Общая надстрока наименьшей длины

Андрей Осипов

15 декабря 2013 г.

1 Постановка задачи

Дан набор строк $S = \{s_1, \ldots, s_n\}$ над конечным алфавитом. Требуется найти, строку s минимальной длины, содержащую как подстроку каждую строку из данного набора.

Пусть язык L это множество пар вида (S, k) для которых верно, что такая строка s существует, и имеет длину не больше k. Тогда задача разрешения языка L является NP-полной. Доказательство этого факта будет приведено позже. А пока, мы ослабим условие следующим образом: пускай теперь требуется найти такую строку t, что она так же как и s содержит всякую строку из S как подстроку, и при этом $|t| \leq 4 * |s|$

2 Алгоритм

Алгоритм для решения этой задачи на первый взгляд может показаться крайне наивным. Но в дальнейшем выяснится, что этого вполне достаточно для достижения даже такой близкой границы. Более того, на практике полученный ответ разрастается не более чем вдвое.

Итак, без ограничения общности будем считать, что среди строк из S нет таких двух x и y, что x подстрока y. В противном случае от x можно спокойно избавиться.

Определение 1. *Пусть даны строки x и y. Представим их как $pr{+}ov$ и $ov{+}su$ соответственно, причем $|pr| > 0$, $|su| > 0$, ov имеет максимальную возможную длину, а опетор $(+)$ - это конкатенация строк. Тогда определим $over(x, y) = ov$, $pref(x, y) = pr$ и $d(x, y) = |pr|$.*

Теперь запустим следующий алгоритм.

1. Если в множестве S осталась ровно одна строка, то выведем её и прекратим работу алгоритма.

2. Иначе, переберём все упорядоченные пары различных строк x и y из S и найдем среди них ту, у которой $|over(x, y)|$ максимален. Если таких несколько можно выбрать любую. Например, лексикографически минимальную.

3. Найдя такую пару, выкинем из S строки x и y, а вместо них положим туда строку $pref(x, y) + y$. И перейдем к первому пункту алгоритма.

3 Доказательство

Давайте посмотрим на то, что на самом деле происходит с исходными строками во время работы алгоритма. Для начала, дадим несколько определений:

Определение 2. *Пусть дана непустая строка s. Тогда $s_{i,j}$, где $1 \leq i \leq j \leq |s|$ - это подстрока s начинающаяся с i-ого символа и заканчивающаяся на j-ом символе строки s включительно.*

Определение 3. *Пусть даны нупустые строки s, t и число pos, тогда:*

$$contains(s, t, pos) = \begin{cases} 1, & \text{если } pos \geq 1, (pos + |t| - 1) \leq |s| \text{ и строка } t \text{ равна } s_{pos,pos+|t|-1} \\ 0, & \text{иначе} \end{cases}$$

Определение 4. *Пусть даны непустые строки s, t, тогда:*

$$index(s, t) = \begin{cases} \text{минимальное число } pos, & \text{такое, что } contains(s,\ t,\ pos) = 1 \\ 0, & \text{если такого числа не существует} \end{cases}$$

Определение 5. *Пусть $T = \{t_1, \ldots, t_k\}, k \geq 1$ - упорядоченное множество непустых строк. Тогда, $orderedSuperstrings(T)$ - это множество таких общих надстрок T, что $\forall i, j$ т.ч. $1 \leq i, j \leq k$ верно: $i \leq j \iff index(t, t_i) \leq index(t, t_j)$. И $superstring(T)$ - это строка из $orderedSuperstrings(T)$ наименьшей длины.*

Теперь заметим, что $superstring(T)$ можно построить жадно:

Лемма 1. *Пусть $T = \{t_1, \ldots, t_k\}, k \geq 1$ - упорядоченное множество непустых строк. Тогда, $superstring(T) = \{pref(t_1, t_2) + pref(t_2, t_3) + \ldots + pref(t_{k-1}, t_k) + t_k\}$.*

Доказательство. Докажем индукцией по k. Для $k = 1$ очевидно(сама строка должна лежать в надстроке и она же является своей надстрокой). Пусть теперь для любого $k \leq K$ лемма верна. Тогда докажем, что она верна и для $k = K + 1$.

Рассмотрим

4 NP-полнота

Теорема 1. *the1*