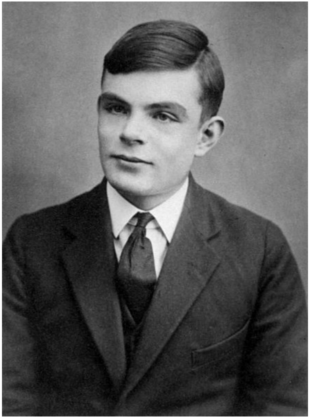
6.5. МАШИНА ТЬЮРИНГА



А́лан Мэ́тисон Тью́ринг
англ. Alan Mathison Turing
(23.06.1912 – 07.06.1954)

Английский математик, логик, криптограф, оказавший существенное влияние на развитие информатики. Кавалер Ордена Британской империи (1945), член Лондонского королевского общества (1951).

Предложенная им в 1936 г. абстрактная вычислительная «Машина Тьюринга», которую можно считать моделью компьютера общего назначения, позволила формализовать понятие алгоритма и до сих пор используется во множестве теоретических и практических исследований.

Научные труды А. Тьюринга – общепризнанный вклад в основания информатики (и, в частности, – теории искусственного интеллекта).

Во время Второй мировой войны Алан Тьюринг работал в Правительственной школе кодов и шифров. Он разработал ряд методов взлома, в том числе теоретическую базу для Bombe – машины, использованной для взлома немецкого шифратора Enigma.

После войны Тьюринг работал в Национальной физической лаборатории, где по его проекту был реализован первый в мире компьютер с хранимой в памяти программой – ACE.

В 1950 году предложил эмпирический тест Тьюринга для оценки искусственного интеллекта компьютера.

В честь учёного названа Премия Тьюринга – самая престижная в мире награда в области информатики. По традиции, лауреат премии Тьюринга при вручении её выступает с докладом, именуемым «Тьюринговская лекция». В этой лекции обычно идёт речь о тех вопросах компьютерной науки, теории и практики использования вычислительной техники, которые лауреат считает достаточно важными, чтобы поделиться своим мнением о них с как можно большим числом специалистов.

На предыдущей лекции были рассмотрены следующие требования к моделям алгоритма:

1. Алгоритм **работает с данными** – применяется к исходным и промежуточным данным и выдает результаты.

Необходимо указывать требования, которым должны удовлетворять данные, чтобы алгоритм мог работать с ними.

2. Данные для своего размещения **требуют памяти.**

Должны быть согласованы единицы измерения данных и организации памяти.

3. Алгоритм состоит из отдельных **элементарных шагов**.

Необходимо определить систему выполняемых алгоритмом действий.

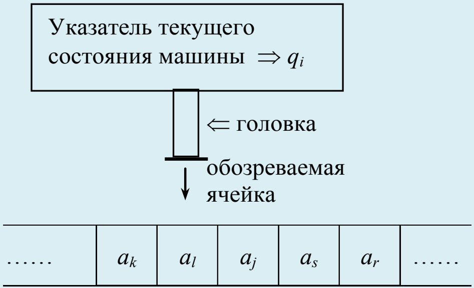
4. Последовательность шагов алгоритма **детерминирована**.

После каждого шага либо должно указываться, какой шаг выполняется дальше, либо должна выдаваться команда остановки, после чего работа алгоритма считается завершенной.

5. Алгоритм должен быть **результативным.**

После конечного числа шагов должна происходить остановка с указанием того, что считать результатом (алгоритм должен сходиться при любых допустимых исходных данных).

Определение машины Тьюринга



Машина Тьюринга

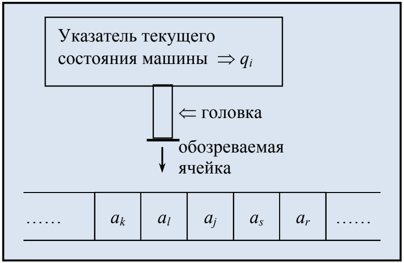
Машина Тьюринга представляет собой некоторое устройство, состоящее из следующих основных узлов:

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

– *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*).



