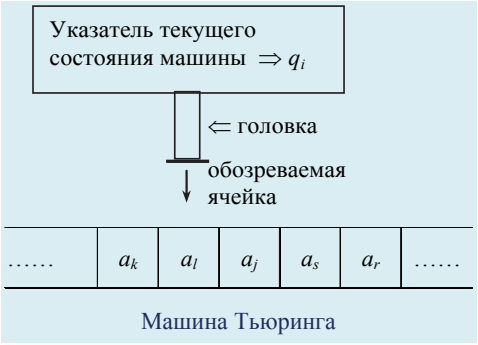
6.5. МАШИНА ТЬЮРИНГА

Определение машины Тьюринга



Машина Тьюринга

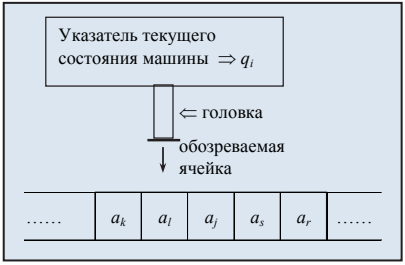
Машина Тьюринга – некоторое устройство, которое состоит из следующих узлов:

– *управляющее устройство*, которое может находиться в одном из состояний из конечного множества $Q = \{q_1, \ldots, q_n\}$;

– *лента, разбитая на ячейки*, в каждой из которой может быть записан ровно один символ конечного *алфавита* $A = \{a_1, \ldots, a_m\}$;

– *устройство обращения к ленте*, т.е. считывающая/пишущая головка.

Память машины Тьюринга – это конечная последовательность состояний, в которых может находиться конкретная машина (*внутренняя память*) и лента (*внешняя память*).



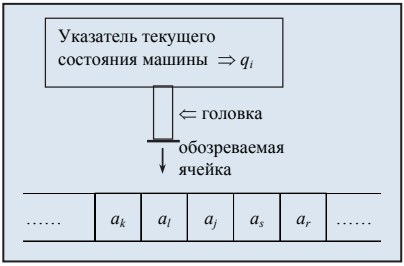
Различают несколько разновидностей машин Тьюринга:

- с лентой, потенциально бесконечной в обе стороны;
- с лентой, бесконечной в одном из направлений.

Различные варианты машин взаимно сводимы.

Во множестве состояний Q обязательно должны быть два состояния:

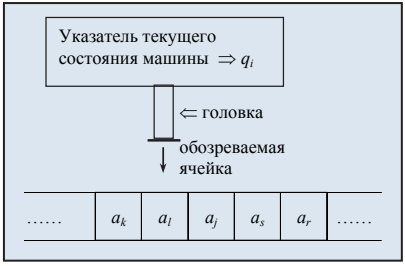
- *начальное состояние* q_1
- *конечное состояние* q_0



В алфавите обязательно присутствует *пустой символ*, который может быть обозначен символами **λ**, **0** или некоторыми другими.

Остальные элементы множества состояний и алфавита каждой машины Тьюринга определяются функцией, которую она вычисляет, и методом, которым она это делает.

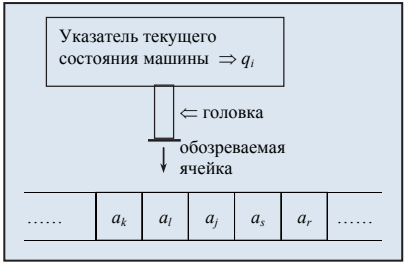
Машина Тьюринга ***функционирует*** следующим образом:



- головка ***считывает символ***, находящийся в очередной ячейке ленты («обозреваемая ячейка», будем ее помечать на рисунках символом ↓ (стрелка вниз);
- в зависимости от прочитанного символа и текущего состояния машины головка ***записывает символ*** в обозреваемую ячейку (новый символ, совпадающий с прежним или пустой);
- ***головка сдвигается*** на одну ячейку вправо или влево либо остается на месте (можно считать, что лента сдвигается соответственно на одну ячейку влево/вправо или остается на месте);
- управляющее устройство ***переходит в новое состояние*** (возможно совпадающее с предыдущим).

Функционирование машины прекращается, если очередное состояние машины заключительное, т.е.

$$q_0.$$



Перед началом работы машина находится в начальном состоянии, т.е.

$$q_1.$$

Элементарные шаги машины Тьюринга:

- считывание/запись символов алфавита;
- сдвиг головки на ячейку влево/вправо;
- переход управляющего устройства в следующее состояние.

Команды машины Тьюринга записываются следующим образом:

$$q_i a_j \longrightarrow q_i' a_j' \alpha_k$$

где q_i – текущее состояние машины; a_j – текущий обозреваемый символ; q_i' – следующее состояние машины; a_j' – символ, который в результате выполнения команды должен быть записан в ячейке вместо символа a_j; $\alpha_k \in \{L, R, E\}$ – направление движения головки (L – влево, R – вправо, E – на месте).

