МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики
Кафедра дискретной математики и алгоритмики

Соболь Сергей Александрович

Временная сложность построения номера машины Тьюринга по её программе

Курсовая работа
студента 3 курса 3 группы

“Допустить к защите”
с предварительной оценкой _____
Руководитель работы
____

“__” ______________ 2012 г.

Руководитель
Мощенский Владимир Андреевич
доцент кафедры ДМА,
канд. физ.-мат. наук

Минск 2012

АННОТАЦИЯ

Рассматривается задача нахождения кода машины Тьюринга по её программе. Строится машина Тьюринга, решающая эту задачу, и оценивается временная сложность её вычисления.

АНАТАЦЫЯ

Разглядаецца задача знаходжання коду машыны Т'юрынга па яе праграме. Будуецца машына Т'юрынга, якая рашае гэту задачу, і ацэньваецца часавая складанасць яе вылічэння.

ANNOTATION

The problem of finding Turing machine code using its program is considered. A Turing machine that solves this problem is built and time complexity of its computation is studied.

РЕФЕРАТ

Курсовая работа, 44 с., 6 табл., 2 источника, 1 приложение.

Ключевые слова: МАШИНА ТЬЮРИНГА, ВРЕМЕННАЯ СЛОЖНОСТЬ, КОД, ПРОГРАММА.

Объект исследования — программы машин Тьюринга.

Цель работы — описание и реализация алгоритма построения кода машины Тьюринга по её программе, оценка временной сложности данного алгоритма.

Методы исследования — методы теории алгоритмов, теории сложности вычислений.

Результатом является реализация алгоритма построения кода машины в виде программы машины Тьюринга и оценка его временной сложности.

Областью применения является теория алгоритмов и формальных вычислительных систем.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Машина Тьюринга — математическое уточнение понятия алгоритма [1, с. 82], которое было предложено Аланом Тьюрингом в 1936 г. Это абстрактное вычислительное устройство позволило формализовать интуитивное понятие алгоритма, ввести строгое понятие временной сложности вычислений, доказать алгоритмическую неразрешимость некоторых проблем.

При исследовании некоторых задач теории алгоритмов необходимо подавать на вход машине Тьюринга другую машину Тьюринга. Чтобы это стало возможно, нужно уметь записывать программу одной машины Тьюринга M_1 на ленте другой машины Тьюринга M_2 в виде слова во внешнем алфавите машины M_2.

В данной работе предлагается кодировать программу любой машины Тьюринга с виде слова в алфавите {‘`0`’, ‘`1`’}, или, другими словами, в виде строки из нулей и единиц. Эта строка, в свою очередь, может интерпретироваться как запись натурального числа в двоичной системе счисления, т. е. как номер машины. Правила построения кода таковы, что в старшем разряде всегда стоит единица, поэтому различным кодам всегда соответствуют различные натуральные числа.

Такой подход оказывается удобным на практике, поскольку правила построения такого кода достаточно просты, кодирование и декодирование выполняется эффективно.

В данной работе анализируется алгоритм кодирования. Строится машина Тьюринга M, которая по представлению программы машины Тьюринга в виде строки в определённом формате записывает двоичный код этой машины. Оценивается временная сложность вычисления машины M.

Оказывается, что если на вход машине M подаётся программа некоторой машины с $(m+1)$ внутренним состоянием и мощностью внешнего алфавита $(k+1)$, то машине M потребуется $O(m^2k^2(m+k)^2)$ шагов для записи кода.

Коды машин Тьюринга применяются в доказательствах алгоритмической неразрешимости некоторых проблем, например проблемы самоприменимости. Коды машин Тьюринга также являются основой для создания универсальной машины Тьюринга, т. е. машины, которая сама исполняет программу другой машины. Универсальными машинами Тьюринга в некотором смысле являются все современные компьютеры.

1 Теоретическая основа

1.1 Общие сведения о машинах Тьюринга

Удобно представлять, что машина Тьюринга состоит из четырёх частей: ленты, считывающей головки, внутренней памяти и устройства управления.

Лента представляет собой полоску бумаги, разделённую на равные клетки.

λ	λ	λ	λ	λ	λ	λ	λ

В каждой клетке может быть записана только одна буква из некоторого конечного множества $A = \{a_0, a_1, \ldots, a_k\}$. Само множество A называется *внешним алфавитом* машины. Одна из букв (обычно a_0) называется *пустой*, её обозначают через λ, все другие буквы из A, кроме λ, называются *непустыми*. Как правило, $|A| \geqslant 2$.

Машина действует в дискретные моменты времени, которые нумеруются числами 1, 2, 3, … В каждый момент времени лента состоит из конечного числа клеток. По ленте передвигается считывающая головка, которая в любой момент времени находится над определённой клеткой ленты; в этом случае говорят, что головка читает букву, написанную в этой клетке. В следующий момент головка остаётся над той же клеткой (что обозначается через Н), или передвигается на одну клетку вправо (что обозначается через П), или передвигается на клетку влево (обозначается через Л). Когда в данный момент времени t считывающая головка находится над крайней клеткой ленты и передвигается в клетку, которой нет в момент t, тогда сразу же пристраивается новая пустая клетка, над которой будет располагаться головка в момент $(t + 1)$. Таким образом, лента потенциально не ограничена, поскольку могут пристраиваться новые пустые клетки на обоих её концах.

Машина обладает внутренней памятью, которая представляет собой некоторое конечное множество внутренних состояний машины $Q = \{q_0, q_1, \ldots, q_m\}$, причём $Q \cap A = \varnothing$. Элемент q_0 называется *заключительным внутренним состоянием*, а элемент q_1 — *начальным внутренним состоянием*. В каждый момент времени машина находится в соответствующем внутреннем состоянии. Движение головки зависит от читаемой буквы на ленте и внутреннего состояния машины.

Устройство управления при незаключительном внутреннем состоянии q_i $(i \neq 0)$ машины выполняет шаг, который полностью определяется этим внутренним состоянием q_i и читаемой буквой a_j. Устройство на каждом шаге:

1) заменяет читаемую букву a_j на другую букву a_s, причём возможно, что $s = j$;
2) переводит машину в другое внутреннее состояние q_r, которое может совпадать с q_i или с заключительным состоянием q_0;
3) передвигает головку на одну клетку вправо, влево или оставляет её на месте.

Говорят, что такой шаг выполняет команду машины, которая определяется выражением

$$«q_i a_j \to a_s d q_r», \tag{1}$$

где «$q_i a_j$» называется *левой частью* этой команды, а «$a_s d q_r$» — её *правой частью*. Подчеркнём, что в левой части каждой команды внутреннее состояние q_0 не встречается.

Таким образом, левая часть каждой команды состоит из величин в момент времени t, а правая часть — величин a_s и q_r в момент $(t + 1)$, а величина сдвига d есть движение головки в момент t, $d \in \{$Л, Н, П$\}$.

Всевозможных команд в алфавитах A и Q с попарно различными левыми частями конечное множество (мощности $|A| \cdot (|Q| - 1)$), и такая их совокупность называется *программой* машины.

Программу машины Тьюринга обычно записывают в виде таблицы. Примером является таблица 1. На пересечении строки q_i $(1 \leqslant i \leqslant m)$ и столбца a_i $(0 \leqslant j \leqslant k)$ пишется правая часть команды (1).

В данной работе мы будет также использовать запись программы машины Тьюринга в виде строки, содержащей все команды машины вида (1), выписанные в произвольном порядке:

$$«q_{i_1}a_{j_1} \to a_{s_1}d_1q_{r_1},\ q_{i_2}a_{j_2} \to a_{s_2}d_2q_{r_2},\ \ldots,\ q_{i_w}a_{j_w} \to a_{s_w}d_wq_{r_w}» \tag{2}$$

Отметим следующий факт. Для того чтобы строка вида (2) представляла собой запись программы некоторой машины Тьюринга с $A = \{a_0, a_1, \ldots, a_k\}$ и $Q = \{q_0, q_1, \ldots, q_m\}$, необходимо и достаточно, чтобы

- для любого индекса n, $1 \leqslant n \leqslant w$, выполняются неравенства $1 \leqslant i_n \leqslant m$, $0 \leqslant j_n \leqslant k$, $0 \leqslant s_n \leqslant k$, $0 \leqslant r_n \leqslant m$, а также $d_n \in \{\text{Л, Н, П}\}$;
- для любой упорядоченной пары (i, j) такой, что $1 \leqslant i \leqslant m$ и $0 \leqslant j \leqslant k$ существовал, и притом единственный, индекс n такой, что $(i_n, j_n) = (i, j)$.

Как следствие отметим, что в записи (2) $w = |A| \cdot (|Q| - 1) = (k + 1) \cdot m$.

Шаг есть выполнение одной команды. *Вычислением* (или *работой*) машины Тьюринга называется последовательность шагов (одного за другим) без пропусков начиная с первого шага. Совокупность трёх величин: слова на ленте, положения считывающей головки и внутреннего состояния машины — называется *конфигурацией* машины в данный момент. Можно показать, что вычисление машины полностью определяется конфигурацией в первый момент. Будем полагать, что в начальной конфигурации внутреннее состояние всегда есть q_1.

Когда в работе машины в некоторый момент времени t выполняется команда, в правой части которой имеется внутреннее состояние q_0, тогда в такой момент работа машины заканчивается. В этом случае говорят, что машина *применима* к слову на ленте в первый момент, и результатом её работы считается слово на ленте в последний момент. Если же в данной работе машины выполняются команды, в правых частях которых внутреннее состояние q_0 не встречается, то говорят, что машина *не применима* к слову на ленте в первый момент, и результат её работы не определяется.

Арифметическая функция $T(n)$ называется *временной сложностью* вычисления машины Тьюринга, если над каждым исходным словом длины n машина выполнит не более чем $T(n)$ шагов [1, с. 113].

1.2 Постановка задачи

Известно [1, с. 99], что программу машины Тьюринга можно задать в виде слова в фиксированном алфавите. Условимся, как и ранее, внешний алфавит каждой машины выписывать из букв $a_0 = \lambda, a_1, a_2, \ldots$, а её внутреннюю память — из букв $q_0, q_1, q_2, \ldots$ Теперь сопоставим этим буквам и буквам Л, Н, П следующие двоичные слова:

$$\begin{gathered} \text{Л} \mapsto 10, \quad \text{Н} \mapsto 100, \quad \text{П} \mapsto 1000, \\ a_i \mapsto 10^{(2i+4)} \quad (i \geqslant 0), \quad q_j \mapsto 10^{(2j+5)} \quad (j \geqslant 0). \end{gathered} \tag{3}$$

Если b есть одна из этих букв, то через $k(b)$ будем обозначать слово, которое ей сопоставлено; например, $k(a_0) = 10^{(4)} = 10000$, $k(q_1) = 10^{(7)} = 10000000$.

Команде вида (1) сопоставим слово

$$K(q_ia_j) = k(q_i)k(a_j)k(a_s)k(d)k(q_t), \tag{4}$$

которое назовём *кодом* этой команды.

Машине $\hat{M}$ с внешним алфавитом $A = \{a_0, a_1, \ldots, a_k\}$ и множеством внутренних состояний $Q = \{q_0, q_1, \ldots, q_m\}$ сопоставим слово

$$\begin{aligned} N(\hat{M}) = K(q_1a_0)K(q_1a_1)\ldots K(q_1a_k)K(q_2a_0)K(q_2a_1)\ldots K(q_2a_k)\ldots \\ \ldots K(q_{m-1}a_0)K(q_{m-1}a_1)\ldots K(q_{m-1}a_k)K(q_ma_0)K(q_ma_1)\ldots K(q_ma_k), \end{aligned} \tag{5}$$

которое назовём её *кодом.* Код машины получается из кодов всех её команд, выписываемых в указанном порядке.

Рассмотрим для примера участника «соревнования по трудолюбию» Радо с двумя незаключительными внутренними состояниями — машину B_2 [2, с. 111]. Программа этой машины Тьюринга приведена в таблице 1.

Таблица 1 – программа машины B_2

	`λ`	`1`
q_1	`1`Пq_2	`1`Лq_2
q_2	`1`Лq_1	`1`Пq_0

Полагая $a_0 = \lambda$, $a_1 =$ ‘`1`’, программу машины B_2 в виде строки (2) можно записать следующим образом:

$$«q_1a_1 \to a_1Лq_2,\ q_2a_0 \to a_1Лq_1,\ q_2a_1 \to a_1Пq_0,\ q_1a_0 \to a_1Пq_2» \qquad (6)$$

Отметим ещё раз, что в этой записи порядок следования команд произволен.

Согласно правилам (4), код машины будет таким:

$$10^{(7)}10^{(4)}10^{(6)}100010^{(9)}10^{(7)}10^{(6)}10^{(6)}1010^{(9)}10^{(9)}10^{(4)}10^{(6)}1010^{(8)}10^{(9)}10^{(6)}10^{(6)}100010^{(5)}, \qquad (7)$$

или же в полной форме (строка длиной в 121 символ)

$$100000001000010000001000100000000010000000\ldots$$
$$\ldots10000001000000101000000000100000000010000\ldots$$
$$\ldots1000000101000000001000000000100000010000001000100000$$

Здесь важно, что порядок следования отдельных закодированных команд фиксирован: если рассматривать все команды в виде (1), то в коде машины команды следуют в порядке возрастания i, при равенстве i — в порядке возрастания j.

