

Глава 6.

Машины Тьюринга

§ 6.1. Неформальное и формальное описания

В этой главе мы рассмотрим еще один тип распознающих устройств — *машины Тьюринга*. Это абстрактное устройство было предложено в качестве математической модели для описания процедур. Так как наше интуитивное понятие процедуры как конечной последовательности инструкций, которые могут выполняться механически, не является математически точным, мы не можем доказать формально, что оно эквивалентно точному понятию машины Тьюринга.

Было предложено много других формализаций процедуры, и было показано, что все они эквивалентны формализации машины Тьюринга.

А.Чёрчем была высказана гипотеза, что любой процесс, который естественным образом мог бы быть назван процедурой, реализуем машиной Тьюринга. Впоследствии вычислимость при помощи машины Тьюринга стала общепризнанным определением процедуры.

В литературе определение машины Тьюринга давалось разными способами. Мы начнем с обсуждения основной модели.

Основная модель (см. рис. [6.1](#)) имеет *конечное управление*, *ленту*, которая разделена на *ячейки*, и *головку ленты*, которая сканирует одну ячейку ленты зараз. Лента имеет крайнюю левую ячейку, но простирается в бесконечность в правую сторону.

Каждая ячейка может содержать ровно один из конечного числа *символов ленты*. Первоначально n крайних левых ячеек для некоторого конечного n содержит *входную цепочку*, строку символов ленты, называемых *входными символами*.

Остальные ячейки до бесконечности содержат *пробел* — специальный символ ленты, который *не считается* входным символом.