Для обеспечения *детерминированности* потребуем, чтобы

- *для любой пары i и j ($i \neq 0$) в системе команд машины имелось не более одной команды с левой частью $q_i a_j$.*
- *в левой части команд не встречалось состояние q_0.*

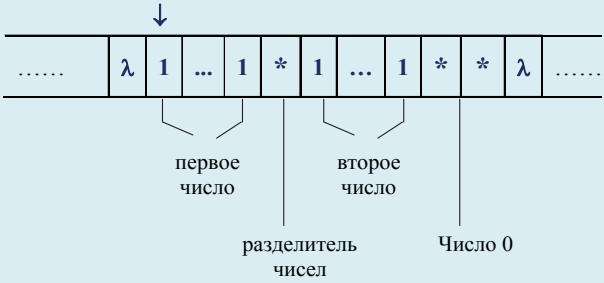
Результатом работы машины Тьюринга является записанное на ленте слово.

Чтобы обеспечить возможность работы машины Тьюринга с числовыми данными, предположим, что один из элементов алфавита A есть **1**, а другой – ***** (символ-разделитель).

Тогда число x можно представить непрерывной последовательностью из x единиц, как показано на рисунке (в тексте будем обозначать это слово как 1^x).

Вариант представления чисел на ленте машины Тьюринга

Два соседних числа на начальном этапе изучения машин Тьюринга будем отделять друг от друга разделителем *. Чтобы отличить число 0 от пустой ленты, будем выделять разделителем ноль ячеек.


……
λ
1
...
1
*
1
…
1
*
*
λ
……
первое
число
второе
число
разделитель
чисел
Число 0

!Результат вычислений зависит от начального положения пишущей/записывающей головки.

Пример 6.7. Дана машина Тьюринга с алфавитом $A = \{\lambda, a, b, c\}$, где λ – символ пустой ячейки и со следующей системой команд:

Состояния управляющего устройства	**Символы алфавита**			
	λ	***a***	***b***	***c***
q_1	$q_2\lambda R$	q_2bR	-	-
q_2	q_0aE	q_2cR	-	-

Входное слово	**Ячейка, обозревая в начальном состоянии**	**Процесс вычислений**	**Выходное слово (результат)**
aaa	1-я слева: ***a***aa	$q_1aaa \Rightarrow bq_2aa \Rightarrow bcq_2a \Rightarrow bccq_2\lambda \Rightarrow bccq_0a$	*bcca*
aaa	1-я справа: aa***a***	$aaq_1a \Rightarrow aabq_2\lambda \Rightarrow aabq_0a$	*aaba*

Можно использовать сокращенную форму записи команд

Форма записи		Примечание
обычная	сокращенная	
$q_1\lambda \to q_2\lambda R$	$q_1\lambda \to q_2 R$	символ λ не заменяется
$q_1\lambda \to q_1 a R$	$q_1\lambda \to a R$	состояние q_1 не заменяется
$q_1 a \to q_1 a R$	$q_1 a \to R$	Символ a и состояние q_1 не заменяются

Вычислимые по Тьюрингу функции

Функция, вычислимая машиной Тьюринга – функция определяется следующим образом. Пусть вычисления начинаются в состоянии q_1 с первоначально обозреваемой ячейки (↓). Тогда значение функции $f(x)$ равно общему числу вхождений символа 1 в обозреваемое слово на ленте в заключительный момент, если вычисление заканчивается (машина на каком-то шаге работы приходит в состояние q_0); в противном случае значение функции $f(x)$ считается неопределенным.

Функция ***$f(x)$* является частичной**, т.е. определенной не для всех значений аргумента x.

Аналогично определяется *n-местная функция* (также в общем случае частичная) $f(x_1, \ldots, x_n)$, вычислимая машиной Тьюринга, подсчетом числа единиц на ленте в заключительном слове, если машина начинает работу в состоянии q_1 и считывает помеченную ячейку на ленте.

Для того чтобы формально определить конкретную машину Тьюринга, необходимо задать тройку конечных множеств:

1) состояний $Q = \{q_1, \ldots, q_n\}$,
2) символов алфавита $A = \{a_1, \ldots, a_m\}$
3) команд $\{q_i a_j \Rightarrow q_i' a_j' \alpha_k\}$.

Функцию f называется ***правильно вычислимой по Тьюрингу***, если существует правильно вычисляющая ее машина T (*правильно вычисляющая функцию f машина Тьюринга начинает вычисления в стандартной начальной конфигурации $q_1 a_r$ и заканчивает вычисления в стандартной заключительной конфигурации $q_0 a_l$, обозревая число, равное значению функции f*).

Например, не является правильно вычисляющей машина с алфавитом $A = \{1, \lambda\}$, состояниями $\{q_0, q_1\}$ и системой из двух команд

$$q_1 1 \to q_1 1R,\ q_1\lambda \to q_1 1R,$$

так как она из любой начальной конфигурации будет работать бесконечно, заполняя всю ленту вправо от начальной ячейки единицами.

Машина Тьюринга зациклится, т.е. также не будет правильно работающей, если в процессе ее функционирования устройство управления придет в некоторое состояние q_i, для которого в системе команд есть пара команд вида

$$q_i 1 \to q_i \lambda E,\ q_i\lambda \to q_i\lambda E.$$

В этом случае, по сравнению с предыдущим примером, неправильность работы является «замаскированной», т.е. в зависимости от остальных команд и исходного слова машина может и не оказаться в состоянии q_i.

Таким образом, всякой правильно работающей машине Тьюринга соответствует вычисляемая ею функция и наоборот. В связи с этим можно ввести понятие *эквивалентных машин Тьюринга*, правильно вычисляющих одну и ту же функцию, а также определить операции над машинами.

