Практика №1 по курсу «Теория алгоритмов»
«Машина Тьюринга»

Группы ФТ-301, ФТ-302

1 Определения

Неформально машина Тьюринга (МТ) состоит из трёх основных компонентов:

- Бесконечная лента, на которой записываются входные данные, результат работы машины и все промежуточные вычисления;
- Каретка — устройство, указывающее на одну конкретную ячейку бесконечной ленты и позволяющее считывать и записывать данные из этой ячейки;
- Состояние машины Тьюринга — определяет текущее поведение и реакцию машины Тьюринга на каждый символ, который она встретит на ленте.

Формально можно определить МТ как кортеж из пяти элементов $M = \langle \Gamma, Q, \delta, q_{start}, q_{finish} \rangle$, где каждый из компонентов имеет следующее значение:

- Γ — множество символов, которые могут быть записаны в ячейках ленты МТ, среди которых всегда есть символ пустой ячейки $\lambda \in \Gamma$;
- Q — конечное множество состояний машины Тьюринга;
- $\delta : (\Gamma \times Q) \mapsto (\Gamma \times Q \times \{\leftarrow, \uparrow, \rightarrow\})$ — функция перехода, которая по заданной паре (символ на ленте, состояние машины) определяет тройку: (символ, который необходимо записать в текущую ячейку, новое состояние МТ, сдвиг карертки МТ);
- q_{start}, q_{finish} — начальное и конечное состояния МТ.

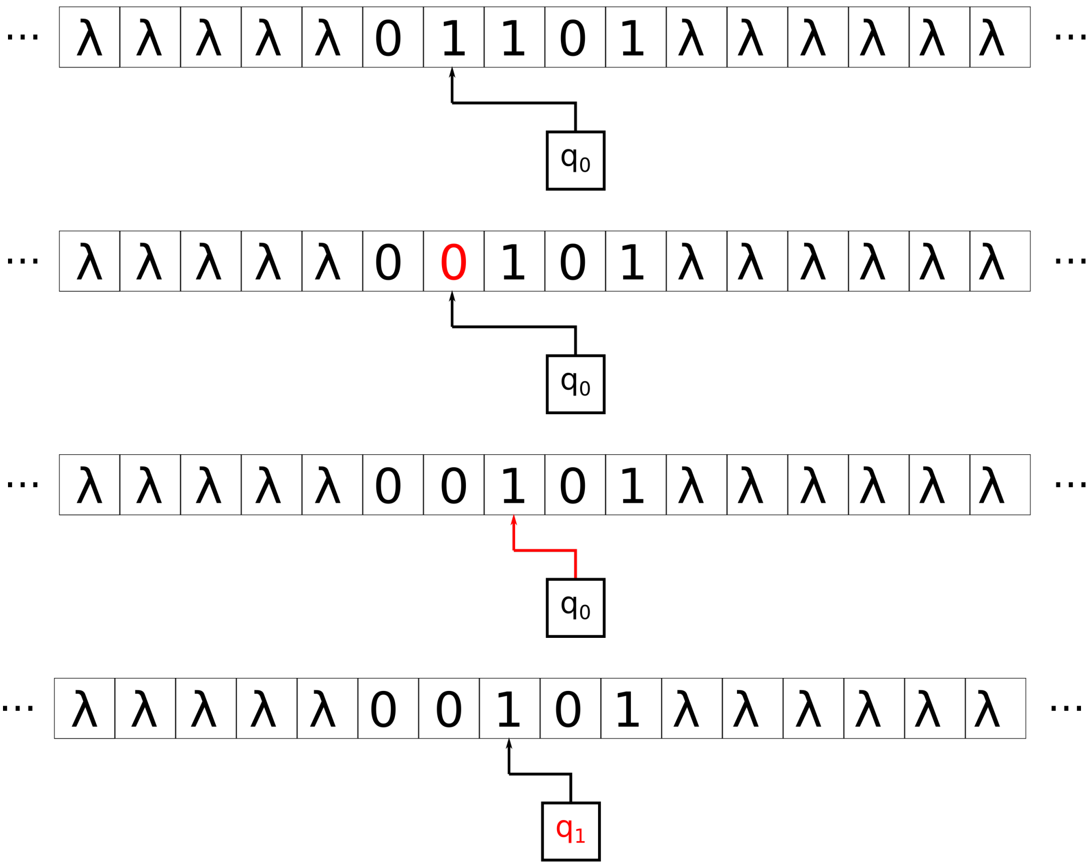

Рис. 1: Пример одной итерации работы МТ, где $\Gamma = \{0, 1, \lambda\}$ и $\delta(1, q_0) = (1, q_1, \rightarrow)$

Можно считать, что в начальный момент времени каретка МТ находится в ячейке первого символа входных данных, а после работы МТ каретка должна указывать на первую ячейку, откуда слева направо можно прочитать выходные данные. Дополнительно можно принять соглашение о том, что слева от конечного положения каретки МТ должны быть только пустые символы.

2 Задача «Чётность числа единиц в строке»

Для примера построим МТ, которая в строке из нулей и единиц будет вычислять значения количества единиц по модулю 2.

