

Абстрактные вычислительные машины

Машина Тьюринга

Материал взят с ресурса [Планета информатики](#)

Машина Тьюринга - абстрактный исполнитель (абстрактная вычислительная машина). Была предложена Аланом Тьюрингом в 1936 году для формализации понятия алгоритма.

Машина Тьюринга является расширением конечного автомата и, согласно тезису Чёрча - Тьюринга, способна имитировать всех исполнителей (с помощью задания правил перехода), каким-либо образом реализующих процесс пошагового вычисления, в котором каждый шаг вычисления достаточно элементарен.

То есть, всякий интуитивный алгоритм может быть реализован с помощью некоторой машины Тьюринга.

Описание

Машина Тьюринга состоит из бесконечной в обе стороны ленты, разделенной на ячейки, и автомата (головки), которая управляется программой.

Программы для машин Тьюринга записываются в виде таблицы, где первые столбец и строка содержат буквы внешнего алфавита и

возможные внутренние состояния автомата (внутренний алфавит). Содержимое таблицы представляет собой команды для машины Тьюринга. Буква, которую считывает головка в ячейке (над которой она находится в данный момент), и внутренне состояние головки определяют, какую команду нужно выполнить. Команда определяется пересечением символов внешнего и внутреннего алфавитов в таблице.

Чтобы задать конкретную машину Тьюринга, требуется описать для нее следующие составляющие:

1. Внешний алфавит. Конечное множество (например, **A**), элементы которого называются буквами (символами). Одна из букв этого алфавита (например, **Q0**) должна представлять собой пустой символ.
2. Внутренний алфавит. Конечное множество состояний головки (автомата). Одно из состояний (например, **Q1**) должно быть начальным (запускающим программу). Еще одно из состояний (**Q0**) должно быть конечным (завершающим программу) – состояние останова.
3. Таблица переходов. Описание поведения автомата (головки) в зависимости от состояния и считанного символа.

Автомат машины Тьюринга в процессе своей работы может выполнять следующие действия:

1. Записывать символ внешнего алфавита в ячейку (в том числе и пустой), заменяя находившийся в ней (в том числе и пустой).

2. Передвигаться на одну ячейку влево или вправо.
3. Менять свое внутреннее состояние.

Одна команда для машины Тьюринга как раз и представляет собой конкретную комбинацию этих трех составляющих: указаний, какой символ записать в ячейку (над которой стоит автомат), куда передвинуться и в какое состояние перейти. Хотя команда может содержать и не все составляющие (например, не менять символ, не передвигаться или не менять внутреннего состояния).

Пример

Пример работы машины Тьюринга

Допустим, на ленте есть слово, состоящее из символов **#**, **\$**, **1** и **0**. Требуется заменить все символы **#** и **\$** на нули. В момент запуска головка находится над первой буквой слова слева. Завершается программа тогда, когда головка оказывается над пустым символом после самой правой буквы слова.

Примечание: длина слова и последовательность символов значения не имеют. На рисунке приводится пример последовательности выполнения команд для конкретного случая. Если на ленте будет другое слово, то и последовательность выполнения команд будет другой. Несмотря на это, данная программа для машины Тьюринга (на рисунке – таблица слева) применима к любым словам описанного внешнего алфавита (соблюдается свойство применимости алгоритма ко всем однотипным задачам – массовость).

Пример работы машины Тьюринга

Последовательность
выполнения команд для
частного случая

Изменения на ленте

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

В ячейке 1.

Не менять символ, сдвинуться вправо,
не менять состояние.

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

В ячейке #.

Записать ноль, сдвинуться вправо,
не менять состояние.

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

В ячейке S.

Записать ноль, сдвинуться вправо,
не менять состояние.

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

В ячейке 1.

Не менять символ, сдвинуться вправо,
не менять состояние.

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

В ячейке 0.

Не менять символ, сдвинуться вправо,
не менять состояние.

