

# Преобразование алгоритма в терминах нормальной схемы алгоритма Маркова в машину Тьюринга

## Алгоритм преобразования

Докажем, что любой алгоритм Маркова можно преобразовать в машину Тьюринга.

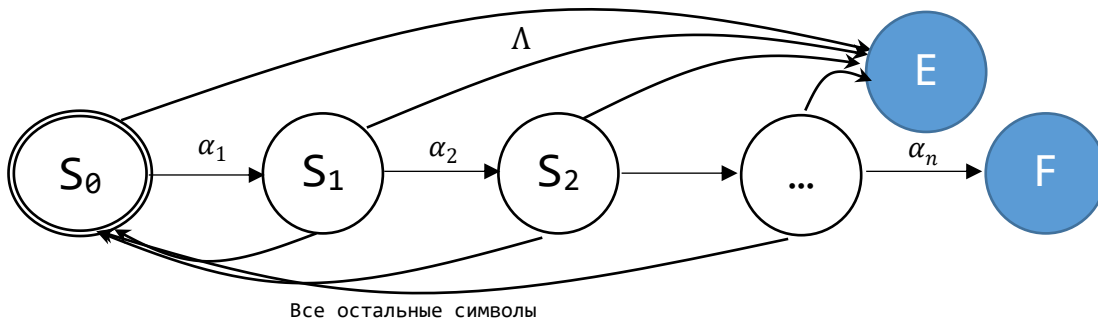
Пусть нам дан алгоритм Маркова, определяемый конечным алфавитом  $\Sigma$ , набором преобразований  $P$  и конечным преобразованием  $P_0$ .

Мы можем построить конечный автомат  $A$ , который ищет в строке слово  $\alpha$ , а также машину Тьюринга  $TM$ , которая заменяет в строке некоторое слово  $\alpha$  на некоторое слово  $\beta$ .

Для дальнейших рассуждений введём обозначение  $q_0$  — начальное состояние нашей машины Тьюринга, реализующей данный алгоритм.

### Поиск слова

Для слова  $\alpha = \alpha_1 \alpha_2 \dots \alpha_n$  автомат  $A$  примет такой вид:



Его конечное состояние  $F$ , означающее, что слово найдено, будет начальным для машины  $TM$ , а достижение состояния  $E$  будет означать, что автомат дошёл до конца строки, но слова не нашёл (после этого необходимо вернуться в начало строки и перейти на начальное состояние автомата поиска следующего слова).

$s_0 \alpha_1 \rightarrow s_1 \alpha_1 R$

$s_1 \alpha_2 \rightarrow s_2 \alpha_2 R$

...

$s_{(n-1)} \alpha_n \rightarrow F \alpha_n E$

Также для каждого состояния будет сгенерированы команды:

$s \Lambda \rightarrow E \Lambda L,$

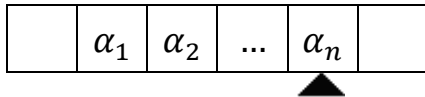
отвечающая за переход к состоянию ошибки по достижению конца строки, и набор команд

$s_i \alpha_j \rightarrow s_0 \alpha_j E,$

где  $j \neq i$  для  $\forall j \in \Sigma$  отвечающая за возврат в начальное состояние, если попал не тот символ, что ожидался.

### Замена слова

В начале работы машина  $TM$  будет находиться в конце слова, которое необходимо заменить.



Поскольку слова  $\alpha$ , длина которого  $n$ , и  $\beta$ , длина которого  $m$ , при построении машины известны, можно рассмотреть три случая:

1)  $n = m$

В этом случае нам нужны  $n$  команд для замены символов одного слова на другое с конца, где последняя команда также установит состояние машины на  $q_0$  (начальное состояние машины)

2)  $n < m$

В этом случае вначале необходимо сдвинуть остаток строки на  $(m - n)$  ячеек влево (подробнее о сдвиге ниже), после чего начать запись справа налево нового слова.

3)  $n > m$

Необходимо сдвинуть остаток строки на  $(m - n)$  ячеек вправо, вернуться в конечную ячейку нового слова и записать его справа налево.

