Классическая модель

Модель

Классическая модель (класс) машин Тьюринга определяется следующей структурой:

Бесконечная **лента** (*tape*), каждая **ячейка** (*cell*) которой содержит один символ из **алфавита ленты** Γ (*tape alphabet*). Все неиспользованные ячейки ленты заполняются **пустым символом** $\sqcup \in \Gamma$ (*blank symbol*). Сам алфавит ленты Γ определяется при задании конкретной машины Тьюринга.

Головка чтения-записи (*read-write head*), которая указывает на ячейку ленты, и может считывать символ из текущей ячейки ленты, записывать новый символ в эту ячейку и перемещаться на одну позицию влево (L) или вправо (R) вдоль ленты.

Управляющее устройство (*control unit*), в котором хранится множество состояний Q, текущее состояние q и функция переходов δ, которая определяет поведение машины.

В **множестве состояний** Q при определении конкретной машины Тьюринга должны быть выделены **стартовое состояние** q_0 (*initial state*), **останавливающие состояния** (*halting states*): q_{accept} – **принимающее состояние** (*accepting state*) и q_{reject} – **отвергающее состояние** (*rejecting state*), причем $q_{\text{accept}} \neq q_{\text{reject}}$.

Классическая модель является детерминированной. Форма **функции перехода** (*transition function*) т.е. программы:

$$\delta : (Q \setminus \{q_{\text{accept}}, q_{\text{reject}}\}) \times \Gamma \to Q \times \Gamma \times \{L, R\}$$

Принцип работы машины Тьюринга:

- Машина считывает символ аа а из текущей ячейки ленты.
- Если $q \in \{q_{\text{accept}}, q_{\text{reject}}\}$, машина **останавливается**.
- Иначе применяется функция перехода $\delta(q, a) = (q', a', D)$:
  - Переход в новое состояние q'.
  - Запись символа a' в текущую ячейку.
  - Движение головки: L или R.

Конкретная машина Тьюринга

Прежде чем определить конкретную машину Тьюринга, необходимо определить спецификацию. Конкретная машина Тьюринга в рамках классической модели задается кортежем из 7 элементов:

$$M = (Q,\ \Sigma,\ \Gamma,\ \delta,\ q_0,\ q_{\text{accept}},\ q_{\text{reject}})$$

1. Определяется множество состояний Q, которое содержит $q_0,\ q_{\text{accept}},\ q_{\text{reject}}$.
2. Определяется ленточный алфавит Γ, который представляет собой конечное множество, содержащее пустой символ $\sqcup$.
3. Определяется входной алфавит $\Sigma \subset \Gamma$. Пустой пустой символ обычно не входит во входной алфавит $\sqcup \notin \Sigma$.
4. Определяется программа в виде функции перехода δ в соответствии с формой заданной классической моделью. Функция перехода определяет, на основе состояния и считываемого символ, в какое состояние из Q перейдет машина, какой символ из Γ будет записан на текущую позицию, и какое действие из $\{L, R\}$ будет произведено головкой. Для финальных состояний $q_{\text{accept}}, q_{\text{reject}}$ функция δ не определяется.

Конфигурация машины

Конфигурация (*configuration* or *instantaneous description*) описывает полное состояние машины в данный момент времени:

$$C = uqv$$

где $q \in Q$ – текущее состояние, $u \in \Gamma^*$ – содержимое ленты слева от головки и $v \in \Gamma^*$ – содержимое ленты от головки и правее. Пустые префиксы и суффиксы u и v опускаются.

В начальной конфигурации C_0 головка находится на первом символе входа, который представляет собой последовательность из Σ^*. Остальные ячейки ленты пустые. Машина стартует в состоянии q_0.

Вычисление представляет собой последовательность конфигураций $C_0, C_1, \ldots$, при котором каждый переход $C_i \to C_{i+1}$ определяется функцией перехода δ.