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

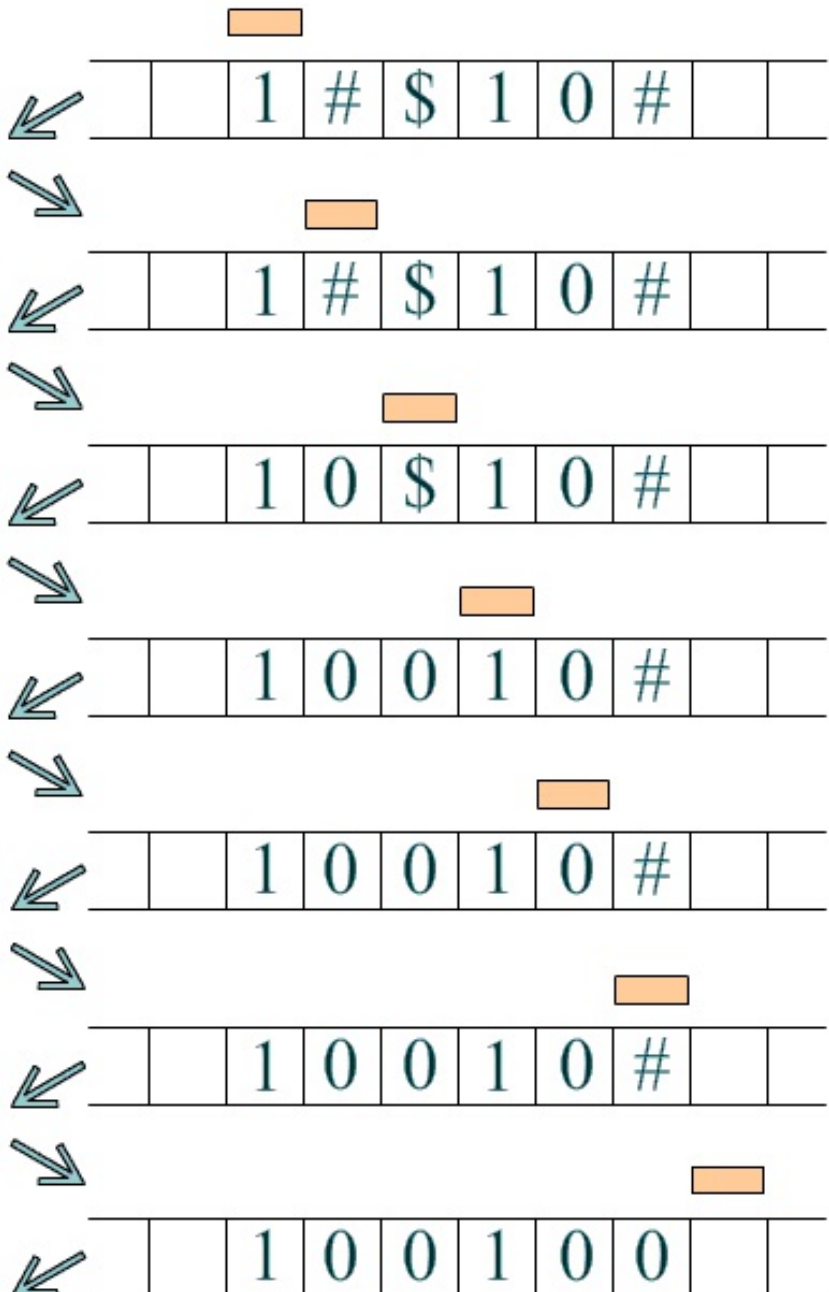
В ячейке #.

Записать ноль, сдвинуться вправо,
не менять состояние.

	#	S	0	1	a ₀
q ₁	0 →	0 →	→	→	q ₀

В ячейке пустота.

Ничего не записывать, стоять на месте,
перейти в состояние **останова**.



Можно усложнить программу. Допустим, головка располагается не

обязательно над первым, а над любым символом слова. Тогда программа для данной машины Тьюринга может быть такой (а могла бы быть и другой):

<http://inf1.info>

Пример программы для машины Тьюринга

	#	\$	0	1	a ₀
q ₁	←	←	←	←	→ q ₂
q ₂	0 →	0 →	→	→	q ₀

Здесь происходит сдвиг головки влево до тех пор, пока она не окажется над пустым символом. После этого машина переходит в состояние **q₂** (команды которого совпадают с командами **q₁** предыдущей программы).