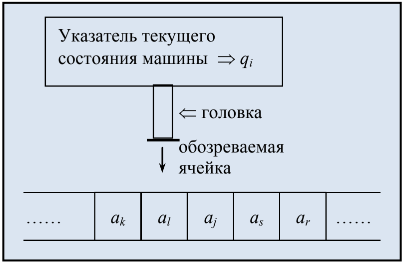
Различают несколько разновидностей машин Тьюринга:

- 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* $\boldsymbol{q_1}$
- *конечное состояние* $\boldsymbol{q_0}$

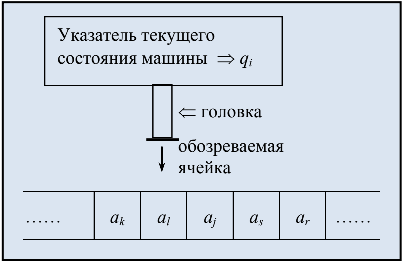


В алфавите обязательно присутствует *пустой символ*, который может быть обозначен символами $\boldsymbol{\lambda}$, **0** или некоторыми другими.

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);



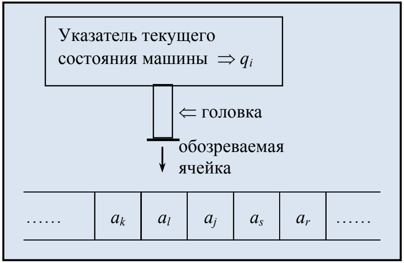
- в зависимости от прочитанного символа и текущего состояния машины головка ***записывает символ*** в обозреваемую ячейку (новый символ, совпадающий с прежним или пустой);

- ***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.

$$q_0.$$

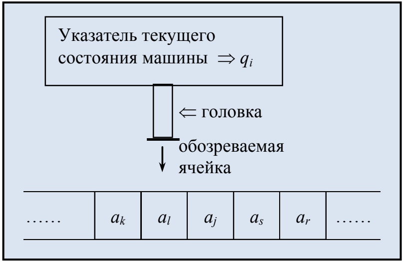


Перед началом работы машина находится в начальном состоянии, т.е.

$$q_1.$$

Данные машины Тьюринга – это записываемые на ленте слова в алфавите A.

Все данные (исходные, промежуточные, заключительные) записываются на ленте, которая выполняет роль *памяти машины*.



Элементарные шаги машины Тьюринга:

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

Детерминированность машины Тьюринга, т.е. однозначность набора последовательности ее шагов, обеспечивается следующим образом: *необходимо, чтобы для любого внутреннего состояния* $\boldsymbol{q_i}$ *и считываемого символа* $\boldsymbol{a_j}$ *однозначно были заданы следующее состояние* $\boldsymbol{q'_i}$, *записываемый вместо* a_j *в ту же ячейку символ* $\boldsymbol{a'_j}$, *направление сдвига головки* $\alpha_k \in \{\boldsymbol{L, R, E}\}$ *(L – влево, R – вправо, E – на месте). Запишем это условие в виде команды*

$$\boldsymbol{q_i a_j \longrightarrow q'_i a'_j \alpha_k}$$

и для обеспечения *детерминированности* потребуем, чтобы

- *для любой пары i и j (i ≠ 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.

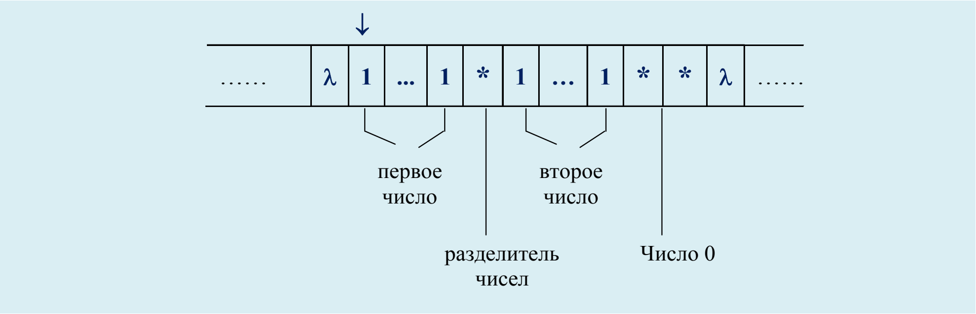
Результатом работы машины Тьюринга является записанное на ленте слово.

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.