### Сдвиг «хвоста» строки

Сдвиг остатка строки на ленте машины Тьюринга на  $n$  ячеек можно реализовать  $n$  повторением сдвига на одну клетку. Рассмотрим сдвиг вправо.

Машина будет находиться в начальном состоянии  $q_{start}$  Первой командой машины Тьюринга для сдвига строки будет:

$q_{start} \ c \rightarrow q_c \ \wedge \ R,$

где  $c$  — первый символ сдвигаемого «хвоста» строки.

Для каждого символа  $c$  алфавита  $\Sigma$  будет сгенерирован набор команд:

$q_c \ b \rightarrow q_b \ c \ R,$

где  $b$  — каждый символ из алфавита  $\Sigma - \{c\}$ .

Таким образом мы записываем значение символа, которое храним в состоянии машины и запоминаем в состоянии значение, которое было в этой ячейке, сдвигаясь вправо.

Также для каждого символа  $c$  алфавита  $\Sigma$  будет сгенерирована команда:

$q_c \Lambda \rightarrow q_{end} \ c \ E,$

где  $q_{end}$  — состояние конца строки.

Теперь необходимо вернуться в ячейку, с которой мы начали сдвигать «хвост». Для этого нам понадобится набор команд

$q_{end} \ c \rightarrow q_{end} \ c \ L,$

для каждого  $c \in \Sigma$ , что означает, что мы двигаемся влево, пока не встречаем пустой символ:

$q_{end} \ \Lambda \rightarrow q_{end2} \ \Lambda \ E$

После этого машина приходит в состояние  $q_{end2}$ , означающее конец её работы.

Для сдвига «хвоста» влево все движение влево нужно заменить на движения вправо и наоборот.

Проблема сдвига «хвоста» решена.

### **Построение машины Тьюринга для замены одного преобразования нормального алгоритма Маркова**

Каждое преобразование нормального алгоритма — это замена одной подстроки на другую. Эта задача разбивается на поиск подстроки и на её замену, реализация которых описана выше. Соединим эти два процесса.

Итак, нам необходимо заменить слово  $\alpha = \alpha_1 \alpha_2 \dots \alpha_n$  на слово  $\beta = \beta_1 \beta_2 \dots \beta_m$ .

Начальное состояние нашей машины обозначим  $prod1\_s\_0$ . Построим машину, описанную в разделе «Поиск слова»:

$prod1\_08.01.2018s\_0 \ \alpha_1 \rightarrow prod1\_s\_1 \ \alpha_1 \ R$

$prod1\_s\_1 \ \alpha_2 \rightarrow prod1\_s\_2 \ \alpha_2 \ R$

...

$prod1\_s\_(n-1) \ \alpha_n \rightarrow prod1\_found \ \alpha_n \ E$

$\forall prod1\_s \in S, \forall \alpha_i, \alpha_j \in \Sigma, j \neq i$ , где  $S$  — набор состояний машины поиска.

$prod1\_s \ \Lambda \rightarrow prod1\_error \ \Lambda \ L,$

$\text{prod1\_s\_i } \alpha_j \rightarrow \text{prod1\_s\_0 } \alpha_j \text{ E,}$

Нам понадобятся также обработка состояния ошибки  $\text{prod1\_error}$ :

$\text{prod1\_error } \Lambda \rightarrow \text{prod2\_s\_0 } \Lambda \text{ R}$

$\forall \alpha \in \Sigma$ :

$\text{prod1\_error } \alpha \rightarrow \text{prod1\_error } \alpha \text{ L,}$

где  $\text{prod2\_s\_0}$  — начальное состояние машины поиска следующего преобразования алгоритма Маркова.

Теперь необходимо запустить замену слова  $\alpha$  на слово  $\beta$ . Мы находимся в ячейке с символом  $\alpha_n$  в состоянии  $\text{prod1\_found}$ .

Если длина второго слова больше первого на  $(m - n) = d$  символов, то необходимо  $d$  раз сдвинуть «хвост» строки вправо. Построим машину Тьюринга, сдвигающую «хвост», с начальным состоянием  $\text{prod1\_found}$ .