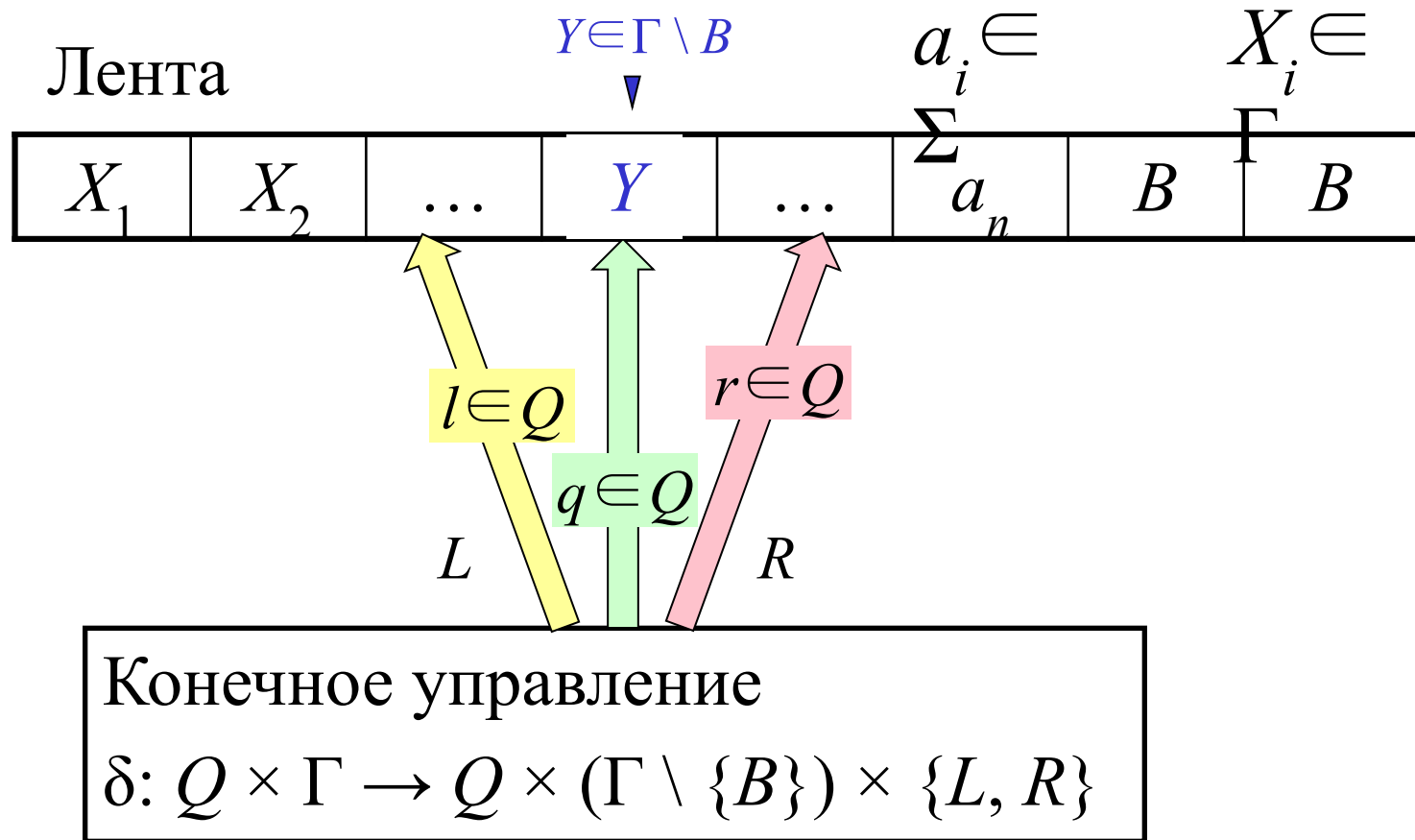


Рис. 6.1.

В один *такт*, зависящий от символа, сканируемого головкой ленты, и состояния конечного управления, машина Тьюринга

- 1) изменяет состояние;
- 2) печатает символ ленты, не являющийся пробелом, в сканируемой ячейке ленты, замещая то, что было там записано;
- 3) сдвигает свою головку влево или вправо на одну ячейку.

Определение 6.1. *Машина Тьюринга (Тм)* является формальной системой:

$$T = (Q, \Sigma, \Gamma, \delta, q_0, F),$$

где Q — конечное множество *состояний*;
 Γ — конечное множество допустимых *символов* ленты, один из них, обычно обозначаемый буквой B , есть *пробел*;
 $\Sigma \subseteq \Gamma \setminus \{B\}$ — множество *входных символов*;
 $\delta: Q \times \Gamma \rightarrow Q \times (\Gamma \setminus \{B\}) \times \{L, R\}$ — функция следующего такта (движения), для некоторых аргументов может быть не определена;
 $q_0 \in Q$ — *начальное состояние*;
 $F \subseteq Q$ — множество *конечных состояний*.

Заметим, что если головка ленты покидает ячейку, она должна напечатать непустой символ в этой ячейке, так что лента всегда содержит непустой блок символов с бесконечным числом пробелов справа от него.

Мы не позволили Tm печатать пробел ради простоты определения конфигураций. Однако, Tm могла бы иметь другой символ, который трактуется точно так же, как пробел, за исключением того, что Tm разрешается печатать этот символ *псевдопробела*.

Разумеется, никакой дополнительной мощности не появляется за счёт введения такого символа.

В неформальном обсуждении мы часто допускаем печать пробела, зная, что вместо него можно использовать другой, но эквивалентный ему СИМВОЛ.

Определим теперь один такт (движение) машины Тьюринга T .

Пусть $(q, A_1 A_2 \dots A_n, i)$ — некоторая её конфигурация, где $q \in Q$, $A_i \in \Gamma \setminus \{B\}$, где $1 \leq i \leq n + 1$.

Случай 1. Если $1 \leq i \leq n$ и $\delta(q, A_i) = (p, A, R)$, то $(q, A_1 A_2 \dots A_{i-1} \textcolor{red}{A}_i A_{i+1} A_n, i) \vdash_T$
 $\vdash_T (p, A_1 A_2 \dots A_{i-1} \textcolor{red}{A} A_{i+1} \dots A_n, i + 1)$,
т. е. T печатает символ A в i -й позиции и двигается вправо.

Случай 2. Если $2 \leq i \leq n$ и $\delta(q, A_i) = (p, A, L)$,
то $(q, A_1 A_2 \dots A_{i-1} \textcolor{red}{A}_i A_{i+1} A_n, \vdash_T$
 $\vdash_T (p, A_1 A_2 \dots A_{i-1} \textcolor{red}{A} A_{i+1} \dots A_n, i-1),$
т. е. T печатает символ A в i -й позиции и
двигается влево, не сходя с левого конца
ленты.

Случай 3. Если $i = n + 1$, то головка сканирует пробел B .

а) Если при этом $\delta(q, B) = (p, A, R)$, то

$$(q, A_1 A_2 \dots A_n \textcolor{red}{B}, n + 1) \mid_{\overline{T}} \\ \mid_{\overline{T}} (p, A_1 A_2 \dots A_n \textcolor{red}{A} B, n + 2).$$

б) Если при этом $\delta(q, B) = (p, A, L)$, то

$$(q, A_1 A_2 \dots A_n \textcolor{red}{B}, n + 1) \mid_{\overline{T}} (p, A_1 A_2 \dots A_n \textcolor{red}{A}, n).$$

Таким образом, мы ввели *отношение* $\mid_T$ *непосредственного следования одной конфигурации за другой*.