Интуитивно понятно, что с помощью указанных правил (4) по программе машины Тьюринга её код строится эффективно. В данной работе процесс построения кода машины по её программе, заданной в виде строки (2), формализуется вновь при помощи машин Тьюринга. Строится машина M, на вход которой подаётся строка вида (2), описывающая программу некоторой машины $\hat{M}$, а результатом является код $N(\hat{M})$ вида (5). Более того, машина M оказывается применимой к строкам, описывающим программу некоторой машины Тьюринга, и только к ним. На входных данных, которые не являются записью программы никакой машины Тьюринга, машина M работает бесконечно долго, никогда не достигая заключительного состояния q_0. В работе также раскрывается смысл фразы «код строится эффективно»: оценивается временная сложность $T(n)$ вычисления машины M и показывается, что функция $T(n)$ ограничена сверху некоторым полиномом от n — размера входа.

1.3 Соглашения о входных и выходных данных

Для задания натуральных чисел на ленте машины Тьюринга мы используем так называемую унарную систему счисления [1, с. 86]. В этой системе натуральное число n задаётся словом «$\mathtt{1}^{n+1}$», или «`111`…`1`» (подряд выписана $(n{+}1)$ единица). Под множеством натуральных чисел здесь и далее понимается множество $\mathbb{N} = \{0, 1, 2, \ldots\}$.

`1`	`λ`	`1`	`1`	`λ`	`1`	`1`	`1`

В данном примере на ленте записаны числа 0, 1 и 2, разделённые пустыми клетками.

Программу машины Тьюринга в виде строки (2) будем задавать на ленте естественным способом — как слово в алфавите $A_1 = \{$‘`q`’, ‘`a`’, ‘`→`’, ‘`1`’, ‘`Л`’, ‘`Н`’, ‘`П`’, ‘`,`’$\}$, при этом индексы i_n, j_n, s_n, r_n будем писать в унарной системе счисления.

Например, программа (6) заишется на ленте так.

q	1	1	a	1	1	→	a	1	1	Л	q	1	1	1	,	q	1	1	1	a	1	→	a	1	1	Л	q	1	1	,	q	1	1	1	

	a	1	1	→	a	1	1	П	q	1	,	q	1	1	a	1	→	a	1	1	П	q	1	1	1

Отметим, что в этой записи команды разделены запятыми, а пробелы и пустые клетки в записи не встречаются.

Результатом работы машины M будет код, созданный по схеме (5). Этот код из нулей и единиц можно очевидным образом представить на ленте машины как слово в алфавите $A_2 = \{$‘`0`’, ‘`1`’$\}$.

При построении машины Тьюринга M нам окажется удобным ввести некоторые служебные буквы. Это позволит облегчить процесс программирования и, вероятно, сделать программу машины M более понятной человеку. Цель минимизировать мощность алфавита не ставится. Введём в качестве служебных буквы алфавита $A_3 = \{$‘`*`’, ‘`k`’, ‘`m`’$\}$.

Таким образом, внешним алфавитом машины M будет множество

$$A = \{\lambda\} \cup A_1 \cup A_2 \cup A_3 = \{\lambda, \text{‘q’}, \text{‘a’}, \text{‘→’}, \text{‘1’}, \text{‘Л’}, \text{‘Н’}, \text{‘П’}, \text{‘,’}, \text{‘0’}, \text{‘*’}, \text{‘k’}, \text{‘m’}\}. \qquad (8)$$

Легко подсчитать, что $|A| = 13$.

2 Построение вспомогательных машин Тьюринга

При написании программы машины M нам потребуется несколько раз выполнять одинаковые последовательности действий (например, чтобы записать код одной команды вида (1), нужно записать отдельно коды q_i и q_r, а это делается по одному и тому же алгоритму). С целью упрощения программирования выделим несколько подзадач, построим машины Тьюринга, их решающие, а затем включим их как составные части в программу машины M.

Чтобы сократить размеры программ вспомогательных машин, будем указывать переходы явно только для некоторых букв из алфавита (8), остальные же буквы объединим в группу, обозначив её знаком ‘`?`’. Команду вида «если головка считывает ‘`?`’ в состоянии q_i, записать на ленту ‘`?`’, перейти в q_j и выполнить перемещение головки d» следует понимать так: «если головка машины в состоянии q_i считывает некоторую букву из A, для которой в программе нет явно прописанных переходов, записать на ленту эту же букву, перейти в q_j и переместиться на d».

Также договоримся в таблицах с программами машин Тьюринга не записывать правые части некоторых команд, которые гарантированно не встретятся в работе машины, и соответствующие ячейки оставлять пустыми.

2.1 Машина $M_{k(q)}$

Посторим машину, которая строит код $k(q)$, действуя согласно соответствующему правилу из списка (4):

$$q_j \mapsto 10^{(2j+5)} \quad (j \geqslant 0), \tag{9}$$

или, как это выглядит на ленте машины,

$$\texttt{q}\underbrace{\texttt{11...1}}_{j+1} \longmapsto \texttt{1}\underbrace{\texttt{000000000...00}}_{2j+5}.$$

Пусть изначально в некотором месте ленты в клетке записана буква ‘`q`’, над ней находится считывающая головка машины. В следующих $(j+1)$ клетках стоят единицы (‘`1`’). Это запись числа j в унарной системе счисления. Далее следует какая-либо отличная от ‘`1`’ буква (возможно, пустая буква λ). Правее этой буквы, равно как и левее ‘`q`’, могут располагаться любые буквы, включая ‘`q`’, ‘`1`’, λ, кроме специального символа ‘`*`’. Важно лишь одно требование: ближайшая справа пустая клетка должна быть началом отрезка пустых клеток достаточной для записи кода $k(q_j)$ длины. Именно в эти пустые клетки машина и запишет код, который представляет собой одну букву ‘`1`’ и $(2j+5)$ букв ‘`0`’. Исходные символы (‘`q`’ и ‘`1`’) будут заменены разделителями ‘`,`’. Головка по завершении исполнения программы остановится сразу за последней ‘`,`’. Например, конфигурацию

q	1	,	q	1	1	1	→	a	1	П	q	1	λ	λ	λ	λ	λ	λ	λ	λ	λ	λ
			$\triangle\ q_1$																			

машина $M_{k(q)}$ преобразует в

q	1	,	,	,	,	,	→	a	1	П	q	1	1	0	0	0	0	0	0	0	0	0
							$\triangle\ q_0$															

Итак, в подаваемой на вход машине строке ‘`q11...1`’ содержится $(j+1)$ единиц, между последней единицей последовательности и первой пустой клеткой располагается l других букв (обозначим их через ‘`?`’). Для наглядности приведём пример, где $j = 3$ и $l = 3$. В начальный момент времени машина находится в состоянии q_1, считывающая головка расположена над буквой ‘`q`’.

q	1	1	1	1	?	?	?
△ q_1							

Машина $M_{k(q)}$ начинает работу. Головка сдвигается вправо, машина переходит в состояние q_2.

q	1	1	1	1	?	?	?
	△ q_2						

Теперь головка считывает первую единицу последовательности. Далее головка сдвигается вправо $(j+1)$ раз, пока считывается единица.

...

q	1	1	1	1	?	?	?
					△ q_2		

Как только на ленте встречается буква, отличная от ‘`1`’, машина переходит в состояние q_3, сдвигая головку влево и возвращая её в позицию над последней единицей последовательности.

q	1	1	1	1	?	?	?
				△ q_3			

Вместо единицы записывается символ ‘`*`’, машина переходит с состояние q_4.

q	1	1	1	*	?	?	?
					△ q_4		

В этом состоянии головка движется вправо, пока не встретит пустую клетку (букву λ). Нетрудно видеть, что головка совершит ровно l перемещений.

...

q	1	1	1	*	?	?	?	λ
								△ q_4

В соответствии с соглашением о входных данных машины $M_{k(q)}$, правее найденной клетки есть ещё некоторое число пустых (возможно, что это край ленты, тогда правее имеется потенциально бесконечное число пустых клеток). Как только первая пустая клетка найдена, машина пишет в неё ‘`1`’, сдвигает головку вправо и переходит в состояние q_5.

q	1	1	1	*	?	?	?	1	λ
									△ q_5

Затем машина записывает один за одним пять символов ‘`0`’, проходя последовательно состояния q_5 — q_9. Когда пятый нуль оказывается записанным, машина переходит в состояние q_{10}.

...

q	1	1	1	*	?	?	?	1	0	0	0	0	0
													△ q_{10}

Сейчас машина начинает осуществление цикла. Конфигурацию в общем случае можно описать так:

q	1	1	*	*	?	?	?	1	0	0	0	0	0	0	0
															△ q_{10}

Пусть изначально после ‘q’ была записана $(j+1)$ единица. Сейчас $(k+1)$ из них заменена на ‘*’, а из букв кода уже записаны ‘1’ и $(2k+5)$ букв ‘0’. Головка считывает последнюю записанную букву ‘0’. Здесь показан пример, когда $j=3$, $l=3$ и $k=1$.

Машина действует следующим образом. Оставаясь в состоянии q_{10}, она двигает головку влево, пока не будет найден первый символ ‘*’. Нетрудно посчитать, что будет сделано $(2k+5+l)$ перемещений.

q	1	1	*	*	?	?	?	1	0	0	0	0	0	0	0
														△ q_{10}	

…

q	1	1	*	*	?	?	?	1	0	0	0	0	0	0	0
				△ q_{10}											

Далее машина переходит в состояние q_{11} и продолжает двигать головку влево, пока не закончится блок из подряд идущих символов ‘*’. На этом этапе будет сделано $(k+1)$ перемещений головки.

q	1	1	*	*	?	?	?	1	0	0	0	0	0	0	0
			△ q_{11}												

q	1	1	*	*	?	?	?	1	0	0	0	0	0	0	0
		△ q_{11}													

1. Пусть головка остановилась на клетке с буквой ‘1’, как показано на примере. Машина пишет на ленту ‘*’, переходит в q_{12} и перемещает головку вправо до тех пор, пока не будет найдена первая пустая клетка (λ). Машине для этого потребуется выполнить ровно $(k+1+l+2k+6)$ шагов.

q	1	*	*	*	?	?	?	1	0	0	0	0	0	0	0
			△ q_{12}												

…

q	1	*	*	*	?	?	?	1	0	0	0	0	0	0	0	λ
																△ q_{12}

В найденную пустую клетку пишется ‘0’, головка двигается вправо, пишется ещё один ‘0’ в состоянии q_{13} (всего два шага), а потом машина переходит вновь в состояние q_{10}, т. е. начинается новая итерация цикла.

q	1	*	*	*	?	?	?	1	0	0	0	0	0	0	0	0	λ
																	△ q_{13}

q	1	*	*	*	?	?	?	1	0	0	0	0	0	0	0	0	0
																	△ q_{10}

2. Если же в состоянии q_{11} головка остановится на букве ‘q’, а не ‘1’, это будет означать, что запись кода завершена.

q	*	*	*	*	?	?	?	1	0	0	0	0	0	0	0	0	0	0	0
△ q_{11}																			

Осталось заменить букву ‘`q`’ и все последующие ‘`*`’ на служебные буквы ‘`,`’, для чего в машине и предусмотрено состояние q_{14}. На данной завершающей фазе машина сделает $(1 + j + 1)$ шаг. Затем будет сделан последний шаг — переход в состояние q_0.

,	*	*	*	*	?	?	?	1	0	0	0	0	0	0	0	0	0	0	0
	△ q_{14}																		

...

,	,	,	,	,	?	?	?	1	0	0	0	0	0	0	0	0	0	0	0
					△ q_0														

Приведем программу машины $M_{k(q)}$ в таблице 2.

Таблица 2 – программа машины $M_{k(q)}$

	λ	*	1	q	?
q_1				qПq_2	
q_2	λЛq_3		1Пq_2	qЛq_3	?Лq_3
q_3			*Пq_4		
q_4	1Пq_5		1Пq_4	qПq_4	?Пq_4
q_5	0Пq_6				
q_6	0Пq_7				
q_7	0Пq_8				
q_8	0Пq_9				
q_9	0Нq_{10}				
q_{10}		*Лq_{11}	1Лq_{10}	qЛq_{10}	?Лq_{10}
q_{11}		*Лq_{11}	*Пq_{12}	,Пq_{14}	
q_{12}	0Пq_{13}	*Пq_{12}	1Пq_{12}	qПq_{12}	?Пq_{12}
q_{13}	0Нq_{10}				
q_{14}		,Пq_{14}	1Нq_0	qНq_0	?Нq_0

Оценим временную сложность вычисления машины $M_{k(q)}$. Учтём, что цикл q_{10} — q_{11} — q_{12} — q_{10} выполнится ровно j раз.

$$\begin{aligned} T(j,l) = 1 + (j+1+1) + 1 + l + 6 + \\ + \sum_{k=0}^{j-1}[(2k+6+l) + (k+1) + (k+1+l+2k+6+1) + 2] + \\ + (2j+6+l) + (j+1) + (1+j+1) + 1, \end{aligned} \quad (10)$$

или, после упрощения с использованием формулы суммы арифметической прогрессии,

$$T(j,l) = 3j^2 + 19j + 20 + 2l(j+1). \quad (11)$$

2.2 Машина $M_{k(a)}$

По аналогии нетрудно построить машину, которая строит код $k(a)$:

$$a_i \mapsto 10^{(2i+4)} \quad (i \geqslant 0), \tag{12}$$

или

$$\texttt{a}\underbrace{\texttt{11...1}}_{i+1} \longmapsto \texttt{1}\underbrace{\texttt{00000000...00}}_{2i+4}.$$

Алгоритм работы машины $M_{k(a)}$ полностью аналогичен алгоритму работы $M_{k(q)}$. Программу машины $M_{k(a)}$ можно получить из программы $M_{k(q)}$ путём замены ‘`q`’ на ‘`a`’ и удаления одного состояния.

Приведём программу в виде таблицы 3 без дополнительных пояснений.