Теорема 6.7. *Если функции* $f_1(x)$ *и* $f_2(x)$ *вычислимы по Тьюрингу, то их суперпозиция* $f_2(f_1(x))$ *также вычислима по Тьюрингу* (общий способ построения такой машины называется *композицией машин Тьюринга*).

Пусть функция $f(x)$ задана следующим описанием:

$$f(x) = \begin{cases} g_1(x), & \text{если } P(x) \text{ истинно;} \\ g_2(x), & \text{если } P(x) \text{ ложно;} \\ \text{не определено, если не определено } P(x). \end{cases}$$

Такая функция может быть названа *разветвлением* или *условным переходом* к $g_1(x)$ и $g_2(x)$ по условию $P(x)$.

Таким образом, **машины Тьюринга обеспечивают возможность выполнения следующих типов действий**:

– вычисление числовых и нечисловых функций;

– вычислений суперпозиций, т.е. конструирование более сложных функций, исходя из каких-либо простейших;

– условные переходы.

Тезис Тьюринга. *Всякий алгоритм* (*в интуитивном понимании этого слова*) *может быть реализован некоторой машиной Тьюринга.*

Тезис Чёрча. *Каждая вычислимая функция частично рекурсивна*

Тезис Тьюринга имеет точно такой же статус, как и тезис Чёрча, т.е. является не точным строгим знанием, а правдоподобной практически обоснованной гипотезой, связывающей интуитивное понятие алгоритма с формальной алгоритмической моделью в виде машины Тьюринга.

Конструирование машин Тьюринга

Создание машин Тьюринга является более сложным процессом по сравнению с процессом применения команд данной машины к словам.

Пример 6.8. Построим машину Тьюринга, которая складывает два числа a и b.

Согласно веденному для машин Тьюринга представлению числовых данных, сложить два числа – это значит слово $1^a * 1^b$ переработать в слово $1^{(a+b)}$. Это можно сделать, например, с помощью удаления разделителя * и перемещения одного из слагаемых к другому.

Пусть в начальной конфигурации обозревается самая левая единица числа 1^a, а в заключительной – самая левая единица числа 1^{a+b}.

Сложение выполняется следующей системой команд

Состояния управляющего устройства	**Символы алфавита**		
	1	**λ**	*
q_1	$q_2 \lambda R$		$q_0 \lambda R$ (если слово $a = 0$)
q_2	R		$q_3 1 L$
q_3	L	$q_0 R$	

Таким образом, состояние q_2 – это просмотр первого слагаемого, q_3 – возврат к начальному символу результирующего слова. При выбранном методе сложения просмотр второго слагаемого не потребовался.

Пример 6.9. Построим машину Тьюринга $T_{коп}$, которая выполняет операцию копирования слова α в слово $\alpha*\alpha$.

Пусть алфавит машины $T_{коп}$ имеет следующий состав:

$$A = \{0, 1, \lambda, *\}.$$

Предположим, что на ленте бесконечной в обоих направлениях записаны α единиц, при этом обозревается крайняя слева единица, все остальные ячейки ленты заполнены пустыми символами. Пусть в начальной и заключительной конфигурации обозревается самая левая единица числа-оригинала 1^a.

Состояния управляющего устройства	**Символы алфавита**			
	0	**1**	**λ**	*****
q_1		$q_2$0R	q_0λE	
q_2		$q_2$1R	q_3*R	q_3*R
q_3		$q_3$1R	$q_4$1L	
q_4	$q_5$0R	$q_4$1L		q_4*L
q_5		$q_2$0R		q_6*L
q_6	$q_6$1L		q_0λR	
q_0				

Наглядно представить систему команд машины Тьюринга можно с помощью диаграммы переходов машины. На рисунке показана диаграмма для построения машины, при этом:

1) если из q_i в q_j ведут два ребра с одинаковой правой частью, то они объединяются в одно ребро, на котором левые части команд записываются через запятую;

2) если символ на ленте не изменяется, то он в правой части команды на диаграмме не пишется.

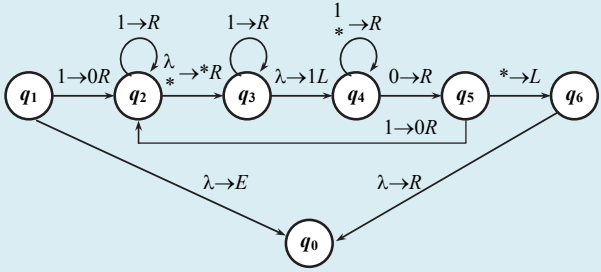


Диаграмма переходов машины Тьюринга $T_{\text{коп}}$

Пример 6.10. Построим машину Тьюринга для вычисления функции $f(x) = 2x$.

Данную машину легко построить как композицию машин T_+ и $T_{\text{коп}}$ из примеров 6.8 и 6.9: $T = T_+(T_{\text{коп}})$.

В качестве алфавита новой машины Тьюринга возьмем объединенный алфавит двух рассматриваемых машин, т.е. $A = \{0, 1, \lambda, *\}$, а в качестве множества состояний машины – объединение соответствующих множеств машин $T_{\text{коп}}$ и T_+.