Применение машины Тьюринга к словам

Проиллюстрируем на примере все введенные понятия, связанные с машиной Тьюринга.

Пример 6.7. Дана машина Тьюринга с внешним алфавитом

$$A = \{\lambda, a, b, c\},$$

где λ – символ пустой ячейки (лента бесконечная в обоих направления и слева, и справа от заданного слова заполнена символам λ), алфавитом внутренних состояний $Q = \{q_0, q_1, q_2\}$ и со следующей функциональной схемой (программой):

$$q_1\lambda \to q_2\lambda R;\ q_2\lambda \to q_0 aE;$$

$$q_1 a \to q_2 bR;\ q_2 a \to q_2 cR.$$

Посмотрим, в какое слово перерабатывает эта машина слово ***aaa*** при различных положениях пишущей/записывающей головки.

$q_1\lambda \to q_2\lambda R;$

$q_1a \to q_2bR;$

$q_2\lambda \to q_0aE;$

$q_2a \to q_2cR.$

А. Пусть в начальном состоянии обозревается крайняя слева заполненная ячейка:

			↓					
……	λ	λ	a	a	a	λ	λ	……

На первом шаге действует команда $q_1a \to q_2bR$.

В результате создается следующая конфигурация:

				↓				
……	λ	λ	b	a	a	λ	λ	……

На втором шаге действует команда: $q_2a \to q_2cR$ и создается конфигурация:

					↓			
……	λ	λ	b	c	a	λ	λ	……

$$q_1\lambda \to q_2\lambda R;$$
$$q_1 a \to q_2 bR;$$
$$q_2\lambda \to q_0 aE;$$
$$q_2 a \to q_2 cR.$$

На третьем шаге действует та же команда, что и на втором шаге:

						↓		
……	λ	λ	b	c	c	λ	λ	……

На четвертом шаге действует команда: $q_2\lambda \to q_0aE$. В результате выполнения этой команды создается конфигурация:

						↓		
……	λ	λ	b	c	c	a	λ	……

Эта конфигурация является заключительной, потому что машина оказалась в состоянии остановки q_0.

Таким образом, исходное слово ***aaa*** было переработано машиной в слово ***bcca***.

$$q_1\lambda \to q_2\lambda R;$$
$$q_1a \to q_2bR;$$
$$q_2\lambda \to q_0aE;$$
$$q_2a \to q_2cR.$$

Б. Пусть в начальный момент времени обозревается последняя заполненная ячейка:

					↓			
……	λ	λ	a	a	a	λ	λ	……

На следующем шаге получаем конфигурацию:

						↓		
……	λ	λ	a	a	b	λ	λ	……

Очередной шаг машины оказывается заключительным, последняя команда создает конфигурацию:

						↓		
……	λ	λ	a	a	b	a	λ	……

Таким образом, исходное слово ***aaa*** было переработано машиной в слово ***aaba***.

Рассмотренный пример показал, что ***результат вычислений зависит от начального положения пишущей/записывающей головки***.

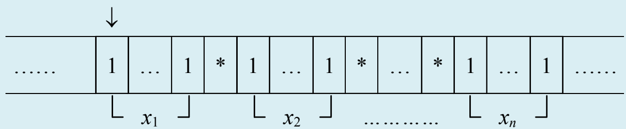
Вычислимые по Тьюрингу функции

Функция, вычислимая машиной Тьюринга, определяется следующим образом.

Пусть вычисления начинаются в состоянии q_1 с первоначально обозреваемой ячейки ($\downarrow$). 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.

Очевидно, такая 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, как показано на рисунке.



Пример записи аргументов n-местной функции 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 $\{q_i a_j \Rightarrow q'_i a'_j \alpha_k\}$.

Функционирование машины – это процесс последовательного преобразования слов на ленте путем перехода из одной конфигурации $q_i a_j$ в другую в соответствии с заданными командами, начиная с начальной конфигурации $q_1 a_r$. Если при этом машина переходит в заключительную конфигурацию $q_0 a_t$, то функционирование конечно, иначе – бесконечно.

В связи с тем, что может быть задано много определяющих машину троек множеств, можно говорить ***о разнообразии машин Тьюринга***.

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 $q_0 a_t$, 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

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

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

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.