Таблица 3 – программа машины $M_{k(a)}$

	λ	*	1	a	?
q_1				aПq_2	
q_2	λЛq_3		1Пq_2	aЛq_3	?Лq_3
q_3			*Пq_4		
q_4	1Пq_5		1Пq_4	aПq_4	?Пq_4
q_5	0Пq_6				
q_6	0Пq_7				
q_7	0Пq_8				
q_8	0Нq_9				
q_9		*Лq_{10}	1Лq_9	aЛq_9	?Лq_9
q_{10}		*Лq_{10}	*Пq_{11}	,Пq_{13}	
q_{11}	0Пq_{12}	*Пq_{11}	1Пq_{11}	aПq_{11}	?Пq_{11}
q_{12}	0Нq_9				
q_{13}		,Пq_{13}	1Нq_0	aНq_0	?Нq_0

Временная сложность вычисления этой машины вычисляется точно так же:

$$\begin{multline} T(i,l) = 1 + (i+1+1) + 1 + l + 5 + \\ + \sum_{k=0}^{i-1}[(2k+5+l) + (k+1) + (k+1+l+2k+5+1) + 2] + \\ + (2i+5+l) + (i+1) + (1+i+1) + 1, \tag{13} \end{multline}$$

после упрощения приходим к формуле

$$T(i,l) = 3i^2 + 17i + 18 + 2l(i+1). \tag{14}$$

2.3 Машина $M_{k(d)}$

Построим машину $M_{k(d)}$, которая строит двоичный код букв ‘`Л`’, ‘`Н`’, ‘`П`’:

$$\text{Л} \mapsto 10, \quad \text{Н} \mapsto 100, \quad \text{П} \mapsto 1000. \tag{15}$$

Будем полагать, что в начальный момент времени считывающая головка располагается над буквой ‘`Л`’, ‘`Н`’ или ‘`П`’. Машина заменяет эту букву на ‘`*`’, перемещает головку на ближайшую справа пустую клетку, записывает соответствующий код (в предположении, что справа от найденной первой пустых клеток достаточно), потом возвращается влево к ‘`*`’, заменяет ‘`*`’ на разделитель ‘`,`’ и сдвигает головку на клетку вправо. Предполагается, что изначально на ленте буква ‘`*`’ не встречается. Например, конфигурацию

a	1	1	Н	q	1	1	1	λ	λ	λ
			$\triangle$ q_1							

машина Тьюринга $M_{k(d)}$ переведёт в

a	1	1	,	q	1	1	1	1	0	0
				$\triangle$ q_1						

Рассмотрим подробнее алгоритм работы машины на таком примере:

Л	?	?	?
$\triangle$ q_1			

Считывается буква ‘`Л`’, машина записывает ‘`*`’, меняет внутреннее состояние на q_2 и движется вправо в поисках пустой клетки.

*	?	?	?
	$\triangle$ q_2		

...

*	?	?	?	λ
				$\triangle$ q_2

Как только пустая клетка обнаружена, машина пишет туда ‘`1`’, сдвигает головку правее, переходя в состояние q_3.

*	?	?	?	1	λ
					$\triangle$ q_3

Теперь записывается ‘`0`’, и машина начинает движение влево. Этот процесс определяется внутренним состоянием q_{11} — машина не изменяет буквы на ленте, двигая считывающую головку влево, пока не будет найдена специальная буква ‘`*`’ (учитывая изложенные выше замечания, можно утверждать, что буква ‘`*`’ единственна на ленте и была записана на первом шаге вычисления машины $M_{k(d)}$).

*	?	?	?	1	0
				$\triangle$ q_{11}	

...

*	?	?	?	1	0
$\triangle$ q_{11}					

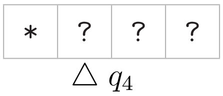
Далее ‘`*`’ заменяется на ‘`,`’, машина сдвигается влево и останавливается.

,	?	?	?	1	0
	$\triangle$ q_0				

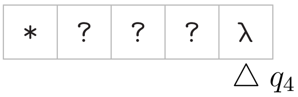
Теперь рассмотрим пример обработки буквы ‘`Н`’.

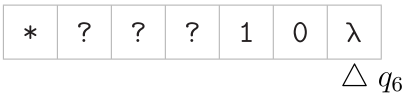
Н	?	?	?
$\triangle$ q_1			

Из q_1 машина переходит в состояние q_4, которое эквивалентно состоянию q_2 за исключением того, что при обнаружении пустой клетки переход будет осуществлён в состояние q_5, после которого последует переход в q_6.

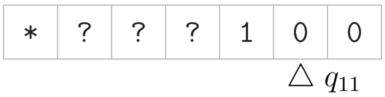


…





Когда машина в состоянии q_6 запишет второй нуль, она в точности так, как описано выше, начнёт движение влево в состоянии q_{11}.



Все дальнейшие действия машины уже описаны выше.

С целью обработки третьей буквы, ‘П’, введены состояния q_7 (движение вправо и запись ‘1’), q_8 (запись ‘0’), q_9 (запись ‘0’), q_{10} (запись ‘0’). Затем вновь машина переходит в состояние q_{11}.

В таблице 4 приводится текст программы.

Таблица 4 – программа машины $M_{k(d)}$

	λ	∗	Л	Н	П	?
q_1			∗Пq_2	∗Пq_4	∗Пq_7	
q_2	1Пq_3		ЛПq_2	НПq_2	ППq_2	?Пq_2
q_3	0Лq_{11}					
q_4	1Пq_5		ЛПq_4	НПq_4	ППq_4	?Пq_4
q_5	0Пq_6					
q_6	0Лq_{11}					
q_7	1Пq_8		ЛПq_7	НПq_7	ППq_7	?Пq_7
q_8	0Пq_9					
q_9	0Пq_{10}					
q_{10}	0Лq_{11}					
q_{11}		,Пq_0	ЛЛq_{11}	НЛq_{11}	ПЛq_{11}	?Лq_{11}

Оценим, сколько шагов сделает машина $M_{k(d)}$. Пусть между буквой ‘Л’, ‘Н’, ‘П’ и первой справа пустой клеткой расположены l других букв, $l \geqslant 0$. Тогда в худшем случае, а именно когда обрабатывается буква ‘П’, машина выполнит

$$T(l) = l + 1 + 3 + 3 + 1 + l + 1 = 2l + 9 \tag{16}$$

шагов.

2.4 Машина $M_{\text{cmp}(y)}$

Построим машину, осуществляющую сравнение двух чисел, точнее, нас будет интересовать только то, равны два числа или не равны.

Пусть в некотором месте ленты записано натуральное число x в унарной системе счисления. Договоримся, что в начале вычисления машины $M_{\text{cmp}(y)}$ её считывающая головка расположена над клеткой, предшествующей клетке с первой единицей числа x. Что именно записано в данной клетке, не важно.

Здесь и далее будем писать букву ‘`y`’ вместо некоторой буквы из алфавита (8), которая встречается на ленте машины $M_{\text{cmp}(y)}$ только один раз. Разумеется, предполагается, что ‘`y`’ отлична от λ и ‘`1`’. Далее при построении машины M мы будем ссылаться на машины $M_{\text{cmp}(k)}$ и $M_{\text{cmp}(m)}$, в которых полагается соответственно ‘`y`’ = ‘`k`’ и ‘`y`’ = ‘`m`’.

Пусть буква ‘`y`’ расположена правее числа x на ленте, а следом за ‘`y`’ идёт непустой блок из единиц, который представляет собой унарную запись некоторого натурального числа y. В примере

λ	1	1	1	1	λ	λ	y	1	1	1	1	1	λ

△ q_1

$x = 3$ (состоит из четырёх единиц) и $y = 4$ (состоит из пяти единиц). Последнюю единицу числа x и букву ‘`y`’ разделяют l ($l \geqslant 0$) произвольных букв (в примере $l = 2$).

В результате работы машины $M_{\text{cmp}(y)}$ головка вернётся в своё первоначальное положение и будет считывать с ленты ‘`0`’, если $x \neq y$, и ‘`1`’, если $x = y$.

Рассмотрим детально алгоритм, по которому вычисляет $M_{\text{cmp}(y)}$. В самом начале работы машины (в состоянии q_1) головка просто смещается на клетку правее и оказывается над первой единицей числа x. Состояние при этом меняется на q_2.

λ	1	1	1	1	λ	λ	y	1	1	1	1	1	λ

△ q_2

Рассмотрим общий случай: для k ($k \geqslant 0$) первых (считая слева направо) разрядов чисел x и y сравнение уже выполнено, и так оказалось, что все эти разряды у обоих чисел оказались единицами и были заменены на ‘`*`’. Это, в свою очередь, значит, что $x \geqslant (k-1)$ и $y \geqslant (k-1)$. Текущее внутреннее состояние — q_2, считывающая головка расположена над клеткой, следующей за последней клеткой с буквой ‘`*`’ в числе x (или над первой единицей числа x, если $k = 0$). Возможны следующие варианты.

1. Головка считывает с ленты ‘`1`’. Приведём пример конфигурации при $k = 2$.

λ	*	*	1	1	λ	λ	y	*	*	1	1	1	λ

△ q_2

Машина заменяет ‘`1`’ на ‘`*`’, перемещает головку вправо и переходит в состояние q_3.

λ	*	*	*	1	λ	λ	y	*	*	1	1	1	λ

△ q_3

Пока не обнаружена буква ‘`y`’, машина двигает головку вправо, при этом данные, записанные на ленте, не изменяются.

λ	*	*	*	1	λ	λ	y	*	*	1	1	1	λ

△ q_3

…

λ	*	*	*	1	λ	λ	y	*	*	1	1	1	λ
							△ q_3						

Всего в состоянии q_3 машина сделает $(x - k + l)$ шагов.

Как только буква ‘y’ найдена, машина смещает головку вправо и меняет состояние на q_4.

λ	*	*	*	1	λ	λ	y	*	*	1	1	1	λ
								△ q_4					

В состоянии q_4 движение в правую сторону продолжается, пока машина считывает ‘*’. Будет сделано ровно k шагов.

λ	*	*	*	1	λ	λ	y	*	*	1	1	1	λ
										△ q_4			

Итак, пусть машина остановилась: встретилась буква, отличная от ‘*’. Здесь вновь возможны варианты.

1.1. В состоянии q_4 головка считывает ‘1’, как это изображено на нашем примере. Тогда машина записывает на ленту ‘*’, смещается влево и переходит в q_5.

λ	*	*	*	1	λ	λ	y	*	*	*	1	1	λ
									△ q_5				

В q_5 машина двигает головку влево, пока не будет прочитана буква ‘y’. Потребуется k перемещений.

...

λ	*	*	*	1	λ	λ	y	*	*	*	1	1	λ
							△ q_5						

Далее состояние сменяется на q_6, движение влево продолжается, пока не будет обнаружена буква ‘*’. Буквы на ленте не меняются, число шагов оказывается равным $(l + x - k)$.

λ	*	*	*	1	λ	λ	y	*	*	*	1	1	λ
						△ q_6							

...

λ	*	*	*	1	λ	λ	y	*	*	*	1	1	λ
			△ q_6										

Теперь головка перемещается на клетку вправо, состояние сменяется на q_2.

λ	*	*	*	1	λ	λ	y	*	*	*	1	1	λ
				△ q_2									

Ситуация полностью аналогична той, которая только что была рассмотрена, только теперь уже $(k + 1)$ единиц в каждом числе заменены на ‘*’. Машина продолжает работу.

1.2. В состоянии q_4 головка машины считывает символ, отличный от ‘1’ и ‘*’. Значит, в числе y не нашлось единицы, соответствующей единице числа x, которая заменена была на ‘*’, когда машина находилась в состоянии q_2. Следовательно, в записи числа y единиц строго меньше, чем в записи числа x. Тогда $x > y$, можно записывать ответ. Для этого случая рассмотрим такой пример:

λ	*	*	*	1	λ	λ	y	*	*	λ	λ	λ	λ
										△ q_4			

Машина смещает головку влево и переходит в состояние q_9.

λ	*	*	*	1	λ	λ	y	*	*	λ	λ	λ	λ
									△ q_9				

Теперь все ‘`*`’ (их ровно k) заменяются обратно на ‘`1`’, чтобы придать ленте первоначальный вид.

…

λ	*	*	*	1	λ	λ	y	1	1	λ	λ	λ	λ
							△ q_9						

Как только обнаруживается ‘`y`’, машина сменяет состояние на q_{10}, движение влево продолжается, но буквы на ленте не меняются. Машина ищет ‘`*`’, делая при этом $(l + x - k)$ шагов.

λ	*	*	*	1	λ	λ	y	1	1	λ	λ	λ	λ
						△ q_{10}							

λ	*	*	*	1	λ	λ	y	1	1	λ	λ	λ	λ
			△ q_{10}										

Когда буква ‘`*`’ найдена, она заменяется на ‘`1`’, машина продолжает двигать головку влево, перейдя в состояние q_{11}, записывая ‘`1`’ вместо ‘`*`’. Т. к. число ‘`*`’ равно $(k + 1)$, то машина сделает $(k + 1)$ шаг.

λ	*	*	1	1	λ	λ	y	1	1	λ	λ	λ	λ
		△ q_{11}											

λ	1	1	1	1	λ	λ	y	1	1	λ	λ	λ	λ
△ q_{11}													

Наконец, когда последовательность ‘`*`’ заканчивается, машина записывает ответ — букву ‘`0`’ — и останавливается.

0	1	1	1	1	λ	λ	y	1	1	λ	λ	λ	λ
△ q_0													

2. Вернёмся к состоянию q_2. Пусть головка считывает символ, отличный от ‘`1`’. Значит, в записи числа x было не больше единиц, чем в записи числа y, т. е. $x \leqslant y$. Чтобы окончательно выяснить вопрос о соотношении между x и y, нужно проверить, остались ли в числе y единицы, которые ещё не заменены на ‘`*`’. Если да, то $x < y$, иначе в обоих числах x и y ровно по k единиц, поэтому $x = y$. Будем для наглядности рассматривать такой пример.