Очевидным образом можно определить *степень* $\mid_T^n$, *транзитивное* $\mid_T^+$ и *рефлексивно-транзитивное* $\mid_T^*$ *замыкания* ЭТОГО ОТНОШЕНИЯ.

Если две конфигурации связаны знаком $\left| \frac{n}{T} \right.$, то мы будем говорить, что вторая получается из первой за n движений.

Соответственно значок $\left| \frac{+}{T} \right.$ обозначает положительное число движений, а $\left| \frac{*}{T} \right.$ — любое число движений, включая нуль.

Определение 6.3. Пусть $T = (Q, \Sigma, \Gamma, \delta, q_0, F)$ — машина Тьюринга.

Язык, распознаваемый машиной T есть $L = \{w \mid w \in \Sigma^* \text{ и } (q_0, w, 1) \vdash_T^* (q, \alpha, i) \text{ для некоторых } q \in F, \alpha \in \Gamma^* \text{ и } i > 0\}$.

Мы предполагаем, что данная T_m , распознающая язык L , *останавливается*, т. е. не имеет никакого следующего движения, всякий раз, как входная цепочка принимается. Но для цепочек, которые не принимаются, T_m может не остановиться.

Два других случая, когда T_m останавливается, однако, не принимая:

- когда значение $\delta(q, X)$ при некоторых $q \in Q$ и $X \in \Gamma$ не определено или
- когда головка ленты “соскакивает” с левого конца ленты (позиция головки ленты $i = 1$ и задано движение влево).

Пример 6.1. Построим Тм, распознающую $\text{cfl } L = \{0^n 1^n \mid n \geq 1\}$.

Положим $T = (Q, \Sigma, \Gamma, \delta, q_0, F)$,

где $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$;

$\Sigma = \{0, 1\}$; $\Gamma = \{0, 1, B, X, Y\}$; $F = \{q_5\}$.

Функцию δ определим следующим образом:

1. $\delta(q_0, 0) = (q_1, X, R)$.

В состоянии q_0 символ 0 заменяется на X и машина сдвигается вправо в состояние q_1 в поисках 1.

Пример 6.1. Тм, распознающая $\text{cfl } L = \{0^n 1^n \mid n \geq 1\}$.

2. а) $\delta(q_1, 0) = (q_1, 0, R);$

б) $\delta(q_1, Y) = (q_1, Y, R);$

в) $\delta(q_1, 1) = (q_2, Y, L).$

Оставаясь в состоянии q_1 , машина продвигается вправо сквозь все нули (п. 2а) и блок Y (п. 2б). Наткнувшись на 1, заменяет её на Y и переходит в состояние q_2 , начав движение влево (п. 2в).

3. а) $\delta(q_2, Y) = (q_2, Y, L);$

б) $\delta(q_2, X) = (q_3, X, R);$

в) $\delta(q_2, 0) = (q_4, 0, L).$

Оставаясь в состоянии q_2 , машина продвигается влево сквозь блок Y (п. 3а).

Если машина встречает X , все еще оставаясь в состоянии q_2 , то больше нет нулей, которые следовало бы заменять на X , и машина переходит в состояние q_3 , начиная движение вправо, чтобы убедиться, что не осталось единиц (п. 3б).

Если же 0 встретился, машина переходит в состояние q_4 , чтобы продолжить движение в поисках крайнего левого 0 (п. 3в).

Пример 6.1. Тм, распознающая $\text{cfl } L = \{0^n 1^n \mid n \geq 1\}$.

4. а) $\delta(q_4, 0) = (q_4, 0, L)$

б) $\delta(q_4, X) = (q_0, X, R)$.

Машина движется сквозь нули (п. 4а). Если встретился X , то машина прошла самый левый нуль. Она должна, сдвинувшись вправо, превратить этот 0 в X (п. 4б). Происходит переход в состояние q_0 , и процесс, только что описанный в п. 1–4, продолжается.

Пример 6.1. Тм, распознающая $\text{cfl } L = \{0^n 1^n \mid n \geq 1\}$.

5. а) $\delta(q_3, Y) = (q_3, Y, R)$

б) $\delta(q_3, B) = (q_5, Y, R)$.

