}$, ставящее в соответствие внутренним состояниям $q_{i_1}^1, q_{i_2}^2, \ldots, q_{i_l}^l$ машин $\mathscr{T}_1, \mathscr{T}_2, \ldots, \mathscr{T}_l$ внутреннее состояние $q_{\mathfrak{m}(i_1, i_2, \ldots, i_l)}$ машины $\mathscr{T}_{1,2,\ldots,l}$.

• $q_{\mathfrak{m}(0,0,\ldots,0)}$ — начальное состояние машины $\mathscr{T}_{1,2,\ldots,l}$.

• $\{q_{\mathfrak{m}(i_1, i_2, \ldots, i_l)} \mid \exists j (j \in \{1, 2, \ldots, l\} \wedge 1 \leq i_j \leq k_j)\}$ — множество допустимых состояний машины $\mathscr{T}_{1,2,\ldots,l}$.

• Программа машины $\mathscr{T}_{1,2,\ldots,l}$ содержит команду

$$q_i \alpha \beta_1 \beta_2 \ldots \beta_{n_1+n_2+\ldots n_l} \to q_j T_1 \gamma_1 T_2 \gamma_2 \ldots T_{n_1+n_2+\ldots n_l} \gamma_{n_1+n_2+\ldots n_l}$$

тогда и только тогда, когда для любого $r$ программа машины $\mathscr{T}_r$ содержит команду

$$q_{i_r}^r \alpha \beta_1' \beta_2' \ldots \beta_{n_r}' \to q_{j_r}^r T_1' \gamma_1' T_2' \gamma_2' \ldots T_{n_r}' \gamma_{n_r}',$$

причем для любого $k \in \{1, 2, \ldots, n_r\}$ выполняются соотношения

$$\beta_k' = \beta_{n_1+n_2+\ldots n_{r-1}+k}, \quad T_k' = T_{n_1+n_2+\ldots n_{r-1}+k}, \quad \gamma_k' = \gamma_{n_1+n_2+\ldots n_{r-1}+k},$$

$$\mathfrak{m}(i_1, i_2, \ldots, i_l) = i, \quad \mathfrak{m}(j_1, j_2, \ldots, j_l) = j.$$

Нетрудно убедиться в том, что если для любого $r$ машина Тьюринга $\mathcal{T}_r$, работающая в реальном времени, распознает множество $L_r(x_r, P_r)$, то машина $\mathcal{T}_{1,2,\ldots,l}$ распознает множество $L = \bigcup\limits_{r=1}^{l} L_r(x_r, P_r)$. Поэтому для доказательства теоремы нам достаточно показать, что для любых $x \in A_n$ и $P \subseteq A_n$ существует машина Тьюринга, работающая в реальном времени и распознающая множество $L(x, P)$.

Пусть $y^0$ — пустой символ для любого символа $y$. Тогда произвольный элемент полугруппы $A_n$ с множеством свободных образующих $a_1, a_2, \ldots, a_n$ можно представить в виде $t_1a_1 + t_2a_2 + \cdots + t_na_n$ и равенство

$$t_1a_1 + t_2a_2 + \cdots + t_na_n = s_1a_1 + s_2a_2 + \cdots + s_na_n$$

выполняется в полугруппе $A_n$ тогда и только тогда, когда для любого $j$ на множестве неотрицательных целых чисел имеет место равенство $t_j = s_j$. Пусть

$$P = \{p_1, p_2, \ldots, p_k\},$$

$$y = s_{1,y}a_1 + s_{2,y}a_2 + \cdots + s_{n,y}a_n,$$

$$x = s_{1,x}a_1 + s_{2,x}a_2 + \cdots + s_{n,x}a_n,$$

$$p_j = s_{1,p_j}a_1 + s_{2,p_j}a_2 + \cdots + s_{n,p_j}a_n, \quad p_j \in P.$$

Тогда $y \in L(x, P)$ в том и только в том случае, если система уравнений

$$\begin{cases} s_{1,y} = s_{1,x} + s_{1,p_1}n_1 + s_{1,p_2}n_2 + \cdots + s_{1,p_k}n_k \\ s_{2,y} = s_{2,x} + s_{2,p_1}n_1 + s_{2,p_2}n_2 + \cdots + s_{2,p_k}n_k \\ \qquad \cdots \\ s_{n,y} = s_{n,x} + s_{n,p_1}n_1 + s_{n,p_2}n_2 + \cdots + s_{n,p_k}n_k \end{cases} \tag{3}$$

разрешима относительно $n_1, n_2, \ldots, n_k$ на множестве неотрицательных целых чисел.