Спецификация машины Тьюринга

Прежде чем определить конкретную машину Тьюринга, следует определить решаемую машиной задачу и установить соглашения о входе-выходе, т.е. семантику и **протокол взаимодействия** (*interface* *protocol*). Таким образом задается **спецификация** работы машины Тьюринга. Спецификация описывает:

- Какую задачу решает машина (какую функцию вычисляет или какой язык распознаёт)
- Формат **входных данных** (*encoding of input*)
- Формат **выходных данных** (*encoding of output*)
- Начальную конфигурацию
- Условия остановки и интерпретацию результата

Конкретная машина Тьюринга данной модели определяется исходя из спецификации. Здесь определяются ленточный алфавит Γ и входной алфавит Σ. Алфавит Γ содержит пустой символ, а $\Sigma \subset \Gamma$ обычно не содержит пустой символ. Определяются множество состояний Q и функция перехода δ (программа). Мощность Q определяется на уровне конкретной машины Тьюринга.

Обобщения и вариации

Машины Тьюринга (*Turing machine*) – это абстрактная вычислительная модель. Помимо классической модели, можно определить неограниченное число моделей, в которых используется несколько лент, или лента с иной геометрией, несколько головок, недетерминированная форма функции перехода и т.д. Конкретная машина Тьюринга всегда относится к той или иной модели машин Тьюринга. В соответствии с тезисом Черча-Тьюринга, всякая модель равносильна по вычислительным возможностям классической модели машин Тьюринга.

При определении модели машин Тьюринга обычно задается следующий набор свойств:

- Характеристики ленты (или лент).
- Характеристики головок чтения-записи.
- Иногда алфавит ленты Γ задается на уровне модели.
- Является ли модель детерминированной.
- Характеристики управляющего устройства. В частности, свойства множества состояний Q и функции перехода δ.
- Принцип работы.
- Могут быть заданы и другие характеристики.

К примеру, вместо двух останавливающих состояний $q_{\text{accept}}, q_{\text{reject}}$ множество Q может характеризоваться только одним останавливающим состоянием или же набором F останавливающих состояний. В последнем случае, если к тому же модель машин Тьюринга вероятностная, то функция перехода может иметь вид (в отличие от классической модели):

$$\delta : (Q \setminus F) \times \Gamma \to \mathcal{P}(Q \times \Gamma \times \{L, R\})$$

где $\mathcal{P}$ определяет распределение вероятностей.

Модель Булоса-Джефри

Модель Булоса-Джефри отличается от классической модели в некоторых моментах:

- Алфавит ленты и входной алфавит совпадают: $\Gamma = \Sigma = \{0, 1\}$, где роль пустого символа играет 0.
- Множество состояний Q использует одно начальное состояние q_1 и одно останавливающее состояние q_n (первый и последний элементы в списке правил перехода).
- Функция перехода имеет вид:

  $$\delta : (Q \setminus \{q_n\}) \times \Gamma \to Q \times A$$

  где $A = \{L, R, S_0, S_1\}$ - набор возможных действий: влево (L), вправо (R), записать ноль (S_0), запись единицу (S_1).

Программа

Программа может быть задана в виде таблицы переходов (*machine table*), графа (*flow chart*) или множества четверок (*set of quadruples*) S C. Четверки образуются из входа и выхода функции перехода. Например, для состояния q_i могут быть определены две четверки, в соответствии с двумя варианты считываемого символа:

$$q_i, 0, a_k, q_j \quad q_i, 1, a_{k'}, q_{j'}$$

При считывании символа 0 производится действие a_k и состояние меняется на q_j, а при считывании символа 1 производится действие $a_{k'}$ и состояние меняется на $q_{j'}$. Для простоты можно записывать с другим вариантом разделителей:

$$q_i 0 a_k q_j, q_i 1 a_{k'} q_{j'}$$

Таким образом всю программу можно записать в виде одной сплошной последовательности.

Также мы хотели бы, для стандартизации, чтобы для каждого состояния, кроме последнего, были определены инструкции для каждого символа. В случае, если для некоторого состояния q_i не определена инструкция при некотором символе s, то в таком случае отсутствие инструкции будет равносильна явной инструкции перехода в состояние q_n с сохранением символа: $q_i s S_s q_n$.