Машина входит в состояние q_3 , когда ни одного 0 не остается (см. п. 3б). Машина должна продвигаться вправо (п. 5а) сквозь блок Y . Если встречается пробел раньше, чем 1, то ни одной 1 не осталось (п. 5б). В этой ситуации машина переходит в конечное состояние q_5 и останавливается, сигнализируя тем самым прием входной цепочки.

Пример 6.1. Тм, распознающая cfl $L = \{0^n 1^n \mid n \geq 1\}$.

6. Во всех случаях, кроме 1–5, функция δ не определена.

Рассмотрим действия машины Тьюринга на входной цепочке 000111.

В табл. 6.1 приведены конфигурации в виде цепочек символов ленты с маркером состояния под сканируемым символом (в конфигурациях 25 и 26 маркер состояния находится под невидимым символом пробела).

Таблица 6.1

Шаг	Конфигурация	Шаг	Конфигурация	Шаг	Конфигурация
1	0 0 0 1 1 1 q_0	10	$X X 0 Y 1 1$ q_1	19	$X X X Y Y Y$ q_2
2	$X 0 0 1 1 1$ q_1	10	$X X 0 Y 1 1$ q_1	19	$X X X Y Y Y$ q_2
3	$X 0 0 1 1 1$ q_1	12	$X X 0 Y Y 1$ q_2	21	$X X X Y Y Y$ q_2
4	$X 0 0 1 1 1$ q_1	12	$X X 0 Y Y 1$ q_2	21	$X X X Y Y Y$ q_3
5	$X 0 0 Y 1 1$ q_2	14	$X X 0 Y Y 1$ q_4	23	$X X X Y Y Y$ q_3
6	$X 0 0 Y 1 1$ q_4	15	$X X 0 Y Y 1$ q_0	24	$X X X Y Y Y$ q_3
7	$X 0 0 Y 1 1$ q_4	16	$X X X Y Y 1$ q_1	25	$X X X Y Y Y$ q_3
8	$X 0 0 Y 1 1$ q_0	17	$X X X Y Y 1$ q_1	26	$X X X Y Y Y Y$ q_5
9	$X X 0 Y 1 1$ q_1	18	$X X X Y Y 1$ q_1		

§ 6.2. Методы построения машин Тьюринга

Машина Тьюринга может “программироваться” во многом так же, как программируются вычислительные машины. Роль программы играет функция δ .

В параграфе § 6.2 Пособия представлена коллекция приемов программирования машины Тьюринга, которые помогут лучше узнать её возможности.

§ 6.3. Машина Тьюринга как процедура

До сих пор мы определяли машину Тьюринга как распознающее устройство. Но можно рассматривать машину Тьюринга и как процедуру.

Например, если мы желаем описать процедуру для определения того, является ли число простым, мы могли бы построить машину Тьюринга, которая принимает множество всех простых чисел. Рассматриваем мы эту машину в данном случае как распознаватель или как процедуру — дело вкуса.

Машина Тьюринга в примере [6.1](#) используется как распознаватель. Заметим, что на некоторых входных цепочках, эта машина со временем достигнет условия, при котором для состояния конечного управления и сканируемого символа функция δ не определена. В таком случае машина Тьюринга останавливается и никакие дальнейшие её движения невозможны.

Если язык принимается машиной Тьюринга, которая останавливается на всех входных цепочках, то говорят, что язык *рекурсивен*.

Следует подчеркнуть, что существуют языки, которые принимаются машинами Тьюринга, не останавливающимися для некоторых цепочек, не содержащихся в языке, но которые не принимаются ни какими машинами Тьюринга, останавливающимися на *всех* входных цепочках.

Язык, который может быть распознан некоторой машиной Тьюринга, называется *рекурсивно перечислимым множеством* (recursively enumerable set — res).

Когда машина Тьюринга рассматривается как процедура и оказывается, что она останавливается для *всех* входных цепочек, то говорят, что такая процедура есть *алгоритм*.

В следующей главе будет показано, что рекурсивно перечислимые множества являются в точности языками, порожаемыми грамматиками типа 0.

Есть процедуры, для которых не существует никакого соответствующего алгоритма.

Примером является процедура для определения, порождает ли контекстно-зависимая грамматика (csg), по крайней мере, одну терминальную цепочку.

Можно построить машину Тьюринга, которая по заданной csg будет порождать все возможные терминальные цепочки в некотором лексикографическом порядке.