При этом с учетом того, что в множествах состояний рассматриваемых машин использованы одинаковые обозначения, выполним переименование:

1) состояния q_i для $i = 2, 3, \ldots, 6$ машины $T_{\text{коп}}$ обозначим через $q_{1,i}$, заключительное состояние – через $q_{1,7}$;

2) в множестве состояний машины T_+ обозначение начального состояния заменим на $q_{2,1}$, остальные состояние (за исключением заключительного) обозначим как $q_{2,j}$ ($j = 2, 3$).

Таким образом, получим множество состояний машины Т:

$$Q = \{q_1, q_{1,2}, q_{1,3}, q_{1,4}, q_{1,5}, q_{1,6}, q_{1,7}, q_{2,1}, q_{2,2}, q_{2,3}, q_0\}.$$

Для получения системы команд машины T необходимо объединить команды машин $T_{\text{коп}}$ и T_+ (с учетом выполненных переименований) и добавить команды перехода из заключительного состояния машины $T_{\text{коп}}$ $(q_{1,7})$ в начальное состояние машины $T_+(q_{2,1})$:

$$q_{1,7}0 \to q_{2,1}0E,\ q_{1,7}1 \to q_{2,1}1E,\ q_{1,7}\lambda \to q_{2,1}\lambda E.$$

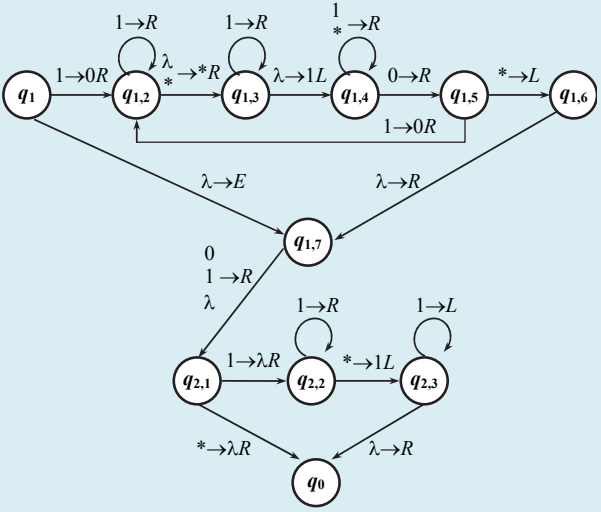
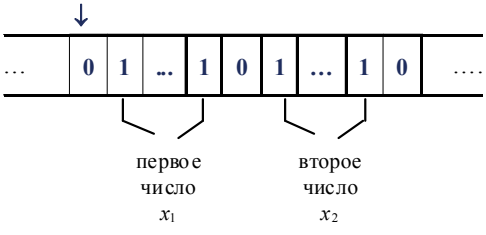


Диаграмма переходов машины Тьюринга $T_{+}(T_{\text{коп}})$,
вычисляющей функцию $f(x) = 2x$

При работе с функциями, заданными на множестве $\boldsymbol{N}\cup\{0\}$ и принимающими также значения из множества $\boldsymbol{N}\cup\{0\}$, в качестве символа-разделителя можно использовать символ 0, который также будет играть роль пустого символа и роль записи числа 0.

Так два числа x_1 и x_2 в унарной системе ($01^{x_1}01^{x_2}$) счисления можно записать на ленте так, как показано на рисунке.



При этом начальная конфигурация машины

$$q_1 01^{x_1} 01^{x_2} \ldots 01^{x_n} 0$$

после вычисления функции $f(x_1, x_2, \ldots, x_n)$ будет такой:

$$q_0 01^{f(x_1,x_2,\ldots,x_n)} 00.$$

Пример 6.11. Составить программу «перенос нуля»: $q_1001^x0 \Rightarrow q_001^x00$. Обозначим эту машину **А**.

В левой колонке таблицы расположим программу, а в правой – результат ее преобразования исходного слова.

Пусть исходным будет слово q_1001^x0.

Команда	**Результат**
$q_10 \to q_20R$	$0q_201^x0$
$q_20 \to q_31R$	$01q_31^x0$
$q_31 \to q_31R$ (x раз)	011^xq_30
$q_30 \to q_40L$	01^xq_410
$q_41 \to q_50E$	01^xq_500
$q_50 \to q_60L$	$01^{x-1}q_6100$
$q_61 \to q_61L$ ($x-1$ раз)	q_601^x00
$q_60 \to q_00E$	q_001^x00

Пример 6.12. Левый сдвиг: $01^x q_1 0 \Rightarrow q_0 01^x 0$. Обозначим эту машину **Б**$^-$.

Команда	Результат
$q_1 0 \to q_2 0 L$	$01^{x-1} q_2 10$
$q_2 1 \to q_2 1 L$ ($x - 1$ раз)	$q_2 01^x 0$
$q_2 0 \to q_0 0 E$	$q_0 01^x 0$

Пример 6.13. Правый сдвиг: $q_1 01^x 0 \Rightarrow 01^x q_0 0$. Обозначим эту машину **Б**$^+$. Программа этой машины получается из предыдущей программы заменой символа L на R. □

Пример 6.14. Транспозиция: $01^x q_1 01^y 0 \Rightarrow 01^y q_0 01^x 0$. Обозначим эту машину **B**.