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
				△ q_2									

Если головка считывает букву ‘`y`’, машина переходит сразу в состояние q_8, двигая головку вправо. Иначе, если машина считывает какую-либо букву из $A \setminus \{$‘`1`’, ‘`y`’$\}$, она, перезаписывая её снова, переходит в состояние q_7.

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
					△ q_7								

Затем, пока машина не встретит ‘y’, она двигает головку вправо, не меняя состояния и символов. Потребуется $(l-1)$ шаг.

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
						△ q_7							

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
							△ q_7						

Как только буква ‘y’ найдена, состояние сменяется на q_8, головка продвигается вправо на одну клетку.

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
								△ q_8					

Нетрудно понять, что сейчас головка располагается над первой буквой ‘*’ в числе y. Она двигается вправо, пока с ленты считывается ‘*’, т. е. k раз.

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
									△ q_8				

…

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
											△ q_8		

Пусть в состоянии q_8 обнаружена буква, отличная от ‘*’. Возможны два случая.

2.1. Эта буква — ‘1’. Именно такой вариант изображён выше. Значит, в числе y строго больше единиц, чем в x, поэтому $x < y$. Необходимо вернуть ленту в изначальное состояние и записать ответ.

Машина, оставляя единицу, смещает головку влево и переходит в состояние q_9.

λ	*	*	*	λ	λ	λ	y	*	*	*	1	1	λ
										△ q_9			

Дальнейшие действия машины уже описаны выше. Результат:

0	1	1	1	λ	λ	λ	y	1	1	1	1	1	λ
△ q_0													

2.2. В состоянии q_8 машина считывает букву, отличную от ‘1’ и ‘*’. Следовательно, все единицы числа y уже заменены на ‘*’, как и единицы числа x. Вывод: $x = y$. Рассмотрим такой пример.

λ	*	*	*	λ	λ	λ	y	*	*	*	λ	λ	λ
											△ q_8		

В этом случае машина не меняет символ на ленте, сдвигает головку влево и переходит в состояние q_{12}.

λ	*	*	*	λ	λ	λ	y	*	*	*	λ	λ	λ
										△ q_{12}			

Далее машина заменяет все буквы ‘*’ на ‘1’, двигаясь справа налево.

$\dots$

λ	*	*	*	λ	λ	λ	y	1	1	1	λ	λ	λ
							△ q_{12}						

Когда встречается буква ‘y’, состояние сменяется на q_{13}. Далее машина, двигая головку в том же направлении, оставляет буквы на ленте без изменения.

λ	*	*	*	λ	λ	λ	y	1	1	1	λ	λ	λ
						△ q_{13}							

$\dots$

λ	*	*	*	λ	λ	λ	y	1	1	1	λ	λ	λ
			△ q_{13}										

Затем блок из подряд идущих символов ‘*’ заменяется блоком из ‘1’ (состояние q_{14}).

λ	*	*	1	λ	λ	λ	y	1	1	1	λ	λ	λ
		△ q_{14}											

$\dots$

λ	1	1	1	λ	λ	λ	y	1	1	1	λ	λ	λ
△ q_{14}													

Последним шагом будет запись символа ‘1’.

1	1	1	1	λ	λ	λ	y	1	1	1	λ	λ	λ
△ q_0													

Программа машины $M_{\mathrm{cmp}(y)}$ приводится в таблице 5.

Таблица 5 – программа машины $M_{\mathrm{cmp}(y)}$

	*	1	y	?
q_1				?Пq_2
q_2		*Пq_3	yПq_4	?Пq_5
q_3		1Пq_3	yПq_6	?Пq_3
q_4	*Пq_4	1Лq_8		?Лq_{10}
q_5		1Пq_5	yПq_4	?Пq_5
q_6	*Пq_6	*Лq_7		?Лq_8
q_7	*Лq_7		yЛq_9	
q_8	1Лq_8		yЛq_{11}	
q_9	*Пq_2	1Лq_9		?Лq_9
q_{10}	1Лq_{10}		yЛq_{13}	
q_{11}	1Лq_{12}	1Лq_{11}		?Лq_{11}
q_{12}	1Лq_{12}			0Нq_0
q_{13}	1Лq_{14}	1Лq_{13}		?Лq_{13}
q_{14}	1Лq_{14}			1Нq_0

Если разобрать три случая ($x < y$, $x > y$, $x = y$) и внимательно проследить за числом шагов машины на каждой стадии описанного алгоритма, затем упростить и объединить результаты, то можно прийти к следующей формуле для временной сложности:

$$T(x, y, l) = 2(\min\{x, y\})^2 + 10\min\{x, y\} + 13 + (l + \max\{x - y, 0\})(2\min\{x, y\} + 4). \quad (17)$$

2.5 Машина $M_{\max(y)}$

Построим ещё одну вспомогательную машину. Она имеет много общего с $M_{\mathrm{cmp}(y)}$.

В начальный момент времени на ленте задаётся два числа x и y так же, как и для машины $M_{\mathrm{cmp}(y)}$. Головка располагается над клеткой левее первой единицы числа x. Число y задано справа от числа x и его начало обозначается некоторой буквой ‘y’, которая больше нигде не встречается на ленте. Между клеткой с последней единицей x и клеткой с буквой ‘y’ имеется l ($l \geqslant 0$) других клеток с произвольными буквами, кроме ‘*’, и при этом первая из данных l клеток не содержит ‘1’. Например, здесь $x = 3$, $y = 1$ и $l = 2$.

?	1	1	1	1	?	?	y	1	1	λ	λ	λ
△ q_1												

В результате вычисления машины $M_{\max(y)}$ число y будет заменено максимумом из x и y, головка разместится над первой единицей x.

?	1	1	1	1	?	?	y	1	1	1	1	λ
	△ q_1											

Машина работает так. Вначале головка позиционируется над первой единицей.

?	1	1	1	1	?	?	y	1	1	λ	λ	λ
	△ q_2											

Вообще, алгоритм, по которому вычисляет машина, прост. Все единицы, составляющие запись числа x в унарной системе счисления, заменяются на символы ‘*’, при этом точно такое же число ‘*’ записывается подряд правее буквы ‘y’. Затем все символы ‘*’ заменяются на ‘1’.

Рассмотрим общий случай: уже k, $k \geqslant 0$, единиц числа x заменены на ‘*’ и после ‘y’ записано ровно k символов ‘*’. Считывающая головка расположена над клеткой, следующей за самой правой буквой ‘*’ числа x (если $k = 0$, то просто над первой единицей x), машина находится в состоянии q_2.

Здесь возможны два принципиально разных случая.

1. Головка считывает с ленты ‘1’ — очередную единицу записи числа ‘x’. Пусть для примера $k = 1$.

?	*	1	1	1	?	?	y	*	1	λ	λ	λ
		△ q_2										

Тогда машина записывает на ленту ‘*’, смещает считывающую головку вправо и переходит в состояние q_3.

?	*	*	1	1	?	?	y	*	1	λ	λ	λ
			△ q_3									

В состоянии q_3 машина ищет на ленте, перемещаясь вправо, букву ‘y’.

?	*	*	1	1	?	?	y	*	1	λ	λ	λ
				△ q_3								

...

?	*	*	1	1	?	?	y	*	1	λ	λ	λ
							△ q_3					

Далее машина изменяет состояние на q_4 и движется вправо, пока на ленте записаны буквы ‘*’.

?	*	*	1	1	?	?	y	*	1	λ	λ	λ
								△ q_4				

?	*	*	1	1	?	?	y	*	1	λ	λ	λ
									△ q_4			

Теперь, когда машина остановилась в состоянии q_4 на букве из множества $A \setminus \{\text{‘*’}\}$, она записывает на ленту ‘*’ и переходит в состояние q_5, переместив головку влево.

?	*	*	1	1	?	?	y	*	*	λ	λ	λ
								△ q_5				

Головка двигается влево, пока не будет считана буква ‘y’.

?	*	*	1	1	?	?	y	*	*	λ	λ	λ
							△ q_5					

Состояние меняется, направление движения головки остаётся прежним.

?	*	*	1	1	?	?	y	*	*	λ	λ	λ
						△ q_6						

...

?	*	*	1	1	?	?	y	*	*	λ	λ	λ
		△ q_6										

Когда головка дойдёт до ‘*’, машина передвинет её вправо и вернётся в состояние q_2.

?	*	*	1	1	?	?	y	*	*	λ	λ	λ
			△ q_2									

Теперь уже $(k+1)$ единица заменена на ‘*’, начинается новая итерация цикла.

2. Возможен вариант, когда в состоянии q_2 головка машины читает букву, отличную от ‘1’. Такая ситуация рано или поздно наступит, т. к. число x конечно.

?	*	*	*	*	?	?	y	*	*	*	*	λ
					△ q_2							

Машине необходимо убрать с ленты все буквы ‘*’, заменив их на ‘1’.

Машина переходит в состояние q_7. В состоянии q_7 она доходит до буквы ‘y’ и перемещается ещё на клетку правее — на первую букву ‘*’ блока, записанного сразу после буквы ‘y’.

?	*	*	*	*	?	?	y	*	*	*	*	λ
						△ q_7						

?	*	*	*	*	?	?	y	*	*	*	*	λ
							△ q_7					

Затем уже в состоянии q_8 машина находит конец этого блока.

?	*	*	*	*	?	?	y	*	*	*	*	λ
								$\triangle\ q_8$				

. . .

?	*	*	*	*	?	?	y	*	*	*	*	λ
												$\triangle\ q_8$

Затем, перейдя в состояние q_9, машина перемещается к началу блока, заменяя ‘*’ на ‘1’.

?	*	*	*	*	?	?	y	*	*	*	*	λ
											$\triangle\ q_9$	

. . .

?	*	*	*	*	?	?	y	1	1	1	1	λ
							$\triangle\ q_9$					

Состояние q_9 сменяется состоянием q_{10}, машина оставляет отличные от ‘*’ буквы без изменения.

?	*	*	*	*	?	?	y	1	1	1	1	λ
						$\triangle\ q_{10}$						

. . .

?	*	*	*	*	?	?	y	1	1	1	1	λ
				$\triangle\ q_{10}$								

Далее в состоянии q_{11} восстанавливается вид числа x.

. . .

?	1	1	1	1	?	?	y	1	1	1	1	λ
$\triangle\ q_{11}$												

Наконец работа машины прекращается, головка переводится на первую единицу.

?	1	1	1	1	?	?	y	1	1	1	1	λ
	$\triangle\ q_0$											

Программа машины приведена в таблице 6.

Таблица 6 – — программа машины $M_{\max(y)}$

	*	1	y	?
q_1				?Пq_2
q_2		*Пq_3	yПq_5	?Пq_4
q_3		1Пq_3	yПq_6	?Пq_3
q_4		1Пq_4	yПq_5	?Пq_4
q_5	*Пq_5	1Лq_9	yЛq_9	?Лq_9
q_6	*Пq_6	*Лq_7		*Лq_7
q_7	*Лq_7		yЛq_8	
q_8	*Пq_2	1Лq_8		?Лq_8
q_9	1Лq_9		yЛq_{10}	
q_{10}	1Лq_{11}	1Лq_{10}		?Лq_{10}
q_{11}	1Лq_{11}	1Пq_0		?Пq_0

Подсчёт числа шагов на каждом этапе приводит нас к такой формуле для временной сложности вычисления машины $M_{\max(y)}$ (заметим, от y она не зависит):

$$T(x,l) = 2x^2 + 10x + 13 + l(2x + 4). \tag{18}$$

3 Посторoение машины M

3.1 Формальная проверка входных данных

Для того чтобы последовательность символов вида (2), записанная на ленте машины Тьюринга по соглашениям из раздела 1.3, была программой некоторой другой машины, необходимо, чтобы эта последовательность удовлетворяла некоторым несложным формальным правилам.

Воспользуемся формой Бэкуса—Наура для определения формальной грамматики Γ, к языку которой обязательно принадлежит слово, задающее программу машины Тьюринга в формате, описанном в разделе 1.3.

⟨*программа*⟩ ::= ⟨*команда*⟩ | ⟨*программа*⟩ ‘`,`’ ⟨*команда*⟩

⟨*команда*⟩ ::= ⟨*левая часть*⟩ ‘`→`’ ⟨*правая часть*⟩

⟨*левая часть*⟩ ::= ⟨*состояние*⟩ ⟨*буква*⟩

⟨*правая часть*⟩ ::= ⟨*буква*⟩ ⟨*направление*⟩ ⟨*состояние*⟩

⟨*состояние*⟩ ::= ‘`q`’ ⟨*число*⟩

⟨*буква*⟩ ::= ‘`a`’ ⟨*число*⟩

⟨*направление*⟩ ::= ‘`Л`’ | ‘`Н`’ | ‘`П`’

⟨*число*⟩ ::= ‘`1`’ | ⟨*число*⟩ ‘`1`’

Нетрудно видеть, что это правила контекстно-свободной грамматики.

Рассмотрим пример программы, заданной на ленте машины Тьюринга, которая хоть и не является в действительности программой никакой машины, но удовлетворяет указанным формальным требованиям.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
△ q_1																									

Понятно, почему эта строка не является программой: здесь встречается внутреннее состояние q_1 и буква a_2, значит, $m \geqslant 1$ и $k \geqslant 2$, поэтому всего должно быть не менее $m(k+1) \geqslant 3$ команд; более того, есть переход из состояния q_0, чего быть не может.