Задачи

Материал взят с ресурса [Планета информатики](#)

Практические задания

1. На ленте машины Тьюринга содержится последовательность символов **+**. Напишите программу для машины Тьюринга,

которая каждый второй символ $+$ заменит на $-$. Замена начинается с правого конца последовательности. Автомат в состоянии $Q1$ обозревает один из символов указанной последовательности. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.

2. Дано число n в восьмеричной системе счисления. Разработать машину Тьюринга, которая увеличивала бы заданное число n на 1 . Автомат в состоянии $Q1$ обозревает некую цифру входного слова. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.
3. Дана десятичная запись натурального числа $n > 1$. Разработать машину Тьюринга, которая уменьшала бы заданное число n на 1 . Автомат в состоянии $Q1$ обозревает правую цифру числа. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.
4. Дано натуральное число $n > 1$. Разработать машину Тьюринга, которая уменьшала бы заданное число n на 1 , при этом в выходном слове старшая цифра не должна быть 0 . Например, если входным словом было 100 , то выходным словом должно быть 99 , а не 099 . Автомат в состоянии $Q1$ обозревает правую цифру числа. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.
5. Дан массив из открывающих и закрывающих скобок. Построить машину Тьюринга, которая удаляла бы пары взаимных скобок,

т.е. расположенных подряд $()$. Например, дано $) (() (()$, надо получить $) . . . (($. Автомат в состоянии Q_1 обозревает крайний левый символ строки. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.

6. Дана строка из букв a и b . Разработать машину Тьюринга, которая переместит все буквы a в левую, а буквы b - в правую части строки. Автомат в состоянии Q_1 обозревает крайний левый символ строки. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.
7. На ленте машины Тьюринга находится число, записанное в десятичной системе счисления. Умножить это число на 2 . Автомат в состоянии Q_1 обозревает крайнюю левую цифру числа. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.
8. Даны два натуральных числа m и n , представленные в унарной системе счисления. Соответствующие наборы символов $|$ разделены пустой клеткой. Автомат в состоянии Q_1 обозревает самый правый символ входной последовательности. Разработать машину Тьюринга, которая на ленте оставит сумму чисел m и n . Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.
9. Даны два натуральных числа m и n , представленных в унарной

системе счисления. Соответствующие наборы символов $|$ разделены пустой клеткой. Автомат в состоянии $Q1$ обозревает самый правый символ входной последовательности. Разработать машину Тьюринга, которая на ленте оставит разность чисел m и n . Известно, что $m > n$. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.

10. На ленте машины Тьюринга находится десятичное число. Определить, делится ли это число на 5 без остатка. Если делится, то записать справа от числа слово **да**, иначе - **нет**. Автомат обозревает некую цифру входного числа. Кроме самой программы-таблицы, описать словами, что выполняется машиной в каждом состоянии.

Решение

Задача 1

	a_0	$+$	$-$
q_1	$a_0 \text{ Л } q_2$	$+$ П q_1	
q_2	$a_0 \text{ Н } q_0$	$+$ Л q_3	
q_3	$a_0 \text{ Н } q_0$	$-$ Л q_2	

В состоянии $Q1$ машина ищет правый конец числа, в состоянии $Q2$ - пропускает знак $+$, при достижении конца последовательности - останавливается. В состоянии $Q3$ машина знак $+$ заменяет на знак $-$, при достижении конца последовательности она останавливается.

Задача 2

	a_0	0	1	2	...	7
q_1	$1 \text{ H } q_0$	$1 \text{ H } q_0$	$2 \text{ H } q_0$	$3 \text{ H } q_0$...	$0 \text{ Л } q_1$

Решение этой задачи аналогично рассмотренному выше примеру.