Сначала переведем слово $01^x q_1 01^y 0$ в слово $01^x q 1^y 00$. При $y > 0$ это достигается следующими действиями: сначала по программе правого сдвига из исходного слова $01^x q_1 01^y 0$ получаем слово $01^x 01^y q_3 0$, затем выполняем следующие команды:

Команда	Результат
$q_3 0 \to q_4 0 L$	$01^x 01^{y-1} q_4 10$
$q_4 1 \to q_5 0 E$	$01^x 01^{y-1} q_5 00$
$q_5 0 \to q_6 0 L$	$01^x 01^{y-2} q_5 100$
$q_6 1 \to q_6 1 L$ (y – 2 раза)	$01^x q_6 01^{y-1} 00$
$q_6 0 \to q_7 1 E$	$01^x q_7 1^y 00$

Чтобы получит слово $01^x q_7 1^y 00$ и при $y = 0$, добавляем команду:

Команда	Результат
$q_4 0 \to q_7 0 E$	$01^x q_7 1^y 00$ ($y \geq 0$)

Теперь из подслова 1^x перебрасываем символ 1 в промежуток между двумя последними нулями. Это достигается следующей программой:

Команда	Результат
$q_7 1 \to q_8 1 L$	$01^{x-1} q_8 11^y 00$
$q_8 1 \to q_9 0 E$	$01^{x-1} q_9 01^y 00$
$q_9 0 \to q_{10} 0 R$	$01^{x-1} 0 q_{10} 1^y 00$
$q_{10} 1 \to q_{10} 1 R$ (y раз)	$01^{x-1} 01^y q_{10} 00$
$q_{10} 0 \to q_{11} 1 E$	$01^{x-1} 01^y q_{11} 10$
$q_{11} 1 \to q_{12} 1 L$	$01^{x-1} 01^{y-1} q_{12} 110$
$q_{12} 1 \to q_{13} 0 E$	$01^{x-1} 01^{y-1} q_{13} 010$
$q_{13} 0 \to q_{14} 0 L$	$01^{x-1} 01^{y-2} q_{14} 1010$
$Q_{14} 1 \to q_{14} 1 L$ ($y-2$ раза)	$01^{x-1} q_{14} 01^{y-1} 010$
$q_{14} 0 \to q_{15} 1 E$	$01^{x-1} q_{15} 1^y 010$

Этот кусок программы работает при $x > 0$, $y > 0$. Чтобы тот же результат получился и при $y = 0$, добавляем к записанной программе еще команды:

Команда	Результат
$q_7 0 \to q_{16} 1E$	$01^x q_{16} 100$
$q_{16} 1 \to q_{17} 1L$	$01^{x-1} q_{17} 1100$
$q_{17} 1 \to q_{15} 0E$	$01^{x-1} 0 q_{15} 1^{y=0} 100$ (или 010)

Теперь зацикливаем программу командами:

Команда	Результат
$q_{15} 1 \to q_7 1E$	$01^{x-1} q_7 1^y 010$ ($y > 0$)
$q_{15} 0 \to q_7 0E$	$01^{x-1} q_7 1^{y=0} 100$ (или 010) ($y = 0$)

Если $x - 1 > 0$, то слово $01^{x-1} q_7 1^y 010$ перерабатывается в слово $01^{x-2} q_7 1^y 0110$. Если же $x - 2 > 0$, то процесс дальнейшей переработки даст слово $01^{x-3} q_7 1^y 01110$ и т.д.

Через x циклом получим слово $0q_7 1^y 01^x 0$.

Команда	Результат
$q_7 1 \to q_8 1L$	$q_8 01^y 01^x 0$
$q_8 0 \to q_{18} 0R$	$0q_{18} 1^y 01^x 0$
$q_{18} 1 \to q_{18} 1R$ (y раз)	$01^y q_{18} 01^x 0$
$q_{18} 0 \to q_0 0E$	$01^y q_0 01^x 0$

Комбинируя между собой полученные машины, можно легко получать машины, выполняющие более сложные преобразования слов. Так, например, применяя дважды подряд программу правого сдвига, получаем машину, выполняющую двойной правый сдвиг:

$$q_1 01^x 01^y 0 \Rightarrow 01^x 01^y q_0 0.$$

Пример 6.15. Программа циклического сдвига (**Ц**):

$$01^x01^y q_1 01^z 0 \Rightarrow 01^z q_0 01^x 01^y 0.$$

Осуществить циклический сдвиг можно с помощью применения друг за другом программы транспозиции, левого сдвига и опять транспозиции: **Ц = ВБ⁻В**. Действительно,

$$01^x01^y q_1 01^z 0 \overset{\text{В}}{\Rightarrow} 01^x01^z q_{19} 01^y 0 \overset{\text{Б}^-}{\Rightarrow} 01^x q_{21} 01^z 01^y 0 \overset{\text{В}}{\Rightarrow} 01^z q_0 01^x 01^y 0.$$

Достаточно научиться строить машины, правильно вычисляющие простейшие функции и функции, возникающие из правильно вычислимых функций с помощью операций суперпозиции, примитивной рекурсии и минимизации.