Перечисленные действия представляют собой не что иное, как основные виды операций, используемых при практическом построении алгоритмов. В связи с этим не выглядит неожиданным следующий тезис.

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. Обозначим ее $\boldsymbol{T}_+$

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.

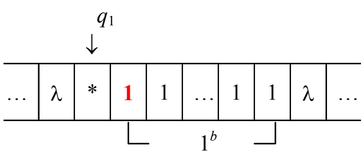
Пусть в начальной конфигурации обозревается самая левая единица числа 1^a, а в заключительной – самая левая единица числа 1^{a+b}.

Сложение двух чисел a и b осуществляется машиной Тьюринга с алфавитом $A = \{1, \lambda\ *\}$, четырьмя состояниями $Q = \{q_0, q_1, q_2, q_3\}$ и следующей системой команд:

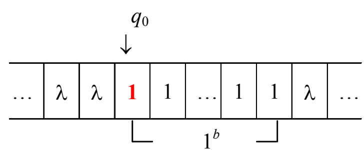
$$\boldsymbol{q_1* \to q_0\lambda R}$$

(команда учитывает случай, когда число $a = 0$ и исходное слово имеет вид $*1^b$; при этом * заменяется на пустой символ λ и головка переводится в первую позицию заключительного слова 1^b; устройство управления переходит в заключительное состояние и машина останавливается);

До выполнения команды



Результат выполнения команды

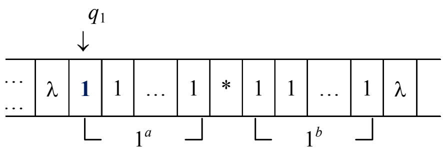


$$\boldsymbol{q_1 1 \rightarrow q_2 \lambda R}$$

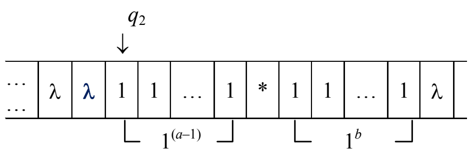
$$\boxed{q_1{*} \rightarrow q_0 \lambda R}$$

(этой командой самая левая единица числа 1^a заменяется пробелом; происходит переход в состояние q_2 и головки сдвигается вправо);

До выполнения команды



Результат выполнения команды



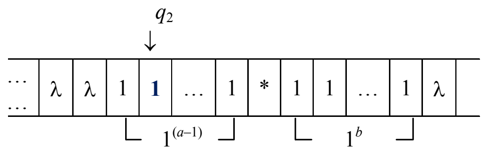
$$q_2 1 \to q_2 1 R$$

(если в состоянии q_2 обозревается символ 1, то головка просто перемещается вправо);

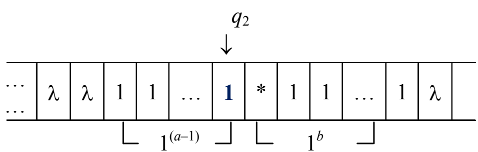
$q_1 * \to q_0 \lambda R$

$q_1 1 \to q_2 \lambda R$

До выполнения команды



Результат выполнения команды (a – 2) раза

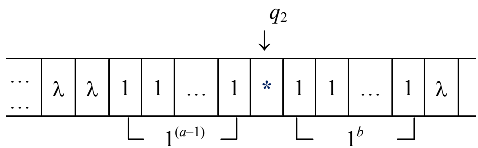


$$q_2* \to q_3 1L$$

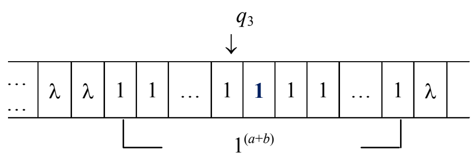
$$q_1* \to q_0\lambda R$$
$$q_1 1 \to q_2\lambda R$$
$$q_2 1 \to q_2 1R$$

(если в состоянии q_2 обозревается символ *, т.е. окончен просмотр числа 1^a, то символ-разделитель * заменяется 1; происходит переход управляющего устройства в новое состояние q_3 и головка сдвигается на одну ячейку влево);

До выполнения команды



Результат выполнения команды



$$q_3 1 \to q_3 1 L$$

(если в состоянии q_3 обозревается символ 1, то головка просто перемещается влево);