Запись в трех последовательных клетках

Простейшим примером можно рассмотреть машину Тьюринга, которая записывает три идущие подряд единицы. Предполагается, что алфавит ленты $\Gamma = \{0, 1\}$, где 0 – пустой символ. Функция переходов записанная в виде таблицы, графа и квадруплов приведена здесь. Код, эмулирующий такую машину Тьюринга представлен здесь. Работа такой машины начинается на пустой ленте. Производится запись тех единиц в соседствующих ячейках, начиная со стартовой позиции, двигаясь влево. Головка останавливается на крайней левой единице.

Конфигурация

Конфигурация машины задается следующим образом. Пусть для данной машины определено множество состояний Q в виде последовательности $q_0, q_1, \ldots, q_n$. Тогда запись $1_2 100111$ определяет конфигурацию, при которой

- на ленте записана последовательность 1100111;
- головка указывает на крайний левый непустой символ;
- в регистре состояний записано состояние q_2.

Остальные ячейки ленты являются пустыми, так что данная запись эквивалентна также и такой записи: $0001_2 10011100$ – можно добавлять произвольное количество нулей с обоих концов.

Спецификация

Пусть машина Тьюринга решают задачу вычисления функции. Для k-местной функции $f : \mathbb{N}^k \to \mathbb{N}$, $(0 \notin \mathbb{N})$ определим следующую спецификацию работы машины Тьюринга:

(a) Аргументы $n_1, \ldots, n_k$ будут представлены в **унарной системе счисления** (*unary numeral system* or *tally numeration system*) блоками заштрихованных клеток (единиц). Каждый блок отделяется от следующего пустой клеткой – пустым символом 0 $(\Gamma = \{0, 1\})$. Таким образом 3 и 2 будут представлены на ленте как 111011. Все остальные ячейки ленты заполнены пустым символом.

(b) В самом начале машина будет сканировать крайнюю левую заштрихованную клетку на ленте, и начальное состояние будет 1. Таким образом, перед вычислением, скажем $3+2$ начальная конфигурация будет $1_1 11011$. Такая конфигурация называется **стандартной начальной конфигурацией** (*standard initial configuration*).

(c) Если значение функции по аргументам вычислено, то машина остановится на ленте, содержащей блок с ответом, который представляет значение (натуральное число). При вычислении $3+2$ на ленте в итоге будет 11111 (остальные ячейки – пустые).

(d) В таком случае машина остановится на крайней левой заштрихованной клетке ленты. Таким образом, при вычислении $3+2$ финальная конфигурация будет $1_m 1111$, где состояние m соответствует инструкции, для которой не определено действие при считывании 1. Такая конфигурация называется **стандартной финальной конфигурацией** (*standard final configuration*).

(e) Если при вычислении значения функции не удается достичь результата, то машина либо никогда не остановится, либо остановится на нестандартной конфигурации, например, $0_m 11111$ или $0111_m 11$.

С такой спецификацией любая машина Тьюринга может быть использована как для вычисления функции от одного аргумента, двух или в общем случае – k аргументов.

Удваивание числа

Программа, производящая удваивание натурального числа представлена здесь в виде графа.

Вычисление четности числа

Пусть $f: \mathbb{N} \to \{0,1\}$ функция, вычисляющая четность числа: если число четное, то функция возвращает 0, если нечетное – 1. Машина Тьюринга начинает работу считывая самый левый символ в конечной последовательности единиц, кодирующих значение натурального аргумента. Машина останавливается на некоторой позиции, указывая на пустой символ 0, если число четное и на символ 1, если число нечетное (все остальные клетки ленты в заключительной конфигурации содержат пустые символы). Граф программы представлен здесь.

Произведение двух натуральных чисел

Программа, производящая умножение двух натуральных чисел представлена здесь в виде графа.