Задача 3

	a_0	0	1	2	...	8	9
q_1		$9 \text{ Л } q_1$	$0 \text{ H } q_0$	$1 \text{ H } q_0$...	$7 \text{ H } q_0$	$8 \text{ H } q_0$

Состояние **Q1** - уменьшаем младшую (очередную) цифру на **1**. Если она не равна нулю, то после уменьшения сразу - останов, если же младшая цифра равна **0**, то вместо нее пишем **9**, смещаемся влево и вновь выполняем вычитание. В клетку [**A0**, **Q1**] машина Тьюринга никогда не попадет, поэтому ее можно не заполнять.

Задача 4 (усложнение задачи 3)

	a_0	0	1	2	...	8	9
q_1		$9 \text{ Л } q_1$	$0 \text{ Л } q_2$	$1 \text{ H } q_0$...	$7 \text{ H } q_0$	$8 \text{ H } q_0$
q_2	$a_0 \text{ П } q_3$	$0 \text{ H } q_0$	$1 \text{ H } q_0$	$2 \text{ H } q_0$...	$8 \text{ H } q_0$	$9 \text{ H } q_0$
q_3		$a_0 \text{ H } q_0$					

Состояние **Q1** - уменьшаем младшую (очередную) цифру на **1**. Если она больше **1**, то после уменьшения - сразу останов, если же младшая цифра равна **0**, то вместо нее пишем **9**, смещаемся влево и вновь

выполняем вычитание. Если уменьшаемая цифра равна **1**, то вместо нее пишем **0** и переходим в состояние **Q2**.

Состояние **Q2** - после записи **0** в каком-либо разряде надо проанализировать, не является ли этот ноль старшей незначащей цифрой (т.е. не стоит ли слева от него в записи выходного слова **A0**).

Состояние **Q3** - если записанный **0** является старшей незначащей цифрой, то его надо удалить из записи выходного слова.

Те клетки, в которые машина Тьюринга никогда не попадает, оставляем пустыми.

Задача 5

	a_0	()
q_1	$a_0 \text{ H } q_0$	(П q_2) П q_1
q_2	$a_0 \text{ H } q_0$	(П q_2) Л q_3
q_3	$a_0 \text{ H } q_0$	$a_0 \text{ П } q_3$	$a_0 \text{ П } q_1$

Состояние **Q1**: если встретили **(**, то сдвиг вправо и переход в состояние **Q2**; если встретили **A0**, то останов.

Состояние **Q2**: анализ символа **(** на парность, в случае парности должны увидеть **)**. Если парная, то возврат влево и переход в состояние **Q3**.

Состояние **Q3**: стираем сначала **(**, затем **)** и переходим в **Q1**.

Задача 6

Решение этой задачи обычно вызывает у школьников затруднение. При разборе решения этой задачи можно пойти, например, следующим путем.

Рассмотрите со школьниками следующие варианты входных слов и попросите их сформулировать, что должна делать машина Тьюринга, каков внешний вид выходного слова, чем с точки зрения машины Тьюринга эти варианты различаются:

aaa -> выходное слово совпадает с входным, просматриваем входное слово до тех пор, пока оно не заканчивается.

a -> выходное слово совпадает с входным, просматриваем входное слово до тех пор, пока оно не заканчивается.

bbb -> выходное слово совпадает с входным, просматриваем входное слово до тех пор, пока оно не заканчивается.

b -> выходное слово совпадает с входным, просматриваем входное слово до тех пор, пока оно не заканчивается.

ab -> выходное слово совпадает с входным, просматриваем входное слово до тех пор, пока оно не заканчивается.

Результат обсуждения. Машина Тьюринга должна “понимать”, по цепочке каких букв она идет, т.е. у нее должно быть как минимум два

состояния. Пусть состояние **Q1** - движение по цепочке из букв **a**, а **Q2** - состояние движения по цепочке из букв **b**. Заметим, что цепочка может состоять и из одной буквы. Если мы дошли до конца строки в состоянии **Q1** или **Q2**, т.е. встретили **A0**, машина должна остановиться, мы обработали всю строку.

