

Машины тьюринга

Алгоритм является точным математическим понятием. В качестве формальной модели алгоритма исторически первой была модель машины, предложенная англичанином Аланом Тьюрингом в 40-х годах 20-го века. Именно машины Тьюринга получили наибольшее распространение в теоретической математике при изучении свойств алгоритмов. Прежде всего, машины Тьюринга позволили решить многие важные вопросы, связанные с проблемой алгоритмической разрешимости или неразрешимости той или иной проблемы. К этому вопросу мы обратимся несколько ниже.

Машина Тьюринга представляет собой **устройство**, содержащее пишущую ленту бесконечной длины, разбитую на ячейки $Я_1, Я_2, \dots, Я_n, \dots$. В каждой ячейке может быть записан один и только один символ из входного алфавита машины Тьюринга. В дальнейшем для простоты будем рассматривать алфавит, состоящий всего из двух символов: 0 и 1. При этом также рассматривается пустой "символ" (именно так следует понимать содержимое ячейки, в которой не записан ни 0, ни 1). Пустой символ обычно обозначают как ε . В начальный момент на ленту записывается какое-то слово. Обычно это слово представляет собой описание некоторой задачи, на которых специализируется данная машина Тьюринга. Кроме ленты, у машины имеется читающая-пишущая головка, которая позволяет считать символ из ячейки или записать в ячейку, непосредственно находящуюся под головкой, какой-то один символ. Машина Тьюринга работает по тактам. Такты машины Тьюринга никак не привязываются к единицам времени, например, секундам. Но именно в тактах измеряется "время", затрачиваемое машиной Тьюринга на то, чтобы выполнить обработку входного слова. Разумно и сложность задачи связывать с числом тактов, которую необходимо затратить на Машине Тьюринга для ее решения.

Машина Тьюринга работает по определенным правилам. По своей сути такие правила однотипны и сводятся к следующему: считать символ из текущей ячейки. По текущему состоянию машины и считанному символу определить новое состояние, в которое переходит машина, записать в ячейку новый символ и либо сдвинуть ленту на одну ячейку влево, либо сдвинуть ленту на одну ячейку вправо, либо оставить ленту в прежнем положении, после чего перейти к следующему такту. Машина завершает работу, если она попадает в **конечное** состояние. Такие правила удобнее всего представлять в виде таблицы. Возьмем в качестве примера следующую таблицу.

	0	1	ε
Q_0	UL Q_0	0L Q_0	UU Q_1

В этой таблице использованы следующие обозначения. Состояния машины Тьюринга обозначены как Q_0, Q_1 . Обычно начальное состояние обозначается как Q_0 (и у нас также начальное состояние машины есть Q_0). Состояние Q_1 в нашем примере есть конечное состояние. В верхней строке записаны символы входного алфавита машины. Таких символов три: 0, 1, ε . Пустой символ тоже считается символом, хотя реально не соответствует никакому символу вообще.

Рассмотрим строку Q_0 . Пусть в текущей ячейке записан 0. На пересечении столбца "0" и строки Q_0 стоит *действие* машины - εLQ_0 . Это действие следует понимать таким образом: первый символ ε означает ничего в ячейку не писать. Второй символ L означает сдвинуть ленту на одну ячейку влево (вперед, иначе говоря). Третий символ Q_0 означает, в какое состояние машина переходит (в нашем случае машина остается в старом состоянии, ячейку не изменяет и сдвигает ленту вперед). Рассмотрим теперь столбец 1. Действие теперь таково $0LQ_0$. По этому действию в текущую ячейку будет записан 0 вместо 1; все остальное – как и в предыдущем примере. Наконец, последнее действие: UUQ_1 . Это действие надо понимать так: ничего в ячейку не писать (первое U), ленту не сдвигать (второе U), перейти в конечное состояние Q_1 . Вот, собственно, и все. Теперь можно догадаться, что представленная выше машина Тьюринга **обнуляет** числа, т.е. просто заменяет 1 на 0. Конечно, это – очень элементарная машина. Но надо сразу оговориться, что решать реальные задачи на машине Тьюринга никто не будет. Машина Тьюринга используется для анализа алгоритмов. Например, вопрос типа :”Можно ли решить эту задачу?” есть по сути вопрос “Можно ли для этой задачи построить машину Тьюринга?”

