

Общая надстрока наименьшей длины

Андрей Осипов

15 декабря 2013 г.

1 Постановка задачи

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

Пусть теперь язык L это множество пар вида (S, k) для которых верно, что такая строка s существует, и имеет длину не больше k . Тогда можно говорить о том, что в таком виде задача разрешения языка L является NP-полной. Доказательство этого факта будет приведено ниже. А пока мы ослабим условие следующим образом: пускай нам теперь нужно 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 NP-полнота

3 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.

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

Определение 1. Пусть даны строки A и B . Представим A как $P+O$, и B как $O+S$, где $|P| > 0$, $|S| > 0$, O имеет максимальную возможную длину, а оператор $(+)$ - это конкатинация строк. Тогда определим $over(A, B) = O$, $pref(A, B) = P$ и $d(A, B) = |P|$.

Теперь запустим на нашем множестве следующий алгоритм.

1. Если в множестве S осталась ровно одна строка, то мы выводим её и прекращаем работу алгоритма.
2. Иначе, перебираем все упорядоченные пары различных строк a и b из S и находим среди них ту, у которой $|over(a, b)|$ максимален. Если таких несколько можно выбрать любую. Например, лексикографически минимальную.
3. Найдя такую пару мы выкидываем из S строки a и b , а вместо них кладем туда строку $pref(a, b) + b$. И переходим к первому пункту алгоритма.

4 Доказательство

Теорема 1. *the1*