

Практикум №1

Регулярные выражения

Мещеряков Евгений, 696

8 ноября 2017г.

Задача 8: Даны α , буква x и натуральное число k , где α - регулярное выражение в обратной польской записи, задающее язык L . Вывести длину кратчайшего слова из языка L , содержащего суффикс x^k .

Алгоритм

Пусть $degree = k$. Будем оперировать объектами класса *Lexeme*:

```
1 class Lexeme:
2     def __init__(self, id, operation, operands):
3         self.id = id
4         self.operation = operation
5         self.operands = operands
6         self.shortest_word_len_with_prefix = []
```

Преобразуем обратную польскую запись регулярного выражения в лист лексем с помощью стека, попутно присваивая каждой лексеме приоритет вычисления (задается листом *calculation_priority* пар $[lexeme_id, priority]$). Сводим нашу задачу к задаче поиска кратчайшего слова с префиксом x^k путём обращения листа *operands* в каждой лексеме с конкатенацией. Полученную задачу решаем динамикой.

Каждая лексема имеет лист *shortest_word_len_with_prefix* размера $degree + 1$. Для краткости обозначим *shortest_word_len_with_prefix* как *list.prefix_len* = длина кратчайшего слова, принадлежащего языку, задаваемому лексемой с префиксом x^k . Вычисление *list.degree* для лексемы, соответствующей всему регулярному выражению дает ответ на задачу. Переберём все лексемы в порядке приоритета, вычисляя для каждой *list*. Для букв и пустого слова вычисление тривиально. Разберем вычисление для операций.

- '+': Перебираем *prefix_len*. Если для обоих операндов значение *list.prefix_len* определено, то выбирается минимум значений. Если значение *list.prefix_len* определено лишь для одного операнда, то выбирается оно. Если для обоих операндов значения не определены, то значение *list.prefix_len* для текущей лексемы тоже не определено.

- ‘.’: Перебираем индексы f_prefix_len и s_prefix_len (f – *first*, s – *second*) для левого и правого операндов конкатенации. Обозначим листы операндов и текущей лексемы соответственно $list_f$, $list_s$ и $list_res$. Вычисление производится согласно утверждениям:

Пусть значения $list_f[f_prefix_len]$ и $list_s[s_prefix_len]$ определены, тогда:

1) $list_res[f_prefix_len] \leq list_f[f_prefix_len] + list_s[s_prefix_len]$ (префикс конкатенации равен префиксу левого операнда конкатенации)

2) Пусть $prefix_len = \min(f_prefix_len + s_prefix_len, degree)$.

Если $f_prefix_len = list_f[f_prefix_len]$, то

$list_res[prefix_len] \leq f_prefix_len + list_s[s_prefix_len]$.

Действительно, если $f_prefix_len = list_f[f_prefix_len]$, то конкатенируемая слева строка есть $x^{f_prefix_len}$, а, значит, конкатенируя справа строку с префиксом длины s_prefix_len , получаем строку с префиксом длины $f_prefix_len + s_prefix_len$.

После перебора, проходя по $list$ текущей лексемы от конца к началу, восстанавливаем следующий ивариант: Пусть значения $list[i]$ и $list[k]$ определены, тогда: $i \leq k \Rightarrow list[i] \leq list[k]$ (Слово, содержащее префикс длины k , содержит префикс меньшей длины)

- ‘*’: Инициализируем $list$ текущей лексемы $listom$ её операнда, так же присваиваем $list[0]$ нуль, так как пустое слово лежит в языке вида L^* . Далее вычисляем $list$ для степеней $2...degree$ операнда алгоритмом для конкатенации.