Стандартными преобразованиями систему уравнений (3) можно привести к виду

$$\begin{cases} n_{i_1} = \delta_{i_1,1}(s_{1,y} - s_{1,x}) + \delta_{i_1,2}(s_{2,y} - s_{2,x}) + \cdots + \delta_{i_1,n}(s_{n,y} - s_{n,x}) \\ n_{i_2} = \delta_{i_2,1}(s_{1,y} - s_{1,x}) + \delta_{i_2,2}(s_{2,y} - s_{2,x}) + \cdots + \delta_{i_2,n}(s_{n,y} - s_{n,x}) \\ \qquad \cdots \\ n_{i_t} = \delta_{i_t,1}(s_{1,y} - s_{1,x}) + \delta_{i_t,2}(s_{2,y} - s_{2,x}) + \cdots + \delta_{i_t,n}(s_{n,y} - s_{n,x}) \end{cases} \tag{4}$$

или к виду

$$\begin{cases} n_{i_1} = \delta_{i_1,1}(s_{1,y} - s_{1,x}) + \delta_{i_1,2}(s_{2,y} - s_{2,x}) + \cdots + \delta_{i_1,n}(s_{n,y} - s_{n,x}) \\ \qquad + \varepsilon_{i_1,j_1}n_{j_1} + \varepsilon_{i_1,j_2}n_{j_2} + \cdots + \varepsilon_{i_1,j_{k-t}}n_{j_{k-t}} \\ n_{i_2} = \delta_{i_2,1}(s_{1,y} - s_{1,x}) + \delta_{i_2,2}(s_{2,y} - s_{2,x}) + \cdots + \delta_{i_2,n}(s_{n,y} - s_{n,x}) \\ \qquad + \varepsilon_{i_2,j_1}n_{j_1} + \varepsilon_{i_2,j_2}n_{j_2} + \cdots + \varepsilon_{i_2,j_{k-t}}n_{j_{k-t}} \\ \qquad \cdots \\ n_{i_t} = \delta_{i_t,1}(s_{1,y} - s_{1,x}) + \delta_{i_t,2}(s_{2,y} - s_{2,x}) + \cdots + \delta_{i_t,n}(s_{n,y} - s_{n,x}) \\ \qquad + \varepsilon_{i_t,j_1}n_{j_1} + \varepsilon_{i_t,j_2}n_{j_2} + \cdots + \varepsilon_{i_t,j_{k-t}}n_{j_{k-t}}, \end{cases} \tag{5}$$

где $1 \le i_1 < i_2 < \cdots < i_t \le k$, $\delta_{x,y}, \varepsilon_{x,y}$ — рациональные константы. Мы рассмотрим только случай системы (4), поскольку случай системы (5) рассматривается аналогично. Пусть $\delta > 0$ — наименьший общий знаменатель чисел $\delta_{x,y}$, где $1 \le x \le k$, $1 \le y \le n$. Тогда, положив $\tau_{x,y} = \delta_{x,y} \cdot \delta$, из системы (4) получим систему уравнений

$$\begin{cases} n_{i_1} = \frac{\tau_{i_1,1}(s_{1,y}-s_{1,x})+\tau_{i_1,2}(s_{2,y}-s_{2,x})+\dots+\tau_{i_1,n}(s_{n,y}-s_{n,x})}{\delta} \\ n_{i_2} = \frac{\tau_{i_2,1}(s_{1,y}-s_{1,x})+\tau_{i_2,2}(s_{2,y}-s_{2,x})+\dots+\tau_{i_2,n}(s_{n,y}-s_{n,x})}{\delta} \\ \dots \\ n_{i_t} = \frac{\tau_{i_t,1}(s_{1,y}-s_{1,x})+\tau_{i_t,2}(s_{2,y}-s_{2,x})+\dots+\tau_{i_t,n}(s_{n,y}-s_{n,x})}{\delta}. \end{cases} \tag{6}$$

Обозначим через $\tau$ число $\max\limits_{1\le x\le k, 1\le y\le n}\{|\tau_{x,y}|\}$.