Пример 6.16. Вычислить функцию $o(x) = 0$, т.е.

$$q_1 01^x 0 \Rightarrow q_0 00^{x+1}.$$

Команда	Результат
$q_1 0 \to q_2 0R$	$0q_2 1^x 0$
$q_2 1 \to q_2 1R$ (x раз)	$01^x q_2 0$
$q_2 0 \to q_3 0L$	$01^{x-1} q_3 10$
$\left.\begin{array}{l} q_3 1 \Rightarrow q_4 0E \\ q_4 0 \Rightarrow q_3 0L \end{array}\right\}$ ($x-2$ раза)	$01^{x-1} q_4 00$ $01^{x-2} q_4 100$
$q_3 0 \to q_0 0E$	$q_0 00^{x+1}$

Машину обозначим через **O**. □

Пример 6.17. Вычислим оператор сдвига $s(x) = x + 1$:

$$q_1 01^x \Rightarrow q_0 01^{x+1}.$$

Команда	Результат
$q_1 0 \to q_2 0R$	$0q_2 1^x 0$
$q_2 1 \to q_2 1R$ (x раз)	$01^x q_2 0$
$q_2 0 \to q_3 1E$	$01^x q_3 10$
$q_3 1 \to q_3 1L$ ($x + 1$ раза)	$q_3 01^{x+1} 0$
$q_3 0 \to q_0 0E$	$q_0 01^{x+1}$

Пример 6.18. Построить машину Тьюринга, правильно вычисляющую функцию $I_m^n(x_1,...,x_n) = x_m \quad (1 \le m \le n)$.

Рассмотрим сначала более простой случай: $I_2^2(x_1, x_2) = x_2$. Исходное слово $q_1 01^{x_1} 01^{x_2} 0$, необходимо получить конечное слово $q_1 01^{x_2}$. Обратимся к рассмотренным ранее примерам. Рассмотрим машины: **Б**$^-$, **Б**$^+$, **В** и **О** (левого и правого сдвигов, транспозиции и вычисления нуль-функции, соответственно), а также циклического сдвига **Ц = ВБ**$^-$**В**.

Тогда для $I_2^2(x_1, x_2) = x_2$ будем иметь $q_1 01^{x_1} 01^{x_2} 0$ $(\mathbf{Б}^+)_1$ $01^{x_1} q_3 01^{x_2} 0$ $(\mathbf{В})_3$ $01^{x_2} q_{21} 01^{x_1} 0$ $(\mathbf{О})_{21}$ $01^{x_2} q_{25} 0 \ldots 0$ $(\mathbf{Б}^-)_{25}$ $q_{27} 01^{x_2} 0$ или сокращенно **Б**$^+$**ВОБ**$^-$.

Эта итоговая программа представляем собой произведение четырех машин Тьюринга. Конечная команда $q_{27} = q_0$. Нижние индексы программ считаются следующим образом: последняя команда q_0 предыдущей программы является первой командой следующей, при этом учитывается число состояний предыдущей машины Тьюринга.

Для машины, реализующей $I_m^n(x_1,...,x_n) = x_m$, начальное слово

$$q_1 01^{x_1} 01^{x_2} \ldots 01^{x_n} 0 \Rightarrow q_0 01^{x_m} 0.$$

Воспользуемся программой циклического сдвига **Ц = ВБ⁻В,**

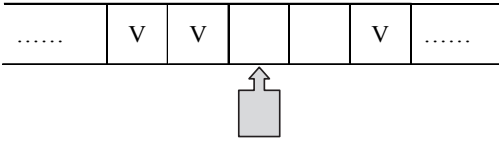
$$q_1 01^{x_1} 01^{x_2} \ldots 01^{x_n} 0 \Rightarrow q_0 01^{x_2} 01^{x_3} \ldots 01^{x_n} 01^{x_1} 0.$$

Если применить ее $(m-1)$ раз, то получим слово $q_0 01^{x_m} 01^{x_{m+1}} \ldots 01^{x_n} 01^{x_1} \ldots 01^{x_{m-1}} 0$, затем применим к этому слову $(n-1)$ раз программу правого сдвига, получим $01^{x_m} 01^{x_{m+1}} \ldots 01^{x_n} 01^{x_1} \ldots\ q_0 01^{x_{m-1}} 0$. Наконец, применим $(n-1)$ раз произведение программ вычисления нуль-функции и левого сдвига (**ОБ$^{-1}$**). В результате будем иметь: $q_0 01^{x_m} 0$. Итак, машина, правильно вычисляющая функцию $I_m^n(x_1,...,x_n) = x_m$, такова

$$(\mathbf{Ц})^{m-1}(\mathbf{Б}^+)^{n-1}(\mathbf{ОБ}^{-1})^{n-1}.$$

МАШИНА ПОСТА

Машина Поста состоит из *ленты* и *каретки* (называемой также считывающей и записывающей головкой). Лента бесконечна в обоих направлениях и разделена на ячейки.



В каждый момент времени каретка указывает на одну из ячеек.

В каждой ячейке ленты может быть либо ничего не записано, либо стоять метка V.

Информация о том, какие ячейки пусты, а какие содержат метки, образует *состояние ленты*. Иными словами, состояние ленты – это распределение меток по ячейкам.

Состояние ленты меняется в процессе работы машины.

Наличие метки в ячейке можно интерпретировать как “1”, а отсутствие – “0”. Такое двоичное представление информации подобно представлению, используемому практически во всех современных ЭВМ.

Каретка может передвигаться вдоль ленты влево и вправо. Когда она неподвижна, она стоит напротив ровно одной ячейки ленты (говорят, что *каретка обозревает одну ячейку*).

За единицу времени каретка может совершить одно из трех действий:

- стереть метку,
- поставить метку,
- совершить движение на соседнюю ячейку.

