



Машина Тьюринга

Одним из способов формализации понятий вычислимости и алгоритма состоит в рассмотрении некоторой идеализированной вычислительной машины, которой должны быть присущи простота и архитектурное строение не должно резко отличаться от реальной вычислительной машины.

Концепция абстрактной вычислительной машины была **предложена Аланом Тьюрингом** в 1936 году. Он показал возможность существования универсальной вычислительной машины (машины Тьюринга), способной выполнить любую эффективную процедуру.

Машина Тьюринга - это гипотетическая вычислительная машина (автомат), использовавшаяся в качестве математической абстракции А.Тьюрингом с целью уточнения эффективного процесса (алгоритма) вычислений.

С точки зрения формального описания машина Тьюринга есть совокупность следующих пяти компонент:

1. Управляющего устройства, способного принимать некоторое число состояний, которые соответствуют различным комбинациям состояний внутренних элементов электронной машины (ячейки памяти, электронные схемы и т.п.);

2. Ленты, разделённой на бесконечное число ячеек (≥ 1). На ленте выбрано направление, называемое направлением слева на право. Соответственно этому на ленте выделяются крайняя слева и крайняя справа ячейки; каждая ячейка, кроме крайней слева, имеет единственную соседнюю



слева, каждая ячейка, кроме крайней справа, имеет единственную соседнюю справа;

3. Головки, которая в каждый момент работы машины находится в некоторой ячейке ленты и считывает информацию из этой ячейки. Головка может работать в каждый момент только с одной ячейкой ленты. При этом она может заменять прочитанный символ другим символом;

4. Двух конечных алфавитов - *внешнего*

$$\Sigma = \{a_{-1}, a_0, \dots, a_n\}, \quad n \geq 1$$

и *внутреннего*

$$S = \{q_0, q_1, \dots, q_l\},$$

называемого также множеством состояний.

Внутренний алфавит - это конечный алфавит символов, которые подаются на вход машины Тьюринга и выдаются на выходе.

Внутренний алфавит - это конечный алфавит символов, комбинации которых определяют внутреннее состояние машины Тьюринга.

a_{-1} - *пограничный маркер*, стоящий справа, q_0 - *начальное состояние* машины Тьюринга.

Командой называется выражение одного из следующих типов:

$$q_i a_j a_k q_m$$

$$q_i a_j \Pi q_m$$

$$q_i a_j \Lambda q_m$$



5. **Программы машины** - конечного множества слов специального вида, называемых командами.

Правильным вычислением машины Тьюринга M мы будем называть последовательность $(s_0, s_1, \dots, s_p)$ конфигураций этой машины такую, что:

Для каждого $i=0, \dots, p-1$ ситуация s_{i+1} получается из s_i выполнением команды машины M ;

s_0 - начальная конфигурация;

s_p - заключительная конфигурация;

Ситуацией называется состояние машины вместе с указанием, какая ячейка обозревается и всего того, что напечатано на ленте.

Формализацией машины Тьюринга называется набор $M = \langle \Sigma, S, P, q_0, q_1, \beta \rangle$, где Σ - внешний алфавит, S - внутренний алфавит, P - программа машины, q_0 - начальное состояние, q_1 - заключительное состояние, β - несобственный символ алфавита, пробел (бланк).

Работа машины Тьюринга. Машина Тьюринга работает следующим образом: в каждой ситуации, определённой внутренним состоянием управляющего устройства и символом, прочитанным на ленте, машина срабатывает, переходя в другое состояние и продвигая ленту определённым образом. Ячейку, обозреваемую головкой в данный момент, называют рабочей ячейкой.

Выполнение каждой команды МТ состоит в следующем:

- Изменить содержимое рабочей ячейки a_j на a_k , т.е. в рабочую ячейку записывается какая-нибудь буква из Σ ;



- Головка передвигается на одну ячейку влево или вправо либо остаётся на месте;
- Головка переводится в новое состояние, которое может и совпадать и со старым.