Частичные, тождественные и пустые фунции

Рассмотрим машину, определенную единственной четверкой $q_1 1 S_1 q_2$. Стартуя в стандартной начальной конфигурации, машина немедленно останавливается, не производя каких либо изменений на ленте. Если на ленте изначально был один блок заштрихованных клеток, то финальная конфигурация будет стандартной, и в таком случае машина вычисляет **тождественную функцию** (*identity function*) от одного аргумента: $\mathrm{id}(n) = n$. В этом случае машина вычисляет **всюду определенную функцию** (*total function*) от одного аргумента. Но если изначально на ленте есть два заштрихованных блока (два аргумента), финальная конфигурация не будет стандартной. В этом случае машина вычисляет экстремально **частично определенную функцию** (*partial function*) от двух аргументов, которая не определена ни для одной пары аргументов: **пустую функцию** (*empty function*) e_2 от двух аргументов. И в общем, для k аргументов эта машина вычисляет пустую функцию e_k от k аргументов.

Здесь представлена, напротив, машина, вычисляющая всюду определенную функцию для любого k, которая возвращает всегда одно конкретное значение, а именно, 1.

Нумерация Гёделя машин Тьюринга

Произведем нумерацию всех машин Тьюринга модели Булоса-Джефри. Будем для определенности использовать спецификацию, в соответствии с которой машина Тьюринга вычисляет функцию $f : \mathbb{N} \to \mathbb{N}$, при этом подразумевается, что $0 \notin \mathbb{N}$. Стартовая конфигурация будет всегда начинаться в состоянии q_1 с одним сплошным блоком из единиц (заштрихованных клеток) и головка будет указывать на крайнюю левую единицу. Все остальные клетки ленты будут пустыми (заполнены нулями). Это соответствует стандартной стартовой конфигурации ❐.

Нумерация, которую мы произведем, в действительности, охватывает все возможные машины Тьюринга данной модели, независимо от спецификации.

Произведем **нумерацию Гёделя** (*Gödel numbering*) машин Тьюринга и соответствующих им функций. Программа, представленная здесь, может быть записана в виде последовательности четверок:

$$q_1 0 R q_3,\ q_1 1 S_0 q_2, q_2 0 R q_1,\ q_2 1 S_1 q_4,\ q_3 0 S_1 q_4,\ q_3 1 S_0 q_2$$

Сам порядок последовательности не имеет значения: программа определяется множеством этих элементов. Заметим, что для состояния остановки q_4 в последовательности нет четверок, так как переходы для него не определяются. Для определенности, расположили правила переходов в порядке возрастания номера состояния, поэтому первые два символа четверок всегда будут предсказуемы: $q_1 0,\ q_1 1,\ q_2 0,\ q_2 1, \ldots, q_n 0,\ q_n 1$. Это обстоятельство позволяет опустить эти префиксы четверок, и записать программу так:

$$R q_3,\ S_0 q_2,\ R q_1,\ S_1 q_4,\ S_1 q_4,\ S_0 q_2$$

Обозначим q_i как i, S_0 как 1, S_1 как 2 (чтобы избежать 0), и L как 3, а R как 4. Тогда программу можно записать еще проще в виде последовательности:

$$4, 3, 1, 2, 4, 1, 2, 4, 2, 4, 1, 2$$

Таким образом, машина Тьюринга может быть полностью представлена как конечная последовательность натуральных чисел, и более того, представлена единственным натуральным числом, например, используя кодирование на основе разложения на простые множители:

$$2^4 \cdot 3^3 \cdot 5 \cdot 7^2 \cdot 11^4 \cdot 13 \cdot 17^2 \cdot 19^4 \cdot 23^2 \cdot 29^4 \cdot 31 \cdot 37^2$$

Обратное, конечно, неверно: не каждому натуральному числу соответствует некоторая машина Тьюринга. Всякое натуральное число имеет уникальное разложение на простые множители, однако далеко не всякая последовательность, которая при этом образуются, определяет машину Тьюринга: большинство последовательностей не будет валидной. Это означает, что мы пронумеровали множество машин Тьюринга с существенными промежутками 210, 420, 630 и т.д. Это и есть нумерация Гёделя.

