

Алгоритм нахождения минимальной длины слова, содержащего подслово x^k и лежащего в языке заданом данным регулярным выражением

Ягудин Амир

7 Ноября 2018

1 Вступление

Данный алгоритм получает на вход регулярное выражение в обратной польской записи и парсит его, посимвольно считывая ввод и кладя прочитанный символ на стек. Опишем структуру **language** которая будет характеризовывать какое-то подвыражение данного регулярного выражения. Соответственно алгоритм будет хранить такие структуры на стеке и, считывая оператор валентности k , будет доставать со стека последние k структур и класть вместо них одну обновленную. Переменные сами по себе представляют валидные структуры **language**.

2 Описание алгоритма

2.1 Описание структуры

Будем хранить в структуре:

- **min_word** - минимальную длину слова в языке, задаваемом данным выражением.
- **min_length** - минимальная длина слова содержащего x^k .

- **complete_lengths** - массив всевозможных длин слов вида x^* , принадлежащих языку (не больших $k + 5$).
- **prefix_length, suffix_length** массивы содержащие минимальную длину слова с суффиксом длины $t < k + 5$.

2.2 Описание операций

При считывании со стека, при нехватке операндов, будем выводить сообщение об ошибке **error**, а также если в конце мы не получим ровно одно выражение.

При описании операций станет понятно, что нам не нужно хранить отдельно информацию о префиксах или суффиксах содержащих больше, чем $k + 5$, следовательно размеры массивов будут линейны по k

При отсутствии слов какого-то вида будет говорить, что их минимальная длина есть **INF**.

В конце ответ будет храниться в поле **min_length**.

Опишем операции которые будут выполнять каждый из возможных операндов:

- **+** При объединении двух языков обновление полей тривиально и выполняется за $O(k)$
- **.** При конкатенации, длины слов x^* и длина минимального слова обновляются тривиально за $O(k^2)$ При обновлении префикса длины t мы смотрим на минимальную длину слова с префиксом t левого операнда и минимальную длину слова правого операнда. Либо (если левое слово лежит полностью в префиксе), на всевозможные длины слов вида x^* из левого и на всевозможные префиксы правого. Так обновляем минимальные длины за $O(k^2)$. Аналогично суффиксы. Ответ инициализируется исходным, а затем обновляется при переборе всех конкатенаций суффикса левого и префикса правого за $O(k^2)$
- ***** При зымакании длины x^* насчитываются с помощью дп по возрастанию длин за $O(k^2)$. Префиксы и суффиксы насчитываются аналогично дп по возрастанию, так как мы можем понять, что (при насчитывании префикса), если самое левое слово используется, то

имеет вид x^* следовательно его можно отрезать (либо осталось одно слово, для которого динамика посчитанна, тк это слово лежит в исходном языке) и применить посчитанное значение динамики, а именно прибавить длину минимального слова, содержащего определенный префикс t . Минимальная длина слова становится 0. Ответ же инициализируется исходным ответом, а затем обновляется при переборе всех конкатенаций суффикса и префикса. Все за $O(k^2)$.

3 Заключение

Получаем, что ответ насчитывается за $O(len(reg_exp) * k^2)$