К каждой цепочке эта машина Тьюринга применяет алгоритм, данный в [гл. 2](#), чтобы увидеть, порождается ли данная цепочка грамматикой.

Если эта грамматика порождает хотя бы одно слово, машина найдет его и остановится в конечном состоянии.

Но если язык, порождаемый этой грамматикой, пуст, то машина будет продолжать порождать слова и проверять их вечно.

Имеет место факт, что не существует машины Тьюринга, которая останавливается на всех входных цепочках и определяет для каждой csg , является ли язык, порождаемый этой грамматикой, пустым.

Другими словами, *проблема распознавания непустоты контекстно-зависимого языка алгоритмически неразрешима.*

§ 6.4. Модификации машин Тьюринга

Одна из причин, по которой машины Тьюринга принимаются в качестве общей модели вычисления, состоит в том, что модель, с которой мы имеем дело, инвариантна по отношению ко многим модификациям, которые, казалось бы, увеличивают вычислительную мощность устройства.

6.4.1. Машина Тьюринга с бесконечной лентой в обе стороны

Теорема 6.1. *Если язык L распознается машиной Тьюринга с бесконечной в обе стороны лентой, то он распознается машиной Тьюринга с полубесконечной лентой.*

6.4.2. Многоленточная машина Тьюринга состоит из конечного управления с k ленточными головками, по одной на каждой ленте.

Каждая лента бесконечна в обоих направлениях. Сначала входная цепочка имеется только на первой ленте, а все другие ленты пусты.

При одном движении, зависящем от состояния конечного управления и сканируемого символа каждой из ленточных головок, машина может:

- 1) изменить состояние;
- 2) напечатать новый символ на каждой из сканируемых ячеек;
- 3) передвинуть каждую из её ленточных головок независимо друг от друга на одну ячейку влево, вправо или оставить её на том же самом месте.

Теорема 6.2. *Если язык L принимается многоленточной машиной Тьюринга, то он принимается одноленточной машиной Тьюринга.*

6.4.3. Недетерминированная машина Тьюринга есть устройство с конечным управлением и одной бесконечной в обе стороны лентой. Для данного состояния и ленточного символа, сканируемого головкой ленты, машина имеет несколько вариантов для следующего движения. Каждый вариант состоит из нового состояния, ленточного символа, который печатается, и направления движения головки. Недетерминированная машина Тьюринга принимает входную цепочку, если какая-нибудь последовательность вариантов движений приводит к принимающему состоянию.

Теорема 6.3. *Если язык L принимается недетерминированной машиной Тьюринга T_1 , то он принимается некоторой детерминированной машиной Тьюринга T_2 .*

Доказательство. Для любого состояния и ленточного символа машины T_1 имеется конечное число вариантов для выбора следующего движения. Варианты могут быть занумерованы числами $1, 2, \dots$.

Пусть r — максимальное число вариантов для любой пары состояние — ленточный символ.

Тогда любая последовательность вариантов движений конечной длины может быть представлена последовательностью цифр от 1 до r .

Не все такие последовательности могут представлять варианты движений, поскольку в некоторых конфигурациях вариантов может быть меньше, чем r .

Можно построить детерминированную машину Тьюринга T_2 , моделирующую машину T_1 . Снабдим её тремя лентами.

Лента 1 будет содержать входную цепочку.

На *ленте 2* машина T_2 будет систематически генерировать последовательность “цифр” от 1 до r , начиная с самой короткой. Среди последовательностей одинаковой длины они генерируются в числовом порядке.

Для каждой последовательности цифр, сгенерированной на *ленте* 2, машина T_2 копирует вход на *ленту* 3 и затем моделирует машину T_1 на *ленте* 3, используя последовательность *ленты* 2 для того, чтобы диктовать движения машины T_1 .

Если машина T_1 входит в принимающее состояние, то машина T_2 также принимает.

Если имеется последовательность вариантов, ведущая к приёму, то она в конце концов будет сгенерирована на *ленте* 2.

Машина T_2 , как модель T_1 , тоже будет принимать входную цепочку.

Но если никакая последовательность вариантов движений машины T_1 не ведёт к приёму входной цепочки, то машина T_2 тоже не примет её.

Заметим, что это доказательство можно обобщить, чтобы показать, как моделировать недетерминированную многоленточную машину Тьюринга обычной (детерминированной) моделью машины Тьюринга.

В пособии рассматриваются многие другие модификации машин Тьюринга, для которых доказано, что они эквиваленты основной модели.