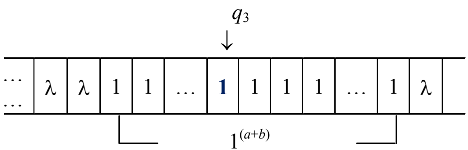
$q_1 * \to q_0 \lambda R$

$q_1 1 \to q_2 \lambda R$

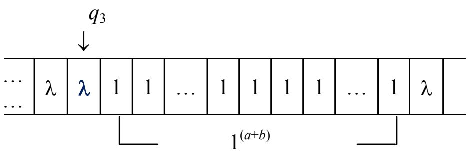
$q_2 1 \to q_2 1 R$

$q_2 * \to q_3 1 L$

До выполнения команды



Результат выполнения команды (a – 1) раз



$$\boldsymbol{q_3\lambda \to q_0\lambda R}$$

(если в состоянии q_3 считан пустой символ λ, т.е. окончен просмотр первого слагаемого 1^a, то осуществляется переход на ячейку вправо и машина останавливается).

$q_1* \to q_0\lambda R$

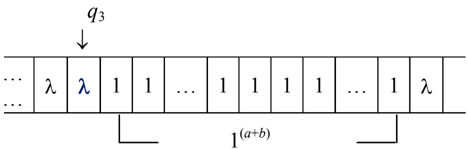
$q_1 1 \to q_2\lambda R$

$q_2 1 \to q_2 1R$

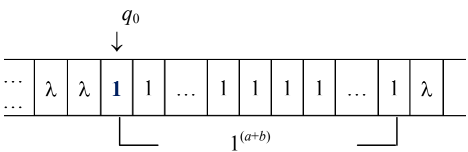
$q_2* \to q_3 1L$

$q_3 1 \to q_3 1L$

До выполнения команды



Результат выполнения команды



Таким образом, система команд машины $\boldsymbol{T}_+$ следующая:

$q_1* \to q_0\lambda R$ $\quad$ $q_2 1 \to q_2 1R$ $\quad$ $q_3 1 \to q_3 1L$

$q_1 1 \to q_2\lambda R$ $\quad$ $q_2* \to q_3 1L$ $\quad$ $q_3\lambda \to q_0\lambda R$

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 $T_{\text{коп}}$, которая выполняет операцию копирования слова α в слово $\alpha*\alpha$.

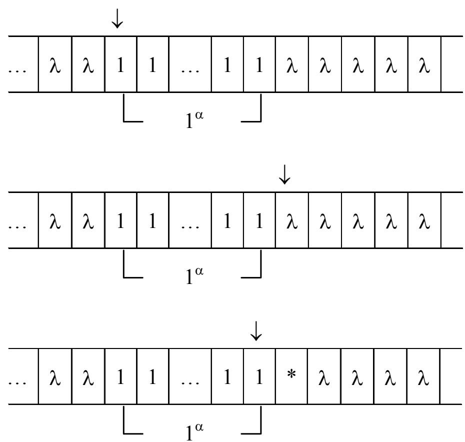
Пусть алфавит машины $T_{\text{коп}}$ имеет следующий состав:

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

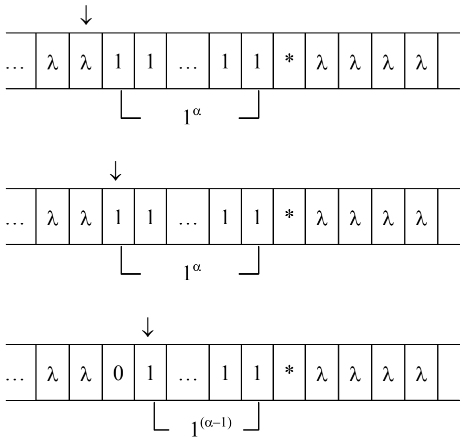
Предположим, что на ленте бесконечной в обоих направлениях записаны α единиц, при этом обозревается крайняя слева единица, все остальные ячейки ленты заполнены пустыми символами. Пусть в начальной и заключительной конфигурации обозревается самая левая единица числа-оригинала 1^a.