Приступим к построению машины Тьюринга $\mathscr{T}$, работающей в реальном времени и распознающей множество $L(x,P)$. Пусть $\mathscr{T}$ — машина Тьюринга, заданная следующим образом.

• Внешний алфавит машины $\mathscr{T}$ состоит из символов $a_1, a_2, \dots, a_n, 1, -1$.

• Машина $\mathscr{T}$ имеет $t\tau$ лент, которые мы будем считать занумерованными натуральными числами $1, 2, \dots, t\tau$.

• $q(s_{1,x}, s_{2,x}, \dots, s_{n,x}, 0, 1, 1, 0, 1, 1, \dots, 0, 1, 1)$ — начальное состояние машины.

• Все внутренние состояния машины, кроме состояния $q_1$, мы будем индексировать последовательностью, состоящей из $3t+n$ чисел: $q(b_1, b_2, \dots, b_{3t+n})$. При этом полагаем, что $b_i \in \{0, 1, 2, \dots, s_{i,x}\}$ для любого $i \in \{1, 2, \dots, n\}$, $b_i \in \{0, 1, 2, \dots, \delta - 1\}$ для любого $i \in \{n+3j-2 \mid j \in \{1, 2, \dots, t\}\}$, $b_i \in \{1, 2, \dots, \tau\}$ для любого $i \in \{n+3j-1 \mid j \in \{1, 2, \dots, t\}\}$, $b_i \in \{-1, 1\}$ для любого $i \in \{n+3j \mid j \in \{1, 2, \dots, t\}\}$.

• Внутреннее состояние $q(b_1, b_2, \dots, b_{3t+n})$ машины будем считать допустимым, если $b_i = 0$ для любого $i \in \{1, 2, \dots, n\}$, $b_i = 0$ для любого $i \in \{n+3j-2 \mid j \in \{1, 2, \dots, t\}\}$, $b_i \in \{1, 2, \dots, \tau\}$ для любого $i \in \{n+3j-1 \mid j \in \{1, 2, \dots, t\}\}$, $b_i = 1$ для любого $i \in \{n+3j \mid j \in \{1, 2, \dots, t\}\}$. Состояние $q_1$ допустимыми не является.

Опишем основные принципы построения программы машины $\mathscr{T}$ и команд, из которых она состоит.

На вспомогательной ленте записано слово длины $l$ во внешнем алфавите. На $i$-м такте работы машины просматривается $i$-я буква слова. Если это 1 или $-1$, то слово, записанное на вспомогательной ленте, не является элементом свободной коммутативной полугруппы $A_n$, а значит, и множества $L(x,P)$. Поэтому машина переходит в состояние $q_1$ и последующие $l-i$ тактов лишь перемещает вспомогательную ленту, не производя никаких преобразований на рабочих лентах и не меняя внутреннего состояния. В результате машина закончит работу в состоянии, которое не является допустимым, что будет означать: слово, записанное на вспомогательной ленте, не является элементом множества $L(x,P)$.

Теперь рассмотрим случай, когда слово, записанное на вспомогательной ленте, не содержит букв 1 и $-1$. В этом случае очевидно, что слово имеет вид

$$a_{h_1} a_{h_2} \dots a_{h_l} \tag{7}$$

и является элементом полугруппы $A_n$, представленным в мультипликативной записи. Следовательно, оно равно слову

$$s_{1,y}a_1 + s_{2,y}a_2 + \dots + s_{n,y}a_n,$$

где $s_{1,y} + s_{2,y} + \cdots + s_{n,y} = l$. Отсюда вытекает, что слово (7) является элементом множества $L(x, P)$ тогда и только тогда, когда $n_{i_1}, n_{i_2}, \ldots, n_{i_t}$, полученные из системы (6), являются неотрицательными целыми числами. Из системы (3) очевидным образом вытекает, что для того чтобы $n_{i_1}, n_{i_2}, \ldots, n_{i_t}$ были неотрицательными целыми числами, необходимо, чтобы выполнялась система неравенств:

$$\begin{cases} s_{1,y} \geq s_{1,x} \\ s_{2,y} \geq s_{2,x} \\ \qquad \cdots \\ s_{n,y} \geq s_{n,x}. \end{cases} \tag{8}$$