Начнём построение машины M. Первоначально она проверяет, принадлежит ли слово на ленте языку грамматики Γ, и если не принадлежит, то это слово точно не является корректной программой, для него невозможно построить код, поэтому машина M к нему неприменима и входит в бесконечный цикл.

Находясь во внутреннем состоянии q_1, машина ожидает увидеть на ленте букву ‘`q`’. Если это не так, машина зацикливается (например, на каждом шаге не меняет внутреннее состояние, записывает на ленту ту же букву, что и была, и не перемещает считывающую голоку). Иначе состояние сменяется на q_2.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
	△ q_2																								

Теперь машина предполагает, что головка располагается над первой единицей числа в унарной системе счисления. Если это не так, машина входит в бесконечный цикл. Иначе смещает головку вправо и переходит в q_3.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
		△ q_3																							

Возможно, число состоит из более чем одной единицы. Машина смещает головку вправо, пока считывается единица, внутреннее состояние не меняется.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
			△ q_3																						

Когда наконец встретится буква из множества $A \setminus \{`1'\}$, это непременно должна быть буква ‘`a`’, иначе строка, записанная на ленте, не является программой и машина $M$ зацикливается. Если же головка оказалась над буквой ‘`a`’, она двигается вправо и машина переходит в q_4.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
				△ q_4																					

Если входные данные представляют собой программу, головка будет располагаться над началом числа. Если считывается ‘`1`’, машина переходит в q_5, иначе зацикливается.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
					△ q_5																				

Теперь машина идёт вправо, оставаясь в состоянии q_5, пока на ленте записаны буквы ‘`1`’.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
						△ q_5																			

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
							△ q_5																		

Если встречается буква ‘`→`’, это означает, что левая часть одной команды программы считана правильно, и машина переходит в состояние q_6. Если же в состоянии q_5 машина считывает букву из $A \setminus \{`1', `\to'\}$, формат команды нарушен (записанная на ленте строка точно не является программой), и машина переходит к осуществлению бесконечного цикла.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
								△ q_6																	

Из q_6 в случае чтения буквы ‘`a`’ есть переход в q_7 и в случае чтения любой другой буквы есть переход в q_6 (зацикливание).

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
									△ q_7																

В состоянии q_7 читается первая единица числа.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
										△ q_8															

Машина затем движется вправо, оставаясь в состоянии q_8.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
											△ q_8														

Если будет обнаружена буква ‘`Л`’, ‘`Н`’ или ‘`П`’, состояние сменится на q_9. Иначе будет начато выполнение бесконечного цикла.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
												△ q_9													

В состоянии q_9 ожидается считывание буквы ‘`q`’.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
													△ q_{10}												

Далее в состоянии q_{10} читается первая единица числа, выполняется переход в q_{11}.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1
														△ q_{11}											

Пока машина находится в состоянии q_{11} и считывается ‘`1`’, головка движется вправо.

Допустим, найдена наконец буква из множества $A \setminus \{$‘`1`’$\}$. Если это λ, можно сделать вывод, что на ленте действительно записано слово из языка грамматики Г.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	λ
																									△ q_{11}

Если на ленте считывается ‘`,`’, можно, сместив головку вправо, перейти к чтению следующей команды программы, т. е. снова к состоянию q_1. Иначе нужно войти в бесконечный цикл.

3.2 Получение m и k

Числа m и k (мощности множества внутренних состояний и внешнего алфавита машины, программа которой подана на вход машине M) можно найти как максимум соответствующих индексов (см. (2)):

$$m = \max_{n=\overline{1,w}} i_n, \tag{19}$$

$$k = \max_{n=\overline{1,w}} j_n. \tag{20}$$

Конфигурация после завершения предыдущей стадии такова.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	λ
																									△ q_{11}

Договоримся, что число m будет записано после имеющихся на ленте данных и будет отделено буквой ‘`m`’, нигде ранее не использовавшейся. Машина пишет на ленту ‘`m`’, ‘`1`’, т. е. устанавливает значение $m = 0$, далее перемещает головку в крайнюю пустую клетку, предшествующую первой занятой клетке. С этой целью применяются состояния q_{12}, q_{13}.

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	λ
																										△ q_{12}

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1
																									△ q_{13}	

q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	λ
																								△ q_{13}			

. . .

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	λ
△ q_{13}																													

Теперь головка перемещается вправо.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	λ
	△ q_{14}																												

Если в состоянии q_{14} машина считывает букву ‘q’, машина, не сдвигая головки, переходит в q_{15}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	λ
	△ q_{15}																												

Сейчас запускается машина $M_{\max(m)}$, которая заменит число m максимумом из m и текущим индексом при ‘q’. Переобозначим внутренние состояния машины $M_{\max(m)}$ так, чтобы они не совпадали с состояниями машины M (за исключением начального и конечного). Внутреннее состояние q_{15} машины M отождествим с внутренним состоянием q_1 машины $M_{\max(m)}$. Заключительное состояние машины $M_{\max(m)}$ отождествим с внутренним состоянием q_{14} машины M.

Для рассматриваемого нами примера после работы $M_{\max(m)}$ получим такую конфигурацию (теперь $m = 1$):

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1
		△ q_{14}																											

Если в состоянии q_{14} считывается любая буква из $A \setminus \{‘\mathtt{q}’, \lambda\}$, машина не изменяет эту букву и двигается вправо.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1
			△ q_{14}																										

…

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1
													△ q_{14}																

Когда будет обнаружена очередная буква ‘q’, машина M снова перейдёт в q_{15} и начнёт выполнять команды машины $M_{\max(m)}$.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1
													△ q_{15}																

…

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1
														△ q_{14}															

Когда наконец головка машины в состоянии q_{14} дойдёт то λ, после буквы ‘m’ окажется записанным число, определяемое формулой (19).

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	λ
																														△ q_{14}

Теперь машина M записывает на ленту ‘k’, переходит в q_{16}, потом пишет ‘1’ (устанавливает $k = 0$), переходит в q_{17} и движется влево.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	λ
																															△ q_{16}

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
																														△ q_{17}	

Головка перемещается, пока не будет найдена λ. Нетрудно видеть, что состояние q_{17} аналогично состоянию q_{13}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
																													△ q_{17}		

...

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
△ q_{17}																															

Теперь машина M переходит в q_{18} и смещается вправо. Это состояние имеет сходство с состоянием q_{14}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
	△ q_{18}																														

Если на ленте записана любая буква из $A \setminus \{\text{‘a’}, \lambda\}$, машина M просто сдвигает головку вправо.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
		△ q_{18}																													

...

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
				△ q_{18}																											

Если в состоянии q_{18} головка считывает ‘a’, машина переходит в состояние q_{19}. Оно является начальным состоянием q_1 для вспомогательной машины $M_{\max(k)}$. Заключительное же внутреннее состояние этой вспомогательной машины отождествляется с внутренним состоянием q_{18} машины M.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1
				△ q_{19}																											

...

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1
					△ q_{18}																												

Таким образом, число k выбирается согласно (20) как максимум из индексов, записанных при ‘a’.

Наконец, когда машина M проанализирует всю исходную строку, поданную ей на вход, в состоянии q_{18} будет считываться λ.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	λ
																																		△ q_{18}

3.3 Проверка неравенств

В разделе 1.1 утверждается, что множество внутренних состояний и внешний алфавит машины Тьюринга состоит по крайней мере из двух элементов, т. е. $m \geqslant 1$ и $k \geqslant 1$. Проверим это. Одновременно сформируем новые значения $\tilde{m} = 1$ и $\tilde{k} = 0$, заменив в числах m и k правые единицы нулями.

Из состояния q_{18} по символу λ есть переход в q_{20}, который сопровождается записью на ленту символа ‘,’ и смещением головки влево.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																																	△ q_{20}	

Машина затем двигает головку влево, находясь пр этом в состоянии q_{20}, пока не будет найдена буква '`m`'.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																																△ q_{20}		

…

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																											△ q_{20}							

Затем головка смещается вправо, состояние сменяется на q_{21}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																												△ q_{21}						

В данный момент головка считывает первую единицу числа m. Головка сдвигается вправо, состояние сменяется на q_{22}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																													△ q_{22}					

Если выполнено условие $m \geqslant 1$, головка будет считывать единицу (вторую в записи числа m в унарной системе счисления). Машина оставит на ленте единицу и перейдёт в состояние q_{23}, переместив при этом головку вправо. Если же $m = 0$, запись числа m состоит из единственной единицы, и в состоянии q_{22} машина прочитает с ленты '`k`'. В этом случае записанная на ленте строка не является программой, поэтому машина M входит в бесконечный цикл.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																														△ q_{23}				

Пока в состоянии q_{23} считываются единицы, они заменяются на нули, головка перемещается по ленте слева направо. Затем, когда будет считана буква '`k`', состояние меняется на q_{24}, головка переместится вправо.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																															△ q_{24}			

Считывается первыя единица числа k, состояние меняется на q_{25}, головка двигается вправо.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	1	1	,
																																△ q_{25}		

Если сейчас головка располагается над клеткой с '`,`', это значит, что число k состоит из одной единицы, т. е. $k = 0$, а следовательно, строка на ленте не является корректной программой. Тогда машина M входит в бесконечный цикл. Иначе на ленту пишется '`0`', состояние меняется на q_{26}, головка двигается снова вправо.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	1	,
																																	△ q_{26}	

Машина заменяет все '`1`' на '`0`'.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
																																		△ q_{26}

Когда она доходит до конца строки на ленте (до '`,`'), она переходит с состояние q_{27}.

3.4 Цикл, который записывает код

Начинается осуществление цикла. Находясь в состоянии q_{27}, машина идёт влево, пока не достигнет пустой клетки.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
																																	△ q_{27}	

. . .

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
△ q_{27}																																		

Когда клетка с λ найдена, машина передвигает головку вправо, переходя в состояние q_{28}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
	△ q_{28}																																	

В состоянии q_{28} осуществляется поиск следующей команды для обработки. Пока с ленты считывается буква ‘`,`’, машина движется вправо. Если же с ленты считывается ‘`m`’, это говорит о том, что требуемая команда на ленте не найдена, записанная программа некорректна, поэтому машина M зацикливается.

Итак, пусть в состоянии q_{28} машина считывает с ленты букву, отличную от ‘`m`’ и ‘`,`’. Мы построим далее программу машины M таким образом, что эта буква будет обязательно буквой ‘`q`’. Машина переходит в состояние q_{29}. В нашем примере переход из q_{28} в q_{29} произойдёт сразу же без перемещения головки.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
	△ q_{29}																																	

В состоянии q_{29} анализируется команда вида «$q_ia_j \to a_sdq_r$», которая начинается в текущей позиции головки. В рассматриваемом нами примере это будет команда «$q_1a_2 \to a_1$Лq_0». Сейчас машина M должна проверить, верно ли, что $i = \tilde{m}$. С этой целью внутреннее состояние q_{29} машины M отождествляется с внутренним состоянием q_1 машины $M_{\mathrm{cmp}(m)}$, а состояние q_0 машины $M_{\mathrm{cmp}(m)}$ — с состоянием q_{30} машины M.

λ	1	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
	△ q_{30}																																	

Теперь после выполнения всех команд вспомогательной машины $M_{\mathrm{cmp}(m)}$ в текущей клетке, над которой располагается головка, записан ‘`0`’, если $i \neq \tilde{m}$, или ‘`1`’, если $i = \tilde{m}$.

1. Если в состоянии q_{30} головка считывает ‘`0`’, то текущая команда точно не является той командой, код которой надо сейчас выписать (напомним, что машина M должна выписать код команды с левой частью «$q_{\tilde{m}}a_{\tilde{k}}$»). Нужно пропускать текущую команду и переходить к следующей.

Машина восстанавливает на ленте букву ‘`q`’, смещается вправо и переходит в состояние q_{31}. В состоянии q_{31} машина движется вправо, пропуская буквы ‘`a`’, ‘`q`’, ‘`1`’, ‘→’, ‘Л’, ‘Н’, ‘П’. Машина остановится, прочитав либо ‘`,`’ (тогда она переходит в q_{28} — к анализу следующей команды), или ‘`m`’ (тогда она зацикливается: требуемой команды на ленте не нашлось).

2. Если головка считает с ленты ‘`1`’ в состоянии q_{30}, как это и получилось в нашем примере, то выполняется равенство $i = \tilde{m}$, нужно проверить равенство $j = \tilde{k}$. Если и оно выполнится, нужно записать код всей команды «$q_ia_j \to a_sdq_r$».

Машина восстанавливает на ленте букву ‘`q`’, смещается влево, переходя в состояние q_{32}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
		△ q_{32}																																

Затем, пока не будет обнаружена буква ‘a’, машина движется вправо в состоянии q_{32}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
				△ q_{32}																														

Потом машина переходит во внутреннее состояние q_{33}. Это состояное отождествляется с начальным состоянием q_1 машины $M_{\text{cmp}(k)}$. Заключительное же состояние q_0 отождествляется с состоянием q_{34} машины M.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
				△ q_{33}																														

...

λ	q	1	1	0	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
				△ q_{34}																														

После выполнения команд вспомогательной машины $M_{\text{cmp}(k)}$ возникает ситуация, аналогичная рассмотренной ранее. Если в текущей клетке (той, над которой располагается головка), записан ‘0’, то $j \neq \tilde{k}$, если же в клетке записана ‘1’, то $j = \tilde{k}$.

1. Пусть машина в состоянии q_{34} считывает с ленты ‘0’, как это произошло в рассматриваемом примере. Значит, нужно восстановить на ленте ‘a’ и переходить к анализу следующей команды. С этой целью выше вводилось состояние q_{31}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
					△ q_{31}																													

...