Рассмотрим следующие варианты входных слов:

bba -> **abb**

bbbaab -> **aabbbb**

aabbbaab -> **aaaabbbb**

Результат обсуждения. Первый вариант входного слова можно последовательно обработать следующим образом: **bba** -> **bbb** -> вернуться к левому концу цепочки из букв **b** -> **abb** (заменить первую букву в этой цепочке на **a**). Для выполнения этих действий нам потребуется ввести два новых состояния и, кроме того, уточнить состояние **Q2**. Таким образом, для решения этой задачи нам нужно построить машину Тьюринга со следующими состояниями:

Q1 - идем вправо по цепочке букв **a**. Если цепочка заканчивается **A0**, то переходим в **Q0**; если заканчивается буквой **b**, то переходим в **Q2**;

Q2 - идем вправо по цепочке букв **b**, если цепочка заканчивается **A0**, то переходим в **Q0**; если заканчивается **a**, то заменяем букву **a** на **b**, переходим в состояние **Q3** (цепочку вида заменили на цепочку вида);

Q3 - идем влево по цепочке букв **b** до ее левого конца. Если встретили **A0** или **a**, то переходим в **Q4**;

Q4 - заменяем **b** на **a** и переходим в **Q1** (цепочку вида заменяем на цепочку вида .

	a_0	a	b
q_1	$a_0 \text{ H } q_0$	$a \text{ П } q_1$	$b \text{ П } q_2$
q_2	$a_0 \text{ H } q_0$	$b \text{ Л } q_3$	$b \text{ П } q_2$
q_3	$a_0 \text{ П } q_4$	$a \text{ П } q_4$	$b \text{ Л } q_3$
q_4			$a \text{ П } q_1$

Задача 7

	a_0	0	1	2	3	4	5	6	7	8	9
q_1	$a_0 \text{ Л } q_2$	$0 \text{ П } q_1$	$1 \text{ П } q_1$	$2 \text{ П } q_1$	$3 \text{ П } q_1$	$4 \text{ П } q_1$	$5 \text{ П } q_1$	$6 \text{ П } q_1$	$7 \text{ П } q_1$	$8 \text{ П } q_1$	$9 \text{ П } q_1$
q_2	$a_0 \text{ H } q_0$	$0 \text{ Л } q_2$	$2 \text{ Л } q_2$	$4 \text{ Л } q_2$	$6 \text{ Л } q_2$	$8 \text{ Л } q_2$	$0 \text{ Л } q_3$	$2 \text{ Л } q_3$	$4 \text{ Л } q_3$	$6 \text{ Л } q_3$	$8 \text{ Л } q_3$
q_3	$1 \text{ H } q_0$	$1 \text{ Л } q_2$	$3 \text{ Л } q_2$	$5 \text{ Л } q_2$	$7 \text{ Л } q_2$	$9 \text{ Л } q_2$	$1 \text{ Л } q_3$	$3 \text{ Л } q_3$	$5 \text{ Л } q_3$	$7 \text{ Л } q_3$	$9 \text{ Л } q_3$

состояние **Q1** - поиск правой (младшей) цифры числа;

состояние **Q2** - умножение очередной цифры числа на **2** без прибавления **1** переноса;

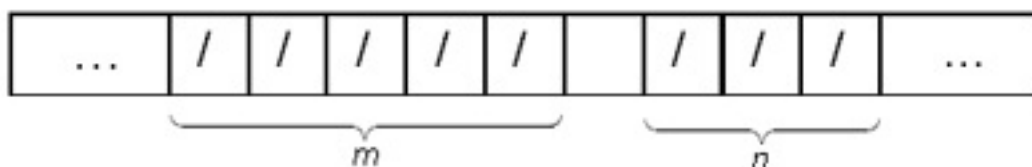
состояние **Q3** - умножение очередной цифры числа на **2** с прибавлением **1** переноса.

Задача 8

Машина Тьюринга для этой программы выглядит тривиально просто - в ней всего одно состояние. Такая машина Тьюринга выполняет следующие действия: стирает самый правый штрих, ищет разделитель (пустую ячейку) и в эту пустую ячейку помещает штрих, тем самым сформирована непрерывная последовательность штрихов длины $n + m$.

	a_0	/
q_1	/ H q_0	a_0 Л q_1

Однако, как ни странно, решение этой задачи вызывает большие трудности. Очень часто ученики строят машину Тьюринга, которая выполняет циклические действия: последовательно пододвигают правые n штрихов к левым.