Для проверки выполнимости системы неравенств (8) предусмотрены первые $n$ индексов внутреннего состояния: если на каком-либо шаге головка машины обозревает на вспомогательной ленте букву $a_i$ и $b_i$ в текущем внутреннем состоянии машины $q(b_1, b_2, \ldots, b_{3t+n})$ отлично от 0, то машина на этом шаге оставляет рабочие ленты без изменения и переходит в состояние

$$q(b_1, b_2, \ldots, b_{i-1}, b_i - 1, b_{i+1}, b_{i+2}, \ldots, b_{3t+n}).$$

Если же $b_i = 0$, то это означает, что при выполнении предыдущих тактов работы буква $a_i$ на вспомогательной ленте встречалась не менее $s_{i,x}$ раз, а значит, $i$-е неравенство системы (8) выполняется. Кроме того, если буква $a_j$ на вспомогательной ленте уже встретилась ровно $s_{j,x}$ раз, то от системы (6) можно перейти к системе

$$\begin{cases} n_{i_1} = \frac{1}{\delta}(\tau_{i_1,1}(s_{1,y} - s_{1,x}) + \tau_{i_1,2}(s_{2,y} - s_{2,x}) + \cdots \\ \qquad + \tau_{i_1,j-1}(s_{j-1,y} - s_{j-1,x}) + \tau_{i_1,j} s'_{j,y} + \tau_{i_1,j+1}(s_{j+1,y} - s_{j+1,x}) \\ \qquad + \tau_{i_1,j+2}(s_{j+2,y} - s_{j+2,x}) + \cdots + \tau_{i_1,n}(s_{n,y} - s_{n,x})) \\ n_{i_2} = \frac{1}{\delta}(\tau_{i_2,1}(s_{1,y} - s_{1,x}) + \tau_{i_2,2}(s_{2,y} - s_{2,x}) + \cdots \\ \qquad + \tau_{i_2,j-1}(s_{j-1,y} - s_{j-1,x}) + \tau_{i_2,j} s'_{j,y} + \tau_{i_2,j+1}(s_{j+1,y} - s_{j+1,x}) \\ \qquad + \tau_{i_2,j+2}(s_{j+2,y} - s_{j+2,x}) + \cdots + \tau_{i_2,n}(s_{n,y} - s_{n,x})) \\ \qquad\qquad \cdots \\ n_{i_t} = \frac{1}{\delta}(\tau_{i_t,1}(s_{1,y} - s_{1,x}) + \tau_{i_t,2}(s_{2,y} - s_{2,x}) + \cdots \\ \qquad + \tau_{i_t,j-1}(s_{j-1,y} - s_{j-1,x}) + \tau_{i_t,j} s'_{j,y} + \tau_{i_t,j+1}(s_{j+1,y} - s_{j+1,x}) \\ \qquad + \tau_{i_t,j+2}(s_{j+2,y} - s_{j+2,x}) + \cdots + \tau_{i_t,n}(s_{n,y} - s_{n,x})) \end{cases} \tag{9}$$

и предполагать, что в системе (9) число $s'_{j,y}$ неотрицательное целое. В процессе работы машины мы либо полностью перейдем от рассмотрения системы (6) к системе

$$\begin{cases} n_{i_1} = \frac{1}{\delta}(\tau_{i_1,1} s'_{1,y} + \tau_{i_1,2} s'_{2,y} + \cdots + \tau_{i_1,n} s'_{n,y}) \\ n_{i_2} = \frac{1}{\delta}(\tau_{i_2,1} s'_{1,y} + \tau_{i_2,2} s'_{2,y} + \cdots + \tau_{i_2,n} s'_{n,y}) \\ \qquad\qquad \cdots \\ n_{i_t} = \frac{1}{\delta}(\tau_{i_t,1} s'_{1,y} + \tau_{i_t,2} s'_{2,y} + \cdots + \tau_{i_t,n} s'_{n,y}) \end{cases}$$

(возможно, на последнем шаге), предполагая, что $s'_{1,y}, s'_{2,y}, \ldots, s'_{n,y}$ — неотрицательные числа, либо по завершении работы машины найдется $k \in \{1, 2, \ldots, n\}$ такое, что $b_k > 0$. В последнем случае машина завершит работу во внутреннем состоянии, не являющемся допустимым. Это будет означать, что слово (7) не принадлежит множеству $L(x, P)$ и будет согласовываться с тем, что $k$-е неравенство в системе (8) не выполняется.