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
															△ q_{31}																			

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	q	1	a	1	→	a	1	П	q	1	1	m	1	1	k	1	0	0	,
																△ q_{28}																		

2. В состоянии q_{34} машина считывает с ленты ‘1’. Значит, выполняются оба равенства $i = \tilde{m}$ и $j = \tilde{k}$. Машина далее запишет код команды «$q_i a_j \to a_s d q_r$», а все буквы на ленте, ранее относившиеся к этой команде, заменит на разделители ‘,’.

Рассмотрим для наглядности другой пример, где знаком ‘?’ обозначен целый отрезок ленты, состояние которого в данный момент несущественно.

λ	q	1	1	1	1	1	1	→	a	1	1	Л	q	1	,	?	λ
				△ q_{34}													

Машина работает так. Она вновь пишет на ленту букву ‘a’, которую удалила вспомогательная машина $M_{\text{cmp}(k)}$, и смещается влево, переходя в состояние q_{35}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	?	λ
			△ q_{35}														

В состоянии q_{35} машина двигает головку влево, пока не встретится ‘q’.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	?	λ
		△ q_{35}															

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	?	λ

△ q_{35}

Далее машина, не перемещая головки, переходит в состояние q_{36}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	?	λ

△ q_{36}

Теперь запускается цепочка вспомогательных машин. Начинает действовать машина $M_{k(q)}$: её начальное состояние отождествляется с состоянием q_{36} машины M, а заключительное — с состоянием q_{37}.

λ	q	1	1	a	1	1	1	→	a	1	1	Л	q	1	,	?	λ

△ q_{36}

...

λ	,	,	,	a	1	1	1	→	a	1	1	Л	q	1	,	?	1	0	0	0	0	0	0	0

△ q_{37}

В свою очередь, состояние q_{37} отождествляется с начальным состоянием вспомогательной машины $M_{k(a)}$, её заключительное состояние соответствует внутреннему состоянию q_{38} машины M.

λ	,	,	,	a	1	1	1	→	a	1	1	Л	q	1	,	?	1	0	0	0	0	0	0	0

△ q_{37}

...

λ	,	,	,	,	,	,	,	→	a	1	1	Л	q	1	,	?	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

△ q_{38}

Машина M, находясь в состоянии q_{38} и считывая '→', пишет на ленту ',', перемещает головку вправо, переходя при этом в состояние q_{39}.

λ	,	,	,	,	,	,	,	,	a	1	1	Л	q	1	,	?	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

△ q_{39}

Дальше вновь запускается машина $M_{k(a)}$. Её начальное состояние отождествляется с состоянием q_{39}, а заключительное — с q_{40}.

...

λ	,	,	,	,	,	,	,	,	,	,	,	Л	q	1	,	?	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	

△ q_{40}

	0	0	0	0	0	0

После этого работает машина $T_{k(d)}$: q_{40} — её начальное состояние, q_{41} — заключительное.

...

λ	,	,	,	,	,	,	,	,	,	,	,	,	q	1	,	?	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	

△ q_{41}

	0	0	0	0	0	0	1	0

Наконец, состояние q_{41} отождествляется с начальным состоянием машины $M_{k(q)}$, а q_{42} — с её заключительным состоянием.

. . .

λ	,	,	,	,	,	,	,	,	,	,	,	,	,	,	,	?	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	

△ q_{42}

	0	0	0	0	0	0	1	0	1	0	0	0	0	0

Сейчас машина движется по ленте вправо, пока не будет прочитана буква ‘`k`’. Рассмотрим упрощённый пример, в котором несущественные на текущем этапе буквы на ленте обозначены просто знаком ‘`?`’.

?	m	1	1	k	1	0	0	,

△ q_{42}

Если $\tilde{k} < k$, машина M выполняет операцию увеличения числа $\tilde{k}$ на единицу и переходит в состояние q_{27}. Если же $\tilde{k} = k$, машина устанавливает значение $\tilde{k} = 0$, а затем переходит к числу $\tilde{m}$. Если $\tilde{m} < m$, к этому числу прибавляется единица и машина переходит в состояние q_{27}. Если же $\tilde{m} = m$, запись кода завершена успешно, осталось только проверить, все ли команды удалены с ленты и заменены на ‘`,`’ (если не все, программа, поданная на вход машине M, содержала несколько переходов из одной пары (q_i, a_j), чего не может быть в программе машины Тьюринга).

Если считывается ‘`k`’, то из состояния q_{42} осуществляется переход в q_{43}, буква ‘`k`’ на ленте сохраняется.

?	m	1	1	k	1	0	0	,

△ q_{43}

В состоянии q_{43} головка движется вправо, пока на ленте записаны единицы.

?	m	1	1	k	1	0	0	,

△ q_{43}

Далее возможны два случая.

1. На ленте следующим за группой подряд идущих ‘`1`’ стоит ‘`0`’. Тогда $\tilde{k} < k$, машина пишет на ленту ‘`1`’ вместо ‘`0`’ и, смещаясь влево, переходит в состояние q_{27}.

?	m	1	1	k	1	1	0	,

△ q_{27}

2. Находясь в состоянии q_{43}, машина обнаруживает на ленте ‘`,`’. Это свидетельствует о том, что $\tilde{k} = k$. Построим такой пример.

?	m	1	1	0	k	1	1	1	,

△ q_{43}

Тогда машина действует так. Символ ‘`,`’ остаётся на прежнем месте, головка смещается влево, состояние сменяется на q_{44}.

?	m	1	1	0	k	1	1	1	,

△ q_{44}

Головка перемещается влево, пока не будет считана буква ‘`k`’, при этом все ‘`1`’ заменяются на ‘`0`’.

?	m	1	1	0	k	1	1	0	,

△ q_{44}

?	m	1	1	0	k	1	0	0	,
						△ q_{44}			

?	m	1	1	0	k	0	0	0	,
					△ q_{44}				

Далее машина сменяет состояние на q_{45} и делает шаг вправо.

?	m	1	1	0	k	0	0	0	,
						△ q_{45}			

Записывается '`1`', головка делает шаг влево. внутреннее состояние изменяется на q_{46}.

?	m	1	1	0	k	1	0	0	,
					△ q_{46}				

Затем вновь шаг влево, переход в q_{47}.

?	m	1	1	0	k	1	0	0	,
				△ q_{47}					

Пока с ленты читается '`0`', головка идёт влево.

?	m	1	1	0	k	1	0	0	,
			△ q_{47}						

Как только встретилась '`1`', головка сдвигается вправо, состояние меняется на q_{48}.

?	m	1	1	0	k	1	0	0	,
				△ q_{48}					

Далее возможны варианты.

1. В состоянии q_{48} с ленты считывается '`0`'. Это говорит о том, что $\tilde{m} < m$. Машина записывает '`1`' вместо '`0`' и переходит в состояние q_{27}.

2. В состоянии q_{48} с ленты читается '`k`'. Значит, $\tilde{m} = m$, процесс записи кода окончен. Рассмотрим простейший пример. Пусть изначально на ленте было записано

q	1	1	a	1	→	a	1	H	q	1
△ q_1										

В интересующий нас момент исполнения программы состояние ленты будет таким:

,	,	,	,	,	,	,	,	,	,	,	m	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	
														△ q_{48}																					

	1	0	0	1	0	0	0	0	0

Тогда машина переходит в состояние q_{49} и начинает движение в левую сторону.

,	,	,	,	,	,	,	,	,	,	,	m	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	
													△ q_{49}																						

	1	0	0	1	0	0	0	0	0

. . .

λ	,	,	,	,	,	,	,	,	,	,	,	m	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	
△ q_{49}																																			

	0	1	0	0	1	0	0	0	0	0

Как только в состоянии q_{49} машина доходит до пустой клетки, она меняет направление движения и движется, стирая с ленты ‘`,`’, оставаясь в состоянии q_{50}, пока не будет обнаружена клетка с ‘`m`’. Если во время движения машина обнаружит какой-либо символ из $A \setminus \{$‘`,`’, ‘`m`’$\}$, она зацикливается.

λ	,	,	,	,	,	,	,	,	,	,	,	m	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	

△ q_{50}

	0	1	0	0	1	0	0	0	0	0

λ	λ	,	,	,	,	,	,	,	,	,	,	m	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	

△ q_{50}

	0	1	0	0	1	0	0	0	0	0

…

m	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0

△ q_{50}

Затем машина переходит в состояние q_{51} и продолжает стирать с ленты буквы, пока не найдёт клетку с ‘`,`’.

λ	1	1	k	1	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0

△ q_{51}

…

λ	λ	λ	λ	λ	,	1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0

△ q_{51}

Наконец, машина стирает этот символ, перемещает головку вправо и останавливается.

1	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0

△ q_0

Запись кода завершена успешно.

4 Оценка временной сложности

Мы построили машину Тьюринга M, решающую поставленную задачу. Оценим временную сложность её вычисления.

Пусть на вход машине M подаётся программа некоторой машины Тьюринга M_0, множество внутренних состояний которой $Q = \{q_0, q_1, \ldots, q_m\}$ имеет мощность $(m+1)$, а внешний алфавит $A = \{a_0, a_1, \ldots, a_k\}$ состоит из $(k+1)$ букв. Оценим длину строки (2), описывающей программу машины M_0.

Строка состоит ровно из $m(k+1)$ команд. Рассмотрим одну команду вида (1). Она состоит из буквы '`q`', унарной записи числа i ($1 \leqslant i \leqslant m$, т. е. число содержит не более $(m+1)$ единиц), буквы '`a`', записи числа j ($0 \leqslant i \leqslant k$, не более $(k+1)$ единиц), буквы '`→`', буквы '`a`', числа s (не более $(k+1)$ единиц), одной из букв '`Л`', '`Н`', '`П`', буквы '`q`', числа r (не более $(m+1)$ единиц). В сумме длина одной команды получается не более $1 + (m+1) + 1 + (k+1) + 1 + 1 + (k+1) + 1 + 1 + (m+1) = 2m + 2k + 10$. Общая длина l_1 всей строки вида (2) включает запись $m(k+1)$ команд и $(m(k+1) - 1)$ букв '`,`', поэтому

$$l_1 \leqslant m(k+1)(2m + 2k + 11) - 1. \tag{21}$$

Оценим, какой может быть длина кода машины M_0. Согласно (4) код одной команды имеет следубщий вид: единица, затем не более чем $(2m+5)$ нулей, единица, не более чем $(2k+4)$ нулей, единица, не более чем 3 нуля, единица, не более чем $(2k+4)$ нулей, наконец единица и вновь не более чем $(2m+5)$ нулей. Получается, что длина кода одной команды не превзойдёт $(4m+4k+26)$, а тогда для длины l_2 кода машины M_0 справедлива оценка

$$l_2 \leqslant m(k+1)(4m + 4k + 26). \tag{22}$$

Машина M строилась так, что помимо исходного и конечного слова на ленте находилась и унарная запись чисел m и k, буквы '`m`', '`k`', '`,`', поэтому общая длина отрезка ленты, по которому за всё время работы машины M передвигается головка, оценивается сверху так:

$$l = l_1 + l_2 + m + k + 7 \leqslant m(k+1)(6m + 6k + 37) + m + k + 6 = O(mk(m+k)). \tag{23}$$

На первоначальной стадии формальной проверки входных данных машина M выполняет $O(l) = O(mk(m+k))$ шагов, т. к. осуществляется всего один проход по ленте и головка движется только вправо.

Далее на стадии проверки неравенств и установки значений «счётчикам цикла» тербуется всего лишь $O(n+k)$ шагов.

Затем следует стадия выписывания m и k. Машина делает два прхода по $O(l)$ клеткам ленты, проводя $m(k+1)$ операций взятия максимума, причём каждая такая операция согласно (18), требует $O(m^2 + k^2 + l(m+k)) = O(mk(m+k)^2)$ шагов.

Всего машине M нужно выписать коды $m(k+1)$ команд. Зафиксируем левую часть команды, которую машине M требуется выписать в данный момент. Чтобы найти в исходной строке вида (2) команду с данной левой частью, требуется выполнить не более $m(k+1)$ сравнений индексов при '`q`' и '`a`'. Согласно (17), сравнение индексов при '`q`' занимает $O(m^2 + lm)$ шагов, а сравнение индексов при '`a`' — $O(k^2 + lk)$ шагов. Получается, что только на поиск нужной команды уйдёт $O(mk(m^2+k^2+l(m+k)))$ шагов, а с учётом оценки на l это можно записать в виде $O(m^2k^2(m+k)^2)$. Непосредственно запись команды занимает много меньшее время: из (14), (11), (16) заключаем, что для записи одной команды потребуется всего $O(m^2 + k^2 + l(m+k)) = O(mk(m+k)^2)$ шагов. А всего команд $m(k+1)$. Тогда после отбрасывания незначительных слагаемых получим оценку $O(m^2k^2(m+k)^2)$

Суммируя оценки на всех четырёх стадиях, приходим к окончательной формуле

$$T(m,k) = O(mk(m+k)) + O(n+k) + O(mk(m+k)^2) + O(m^2k^2(m+k)^2), \tag{24}$$

или, что эквивалентно,

$$T(m,k) = O(m^2k^2(m+k)^2). \tag{25}$$

Оценка осуществлялась весьма грубо, однако она позволяет заключить, что построенный алгоритм на любых корректных входных данных завершится за конечное число шагов, и время его работы ограничивается полиномом от m и k.

ЗАКЛЮЧЕНИЕ

В ходе выполнения работы была построена машина Тьюринга M, решающая поставленную задачу: по программе вида (2) построить двоичный код машины Тьюринга вида (5). Построенная машина M имеет 159 внутренних состояний и внешний алфавит мощности 13, итого в программе $13 \times (159 - 1) = 2054$ команды. Отметим, что цель сократить число внутренних состояний или мощность внешнего алфавита не ставилась. Программа машины M полностью приведена в приложении А.