Копирование слова 1^{α} (т.е. числа α) выполняется по следующему алгоритму:

1. Вставка в первую пустую ячейка после числа α разделителя * (т.е. определение конца числа α).



2. Замена самой левой единицы слова α нулем для обозначения границы нескопированного участка числа-оригинала.



3. Поиск ближайшей справа пустой ячейки и запись в нее единицы.

							↓					
…	λ	λ	0	1	…	1	1	*	λ	λ	λ	λ
					$1^{(\alpha-1)}$							

								↓				
…	λ	λ	0	1	…	1	1	*	λ	λ	λ	λ
					$1^{(\alpha-1)}$							

									↓			
…	λ	λ	0	1	…	1	1	*	λ	λ	λ	λ
					$1^{(\alpha-1)}$							

								↓				
…	λ	λ	0	1	…	1	1	*	1	λ	λ	λ
					$1^{(\alpha-1)}$							

4. Поиск слева от символа * символа 1.

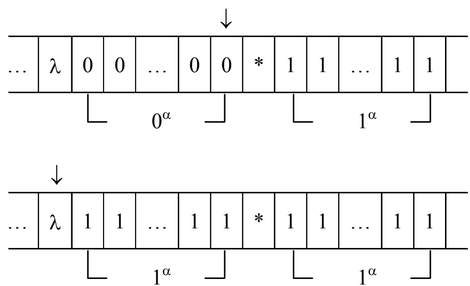
							↓						
…	λ	λ	0	1	…	1	1	*	1	λ	λ	λ	
					$1^{(\alpha-1)}$								

5. Если слева от символа * стоит символ 1, то необходимо перейти к выполнению п. 2.

6. Если слева от символа * стоит ноль (т.е. все единицы, составляющие число α, уже скопированы), то перейти к выполнению п. 7.

						↓							
…	λ	0	0	0	…	0	*	1	1	…	1	1	
				0^{α}						1^{α}			

7. Заменить все нули единицами, двигаясь влево от символа-разделителя *.



8. Если все нули закончились (т.е. обозревается первая пустая ячейка слева от числа-оригинала), то перейти в заключительное состояние и переместиться вправо (для того, чтобы обозреваемой стала первая левая единица числа-оригинала).

С учетом построенного алгоритма распределим действия между состояниями устройства управления:

q_1 – начало работы; анализ обозреваемого начального символа;

q_2 – просмотр числа-оригинала слева направо (правый просмотр); при первом просмотре – установка разделителя *;

q_3 – правый просмотр числа-копии; запись единицы в ближайшую пустую ячейку;

q_4 – проход по числу-копии и числу-оригиналу справа налево до ближайшего нуля; возврат от нуля на одну ячейку вправо;

q_5 – анализ символа – «левого соседа» нуля, считанного в состоянии q_4;

q_6 – восстановление числа-оригинала (замена нулей единицами);

q_0 – заключительное состояние.

Тогда система команд машины Тьюринга $\boldsymbol{T}_{\mathbf{коп}}$ может быть следующей:

$$q_1\lambda \to q_0\lambda E, \qquad q_1 1 \to q_2 0R, \qquad q_2\lambda \to q_3{*}R,$$

$$q_2{*} \to q_3{*}R, \qquad q_2 1 \to q_2 1R, \qquad q_3 1 \to q_3 1R,$$

$$q_3\lambda \to q_4 1L, \qquad q_4 1 \to q_4 1L, \qquad q_4{*} \to q_4{*}L,$$

$$q_4 0 \to q_5 0R, \qquad q_5 1 \to q_2 0R, \qquad q_5{*} \to q_6{*}L.$$

$$q_6 0 \to q_6 1L, \qquad q_6\lambda \to q_0\lambda R,$$

Исходное слово $\quad \dots \mid \lambda \mid \lambda \mid \underbrace{\overset{\downarrow}{1} \mid 1 \mid \dots \mid 1 \mid 1}_{1^{\alpha}} \mid \lambda \mid \lambda \mid \lambda \mid \lambda \mid \lambda \mid$

Результат $\quad \dots \mid \lambda \mid \underbrace{\overset{\downarrow}{1} \mid 1 \mid \dots \mid 1 \mid 1}_{1^{\alpha}} \mid * \mid \underbrace{1 \mid 1 \mid \dots \mid 1 \mid 1}_{1^{\alpha}} \mid \lambda$