Первая команда нашей машины будет:

$\text{prod1\_found } \alpha_n \rightarrow \text{prod1\_shift1\_c } \Lambda \text{ R,}$

где  $c = \alpha_n$ .

Затем будет набор команд:

$\text{prod1\_shift1\_c } b \rightarrow \text{prod1\_shift1\_b } c \text{ R,}$

Где  $b$  — каждый символ из алфавита алгоритма.

Последней командой будет

$\text{prod1\_shift1\_c } \Lambda \rightarrow \text{prod1\_shift1\_end } c \text{ E,}$

где  $\text{prod1\_shift1\_end}$  — состояние конца строки.

Теперь необходимо вернуться в ячейку, с которой мы начали сдвигать «хвост». Для этого нам понадобится набор команд

$\text{prod1\_shift1\_end } c \rightarrow \text{prod1\_shift1\_end } c \text{ L,}$

для каждого  $c \in \Sigma$ , что означает, что мы двигаемся влево, пока не встречаем пустой символ:

$\text{prod1\_shift1\_end } \Lambda \rightarrow \text{prod1\_shift1\_end2 } \Lambda \text{ E}$

После этого машина приходит в состояние  $q\_end2$ , означающее конец её работы.

В случае, если необходимо сдвинуть «хвост» более, чем на одну позицию, переходим на начальное состояние аналогичной машины  $\text{prod1\_shift2\_start}$ .

$\text{prod1\_shift2\_start } c \rightarrow \text{prod1\_shift2\_c } \Lambda$

Далее аналогично.

Для последнего сдвига состояние  $\text{prod1\_shiftL\_end2}$  будет начальным для машины, записывающей слово  $\beta$ :

$\text{prod1\_shiftL\_end2 } \Lambda \rightarrow \text{prod1\_write\_m } \beta_m \text{ } L$

$\text{prod1\_write\_m } * \rightarrow \text{prod1\_write\_}(m-1) \beta_{m-1} \text{ } L,$

где  $*$  — любой символ (значит, таких команд будет столько же, сколько символов в алфавите).

Машина закончит свою работу в состоянии  $\text{prod1\_write\_1}$ . Вернём каретку на начало строки.

$\text{prod1\_write\_1 } * \rightarrow \text{prod1\_write\_1 } * \text{ } L,$

где  $*$  — любой символ (значит, таких команд будет столько же, сколько символов в алфавите).

Если преобразование алгоритма Маркова, которое мы разбираем, не конечное, установим начальное состояние машины.

$\text{prod1\_write\_1 } \Lambda \rightarrow \text{prod1\_s\_0 } \Lambda \text{ } R$

Если преобразование конечное («с точкой»), установим конечное состояние машины:

$\text{prod1\_write\_1 } \Lambda \rightarrow q\_end \text{ } \Lambda \text{ } R$

### **Построение машины Тьюринга, выполняющей нормальный алгоритм Маркова**

Чтобы получить конечную машину Тьюринга, выполняющую заданный алгоритм, необходимо каждое преобразование алгоритма Маркова «разобрать» на машину Тьюринга для поиска и замены слов данного преобразования.

Начальным состоянием всей машины будет  $\text{prod1\_s\_0}$  — начальное состояние машины поиска слова из первого преобразования. Переход к последующему преобразованию будет происходить через состояние  $\text{prodX\_error}$ , означающему, что искомое слово в строке не найдено, и нужно искать следующее.

Таким образом, можно построить такую машину Тьюринга  $TM$ , что  $TM = \{Q, \Sigma, E, q_0, F\}$ , где  $Q$  — конечный набор состояний машины, складываемый из состояний всех автоматов поиска и машин замены слова,  $\Sigma$  — алфавит данного алгоритма Маркова,  $E$  — конечный набор преобразований машины Тьюринга, складывающийся из описанных выше преобразований для каждого преобразования нормального алгоритма,  $q_0$  — начальное состояние машины (`prod1_s_0`),  $F$  — набор финальных состояний машины Тьюринга.

## Литература

1. CHEN Yuanmi “Markov Algorithm”, 2007  
(<https://www.irif.fr/~carton/Enseignement/Complexite/ENS/Redaction/2007-2008/yuanmi.chen.pdf>)