Показано, что если машине M на вход подать программу некоторой машины с $(m+1)$ внутренними состояниями и $(k+1)$ буквами во внешнем алфавите, то машина M выполнит $O(m^2k^2(m+k)^2)$ шагов, т. е. время её работы полиномиально зависит от k и m. А поскольку непосредственная длина входной строки не превосходит k и m, то время работы машины M полиномиально зависит от длины входа.

Дальнейшие исследования по данной теме могут быть связаны с разработкой более эффективного алгоритма, решающего ту же задачу, с доказательством какой-либо нижней оценки на число шагов, с минимизацией числа внутренних состояний в машине M, а также с исследованием того, как меняется код машины при добавлении или удалении её состояний.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. *Мощенский А. В.* Курс математической логики: Учеб. пособие / А. В. Мощеский, В. А. Мощенский; Пер. с бел. яз. авторов. — Мн.: БГУ, 2001. — 129 с.
2. *Мощенский А. В.* Математические основы информатики: Пособие для студентов спец. 1-31 03 04 «Информатика» / А. В. Мощеский, В. А. Мощенский. — 2-е изд., перераб. и доп. — Мн. : БГУ, 2008. — 155 с.

ПРИЛОЖЕНИЕ А Программа построенной машины

Таблица А.1

	λ	*	,	→	0	1	Л	Н	П	a	k	m	q
q_1	λНq_1	*Нq_1	,Нq_1	→Нq_1	0Нq_1	1Нq_1	ЛНq_1	ННq_1	ПНq_1	aНq_1	kНq_1	mНq_1	qПq_2
q_2	λНq_2	*Нq_2	,Нq_2	→Нq_2	0Нq_2	1Пq_3	ЛНq_2	ННq_2	ПНq_2	aНq_2	kНq_2	mНq_2	qНq_2
q_3	λНq_3	*Нq_3	,Нq_3	→Нq_3	0Нq_3	1Пq_3	ЛНq_3	ННq_3	ПНq_3	aПq_4	kНq_3	mНq_3	qНq_3
q_4	λНq_4	*Нq_4	,Нq_4	→Нq_4	0Нq_4	1Пq_5	ЛНq_4	ННq_4	ПНq_4	aНq_4	kНq_4	mНq_4	qНq_4
q_5	λНq_5	*Нq_5	,Нq_5	→Пq_6	0Нq_5	1Пq_5	ЛНq_5	ННq_5	ПНq_5	aНq_5	kНq_5	mНq_5	qНq_5
q_6	λНq_6	*Нq_6	,Нq_6	→Нq_6	0Нq_6	1Нq_6	ЛНq_6	ННq_6	ПНq_6	aПq_7	kНq_6	mНq_6	qНq_6
q_7	λНq_7	*Нq_7	,Нq_7	→Нq_7	0Нq_7	1Пq_8	ЛНq_7	ННq_7	ПНq_7	aНq_7	kНq_7	mНq_7	qНq_7
q_8	λНq_8	*Нq_8	,Нq_8	→Нq_8	0Нq_8	1Пq_8	ЛПq_9	НПq_9	ППq_9	aНq_8	kНq_8	mНq_8	qНq_8
q_9	λНq_9	*Нq_9	,Нq_9	→Нq_9	0Нq_9	1Нq_9	ЛНq_9	ННq_9	ПНq_9	aНq_9	kНq_9	mНq_9	qПq_{10}
q_{10}	λНq_{10}	*Нq_{10}	,Нq_{10}	→Нq_{10}	0Нq_{10}	1Пq_{11}	ЛНq_{10}	ННq_{10}	ПНq_{10}	aНq_{10}	kНq_{10}	mНq_{10}	qНq_{10}
q_{11}	mПq_{12}	*Нq_{11}	,Пq_1	→Нq_{11}	0Нq_{11}	1Пq_{11}	ЛНq_{11}	ННq_{11}	ПНq_{11}	aНq_{11}	kНq_{11}	mНq_{11}	qНq_{11}
q_{12}	1Лq_{13}												
q_{13}	λПq_{14}		,Лq_{13}	→Лq_{13}		1Лq_{13}	ЛЛq_{13}	НЛq_{13}	ПЛq_{13}	aЛq_{13}		mЛq_{13}	qЛq_{13}
q_{14}	kПq_{16}		,Пq_{14}	→Пq_{14}		1Пq_{14}	ЛПq_{14}	НПq_{14}	ППq_{14}	aПq_{14}		mПq_{14}	qНq_{15}
q_{15}	λПq_{53}		,Пq_{53}	→Пq_{53}	0Пq_{53}		ЛПq_{53}	НПq_{53}	ППq_{53}	aПq_{53}	kПq_{53}		qПq_{53}
q_{16}	1Лq_{17}												
q_{17}	λПq_{18}		,Лq_{17}	→Лq_{17}		1Лq_{17}	ЛЛq_{17}	НЛq_{17}	ПЛq_{17}	aЛq_{17}	kЛq_{17}	mЛq_{17}	qЛq_{17}
q_{18}	,Лq_{20}		,Пq_{18}	→Пq_{18}		1Пq_{18}	ЛПq_{18}	НПq_{18}	ППq_{18}	aНq_{19}	kПq_{18}	mПq_{18}	qПq_{18}
q_{19}	λПq_{63}		,Пq_{63}	→Пq_{63}	0Пq_{63}		ЛПq_{63}	НПq_{63}	ППq_{63}	aПq_{63}		mПq_{63}	qПq_{63}
q_{20}						1Лq_{20}					kЛq_{20}	mПq_{21}	
q_{21}						1Пq_{22}							
q_{22}						1Пq_{23}					kНq_{22}		
q_{23}						0Пq_{23}					kПq_{24}		
q_{24}						1Пq_{25}							
q_{25}			,Нq_{25}			0Пq_{26}							
q_{26}			,Лq_{27}			0Пq_{26}							
q_{27}	λПq_{28}		,Лq_{27}	→Лq_{27}	0Лq_{27}	1Лq_{27}	ЛЛq_{27}	НЛq_{27}	ПЛq_{27}	aЛq_{27}	kЛq_{27}	mЛq_{27}	qЛq_{27}
q_{28}			,Пq_{28}									mНq_{28}	qНq_{29}
q_{29}	λПq_{73}		,Пq_{73}	→Пq_{73}	0Пq_{73}		ЛПq_{73}	НПq_{73}	ППq_{73}	aПq_{73}	kПq_{73}		qПq_{73}
q_{30}					qПq_{31}	qПq_{32}							
q_{31}			,Пq_{28}	→Пq_{31}		1Пq_{31}	ЛПq_{31}	НПq_{31}	ППq_{31}	aПq_{31}		mНq_{31}	qПq_{31}
q_{32}						1Пq_{32}				aНq_{33}			
q_{33}	λПq_{86}		,Пq_{86}	→Пq_{86}	0Пq_{86}		ЛПq_{86}	НПq_{86}	ППq_{86}	aПq_{86}		mПq_{86}	qПq_{86}
q_{34}					aПq_{31}	aЛq_{35}							
q_{35}						1Лq_{35}							qНq_{36}
q_{36}													qПq_{99}
q_{37}										aПq_{112}			
q_{38}				,Пq_{39}									
q_{39}										aПq_{124}			
q_{40}							*Пq_{136}	*Пq_{138}	*Пq_{141}				

Продолжение таблицы A.1

	λ	*	,	→	0	1	Л	Н	П	a	k	m	q
q_{41}													qПq_{146}
q_{42}			,Пq_{42}	→Пq_{42}	0Пq_{42}	1Пq_{42}	ЛПq_{42}	НПq_{42}	ППq_{42}	aПq_{42}	kПq_{43}	mПq_{42}	qПq_{42}
q_{43}			,Лq_{44}		1Лq_{27}	1Пq_{43}							
q_{44}						0Лq_{44}					kПq_{45}		
q_{45}					1Лq_{46}								
q_{46}											kЛq_{47}		
q_{47}					0Лq_{47}	1Пq_{48}							
q_{48}					1Лq_{27}						kЛq_{49}		
q_{49}	λПq_{50}		,Лq_{49}	→Лq_{49}	0Лq_{49}	1Лq_{49}	ЛЛq_{49}	НЛq_{49}	ПЛq_{49}	aЛq_{49}		mЛq_{49}	qЛq_{49}
q_{50}			λПq_{50}	→Нq_{50}		1Нq_{50}	ЛНq_{50}	ННq_{50}	ПНq_{50}	aНq_{50}		λПq_{51}	qНq_{50}
q_{51}			λПq_{0}		λПq_{51}	λПq_{51}					λПq_{51}		
q_{52}													
q_{53}	λПq_{58}		,Пq_{58}	→Пq_{58}	0Пq_{58}	*Пq_{54}	ЛПq_{58}	НПq_{58}	ППq_{58}	aПq_{58}	kПq_{58}	mПq_{59}	qПq_{58}
q_{54}	λПq_{54}		,Пq_{54}	→Пq_{54}	0Пq_{54}	1Пq_{54}	ЛПq_{54}	НПq_{54}	ППq_{54}	aПq_{54}	kПq_{54}	mПq_{55}	qПq_{54}
q_{55}	*Лq_{56}	*Пq_{55}	*Лq_{56}	*Лq_{56}	*Лq_{56}	*Лq_{56}	*Лq_{56}	*Лq_{56}	*Лq_{56}	*Лq_{56}	*Лq_{56}		*Лq_{56}
q_{56}		*Лq_{56}										mЛq_{57}	
q_{57}	λЛq_{57}	*Пq_{53}	,Лq_{57}	→Лq_{57}	0Лq_{57}	1Лq_{57}	ЛЛq_{57}	НЛq_{57}	ПЛq_{57}	aЛq_{57}	kЛq_{57}		qЛq_{57}
q_{58}	λПq_{58}		,Пq_{58}	→Пq_{58}	0Пq_{58}	1Пq_{58}	ЛПq_{58}	НПq_{58}	ППq_{58}	aПq_{58}	kПq_{58}	mПq_{59}	qПq_{58}
q_{59}	λЛq_{60}	*Пq_{59}	,Лq_{60}	→Лq_{60}	0Лq_{60}	1Лq_{60}	ЛЛq_{60}	НЛq_{60}	ПЛq_{60}	aЛq_{60}	kЛq_{60}	mЛq_{60}	qЛq_{60}
q_{60}		1Лq_{60}										mЛq_{61}	
q_{61}	λЛq_{61}	1Лq_{62}	,Лq_{61}	→Лq_{61}	0Лq_{61}	1Лq_{61}	ЛЛq_{61}	НЛq_{61}	ПЛq_{61}	aЛq_{61}	kЛq_{61}		qЛq_{61}
q_{62}	λПq_{14}	1Лq_{62}	,Пq_{14}	→Пq_{14}	0Пq_{14}	1Пq_{14}	ЛПq_{14}	НПq_{14}	ППq_{14}	aПq_{14}	kПq_{14}		qПq_{14}
q_{63}	λПq_{68}		,Пq_{68}	→Пq_{68}	0Пq_{68}	*Пq_{64}	ЛПq_{68}	НПq_{68}	ППq_{68}	aПq_{68}	kПq_{69}	mПq_{68}	qПq_{68}
q_{64}	λПq_{64}		,Пq_{64}	→Пq_{64}	0Пq_{64}	1Пq_{64}	ЛПq_{64}	НПq_{64}	ППq_{64}	aПq_{64}	kПq_{65}	mПq_{64}	qПq_{64}
q_{65}	*Лq_{66}	*Пq_{65}	*Лq_{66}	*Лq_{66}	*Лq_{66}	*Лq_{66}	*Лq_{66}	*Лq_{66}	*Лq_{66}	*Лq_{66}		*Лq_{66}	*Лq_{66}
q_{66}		*Лq_{66}									kЛq_{67}		
q_{67}	λЛq_{67}	*Пq_{63}	,Лq_{67}	→Лq_{67}	0Лq_{67}	1Лq_{67}	ЛЛq_{67}	НЛq_{67}	ПЛq_{67}	aЛq_{67}		mЛq_{67}	qЛq_{67}
q_{68}	λПq_{68}		,Пq_{68}	→Пq_{68}	0Пq_{68}	1Пq_{68}	ЛПq_{68}	НПq_{68}	ППq_{68}	aПq_{68}	kПq_{69}	mПq_{68}	qПq_{68}
q_{69}	λЛq_{70}	*Пq_{69}	,Лq_{70}	→Лq_{70}	0Лq_{70}	1Лq_{70}	ЛЛq_{70}	НЛq_{70}	ПЛq_{70}	aЛq_{70}	kЛq_{70}	mЛq_{70}	qЛq_{70}
q_{70}		1Лq_{70}									kЛq_{71}		
q_{71}	λЛq_{71}	1Лq_{72}	,Лq_{71}	→Лq_{71}	0Лq_{71}	1Лq_{71}	ЛЛq_{71}	НЛq_{71}	ПЛq_{71}	aЛq_{71}		mЛq_{71}	qЛq_{71}
q_{72}	λПq_{18}	1Лq_{72}	,Пq_{18}	→Пq_{18}	0Пq_{18}	1Пq_{18}	ЛПq_{18}	НПq_{18}	ППq_{18}	aПq_{18}		mПq_{18}	qПq_{18}
q_{73}	λПq_{76}		,Пq_{76}	→Пq_{76}	0Пq_{76}	*Пq_{74}	ЛПq_{76}	НПq_{76}	ППq_{76}	aПq_{76}	kПq_{76}	mПq_{75}	qПq_{76}
q_{74}	λПq_{74}		,Пq_{74}	→Пq_{74}	0Пq_{74}	1Пq_{74}	ЛПq_{74}	НПq_{74}	ППq_{74}	aПq_{74}	kПq_{74}	mПq_{77}	qПq_{74}
q_{75}	λЛq_{81}	*Пq_{75}	,Лq_{81}	→Лq_{81}	0Лq_{81}	1Лq_{79}	ЛЛq_{81}	НЛq_{81}	ПЛq_{81}	aЛq_{81}	kЛq_{81}		qЛq_{81}
q_{76}	λПq_{76}		,Пq_{76}	→Пq_{76}	0Пq_{76}	1Пq_{76}	ЛПq_{76}	НПq_{76}	ППq_{76}	aПq_{76}	kПq_{76}	mПq_{75}	qПq_{76}
q_{77}	λЛq_{79}	*Пq_{77}	,Лq_{79}	→Лq_{79}	0Лq_{79}	*Лq_{78}	ЛЛq_{79}	НЛq_{79}	ПЛq_{79}	aЛq_{79}	kЛq_{79}		qЛq_{79}
q_{78}		*Лq_{78}										mЛq_{80}	
q_{79}		1Лq_{79}										mЛq_{82}	
q_{80}	λЛq_{80}	*Пq_{73}	,Лq_{80}	→Лq_{80}	0Лq_{80}	1Лq_{80}	ЛЛq_{80}	НЛq_{80}	ПЛq_{80}	aЛq_{80}	kЛq_{80}		qЛq_{80}