Удобной формой представления системы команд машины Тьюринга является таблица. Ниже показано, как система команд машины $\boldsymbol{T}_{\textbf{коп}}$ представлена в таблице.

$$
\begin{array}{lll}
q_1\lambda \to q_0\lambda E, & q_1 1 \to q_2 0R, & q_2\lambda \to q_3{*}R, \\
q_2{*} \to q_3{*}R, & q_2 1 \to q_2 1R, & q_3 1 \to q_3 1R, \\
q_3\lambda \to q_4 1L, & q_4 1 \to q_4 1L, & q_4{*} \to q_4{*}L, \\
q_4 0 \to q_5 0R, & q_5 1 \to q_2 0R, & q_5{*} \to q_6{*}L. \\
q_6 0 \to q_6 1L, & q_6\lambda \to q_0\lambda R, &
\end{array}
$$

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

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.

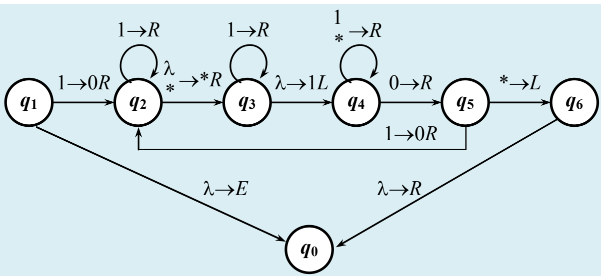


Диаграмма переходов машины Тьюринга $T_{\text{коп}}$

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

Данную машину легко построить как композицию машин T_+ и $T_{коп}$ из примеров 6.8 и 6.9: $\boldsymbol{T_{2x} = T_+(T_{коп})}$.

Процесс вычисления функции $f(x) = 2x$ состоит в том, чтобы сначала машина $T_{коп}$ построила двухкомпонентный вектор (x, x), а затем машина T_+ выполнила операцию сложения двух переменных.

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 $T_{коп}$ 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_+ ($q_{2,1}$):

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

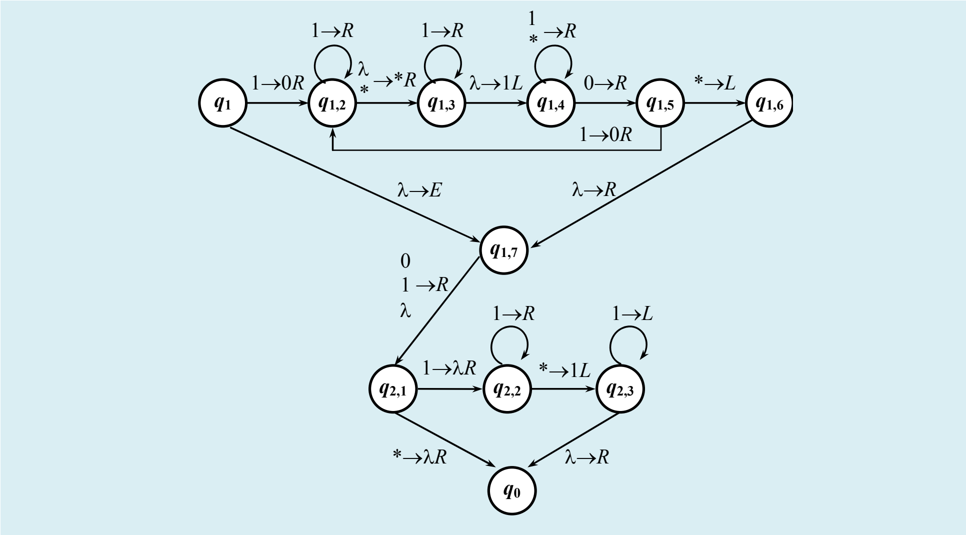


Диаграмма переходов машины Тьюринга $T_{+}(T_{\text{коп}})$,
вычисляющей функцию $f(x) = 2x$

Машина Тьюринга тренажер для изучения универсального исполнителя:

http://kpolyakov.spb.ru/prog/turing.htm

Пример использования тренажера:

https://www.youtube.com/watch?v=clrdEuTX9r8