

# Описание алгоритма

Вдовин Максим

25 ноября 2020 г.

Дано  $\alpha$  и слово  $u \in \{a, b, c\}^*$ . Нужно найти длину наибольшего префикса  $u$ , принадлежащего  $L$ .

Решать эту задачу буду методом динамического программирования. Пусть у нас есть строки:  $reg$  - регулярное выражение,  $str$  - слово, для которого нужно найти наибольший общий префикс принадлежащий языку. Также пусть длинна  $|reg| = m$ , а  $|str| = n$ . Суть алгоритма в том, чтобы для каждого символа регулярки понять какую часть слова она распознаёт. Заведём массив булей  $dp[l][i][j]$ , где  $l$  - номер текущего символа из  $reg$ ,  $i$  и  $j$  - левая и правая граница, той части слова, которая считается регулярным выражением. Значение  $true$  в  $dp[l][i][j]$  означает, что  $reg[l]$  принимает в себя подслово  $[i, j]$ .

Далее я покажу как подсчитать  $dp$  в зависимости от разных случаев  $reg[l]$

1. Если  $reg[l] \rightarrow 1$ , то  $\forall i \quad dp[l][i][i] = true$ .

Очевидно, что таким образом мы задаём пустое слово.

2. Если  $reg[l] \rightarrow$  буква, то  $\forall i$ , если  $(str[i] == reg[l])$  мы записываем  $dp[l][i][i+1] = true$
3. Если  $reg[l] = "+"$ , то мы будем искать такие  $l_0$  и  $l_1$ , что они являются операндами для  $reg[l]$ . Далее подсчитаем  $\forall i, j \quad dp[l][i][j] = dp[l_0][i][j] \parallel dp[l_1][i][j]$
4. Если  $reg[l] = "."$ , то мы будем искать такие  $l_0$  и  $l_1$ , что они являются операндами для  $reg[l]$ . Далее подсчитаем возьмем некоторый разделитель —  $pivot$ , который будет изменяться  $\forall i, j$ , так что  $i \leq pivot \leq j$ . Массив  $dp$  теперь будет считаться по следующей формуле:

$$\forall i, j \quad dp[l][i][j] = dp[l][i][j] \parallel (dp[l_0][i][pivot] \parallel dp[l_1][pivot][j])$$

.

Здесь мы как бы говорим, что подслово  $str[i..j]$  может быть получено из регулярного выражения, соответствующего  $reg[l]$ , если существует разбиение этого слова на два слова:  $str[i..j] = str[i..pivot]str[pivot..j]$  и каждое из этих двух слов может быть получено из соответствующих операндов для  $reg[l]$ .

5. Если  $reg[l] = "*" \rightarrow$ , то мы будем искать такое  $l$ , что оно является операндом для  $reg[l]$ . Далее подсчитаем возьмем некоторый разделитель —  $pivot$ , который будет изменяться  $\forall i, j$ , так что  $i \leq pivot \leq j$ . Массив  $dp$  теперь будет считаться по следующей формуле:

$$\forall i, j \quad dp[l][i][j] = dp[l][i][j] \parallel (dp[l_0][i][pivot] \parallel dp[l_1][pivot][j])$$

Здесь мы как бы говорим, что подслово  $str[i..j]$  может быть получено из регулярного выражения для  $reg[l]$ , если существует разбиение этого слова на несколько слов:  $str[i..j] = str[i..i_1]str[i_1..i_2] \dots str[i_k..j]$  и каждое из этих слов может быть получено из регулярного выражения для  $reg[l_0]$ . Это в свою очередь эквивалентно тому, что существует разбиение слова  $str[i..j]$  на два слова:  $str[i..j] = str[i..i_l]str[i_l..j]$  и первое слово может быть получено из регулярного выражения для  $reg[l]$ , а второе из РВ для  $reg[l_0]$ .