Нумерация машин Тьюринга и универсальная машина Тьюринга.

Для формального анализа различных проблем, связанных с машинами Тьюринга, удобно каждой машине присвоить числовой номер – целое положительное число. Такую нумерацию можно задать разными способами. Одним из известных способов является способ К.Геделя – австрийского математика, доказавшего несколько знаменитых теорем в области логики, теории множеств и теории вычислений. Способ Геделя основан на том, что, как известно из арифметики, любое целое положительное число A можно единственным способом представить в виде произведения простых множителей в виде:

$$A = p_1^{n_1} * p_2^{n_2} * \dots * p_z^{n_z}$$

Здесь $p_i^{n_i}$ - i -й простой множитель числа A ; n_i – степень, с которой данный множитель входит в разложение числа A . Пример

$$126 = 2^1 * 3^2 * 7^1.$$

Ясно, что порядок множителей в разложении числа значения не имеет. Теперь посмотрим, как получить геделевский номер машины Тьюринга. Такая машина задана своими правилами. Каждое правило можно представить как строку символов

$$Q_i S_j \rightarrow Q_k S_z A$$

Мы уже знаем, как читать такую строку: если машина находится в состоянии Q_i и читает символ S_j , то она переходит в состояние Q_k , записывает символ S_z вместо S_j и выполняет действие $A = \{\text{оставить ленту на месте; сдвинуть ленту влево на одну ячейку; сдвинуть ленту вправо на одну ячейку}\}$. Получим геделев номер этого правила, используя номера символов в алфавите. Например, пусть используются только заглавные буквы. Каждая буква имеет свой номер:

A(1),B(2),C(3),D(4),E(5),F(6),G(7),H(8),I(9),J(10),Q(11),K(12),L(13),M(14),N(15),O(16),

P(17),R(18),S(19),T(20),U(21),W(22),X(23),Y(24),Z(25),→(26),0(27),1(28),2(29),3(30),4(31),5(32),6(33),7(34),8(35),9(36),0(37).

Например, правило вида $Q_2S_9 \rightarrow Q_{11}S_0A$ дает последовательность чисел: 11,29,19,36,26,11,28,28,19,27,1. Это правило можно заменить однозначным образом на число

$$2^1 * 3^{29} * 5^{19} * 7^{36} * 11^{26} * 13^{11} * 17^{28} * 19^{28} * 23^{19} * 29^{27} * 31^1$$

Несмотря на гигантский размер этого числа, нам достаточно знать, как его получить и как по числу восстановить запись правила.

Таким образом, можно получить геделевы номера для правил машины Тьюринга. Теперь расположим эти номера по возрастанию. Пусть при этом получена последовательность чисел n_1, n_2, \dots, n_x . Для этой последовательности чисел снова определяем ее геделев номер. Таким образом, окончательно получим геделев номер машины Тьюринга. Рассмотрим пару чисел (x, y) . Здесь x – Геделев номер машины Тьюринга, y – произвольное двоичное число на входе машины Тьюринга. Такую пару чисел можно рассматривать как функцию $f_x(y)$, вычисляемую следующим образом. По номеру x определяется набор правил машины Тьюринга. Если x дает синтаксически некорректную спецификацию (неверную запись для правил), то проверяем последовательно номера $x+1$, $x+2$, и т.д., пока не получим правильную запись правил. Используя правила полученной машины Тьюринга моделируем ее работу на входе y . В результате, если машина остановится, получим записанное на ленте слово, которое и будем рассматривать, как значение функции $f_x(y)$. Ясно, что функция $f_x(y)$ может не завершить вычисление за конечное время. Тогда говорим, что ее значение на входе y не определено.

Определение. Класс всех одноместных функций $f_x(y)$, для каждой из которых имеется вычисляющая ее машина Тьюринга, называется классом **вычислимых** функций.

Теорема. Имеется функция, не являющаяся вычислимой.