Если команда не меняет буквы, записанной в рабочей ячейке, не сдвигает головки и не меняет её состояние, то такую команду будем называть *командой останова*.

Перед началом работы МТ на её ленту записываются начальные данные: одно или несколько слов над алфавитом Σ . Количество слов, составляющих начальные данные, зависит от алгоритма, но всегда конечно.

Слова отделяются друг от друга несобственной буквой β .

В начале работы МТ головка приводится в начальное состояние q_0 и помещается над начальной рабочей ячейкой, которая определённым и фиксированным для каждой конкретной МТ образом расположена относительно начальных данных. Начав работу МТ может, в зависимости от начальных данных:

- Через конечное число тактов (шагов) приступить к выполнению команды

$$(q, a, a, q)$$

которая ничего не изменяет, и остановится. При этом на ленте оказывается записанной новая последовательность слов, которая представляет собой результат работы МТ.



• МТ никогда не останавливается либо возникает пара (q, a) , для которой не определена команда МТ. В этом случае считается, что МТ неприменима к соответствующим начальным данным.

На множество команд, задающих машину Тьюринга, наложено требование, чтобы начальные пары символов всех команд были различны. Машину, для которой это требование выполнено, мы назовём **детерминированной**. **Конфигурацией** называется выражение, удовлетворяющее следующим условиям:

- Оно не содержит вхождений символов Π и $\mathbb{L}$;
- Оно содержит ровно одно вхождение символа типа q_i ;
- Это вхождение символа q_i не занимает в выражении крайней правой позиции.

Вычислимость по Тьюрингу.

С помощью машины Тьюринга формализуется интуитивное понятие вычислимости (а тем самым и интуитивное понятие алгоритма).

Пусть даны два алфавита A и B , такие , что $A \cap B \neq \{\emptyset\}$. Рассмотрим подмножество $L \subseteq (A^*)^n$, т.е. совокупности из n слов в алфавите A

$$(A^*)^n = \{(w_1, w_2, \dots, w_n \mid w_i \in A^*), \text{ где } A^* - \text{множество слов над алфавитом } A.$$

n -местной частичной функцией из $(A^*)^n$ (совокупности n слов над алфавитом A) в B^* называется правило:

$$f(w_1, w_2, \dots, w_n) = w \in B^*$$



т.е. совокупности n слов $(w_1, w_2, \dots, w_n) \in (A^*)^n$ ставится в соответствие слово $w \in B^*$, где $A \cap B \neq \{\emptyset\}$.

Область определения функции f : $\text{Def}(f) = L \subseteq (A^*)^n$, т.е. совокупности n слов над алфавитом A .

Область значений функции f : $\text{Im}(f) = \{w \in B^* \mid \text{если существует совокупность из } n \text{ слов } (w_1, w_2, \dots, w_n) \in (A^*)^n, w = f(w_1, w_2, \dots, w_n)\}$.

Под вычислимой функцией понимается функция, которая может быть вычислена с помощью некоторого алгоритма.

Пусть МТ T имеет рабочий алфавит $A \cup B$. будем говорить, что МТ T **определяет n -местную частичную функцию f_T** , отображающую $(A^*)^n$ в B^* , если выполняются следующие условия:

1. Совокупность n слов $(w_1, w_2, \dots, w_n) \in \text{Def}(f_T)$ тогда и только тогда, когда имеет место отношение $\langle \beta w_1 \beta w_2 \beta \dots \beta w_n (\beta) \beta \rangle \Rightarrow^* \langle \beta \phi \beta w (\beta) \psi \beta \rangle$ (т.е. МТ T начав работу в начальной ситуации S_0 за конечное число шагов остановится в ситуации S) $S_0 \Rightarrow^* S$, где слово $\phi, \psi \in (A \cup B \cup \{\beta\})^*$ - расширение рабочего алфавита МТ T . $w \in B^*$, при этом слово w является значением функции f_T , т.е. $w = f_T(w_1, w_2, \dots, w_n)$
2. Совокупность n слов $(w_1, w_2, \dots, w_n) \notin \text{Def}(f_T)$, если: МТ неприменима к начальным данным, либо $w \notin B^*$, либо в рабочей ячейке в результате останова МТ содержится не буква β , либо рабочая ячейка не следует за словом w .