Состояние машины Поста складывается из состояния ленты и положения каретки.

Действия каретки подчинены программе, состоящей из **пронумерованного** набора команд.

Команды бывают шести типов:

- записать 1 (метку), перейти к *i*-й строке программы;
- записать 0 (стереть метку), перейти к *i*-й строке программы;
- сдвиг влево, перейти к *i*-й строке программы;
- сдвиг вправо, перейти к *i*-й строке программы;
- останов;
- если 0, то перейти к *i*, иначе перейти к *j*.

Недопустимые действия, ведущие к аварийной остановке машины:

- попытка записать 1 (метку) в заполненную ячейку;
- попытка стереть метку в пустой ячейке;
- бесконечное выполнение группы команд.

Команды машины будем обозначать следующим образом:

Обозначение команды	Действие, выполняемое командой
$\rightarrow$	Шаг вправо
$\leftarrow$	Шаг влево
V	Записать отметку
X	Стереть отметку
? *a*; *b*	Просмотреть ячейку: если в ячейке находится 0, то перейти на команду с номером *a*, иначе на команду с номером *b*
!	Останов

Будем говорить, что мы можем применить программу к текущему состоянию машины Поста, если выполнение программы не приведет к зацикливанию, т.е. рано или поздно мы выполним команду останов.

Пример программы, которая не применима ни к одному состоянию машины Поста:

1. $\rightarrow 1$
2. !

Рассмотрим задачу для машины Поста и ее решение.

Задача. На ленте проставлена метка в единственной ячейке. Каретка стоит на некотором расстоянии левее этой ячейки. Необходимо подвести каретку к ячейке, стереть метку и остановить каретку слева от этой ячейки.

Решение. Сначала попробуем описать алгоритм обычным языком. Поскольку нам известно, что каретка стоит напротив пустой ячейки, но неизвестно, сколько шагов нужно совершить до пустой ячейки, мы можем сразу сделать шаг вправо; проверить, заполнена ли ячейка; если она пустая, то повторять эти действия до тех пор, пока не наткнемся на заполненную ячейку. Как только мы ее найдем, мы выполним операцию стирания, после чего нужно будет лишь сместить каретку влево и остановить выполнение программы.

Программа для машины Поста:

1. → 2
2. ? 1; 3
3. X 4
4. ← 5
5. !

→	Шаг вправо
←	Шаг влево
V	Записать отметку
X	Стереть отметку
? *a*; *b*	Просмотреть ячейку: если в ячейке находится 0, то перейти на команду с номером *a*, иначе на команду с номером *b*
!	Останов

Рассмотрим еще несколько задач

Пояснения к условиям задач

1. В задачах под *массивом* понимается последовательность подряд идущих меток, ограниченная пустыми ячейками.

2. Если в задаче говорится, что на ленте задано число в унарной системе, то имеется в виду, что натуральное число n закодировано с помощью массива длины n.

3. В задачах при описании начального состояния ленты будем указывать то, что записано начиная с самой левой непустой ячейки и заканчивая самой правой непустой ячейкой. При этом будем использовать следующие обозначения:

n подряд идущих меток будем обозначать 1^n,

m пустых ячеек – 0^m.

При обозначении одной заполненной или пустой ячейки будем писать просто 1 или 0, соответственно.

К примеру, запись “$1^2 0 1^2$” будет соответствовать записи “11011” на ленте.

4. Если не сказано ничего о местонахождении каретки в начальный момент времени, то будем считать, что каретка обозревает ячейку с самой левой меткой.

1. Применимость программ. Определение результата выполнения программ

1. Выяснить, применимы ли программа к заданным состояниям машины Поста, указать результат работы машины Поста для каждого состояния.

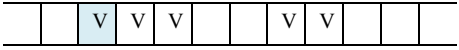
1. ? 3; 2
2. $\rightarrow$ 1
3. $\rightarrow$ 4
4. ? 6; 5
5. $\leftarrow$ 1
6. $\rightarrow$ 7
7. ? 8; 9
8. !
9. $\rightarrow$ 4

1. ? 3; 2
2. → 1
3. → 4
4. ? 6; 5
5. ← 1
6. → 7
7. ? 8; 9
8. !
9. → 4

Начальные состояния ленты:

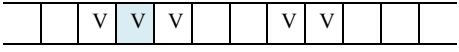
1) $1^3 0^2 1^2$
2) $1^5 0 1^2$
3) 1^6

Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2.



Команда 2. Сдвинуть каретку вправо. Перейти к команде 1.

Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2.



Команда 2. Сдвинуть каретку вправо. Перейти к команде 1.

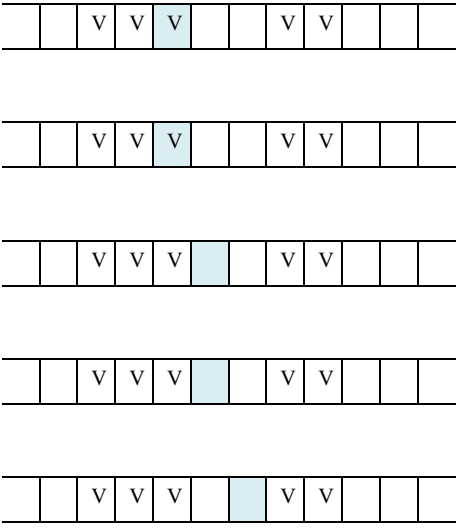
Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2.