Принцип работы МТ для этой задачи простой — необходимо считать строку слева направо, поддерживая при этоя чётность количества единиц для каждого префикса. Для поддержания этого значения можно завести пару отдельных состояний `even` и `odd`.

Таким образом легко определить следующие компоненты МТ:

- $\Gamma = \{0, 1, \lambda\}$;
- $Q = \{\texttt{even}, \texttt{odd}, \texttt{finish}\}$;
- $q_{start} = \texttt{even}, q_{finish} = \texttt{finish}$.

Функция перехода действует тоже достаточно просто — при обнаружении символа 0 под кареткой, состояние не меняется, а при обнаружении символа 1 — состояние инвертируется и каретка в любом случае двигается вправо. Полное множество переходов можно записать в виде таблицы следующим образом:

состояние \ символ	0	1	λ
`even_count`	$\lambda \rightarrow$	$\lambda \rightarrow$ `odd`	0 `finish`
`odd_count`	$\lambda \rightarrow$	$\lambda \rightarrow$ `even`	1 `finish`

Для описания перехода для конкретной пары $(c, q) \in \Gamma \times Q$ в таблице используется порядок (символ, направление, новое состояние), где каждую из компонент можно опустить, в результате чего будет использовано поведение по умолчанию (по умолчанию МТ не меняет символ, который написан в текущей ячейке, стоит на месте и остается в текущем состоянии).

Упражнение: Задача B из контеста про машины Тьюринга. Чтобы получить доступ к контесту отправь свой `JudgeID` на сайте `acm.timus.ru` в письме с темой «`Теория алгоритмов. Контест. %ФИО%. %Группа%`» на почту `sivukhin.nikita@yandex.ru`. Чтобы сдать задачу в этом контесте необходимо описать МТ в следующем формате:

```
[количество состояний, |Q|]
[идентификаторы состояний МТ]...
[количество определённых значений функции перехода]
[состояние] [символ] -> [новое состояние] [новый символ] [направление|L, S, R]...
```

3 Распознаватель

Грубо говоря можно считать, что МТ построенная по определению из предыдущего раздела, задаёт функцию $f : \Sigma^* \mapsto \Sigma^* \cup \{\perp\}$, где $\Sigma = \Gamma \setminus \{\lambda\}$, а $\perp$ — специальный символ, который обозначает, что машина Тьюринга зациклилась или сломалась на заданном входе (мы будем стараться строить МТ, которые не ломаются и не зацикливаются, но формально никто не запрещает подобрать такую функцию перехода, которая будет зацикливаться или ломаться на некоторых входных данных).

Иногда имеет ограничить выходные значения МТ, для упрощения её построения и анализа. Для этого модифицируем формальное определение МТ, заменив одного терминальное состояние q_{finish} на пару терминальных состояний: q_{accept} и q_{reject}. В случае завершения своей работы в состоянии q_{accept} МТ «принимает» исходную входную строку, а в случае завершения работы в q_{reject} — отвергает. Такая модификация определяет МТ-распознаватель, который задаёт функцию $f : \Sigma^* \mapsto \{0, 1, \perp\}$.

4 Задача «Копия исходной строки»

Задача: Построим МТ, которая по входной строке $s \in \{0, 1\}^*$ построит на ленте строку $s\#s$.

Процесс работы МТ достаточно прост: машина сначала должна поставить решётку после конце исходной строки, после чего начать копировать символы слева направо. Значение очередного символа для копирования сохраняется в состоянии, а чтобы МТ смогла вернуться к следующему символу можно временно стереть символ в исходной строке, который в данный момент копируется. Формально, МТ будет выглядеть следующим образом:

- $\Gamma = \{0, 1, \lambda\}$;
- $Q = \{\texttt{sharp}, \texttt{unk}, \texttt{set}, \texttt{left\_0}, \texttt{left\_1}, \texttt{right\_0}, \texttt{right\_1}, \texttt{finish}\}$;

- $q_{start} = \texttt{sharp}, q_{finish} = \texttt{finish}$.

состояние \ символ	0	1	#	λ
`sharp`	$\rightarrow$	$\rightarrow$	X	$\# \leftarrow$ `unk`
`unk`	$\leftarrow$	$\leftarrow$	X	$\rightarrow$ `set`
`set`	$\lambda \rightarrow$ `right_0`	$\lambda \rightarrow$ `right_1`	`finish`	X
`right_0`	$\rightarrow$	$\rightarrow$	$\rightarrow$	$0 \leftarrow$ `left_0`
`right_1`	$\rightarrow$	$\rightarrow$	$\rightarrow$	$1 \leftarrow$ `left_1`
`left_0`	$\leftarrow$	$\leftarrow$	$\leftarrow$	$0 \rightarrow$ `set`
`left_1`	$\leftarrow$	$\leftarrow$	$\leftarrow$	$1 \rightarrow$ `set`

Таблица 1: недостижимые состояния МТ обозначаются крестиком(X) в таблице переходов

Заметим, что функция перехода δ упрощена, так как в конце своей работы каретка МТ находится в ячейке с символом $\#$. Чтобы передвинуть МТ к началу выходных данных понадобятся ещё состояния.