Перед началом работы машины все рабочие ленты пусты. Это интерпретирует, что текущие значения решений системы (6) равны:

$$n_{i_1} = 0,\ n_{i_2} = 0,\ \ldots,\ n_{i_t} = 0.$$

При каждом прочтении головкой машины буквы $a_j$ на вспомогательной ленте начиная с $s_{j,x}+1$-го машина осуществляет действия, интерпретирующие прибавление к уже имеющимся текущим значениям решений $n_{i_1}, n_{i_2}, \ldots, n_{i_t}$ чисел $\tau_{i_1,j}, \tau_{i_2,j}, \ldots, \tau_{i_t,j}$ соответственно. Для этого для любого $k \in \{1, 2, \ldots, t\}$ индекс $b_{n+3k-2}$ внутреннего состояния машины $q(b_1, b_2, \ldots, b_{3t+n})$, интерпретирующий остаток от деления на $\delta$ текущего значения $n_{i_k}$, заменяется на $b'_{n+3k-2} = b_{n+3k-2} + \tau_{i_t,j} (\mathrm{mod}\, \delta)$. Для любого $k \in \{1, 2, \ldots, t\}$ следует рассмотреть два случая в зависимости от знака $\tau_{i_k,j}$. Предположим, что $\tau_{i_k,j} \geq 0$, $b_{n+3k-1} > 1$ и на ленте с номером $\tau(k-1) + b_{n+3k-1} - 1$ головка машины обозревает $-1$ или $b_{n+3k-1} = 1$ и $b_{n+3k} = -1$. Тогда если $\tau_{i_k,j} \leq b_{n+3k-1} - 1$, то в текущих ячейках лент с номерами $(k-1)\tau + b_{n+3k-1} - \tau_{i_k,j}$, $(k-1)\tau + b_{n+3k-1} - \tau_{i_k,j} + 1, \ldots$, $(k-1)\tau + b_{n+3k-1} - 1$ буква $-1$ заменяется пустым символом и каждая из этих лент сдвигается на 1 ячейку влево. Остальные ленты с номерами от $(k-1)\tau + 1$ до $k\tau$ оставляются без изменения. Индекс $b_{n+3k-1}$, интерпретирующий номер ленты, относительно которой производятся текущие преобразования, заменяется на $b'_{n+3k-1} = b_{n+3k-1} - \tau_{i_k,j}$. Если на одной из лент с номерами от $(k-1)\tau + 1$ до $k\tau$ остается $-1$, то $b_{n+3k}$ — индекс, интерпретирующий знак текущего значения $n_{i_k}$, заменяется на $b'_{n+3k} = -1$, иначе $b_{n+3k}$ заменяется на $b'_{n+3k} = 1$. Если $\tau_{i_k,j} > b_{n+3k-1} - 1$ и на ленте с номером $k\tau$ головка машины обозревает символ $-1$, то в текущих ячейках лент с номерами от $(k-1)\tau + 1$ до $(k-1)\tau + b_{n+3k-1} - 1$ и от $k\tau + b_{n+3k-1} - 1 - \tau_{i_k,j}$ до $k\tau$ буква $-1$ заменяется пустым символом и каждая из этих лент сдвигается на 1 ячейку влево. Остальные ленты с номерами от $(k-1)\tau + 1$ до $k\tau$ остаются без изменения. Индекс $b_{n+3k-1}$ заменяется на $b'_{n+3k-1} = \tau + b_{n+3k-1} - \tau_{i_k,j}$. Если на одной из лент с номерами от $(k-1)\tau + 1$ до $k\tau$ остается $-1$, то $b_{n+3k}$ заменяется на $b'_{n+3k} = -1$, иначе $b_{n+3k}$ заменяется на $b'_{n+3k} = 1$. Если $\tau_{i_k,j} > b_{n+3k-1} - 1$ и на ленте с номером $k\tau$ головка машины обозревает пустой символ, то в текущих ячейках лент с номерами от $(k-1)\tau + 1$ до $(k-1)\tau + \tau_{i_k,j} - b_{n+3k-1} + 1$ буква $-1$ заменяется буквой 1 и каждая из этих лент сдвигается на 1 ячейку вправо. В текущих ячейках лент с номерами от $(k-1)\tau + \tau_{i_k,j} - b_{n+3k-1} + 2$ до $(k-1)\tau + b_{n+3k-1} - 1$ буква $-1$ заменяется пустым символом и каждая из этих лент сдвигается на 1 ячейку влево. Остальные ленты с номерами от $(k-1)\tau + 1$ до $k\tau$ остаются без изменения. Индекс $b_{n+3k}$ заменяется на $b'_{n+3k} = 1$, индекс $b_{n+3k-1}$ заменяется на $b'_{n+3k-1} = \tau_{i_k,j} - b_{n+3k-1} + 2$ и индекс $b'_{n+3k-2} = b_{n+3k-2} + \tau_{i_k,j} (\mathrm{mod}\, \delta)$.