В этом случае их программа выглядит следующим образом:

	a_0	/
q_1	$a_0 \text{ H } q_2$	/ Л q_1
q_2	/ П q_2	a_0 П q_3
q_3	$a_0 \text{ H } q_0$	/ Л q_2

состояние **Q1** - поиск разделителя;

состояние **Q2** - передвинули штрих;

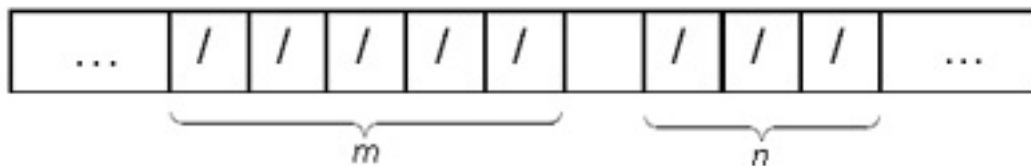
состояние **Q3** - проверка на конец (все ли штрихи передвинули).

На примере этой задачи четко видно, как часто дети пытаются решить задачу уже знакомыми способами. Мне кажется, что, предлагая ученикам задачи на составление машин Тьюринга, мы развиваем способность к нахождению необычных решений, развиваем способность творчески думать!

Задача 9

Эта задача кажется школьникам достаточно легкой, но трудности возникают с остановом машины Тьюринга. Ниже приведен один из возможных вариантов машины Тьюринга для этой задачи.

Идея решения (условие останова). На ленте есть два исходных массива штрихов.



Штрихи начинаем стирать с левого конца массива **m**. И поочередно стираем самый левый штрих в массиве **m** и самый правый штрих в массиве **n**. Но прежде чем стереть правый штрих в массиве **n**, проверяем, единственный он (т.е. последний, который надо стереть) или нет.

Опишем сначала состояния машины Тьюринга, которые необходимы для решения нашей задачи, а затем составим программу-таблицу.

Состояние **Q1** - поиск разделителя между массивами штрихов при движении справа налево;

состояние **Q2** - поиск левого штриха в массиве **m**;

состояние **Q3** - удаление левого штриха в массиве **m**;

состояние **Q4** - поиск разделителя при движении слева направо;

состояние **Q5** - поиск правого штриха в массиве **n**;

состояние **Q6** - проверка единственности этого штриха в массиве **n**, т.е. определяем, был ли он последним;

состояние **q7** - если он был последним, то останов, иначе переход на новый цикл выполнения алгоритма.

	a_0	$/$
q_1	$a_0 \text{ Л } q_2$	$/ \text{ Л } q_1$
q_2	$a_0 \text{ П } q_3$	$/ \text{ Л } q_2$
q_3		$a_0 \text{ П } q_4$
q_4	$a_0 \text{ П } q_5$	$/ \text{ П } q_4$
q_5	$a_0 \text{ Л } q_6$	$/ \text{ П } q_5$
q_6		$a_0 \text{ Л } q_7$
q_7	$a_0 \text{ Н } q_0$	$/ \text{ Л } q_1$

Задача 10

При решении этой задачи следует обратить внимание на правильное выписывание алфавита:

$$A = \{ A0, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, Д, А, Н, Е, Т \}.$$

	a_0	0	1	2	...	5	...	9	Д	А	Н	Е	Т
q_1	$a_0 \text{ П } q_2$	$0 \text{ П } q_1$	$1 \text{ П } q_1$	$2 \text{ П } q_1$		$9 \text{ П } q_1$					
q_2		$0 \text{ П } q_3$	$1 \text{ П } q_5$	$2 \text{ П } q_3$		$5 \text{ П } q_3$		$9 \text{ П } q_5$					
q_3	$Д \text{ П } q_4$												
q_4	$А \text{ Н } q_0$												
q_5	$Н \text{ П } q_6$												
q_6	$Е \text{ П } q_7$												
q_7	$Т \text{ Н } q_0$												