Частичная функция $f: (A^*)^n \rightarrow B^*$ (задающая отображение) называется **вычислимой по Тьюрингу** (ВТ-функцией), если существует МТ T с рабочим алфавитом $A \cup B$, такая что n -местная частичная функция $f_T: (A^*)^n \rightarrow B^*$,



определяемая этой МТ, совпадает с f (т.е. $f_T = f$). О любой такой МТ говорят, что она вычисляет функцию f .

Если МТ T' моделирует МТ T , то эти машины вычисляют одну и ту же функцию f .

Практическая вычислимость - это вычислимость, которая реализуется с помощью физического автомата и является значительно более узким понятием, чем вычислимость по Тьюрингу.

Функция $f: (A^*)^n \rightarrow B^*$ называется **нормированно вычислимой по Тьюрингу** (НВТ-функцией), если существует МТ T , которая вычисляет f и при этом удовлетворяет следующим требованиям:

1. Если совокупность из n слов $(w_1, w_2, \dots, w_n) \notin \text{Def}(f)$, то МТ начав работу с ситуации $\langle \beta w_1 \beta w_2 \beta \dots \beta w_n(\beta) \beta \rangle$ никогда не остановится.
2. Если $(w_1, w_2, \dots, w_n) \in \text{Def}(f)$, то справедливо отношение $\langle \beta w_1 \beta w_2 \beta \dots \beta w_n(\beta) \beta \rangle \Rightarrow^* \langle \beta w_1 \beta w_2 \beta \dots \beta w_n(\beta) w \beta \rangle$, где $w = f_T(w_1 \dots w_n)$

Смысл нормированного вычисления состоит в том, что при таком вычислении часть ленты, расположенная левее символа β , непосредственно предшествующего слову w_1 , не меняется и не влияет на работу МТ T . при нормированном вычислении головка не должна заходить левее указанного символа β .

Теорема. Всякая вычислима по Тьюрингу функция (ВТ-функция) является НВТ-функцией.

Пример 1 (вычислимой функции). Моделирование работы Машины Поста.



Функция $f(x, y) = x + y$, определённая на множестве пар неотрицательных целых чисел, является "вычислимой в интуитивном смысле"; покажем что она вычислима и в смысле Тьюринга. Для этого построим МТ M с алфавитом $\{\beta, | \}$ такую, что $x + y = f_T(x, y)$.

Подобная машина должна, имея на ленте коды $\bar{x}$ и $\bar{y}$ ($\bar{0} = |$, $\bar{4} = ||||$), разделённые пустой ячейкой, получать из этой записи $x + y$ палочек (не обязательно подряд). Заметим, что $x + y = \langle \bar{x} \rangle + \langle \bar{y} \rangle - 2$

Легко убедиться, что следующая машина выполняет поставленную задачу:

$q_1 \beta q_2$	стирается первая палочка первого кода
$q_2\beta l q_3$	
$q_3 \beta q_4$	стирается вторая палочка первого кода
$q_3\beta\beta q_3$	
$q_4\beta l q_5$	} лента сдвигается до тех пор, пока головка не дойдёт до несобственного символа
$q_5 \beta q_5$	
$q_5\beta q_6$	заменяем несобственный символ палочкой
$q_4 \beta q_4$	останов

Здесь машина останавливается, поскольку ситуации $q_4\beta$ нет ни в одной команде. В этот момент на ленте ровно $x + y$ палочек.

Пример 2 (пример частично вычислимой функции).