Предположим, что $\tau_{i_k,j} \geq 0$, $b_{n+3k-1} = 1$ и $b_{n+3k} = 1$. Тогда на лентах с номерами $(k-1)\tau + 1, (k-1)\tau + 2, \ldots, (k-1)\tau + \tau_{i_k,j}$ в обозреваемых ячейках расположены пустые символы. Машина на этих лентах заменяет пустые символы символами 1 и сдвигает их на одну ячейку вправо. Ленты с номерами $(k-1)\tau + \tau_{i_k,j} + 1, (k-1)\tau + \tau_{i_k,j} + 2, \ldots, k\tau$ остаются без изменения. Индекс $b_{n+3k}$ заменяется на $b'_{n+3k} = 1$, индекс $b_{n+3k-1}$ заменяется на $b'_{n+3k-1} = \tau_{i_k,j}$ и индекс $b_{n+3k-2}$ заменяется на $b'_{n+3k-2} = \tau_{i_k,j} (\mathrm{mod}\, \delta)$.

Предположим, что $\tau_{i_k,j} \geq 0$, $b_{n+3k-1} > 1$ и на ленте с номером $(k-1)\tau + b_{n+3k-1} - 1$ головка машины обозревает 1. Тогда если $b_{n+3k-1} - 1 + \tau_{i_k,j} \leq \tau$, то в текущих ячейках лент с номерами $(k-1)\tau + b_{n+3k-1}, (k-1)\tau + b_{n+3k-1} + 1, \ldots, (k-1)\tau + b_{n+3k-1} + \tau_{i_k,j}$ ставится символ 1 и каждая из этих лент сдвигается на одну ячейку вправо. Остальные ленты с номерами от $(k-1)\tau + 1$ до $k\tau$

остаются без изменения. При этом индекс $b_{n+3k}$ заменяется на $b'_{n+3k} = 1$, индекс $b_{n+3k-1}$ заменяется на $b'_{n+3k-1} = b_{n+3k-1} + \tau_{i_k,j}$ и индекс $b_{n+3k-2}$ заменяется на $b'_{n+3k-2} = b_{n+3k-2} + \tau_{i_k,j} (\mathrm{mod}\, \delta)$. Если $b_{n+3k-1} - 1 + \tau_{i_k,j} > \tau$, то в текущих ячейках лент с номерами $(k-1)\tau + b_{n+3k-1}, (k-1)\tau + b_{n+3k-1} + 1, \dots, k\tau$ и номерами $(k-1)\tau+1, (k-1)\tau+2, \dots, (k-2)\tau+b_{n+3k-1}-1+\tau_{i_k,j}$ ставится символ 1 и каждая из этих лент сдвигается на одну ячейку вправо. Остальные ленты с номерами от $(k-1)\tau+1$ до $k\tau$ оставляются без изменения. При этом индекс $b_{n+3k}$ заменяется на $b'_{n+3k} = 1$, индекс $b_{n+3k-1}$ заменяется на $b'_{n+3k-1} = b_{n+3k-1} - 1 + \tau_{i_k,j} - \tau$ и индекс $b_{n+3k-2}$ заменяется на $b'_{n+3k-2} = b_{n+3k-2} + \tau_{i_k,j} (\mathrm{mod}\, \delta)$.

Случай, когда $\tau_{i_k,1} < 0$, рассматривается аналогично.

