

Тряп-3

Волынцев Дмитрий 676 гр.

25 сентября 2017

1

По определению префикс-функция возвращает максимальный по длине собственный префикс, являющийся так же суффиксом слова. Пусть $a_1 = l(\omega)$ - такой префикс. Второй по длине префикс исходного слова a_2 , являющийся суффиксом, будет частью a_1 , т.к. a_1 содержит начало и конец исходного слова и является максимальным. Так как каждый последующий "префикс-суффикс" обладает теми же свойствами, получим последовательность вложенных друг в друга подслов, имеющих общее начало(новый префикс) и конец(новый суффикс), которые совпадают сами. Новые же префикс-суффиксы являются новыми префикс-функциями от предыдущих префикс-функций, т.е.:

$$a_1 = l(\omega)$$

$$a_2 = l(a_1) = l(l(\omega))$$

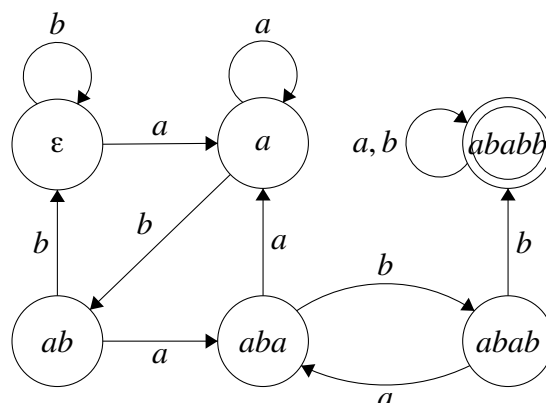
$$a_3 = l(a_2) = l(l(a_1)) = l(l(l(\omega)))$$

и т.д.

Утверждение доказано.

2

В соответствии с определением КМП-автомата построим автомат, корректность которого следует из построения и определения.



Протокол работы алгоритма КМП:

ababb#a - префикс подслова = суффикс слова = *a*; префикс-функция = 1

ababb#aa - префикс подслова = суффикс слова = *a*; префикс-функция = 1

ababb#aab - префикс подслова = суффикс слова = *ab*; префикс-функция = 2

ababb#aaba - префикс подслова = суффикс слова = *aba*; префикс-функция = 3

ababb#aabab - префикс подслова = суффикс слова = *abab*; префикс-функция = 4

ababb#aababa - префикс подслова = суффикс слова = *aba*; префикс-функция = 3

ababb#aababab - префикс подслова = суффикс слова = *abab*; префикс-функция = 4

ababb#aabababb - префикс подслова = суффикс слова = *ababb*; префикс-функция = 5

Есть вхождение слова

ababb#aabababbb - префикс подслова = суффикс слова = ϵ ; префикс-функция = 0

3

Реализуем алгоритм добавления нового слова:

Возможны 2 варианта:

1) Для нового слова уже есть состояние в автомате, то есть это слово является подсловом какого-то принимаемого слова. В этом случае просто

делаем данное состояние принимающим, значит слово теперь принимается автоматом.

2) Для нового слова нет состояния в автомате. В данном случае ищем среди существующих состояний то, которое является максимальным префиксом $\omega[0, i]$ добавляемого слова (идем из состояния ϵ по символам, пока не дойдем до состояния, из которого нельзя пройти дальше с нашим словом). Далее добавляем посимвольный переходы (сначала символ $\omega[i + 1]$ и т.д.) и необходимые состояния $\omega[0, i + 1]$ до того момента, пока все слово не станет состоянием. Делаем это состояние принимающим. Таким образом новое слово принимается автоматом, который остается детерминированным по построению (переходы определены однозначно).

Реализуем алгоритм удаления слова:

Также возможны 2 варианта:

1) Принимающее состояние слова имеет переходы к другим принимающим состояниям. В этом случае просто делаем это состояние непринимающим. Таким образом мы удалили слово из словаря, не нарушив вхождений других слов.

2) Принимающее состояние слова не имеет переходов. В этом случае удаляем данное состояние, переходы к нему и те НЕПРИНИМАЮЩИЕ состояния, из которых эти переходы были осуществлены. Т.е. удаляем цепочку состояний и переходов до ближайшего принимающего состояния (по стрелкам, но в обратную сторону). Таким образом мы удалили слово, не нарушив вхождений других слов.

4

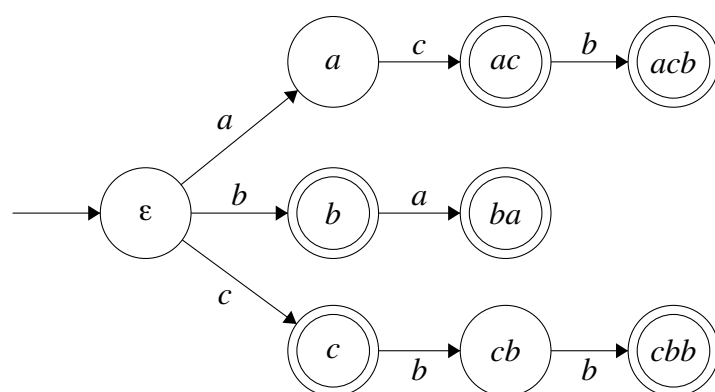
В первую очередь строим состояние для пустого слова ϵ и в зависимости от того, входит оно во множество S или нет, делаем его принимающим/непринимающим.