Функция $g(x, y) = x - y$ определена лишь для тех пар, у которых $x \geq y$, покажем, что она частично вычислима по Тьюрингу. Для этого мы построим машину Тьюринга M , выполняющую вычитание. Пусть на ленте имеются коды $\bar{x}$ и $\bar{y}$, разделённые пустой ячейкой; $\bar{x}$ слева, $\bar{y}$ справа. Машина будет работать последовательными циклами; каждый цикл состоит в стирании одной (самой левой) палочки в $\bar{x}$ и одной (самой правой) палочки $\bar{y}$.

$q_1 \beta q_1$	стирается самая левая палочка в $\bar{x}$,
-----------------	---



$q_1\beta\sqcup q_2$ лента сдвигается на одну ячейку влево,
 $q_2|\sqcup q_2$ лента сдвигается влево, пока головка не дойдёт до конца кода $\bar{x}$,
 $q_1\beta\sqcup q_3$ головка переходит через пробел, разделяющий $\bar{x}$ и $\bar{y}$,
 $q_3|\sqcup q_3$ лента сдвигается влево, пока головка не дойдёт до конца кода $\bar{y}$,
 $q_3\beta\Pi q_4$ шаг назад на первый символ $\bar{y}$,
 $q_4|\beta q_4$ стираем обозреваемый символ,
 $q_4\beta\Pi q_5$ делаем шаг вправо,
 $q_5|\Pi q_6$ проверяем, если обозреваем палочку, то продолжаем вычисления, иначе машина останавливается, так как отсутствует команда $q_5\beta$,
 $q_6|\Pi q_6$ сдвигаемся вправо, пока не встретим начало кода $\bar{y}$,
 $q_6\beta\Pi q_7$ переходим через пробел, разделяющий $\bar{x}$ и $\bar{y}$,
 $q_7|\Pi q_8$ если в $\bar{x}$ осталось хоть одна палочка, вычисления продолжаются,
 $q_7\beta\Pi q_7$ если в $\bar{x}$ не осталось ни одной палочки (случай $x < y$), машина будет работать вечно, сдвигая ленту вправо,
 $q_8|\Pi q_8$ лента сдвигается до начала кода $\bar{x}$,
 $q_8\beta\sqcup q_1$ головка устанавливается так, что обозревается самая левая палочка "остатка" кода $\bar{x}$; машина переходит в начальное состояние, тем самым оказывается готовой начать очередной цикл.

Пример 3 (нормированные вычисления).

Построить МТ M , делающую нормированные вычисления $\langle \beta w(\beta) \rangle \Rightarrow^* \langle \beta w \beta w_1(\beta) \rangle$, где $w, w_1 \in \bar{A}^*$, $A = \{1, 0\}$, $w_1 = w + 1$, т.е. должны быть сохранены исходные данные.

Для этого необходимо вначале скопировать исходные данные, а затем осуществлять прибавление 1, т.е. вначале необходимо построить копирующую машину $\langle \beta w(\beta) \rangle \Rightarrow^* \langle \beta w \beta w(\beta) \rangle$, а затем строить МТ $\langle \beta w \beta w(\beta) \rangle \Rightarrow^* \langle \beta w \beta w_1(\beta) \rangle$, делающую нормированные вычисления и прибавляющую 1 к любому второму числу за конечное число тактов.



Связь МТ и РАМ-машин.

Основное применение МТ находят в установлении нижних оценок на время и память, необходимые для решения данной задачи. В большинстве случаев установить нижние оценки можно только с точностью до полиномиальной связанности. Для установления точных оценок потребуется рассматривать более специфические детали конкретных моделей.

Рассмотрим связь между РАМ и МТ.

РАМ может моделировать работу k -ленточной МТ, помещая в регистр одну ячейку с ленты МТ. В частности, i -ю ячейку j -той ленты можно записать в регистре $k*i+j+c$, $c=\text{const}$. k регистрами РАМ пользуются для запоминания k состояний головок МТ. РАМ может считывать исходные слова (буквы) из ячеек МТ, используя косвенную адресацию с помощью регистров, содержащих состояния головок.