Упражнение на дом: решить задачу дублирования без символа $\#$. Задача C из контеста про машины Тьюринга

5 Задача «Проверить строку на палиндромность»

*Палиндром, **перевертень** — число, буквосочетание, слово или текст, одинаково читающееся в обоих направлениях*

Задача: Построить распознаватель, который определит — является ли записанная на ленте строка $s \in \{0, 1\}^*$ палиндромом или нет?

Для решения задачи достаточно построить МТ, которая будет сравнивать между собой символы с обоих концов строки. Для удобства лучше стирать символы, которые были ужа сравнены, чтобы уменьшить размер описания МТ.

Упражнение на дом: написать таблицу переходов для решения этой задачи. Задача D из контеста про машины Тьюринга.

6 Задача «A + B»

Задача: Построить машину Тьюринга, которая для пары двоичных чисел, записанных через знак `+` вычислит их сумму и запишет на ленту в двоичной системе счисления.

Для решения этой задачи полезно расширить множество допустимых символов, добавив туда помеченные символы для двоичных цифр: $\Gamma = \{0, 1, 0', 1', \lambda, +\}$. С помощью данных меток МТ может «запоминать» позицию разряда в одном числе, пока происходит процесс обработки второго числа.

Упражнение на дом: написать таблицу переходов для решения этой задачи. Задача E из контеста про машины Тьюринга.

7 Поиграем со множеством перемещений МТ

Рассмотрим три модификации МТ, которые будут отличаться только определением функции перехода:

- MT_1: машины Тьюринга такие, что $\delta : \Gamma \times Q \mapsto \Gamma \times Q \times \{\leftarrow, \uparrow, \rightarrow\}$
- MT_2: машины Тьюринга такие, что $\delta : \Gamma \times Q \mapsto \Gamma \times Q \times \{\leftarrow, \rightarrow\}$
- MT_3: машины Тьюринга такие, что $\delta : \Gamma \times Q \mapsto \Gamma \times Q \times \{\uparrow, \rightarrow\}$

Понятно, что класс функций, которые можно вычислить с помощью MT_1, не меньше, чем классы функций, которые можно вычислить с помощью MT_2 и MT_3. Постараемся ответить на такой вопрос: существует ли функция f, которую можно вычислить с помощью некоторой МТ из класса MT_1, но нельзя вычислить с помощью МТ из класса MT_2. Легко заметить, что такой функции не существует, потому что любую МТ $T \in MT_1$ можно легко преобразовать в T' таким образом, чтобы функция переходов всегда передвигала каретку либо вправо либо влево, а значит $T' \in MT_2$. Идея преобразования достаточно простая — для каждого перехода с неподвижной кареткой необходимо сделать два последовательных перехода, где каретка двигалась бы в две противоположные стороны. Формально, данное преобразование можно описать следующим образом:

- $T = \langle \Gamma, Q, \delta, q_{start}, q_{finish} \rangle$
- $T' = \langle \Gamma, Q', \delta', q_{start}, q_{finish} \rangle$
- $Q' = Q \cup \{q' \mid q \in Q\}$
-

$$\begin{aligned}\delta' =& \{(c_1, q_1) \to (c_2, q_2, d) \mid (c_1, q_1) \to (c_2, q_2, d) \in \delta \wedge d \in \{\leftarrow, \to\}\} \cup \\ & \{(c_1, q_1) \to (c_2, q_2', \leftarrow) \mid (c_1, q_1) \to (c_2, q_2, \uparrow) \in \delta\} \cup \\ & \{(c_1, q') \to (c_1, q, \to) \mid q \in Q\}\end{aligned}$$

Таким образом классы машин Тьюринга MT_1 и MT_2 эквиваленты с точки зрения функций, которые они могут вычислить.

Несложно также понять, что класс MT_3 неэквивалентен классам MT_1 и MT_2. Если внимательно присмотреться, то МТ из MT_3 представляют из себя конечные автоматы и для доказательства неэквивалентности классов достаточно привести пример нерегулярного языка, который можно распознать с помощью обычной МТ (например язык палиндромных строк).

8 Упражнения на дом по желанию

- Написать МТ, распознающую степени двойки, записанные в унарной системе счисления (задача F из контеста про машины Тьюринга)
- Написать МТ, подсчитывающую количество вхождений подстроки в строку и выводяющую данное число в двоичной системе счисления (задача G из контеста про машины Тьюринга)

Писать таблицы переходов может быть утомительно, поэтому можно попробовать упростить себе задачу, написав несложный транслятор из примитивного языка программирования в формальное определение МТ. Пример синтаксиса, который можно было бы поддержать (за основу взять `javascript`):

```
export const Gamma = ['0', '1', '_'];
let sum = '0';
while (head != '_') {
    let current = sum;
    if (head == '1' && current == '0') {
        sum = '1';
    }
    if (head == '1' && current == '1') {
        sum = '0';
    }
    write('_');
    move_right();
}
write(sum);
```