Продолжение таблицы А.1

	λ	*	,	→	0	1	л	н	п	a	k	m	q
q_{81}		1Лq_{81}										mЛq_{84}	
q_{82}	λЛq_{82}	1Лq_{83}	,Лq_{82}	→Лq_{82}	0Лq_{82}	1Лq_{82}	лЛq_{82}	нЛq_{82}	пЛq_{82}	aЛq_{82}	kЛq_{82}		qЛq_{82}
q_{83}	0Нq_{30}	1Лq_{83}	0Нq_{30}	0Нq_{30}	0Нq_{30}		0Нq_{30}	0Нq_{30}	0Нq_{30}	0Нq_{30}	0Нq_{30}		0Нq_{30}
q_{84}	λЛq_{84}	1Лq_{85}	,Лq_{84}	→Лq_{84}	0Лq_{84}	1Лq_{84}	лЛq_{84}	нЛq_{84}	пЛq_{84}	aЛq_{84}	kЛq_{84}		qЛq_{84}
q_{85}	1Нq_{30}	1Лq_{85}	1Нq_{30}	1Нq_{30}	1Нq_{30}		1Нq_{30}	1Нq_{30}	1Нq_{30}	1Нq_{30}	1Нq_{30}		1Нq_{30}
q_{86}	λПq_{89}		,Пq_{89}	→Пq_{89}	0Пq_{89}	*Пq_{87}	лПq_{89}	нПq_{89}	пПq_{89}	aПq_{89}	kПq_{88}	mПq_{89}	qПq_{89}
q_{87}	λПq_{87}		,Пq_{87}	→Пq_{87}	0Пq_{87}	1Пq_{87}	лПq_{87}	нПq_{87}	пПq_{87}	aПq_{87}	kПq_{90}	mПq_{87}	qПq_{87}
q_{88}	λЛq_{94}	*Пq_{88}	,Лq_{94}	→Лq_{94}	0Лq_{94}	1Лq_{92}	лЛq_{94}	нЛq_{94}	пЛq_{94}	aЛq_{94}		mЛq_{94}	qЛq_{94}
q_{89}	λПq_{89}		,Пq_{89}	→Пq_{89}	0Пq_{89}	1Пq_{89}	лПq_{89}	нПq_{89}	пПq_{89}	aПq_{89}	kПq_{88}	mПq_{89}	qПq_{89}
q_{90}	λЛq_{92}	*Пq_{90}	,Лq_{92}	→Лq_{92}	0Лq_{92}	*Лq_{91}	лЛq_{92}	нЛq_{92}	пЛq_{92}	aЛq_{92}		mЛq_{92}	qЛq_{92}
q_{91}		*Лq_{91}									kЛq_{93}		
q_{92}		1Лq_{92}									kЛq_{95}		
q_{93}	λЛq_{93}	*Пq_{86}	,Лq_{93}	→Лq_{93}	0Лq_{93}	1Лq_{93}	лЛq_{93}	нЛq_{93}	пЛq_{93}	aЛq_{93}		mЛq_{93}	qЛq_{93}
q_{94}		1Лq_{94}									kЛq_{97}		
q_{95}	λЛq_{95}	1Лq_{96}	,Лq_{95}	→Лq_{95}	0Лq_{95}	1Лq_{95}	лЛq_{95}	нЛq_{95}	пЛq_{95}	aЛq_{95}		mЛq_{95}	qЛq_{95}
q_{96}	0Нq_{34}	1Лq_{96}	0Нq_{34}	0Нq_{34}	0Нq_{34}		0Нq_{34}	0Нq_{34}	0Нq_{34}	0Нq_{34}		0Нq_{34}	0Нq_{34}
q_{97}	λЛq_{97}	1Лq_{98}	,Лq_{97}	→Лq_{97}	0Лq_{97}	1Лq_{97}	лЛq_{97}	нЛq_{97}	пЛq_{97}	aЛq_{97}		mЛq_{97}	qЛq_{97}
q_{98}	1Нq_{34}	1Лq_{98}	1Нq_{34}	1Нq_{34}	1Нq_{34}		1Нq_{34}	1Нq_{34}	1Нq_{34}	1Нq_{34}		1Нq_{34}	1Нq_{34}
q_{99}	λЛq_{100}		,Лq_{100}	→Лq_{100}	0Лq_{100}	1Пq_{99}	лЛq_{100}	нЛq_{100}	пЛq_{100}	aЛq_{100}	kЛq_{100}	mЛq_{100}	qЛq_{100}
q_{100}						*Пq_{101}							
q_{101}	1Пq_{102}		,Пq_{101}	→Пq_{101}	0Пq_{101}	1Пq_{101}	лПq_{101}	нПq_{101}	пПq_{101}	aПq_{101}	kПq_{101}	mПq_{101}	qПq_{101}
q_{102}	0Пq_{103}												
q_{103}	0Пq_{104}												
q_{104}	0Пq_{105}												
q_{105}	0Пq_{106}												
q_{106}	0Нq_{107}												
q_{107}		*Лq_{108}	,Лq_{107}	→Лq_{107}	0Лq_{107}	1Лq_{107}	лЛq_{107}	нЛq_{107}	пЛq_{107}	aЛq_{107}	kЛq_{107}	mЛq_{107}	qЛq_{107}
q_{108}		*Лq_{108}				*Пq_{109}							,Пq_{111}
q_{109}	0Пq_{110}	*Пq_{109}	,Пq_{109}	→Пq_{109}	0Пq_{109}	1Пq_{109}	лПq_{109}	нПq_{109}	пПq_{109}	aПq_{109}	kПq_{109}	mПq_{109}	qПq_{109}
q_{110}	0Нq_{107}												
q_{111}		,Пq_{111}	,Нq_{37}	→Нq_{37}	0Нq_{37}	1Нq_{37}	лНq_{37}	нНq_{37}	пНq_{37}	aНq_{37}	kНq_{37}	mНq_{37}	qНq_{37}
q_{112}	λЛq_{113}		,Лq_{113}	→Лq_{113}	0Лq_{113}	1Пq_{112}	лЛq_{113}	нЛq_{113}	пЛq_{113}	aЛq_{113}	kЛq_{113}	mЛq_{113}	qЛq_{113}
q_{113}						*Пq_{114}							
q_{114}	1Пq_{115}		,Пq_{114}	→Пq_{114}	0Пq_{114}	1Пq_{114}	лПq_{114}	нПq_{114}	пПq_{114}	aПq_{114}	kПq_{114}	mПq_{114}	qПq_{114}
q_{115}	0Пq_{116}												
q_{116}	0Пq_{117}												
q_{117}	0Пq_{118}												
q_{118}	0Нq_{119}												
q_{119}		*Лq_{120}	,Лq_{119}	→Лq_{119}	0Лq_{119}	1Лq_{119}	лЛq_{119}	нЛq_{119}	пЛq_{119}	aЛq_{119}	kЛq_{119}	mЛq_{119}	qЛq_{119}
q_{120}		*Лq_{120}				*Пq_{121}				,Пq_{123}			

Продолжение таблицы А.1

	λ	*	,	→	0	1	Л	Н	П	a	k	m	q
q_{121}	0Пq_{122}	*Пq_{121}	,Пq_{121}	→Пq_{121}	0Пq_{121}	1Пq_{121}	ЛПq_{121}	НПq_{121}	ППq_{121}	aПq_{121}	kПq_{121}	mПq_{121}	qПq_{121}
q_{122}	0Нq_{119}												
q_{123}		,Пq_{123}	,Нq_{38}	→Нq_{38}	0Нq_{38}	1Нq_{38}	ЛНq_{38}	ННq_{38}	ПНq_{38}	aНq_{38}	kНq_{38}	mНq_{38}	qНq_{38}
q_{124}	λЛq_{125}		,Лq_{125}	→Лq_{125}	0Лq_{125}	1Пq_{124}	ЛЛq_{125}	НЛq_{125}	ПЛq_{125}	aЛq_{125}	kЛq_{125}	mЛq_{125}	qЛq_{125}
q_{125}						*Пq_{126}							
q_{126}	1Пq_{127}		,Пq_{126}	→Пq_{126}	0Пq_{126}	1Пq_{126}	ЛПq_{126}	НПq_{126}	ППq_{126}	aПq_{126}	kПq_{126}	mПq_{126}	qПq_{126}
q_{127}	0Пq_{128}												
q_{128}	0Пq_{129}												
q_{129}	0Пq_{130}												
q_{130}	0Нq_{131}												
q_{131}		*Лq_{132}	,Лq_{131}	→Лq_{131}	0Лq_{131}	1Лq_{131}	ЛЛq_{131}	НЛq_{131}	ПЛq_{131}	aЛq_{131}	kЛq_{131}	mЛq_{131}	qЛq_{131}
q_{132}		*Лq_{132}				*Пq_{133}				,Пq_{135}			
q_{133}	0Пq_{134}	*Пq_{133}	,Пq_{133}	→Пq_{133}	0Пq_{133}	1Пq_{133}	ЛПq_{133}	НПq_{133}	ППq_{133}	aПq_{133}	kПq_{133}	mПq_{133}	qПq_{133}
q_{134}	0Нq_{131}												
q_{135}		,Пq_{135}	,Нq_{40}	→Нq_{40}	0Нq_{40}	1Нq_{40}	ЛНq_{40}	ННq_{40}	ПНq_{40}	aНq_{40}	kНq_{40}	mНq_{40}	qНq_{40}
q_{136}	1Пq_{137}		,Пq_{136}	→Пq_{136}	0Пq_{136}	1Пq_{136}	ЛПq_{136}	НПq_{136}	ППq_{136}	aПq_{136}	kПq_{136}	mПq_{136}	qПq_{136}
q_{137}	0Лq_{145}												
q_{138}	1Пq_{139}		,Пq_{138}	→Пq_{138}	0Пq_{138}	1Пq_{138}	ЛПq_{138}	НПq_{138}	ППq_{138}	aПq_{138}	kПq_{138}	mПq_{138}	qПq_{138}
q_{139}	0Пq_{140}												
q_{140}	0Лq_{145}												
q_{141}	1Пq_{142}		,Пq_{141}	→Пq_{141}	0Пq_{141}	1Пq_{141}	ЛПq_{141}	НПq_{141}	ППq_{141}	aПq_{141}	kПq_{141}	mПq_{141}	qПq_{141}
q_{142}	0Пq_{143}												
q_{143}	0Пq_{144}												
q_{144}	0Лq_{145}												
q_{145}		,Пq_{41}	,Лq_{145}	→Лq_{145}	0Лq_{145}	1Лq_{145}	ЛЛq_{145}	НЛq_{145}	ПЛq_{145}	aЛq_{145}	kЛq_{145}	mЛq_{145}	qЛq_{145}
q_{146}	λЛq_{147}		,Лq_{147}	→Лq_{147}	0Лq_{147}	1Пq_{146}	ЛЛq_{147}	НЛq_{147}	ПЛq_{147}	aЛq_{147}	kЛq_{147}	mЛq_{147}	qЛq_{147}
q_{147}						*Пq_{148}							
q_{148}	1Пq_{149}		,Пq_{148}	→Пq_{148}	0Пq_{148}	1Пq_{148}	ЛПq_{148}	НПq_{148}	ППq_{148}	aПq_{148}	kПq_{148}	mПq_{148}	qПq_{148}
q_{149}	0Пq_{150}												
q_{150}	0Пq_{151}												
q_{151}	0Пq_{152}												
q_{152}	0Пq_{153}												
q_{153}	0Нq_{154}												
q_{154}		*Лq_{155}	,Лq_{154}	→Лq_{154}	0Лq_{154}	1Лq_{154}	ЛЛq_{154}	НЛq_{154}	ПЛq_{154}	aЛq_{154}	kЛq_{154}	mЛq_{154}	qЛq_{154}
q_{155}		*Лq_{155}				*Пq_{156}							,Пq_{158}
q_{156}	0Пq_{157}	*Пq_{156}	,Пq_{156}	→Пq_{156}	0Пq_{156}	1Пq_{156}	ЛПq_{156}	НПq_{156}	ППq_{156}	aПq_{156}	kПq_{156}	mПq_{156}	qПq_{156}
q_{157}	0Нq_{154}												
q_{158}		,Пq_{158}	,Нq_{42}	→Нq_{42}	0Нq_{42}	1Нq_{42}	ЛНq_{42}	ННq_{42}	ПНq_{42}	aНq_{42}	kНq_{42}	mНq_{42}	qНq_{42}