Предположим, что данная МТ имеет временную сложность $T(n) \geq n$. Тогда РАМ может прочесть её вход, запомнить его в регистрах, представляющих первую ленту, и смоделировать эту МТ за время $O(T(n))$ при равномерном весовом критерии и $O(T(n)\log T(n))$ при логарифмическом.

В любом случае время на РАМ ограничено сверху полиномом от времени МТ, т.к. любая функция $O(T(n)\log T(n)) \leq O(T^2(n))$. Обратное утверждение верно только при логарифмическом критерии.

Утверждение. Пусть L - язык, допускаемый некоторой РАМ за время $T(n)$ при логарифмическом критерии. Если в РАМ-программе нет умножений и делений, то на многоленточных МТ язык L имеет временную сложность не более $O(T^2(n))$.



Моделирование РАМ машиной Тьюринга.

Представим все, не содержащие нуля, регистры РАМ на ленте МТ следующим образом:

β	β	i_1	β	c_1	β	β	i_2	β	c_2	...	β	β	i_k	β	c_k	β	β	...
---------	---------	-------	---------	-------	---------	---------	-------	---------	-------	-----	---------	---------	-------	---------	-------	---------	---------	-----

На ленте МТ помещена последовательность пар чисел (i_j, c_j) , записанных в двоичной системе счисления без нулей в начале слова и разделённых несобственной буквой β , где i_j - номер регистра, c_j - содержимое регистра в двоичной системе счисления ($i=1, n$).

Содержимое сумматора РАМ хранится в двоичном коде на второй ленте. Третья лента МТ играет роль рабочей памяти. Ещё две ленты служат для записи входа и выхода РАМ. Каждый шаг РАМ-программы представлен числом состояний МТ.

Рассмотрим как можно смоделировать команды РАМ на МТ.

ADD *20

1. На ленте 1 отыскивается входное слово $w_j = \beta\beta 10100\beta$, соответствующее регистру 20 в РАМ. Если такое слово w_j находится, то на ленту 3 (которая играет роль рабочей памяти помещается следующее за ним число c_j , которое будет содержимым регистра 20. Если же слова w_j на ленте нет, машина останавливается, т.к. содержимое регистра 20 равно 0, и поэтому косвенную адресацию провести нельзя (на ленту были записаны все регистры с содержимым $\neq 0$).

2. На ленте 1 отыскивается слово w_{c_j} , записанное на ленте 3. Если оно находится, то следующее за ним слово, отделённое несобственной буквой β записывается на ленту 3. Если этого символа нет, то на ленту помещается 0.



3. Число, записанное на ленту 3 на шаге 2, прибавляется к содержимому сумматора, которое хранится на ленте 2.

Для моделирования команды STORE 30 ($C(30) \leftarrow C(0)$) можно построить МТ, чтобы она работала следующим образом:

1. На ленте 1 отыскивается слово w_j , соответствующее регистру 30 в РАМ, т.е. $w_j = \beta\beta 11110\beta$.

2. Если оно находится, то на ленту 3 копируется лента 1, но слово c_j непосредственно следующее за словом w_j (т.е. указывающее на содержимое регистра 30) заменяется на слово, записанное на ленте 2.

3. Если на ленте 1 нет слова w_j , то после того как на ленту 3 скопированы все исходные данные машина печатает слово $w_j = \beta\beta 11110\beta$, затем содержимое сумматора и конец $\beta\beta$.

Аналогично можно смоделировать любые команды РАМ.

Рис. Время, требуемое машиной А для моделирования работы алгоритма сложности $T(n)$ на машине В.

Моделируемая машина В	Моделирующая машина А		
	МТ (1 лента)	МТ (к лент)	РАМ
МТ (1 лента)	-	$O(T(n))$	$O(T(n)\log T(n))$
МТ (к лент)	$O(T^2(n))$	-	$O(T(n)\log T(n))$
РАМ	$O(T^3(n))$	$O(T^2(n))$	-