Далее используем алгоритм добавления слова, не имеющего состояния, в словарь из задачи 3. Все последующие слова из множества S проверяем посредством того же алгоритма и строим соответствующий автомат, который будет являться детерминированным и корректным по построению (задача 3).

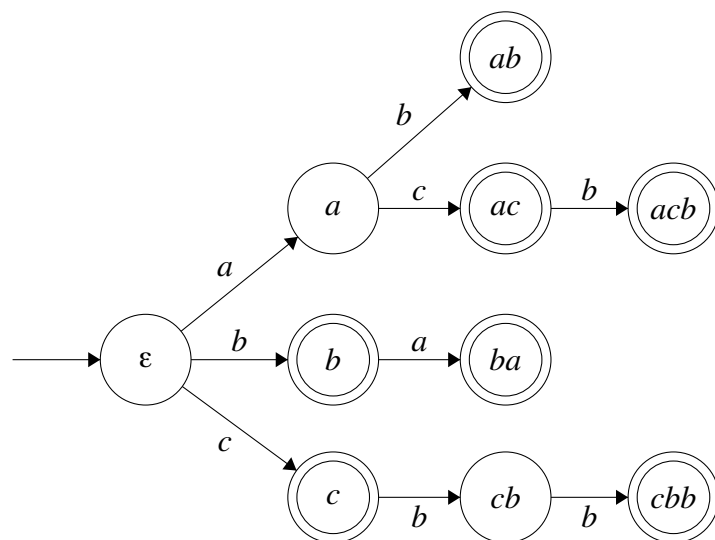
5

6

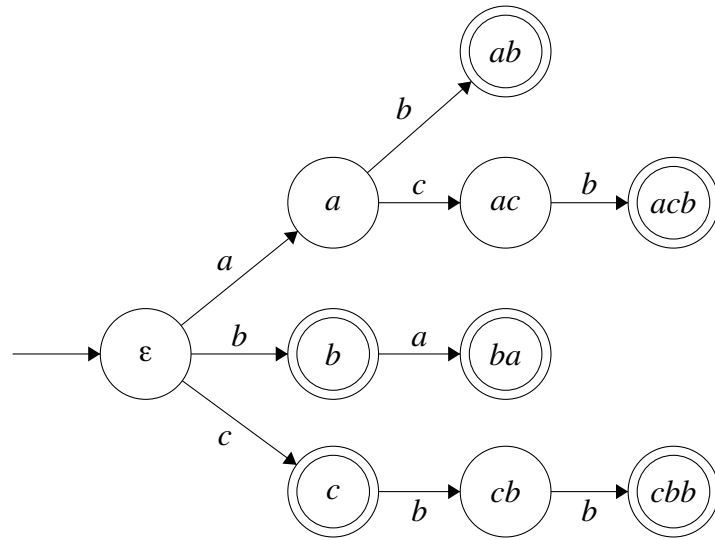
Искомый словарь:



Чтобы добавить слово *ab* в словарь, добавим новое принимающее состояние:



Чтобы удалить слово *ac* из словаря, сделаем соответствующее состояние непринимающим:



7

Будем искать подслово *abbaa* в слове *ababbabbaab*. Пользуясь алгоритмом из Шеня, запишем слова в виде *abbaa#ababbabbaab* и вычислим префикс-функцию, причем если какая-нибудь префикс-функция будет равна 5 (т.к. подслово длины 5), это будет означать вхождение подслова в слово. Получим следующую запись (здесь индексы - соответствующая символу префикс-функция): $a_1b_2a_1b_2b_3a_4b_2b_3a_4a_5b_2$. Как видно, подслово входит в слово.

Протокол работы алгоритма КМП:

abbaa#a - префикс подслова = суффикс слова = *a*; префикс-функция = 1

abbaa#ab - префикс подслова = суффикс слова = *ab*; префикс-функция = 2

abbaa#aba - префикс подслова = суффикс слова = *a*; префикс-функция = 1

abbaa#abab - префикс подслова = суффикс слова = *ab*; префикс-функция = 2

abbaa#ababb - префикс подслова = суффикс слова = *abb*; префикс-функция = 3

abbaa#ababba - префикс подслова = суффикс слова = *abba*; префикс-функция = 4

abbaa#ababbab - префикс подслова = суффикс слова = *ab*; префикс-

функция = 2

abbaa#ababbabb - префикс подслова = суффикс слова = *abb*; префикс-функция = 3

abbaa#ababbabba - префикс подслова = суффикс слова = *abba*; префикс-функция = 4

abbaa#ababbabbaa - префикс подслова = суффикс слова = *abbaa*; префикс-функция = 5

abbaa#ababbabbaab - префикс подслова = суффикс слова = *a*; префикс-функция = 2