Команда 2. Сдвинуть каретку вправо. Перейти к команде 1.

Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2.

Команда 3. Сдвинуть каретку вправо. Перейти к команде 4.

Команда 4. Если метки нет, то перейти к команде 6. Иначе к 5.



1. ? 3; 2
2. → 1
3. → 4
4. ? 6; 5
5. ← 1
6. → 7
7. ? 8; 9
8. !
9. → 4

Команда 6. Сдвинуть каретку вправо. Перейти к команде 7.

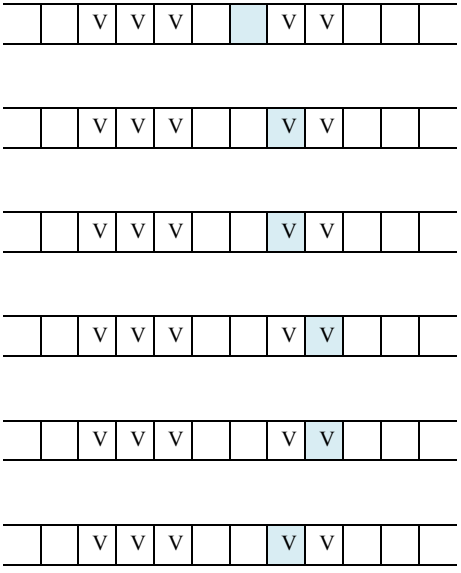
Команда 7. Если метки нет, то перейти к команде 8. Иначе к 9.

Команда 9. Сдвинуть каретку вправо. Перейти к команде 4.

Команда 4. Если метки нет, то перейти к команде 6. Иначе к 5.

Команда 5. Сдвинуть каретку влево. Перейти к команде 1.

Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2.



1. ? 3; 2
2. → 1
3. → 4
4. ? 6; 5
5. ← 1
6. → 7
7. ? 8; 9
8. !
9. → 4

Команда 2. Сдвинуть каретку вправо. Перейти к команде 1

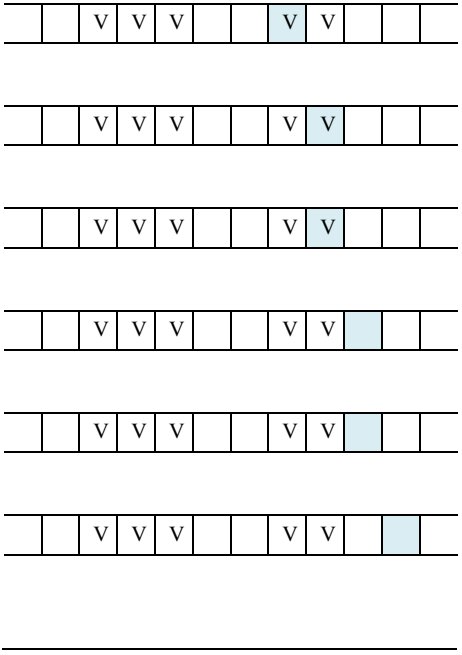
Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2

Команда 2. Сдвинуть каретку вправо. Перейти к команде 1

Команда 1. Если метки нет, то перейти к команде 3. Иначе к 2

Команда 3. Сдвинуть каретку вправо. Перейти к команде 4.

Команда 4. Если метки нет, то перейти к команде 6. Иначе к 5.

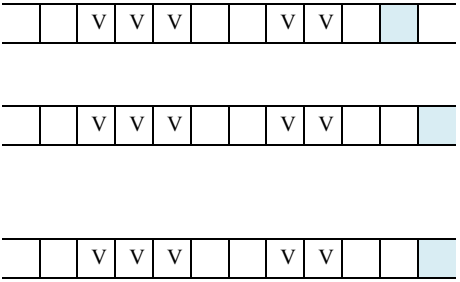


1. ? 3; 2
2. → 1
3. → 4
4. ? 6; 5
5. ← 1
6. → 7
7. ? 8; 9
8. !
9. → 4

Команда 6. Сдвинуть каретку вправо. Перейти к команде 7.

Команда 7. Если метки нет, то перейти к команде 8. Иначе к 9.

Команда 8. Стоп.



1. ? 3; 2
2. → 1
3. → 4
4. ? 6; 5
5. ← 1
6. → 7
7. ? 8; 9
8. !
9. → 4

1) Ответ: 1110011000
2) Ответ: зацикливание
3) Ответ 1001011000

2. Арифметические задачи

2. На ленте задан массив меток. Увеличить длину массива на 1 метку. Каретка находится либо слева от массива, либо над одной из ячеек самого массива

Решение

1. ? 2; 3 (с помощью команд 1 и 2 передвигаем каретку к массиву)
2. $\rightarrow$ 1
3. $\rightarrow$ 4 (команды 3 и 4 – передвигаем каретку к концу массива)
4. ? 5; 3
5. V 6 (ставим 1 метку в конце массива).
6. !

3. Даны два массива меток, которые находятся на некотором расстоянии друг от друга. Требуется соединить их в один массив. Каретка находится над крайней левой меткой массива.

1. X 2
2. → 3
3. ? 4; 2
4. V 5
5. → 6
6. ? 8; 7
7. !
8. ← 9
9. ? 10; 8
10. → 1