Доказательство. Доказательство проводится методом, предложенным Кантором и называется **канторовским диагональным методом**.

Прежде всего, каждую функцию можно рассматривать как отображение одного множества чисел на другое (или на это же множество). Первое множество называется областью определения функции, а второе – областью значений. Мы будем рассматривать только такие функции, у которых область определения и область значений суть множество натуральных чисел. Например, рассмотрим функцию $f(x) = x^2$. Область определения этой функции есть натуральный ряд $1, 2, 3, \dots$. Натуральный ряд удобно обозначить буквой \mathbb{N} , а отображение, реализуемое функцией, в виде $\mathbb{N} \rightarrow \mathbb{N}$. Множество значений также принадлежит этому ряду (хотя и не все числа натурального ряда дают эти значения). Вычислимая функция определяет вычисляемое на машине Тьюринга отображение. Номер вычислимой функции можно связать с номером машины

Тьюринга, которая эту функцию вычисляет. Запись $f_n(x) = y$ можно теперь читать таким образом:

вычислимая функция с номером n отображает число x в число y .

Важно заметить, что вычислимые функции не обязательно всюду определены. Например, у нас есть функция отыскания корня квадратного. Однако нет целого корня, например, у 23 или 47. Следовательно, на этих числах в рассматриваемом нами классе функций от натуральных аргументов данная функция не определена. Однако для этой функции есть машина Тьюринга, т.е. ее правильно назвать **частично-вычислимой**.

Пусть задан пересчет частично вычислимых функций (вычислимых отображений)

$$f_1(y), f_2(y), f_3(y), \dots, f_n(y), \dots$$

Рассмотрим следующую таблицу:

	$f_1(y)$	$f_2(y)$	$f_3(y)$...
1	$f_1(1)_{+1}$	$f_2(1)$	$f_3(1)$	
2	$f_1(2)$	$f_2(2)_{+1}$	$f_3(2)$	
3	$f_1(3)$	$f_2(3)$	$f_3(3)_{+1}$	
...				
k	$f_1(4)$	$f_2(4)$	$f_3(4)$	
...				

На основании данной таблицы определим следующую функцию

$$\varphi(x,y) = \begin{cases} f_x(y), & \text{если } f_x(y) \text{ определена на входе } y \neq x; \\ f_x(y)+1, & \text{если } f_x(y) \text{ определена на входе } y = x; \\ 0, & \text{в противном случае} \end{cases}$$

$$\varphi(x,y)$$

Ясно, что наша новая функция отличается от каждой вычислимой функции $f_w(x)$ в точках $x=w$ и является всюду определенной. Покажем теперь формально, что наша функция $\varphi(x,y)$ не может быть вычислимой никакой машиной Тьюринга. Допустим, что это не так и машина с номером z вычисляет нашу функцию. На основании этой машины можно построить машину для вычисления $\varphi(x,x)$. Тогда получим что $\varphi(x,x) = f_z(x)$. Но это противоречит определению $\varphi(x,y)$ при $x=y$.

Поскольку $\varphi(x,x)$ определена для всех x , то $\varphi(z,z) = f_z(z) = f_z(z) + 1$, чего быть не может.

Теорема. Для любых целых положительных чисел A и B взаимно однозначно определяется число

$$C = 0.5 * ((A+B)^2 + 3A+B)$$

Теорема. Число невычислимых функций несчетно.

Эта теорема доказывается с привлечением знаний из теории множеств. Число всех вычислимых функций счетно, т.к. им соответствуют уникальные номера из натурального ряда чисел. Сколько же всего функций? Каждая функция задает отображение из множества натуральных чисел \aleph в множество натуральных чисел \aleph . При этом если рассматривать всюду определенные тотальные отображения, то 1 можно отобразить в любое из $|\aleph|$ чисел, далее 2 можно отобразить в любое из $|\aleph|$ чисел, и т.д, т.е. всех мыслимых отображений $|\aleph|^{|\aleph|}$. В теории множеств доказывается, что $2^{|\aleph|} > |\aleph|$, что и дает несчетность числа всех определенных отображений (функций).