Состояние **Q1** - поиск правого конца числа;

состояние **Q2** - анализ младшей цифры числа; если она равна **0** или **5**, т.е. число делится на **5**, то переход в состояние **Q3**, иначе переход в состояние **Q5**;

состояние **Q3** - запись буквы **Д** справа от слова на ленте;

состояние **Q4** - запись буквы **А** справа от слова и останов машины;

состояние **Q5** - запись буквы **Н** справа от слова;

состояние **Q6** - запись буквы **Е** справа от слова;

состояние **Q7** - запись буквы **Т** справа от слова и останов машины.

Машина Поста

Машина Поста - абстрактная вычислительная машина, предложенная Эмилем Постом в 1936 году, создана независимо от машины Тьюринга, но сообщение о машине Поста опубликовано на несколько месяцев позднее. Отличается от машины Тьюринга большей простотой, при том обе машины алгоритмически “эквивалентны” и обе разработаны для формализации понятия алгоритма и решения задач об алгоритмической разрешимости, то есть, демонстрации алгоритмического решения задач в форме последовательности команд для машины Поста.

Принцип работы

Машина Поста состоит из каретки (или считывающей и записывающей головки) и разбитой на ячейки бесконечной в обе стороны ленты. Каждая ячейка ленты может находиться в 2 состояниях - быть либо пустой - **0**, либо помеченной меткой **1**. За такт работы машины каретка может сдвинуться на одну позицию влево или вправо, считать, изменить символ в своей текущей позиции.

Работа машины Поста определяется программой, состоящей из конечного числа строк. Для работы машины нужно задать программу и её начальное состояние (то есть состояние ленты и позицию каретки). Кареткой управляет программа, состоящая из пронумерованных не

обязательно упорядоченных строк команд, если в каждой команде указана строка, на которую нужно перейти. Обычно принимается, что если в команде переход не указан, то переход происходит на следующую строку. Каждая команда имеет следующий синтаксис:

```
i. K j
```

где **i** - номер команды, **K** - действие каретки, **j** - номер следующей команды (отсылка).

Всего для машины Поста существует шесть типов команд:

- **V j** - поставить метку, перейти к **j**-й строке программы;
- **X j** - стереть метку, перейти к **j**-й строке;
- **<- j** - сдвинуться влево, перейти к **j**-й строке;
- **-> j** - сдвинуться вправо, перейти к **j**-й строке;
- **? j1; j2** - если в ячейке нет метки, то перейти к **J1**-й строке программы, иначе перейти к **J2**-й строке;
- **!** - конец программы (“стоп”, останов).

В команде «стоп» переход на следующую строку не указывается.

После программы запуска возможны варианты:

- работа может закончиться невыполнимой командой (стирание несуществующей метки или запись в помеченное поле);
- работа может закончиться командой “стоп”;
- работа никогда не закончится.
-

Пример

Для сложения и вычитания натуральных (целых неотрицательных) чисел P и Q их можно представить на ленте набором из P единиц и Q , отделённых друг от друга одним нулём; пусть исходное положение каретки находится на крайней левой 1 группы единиц Q :

```
001111101111000
```

Сложение двух чисел тривиально - достаточно поставить 1 между числами и стереть одно крайнее правое 1 у представления Q .

Программа вычитания таких чисел состоит из последовательного изменения крайних левых 1 у представления Q и правых 1 у представления P . В начале программы каретка установлена на крайнюю левую 1 у Q :

1. $<-$ - шаг влево
2. $? 1; 3$ - если в ячейке пусто, перейти к 1 -шагу, если нет - к 3
3. X - удалить метку
4. $->$ - шаг вправо
5. $? 4; 6$ - если в ячейке пусто, перейти к 4 -шагу, если нет - к 6
6. X - удалить метку
7. $->$ - шаг вправо
8. $? 9; 1$ - если в ячейке пусто, перейти к 9 шагу, если нет - к 1
9. $!$ - конец

В 5 -й строке возможно заикливание, если $Q > P$.