По мере возрастания номера, мы можем обозначить машины Тьюринга $M_1, M_2, M_3, \ldots$, и каждая такая машина может быть запущена в стандартной стартовой конфигурации, с произвольным натуральным числом на входе. При заданном входе n машина Тьюринга M_i либо останавливается в стандартной конфигурации – выдает значение $f_i(n)$, либо не останавливается или останавливается в нестандартной конфигурации – значение $f_i(n)$ не определено при n. Так, всякой машине Тьюринга соответствует некоторая функция $f_i : \mathbb{N} \to \mathbb{N}$, всюду или частично определенная, а последовательности $M_1, M_2, M_3, \ldots$ соответствует последовательность функций $f_1, f_2, f_3, \ldots$

В этом списке $f_1, f_2, f_3, \ldots$ находятся все функции $f : \mathbb{N} \to \mathbb{N}$, которые могут быть вычислены машинами Тьюринга данной модели. Согласно тезису Черча-Тьюринга, набор этих функций совпадает для любой вычислительной модели, будь это иная модель машин Тьюринга, рекурсивные функции или иные вычислительные модели. Множество функций $f : \mathbb{N} \to \mathbb{N}$ более чем счётно, поэтому существует множество невычислимых функций.

Вот первые три машины M_1, M_2, M_3:

M_1

Первый пример можно получить рассматривая машину, представленную последовательностью $(1,1,1,1)$ или $2 \cdot 3 \cdot 5 \cdot 7 = 210$. В четверках это будет представлено как $q_1 0 S_0 q_1$, $q_1 1 S_0 q_1$. Начиная считывать заштрихованную клетку (1), машина стирает ее и навсегда остается в состоянии q_1, никогда не переходя в состояние остановки q_2. Фактически, машина не останавливается, хотя ничего при этом не меняется: она бесконечно записывает 0 в одну и ту же ячейки.

M_2

Второй пример $(2,1,1,1)$ или $2^2 \cdot 3 \cdot 5 \cdot 7 = 420$, которому соответствует программа $q_1 0 S_1 q_1$, $q_1 1 S_0 q_1$. Стартуя с заштрихованной клетки, машина затирает ее, затем снова заштриховывает, и так далее, и снова машина никогда не останавливается.

M_3

Третий пример $(1,2,1,1)$ или $2 \cdot 3^2 \cdot 5 \cdot 7 = 630$, которому соответствует программа $q_1 0 S_0 q_2$, $q_1 1 S_0 q_1$. Стартуя с заштрихованной клетки, машина затирает ее и переходит в состояние остановки q_2. Головка при этом указывает на пустую клетку. Это означает, что машина завершается в нестандартной финальной конфигурации. Номера Гёделя 210, 420 и 630 – наименьшие числа, которым соответствуют машины Тьюринга, так что именно эти три машины обозначаются M_1, M_2, M_3 и мы имеем три пустые (нигде не определенные) функции $f_1 = f_2 = f_3 = e$.

❐ Диагональная функция – пример невычислимой функции

Универсальная машина Тьюринга

Универсальная машина Тьюринга (*universal Turing machine*, UTM) – это особая машина Тьюринга, которая может симулировать работу любой другой машины Тьюринга заданной модели.

Обычная машина Тьюринга решает одну конкретную задачу – ее программа фиксирована. Универсальная машина Тьюринга U принимает на вход закодированное описание $\langle M \rangle$ произвольной машины Тьюринга M (таблицу переходов), входные данные для этой машины w и симулирует работу машины M на входе w. Таким образом U получает на вход $\langle M \rangle, w$ и вычисляет ту же частичную функцию, что и M:

$$U(\langle M \rangle, w) \simeq M(w)$$

где $\simeq$ означает, что U останавливается и выдает тот же результат, что и M, если M останавливается при w, либо не останавливается, если M не останавливается на w.

Важно определить заранее для данной модели машин Тьюринга, как читается вход/выход, как кодируется пара $\langle M \rangle, w$. В зависимости от этого, определяются переходы в U. Например, $\langle M \rangle$ может представлять собой двоичный код, составленный из последовательности переходов, или же номер гёделевой нумерации. Во всяком случае, парсер U должен суметь восстановить функцию перехода M.

Универсальная машина Тьюринга служит интерпретатором всех машин Тьюринга заданной модели, включая саму себя $U(\langle U \rangle, x)$; в роли x выступает некоторая последовательность $\langle M \rangle, w$. Нет проблем и в том,

```
чтобы универсальной машиной симулировать универсальную машину,
симулирующую саму себя и т.д.
```