Завершить построение набора команд программы машины $\mathscr{T}$ и провести проверку корректности работы этой программы можно стандартными рассуждениями по схеме доказательства теоремы 2 из п. 15.2 книги [18] (см. также разделы 7.2–7.4 обзора [2] и, например, работы [19–21]).

Теорема доказана.

## *ЛИТЕРАТУРА*


1. *Garey M. R., Johnson D. S.* Computers and intractibility. A guide to the theory of NP-completeness. San Francisco: W. H. Freeman and Co., 1979.
2. *Kharlampovich O. G., Sapir M. V.* Algorithmic problems in varieties // Internat. J. Algebra Comput. 1995. V. 5, N 4–5. P. 379–602.
3. *Сапир М. В.* Алгоритмические проблемы в многообразиях полугрупп // Изв. вузов. Математика. 1985. № 12. С. 71–74.
4. *Сапир М. В.* Алгоритмические проблемы в многообразиях полугрупп // Алгебра и логика. 1988. Т. 27, № 4. С. 440–463.
5. *Емеличев В. А.* Коммутативные полугруппы с одним определяющим соотношением // Уч. зап. Шуйск. пед. ин-та. 1958. Т. 6. С. 227–242.
6. *Мальцев А. И.* О гомоморфизмах на конечные группы // Уч. зап. Ивановск. пед. ин-та. 1958. Т. 18. С. 49–60.
7. *Тайцлин М. А.* Об элементарных теориях коммутативных полугрупп с сокращением // Алгебра и логика. 1966. Т. 5, № 1. С. 51–69.
8. *Тайцлин М. А.* Об элементарных теориях коммутативных полугрупп // Алгебра и логика. 1966. Т. 5, № 4. С. 55–89.
9. *Cardoza E. W.* Computational complexity of the word problem for commutative semigroups. MAC technical memorandum 67. MIT, 1975.
10. *Mayr E. W., Meyer A. R.* The complexity of the word problem for commutative semigroups and polynomial ideals // Adv. Math. 1982. V. 46, N 3. P. 305–329.
11. *Alonso J., Brady T., Cooper D., Ferlini V., Lustig M., Mihalik M., Shapiro M., Short H.* Notes on word hyperbolic groups // Proc. Conf. Group Theory from a Geometrical Viewpoint. Eds. E. Ghys, A. Haefliger and A. Verjovsky, Held in I.C.T.P., Trieste, March 1990. Singapore: World Scientific, 1991. P. 3–63.
12. *Holt D. F.* Word-hyperbolic groups have real-time word problem // Internat. J. Algebra Comput. 2000. V. 10, N 2. P. 221–227.
13. *Rabin M. O.* Real time computation // Israel J. Math. 1963. V. 1. P. 203–211.
14. *Rosenberg A.* Real-Time Definable Languages // J. Assoc. Comput. Mach. 1967. V. 14. P. 645–662.
15. *Тайцлин М. А.* Об алгоритмических проблемах для коммутативных полугрупп // Докл. АН СССР. 1968. № 9. С. 201–204.
16. *Ginsburg S., Spanier E. H.* Semigroups, Presburger formulas, and languages // Pacific J. Math. 1965. V. 16, N 2. P. 285–296.
17. *Лисовик Л. П.* Полулинейные множества и разрешимые проблемы // Мат. вопросы кибернетики. 1988. № 1. С. 201–222.
18. *Мальцев А. И.* Алгоритмы и рекурсивные функции. М.: Наука, 1965.
19. *Гуревич Ю. Ш.* Проблема равенства слов для некоторых классов полугрупп // Алгебра и логика. 1966. Т. 5, № 5. С. 25–35.

**20.** *Мурский В. Л.* Несколько примеров многообразий полугрупп // Мат. заметки. 1968. Т. 3, № 6. С. 663–670.

**21.** *Попов В. Ю.* Об эквациональных теориях классов конечных полугрупп // Алгебра и логика. 2001. Т. 40, № 1. С. 97–116.

*Статья поступила 19 ноября 2001 г.*

*Попов Владимир Юрьевич*
*Уральский гос. университет им. А. М. Горького,*
*математико-механический факультет, кафедра алгебры и дискретной математики,*
*пр. Ленина, 51, Екатеринбург 620083*
Vladimir.Popov@usu.ru