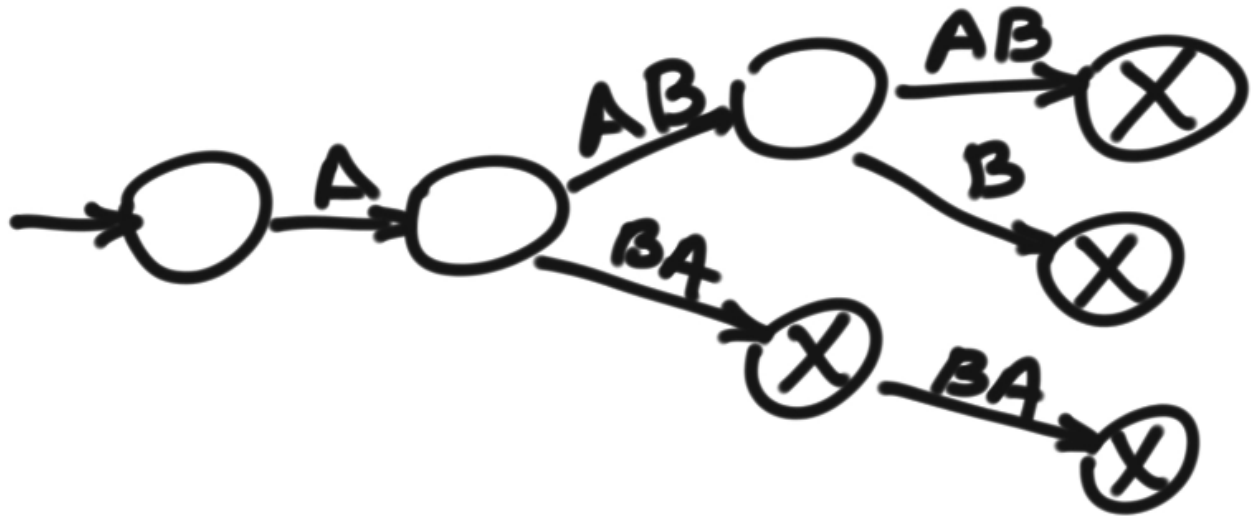
Суффиксное дерево

Сжатый бор

Сжатый бор - бор, в котором все последовательности несущественных ребер кроме тех, что проходят через терминальную вершину, заменены одним ребром.

$$P = \{ABA, AABAB, ABABA, AABB\}$$

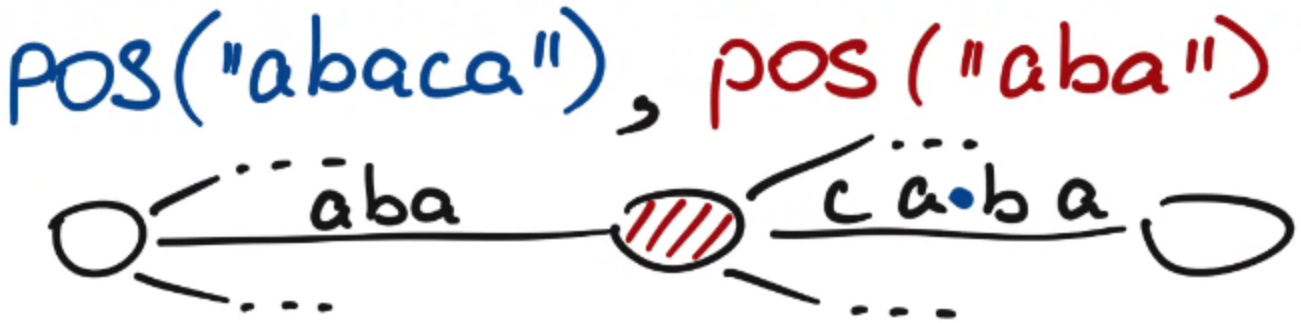


На ребрах достаточно хранить пару индексов (начало и конец подстроки), а не строку целиком.

Сжатый бор: свойства

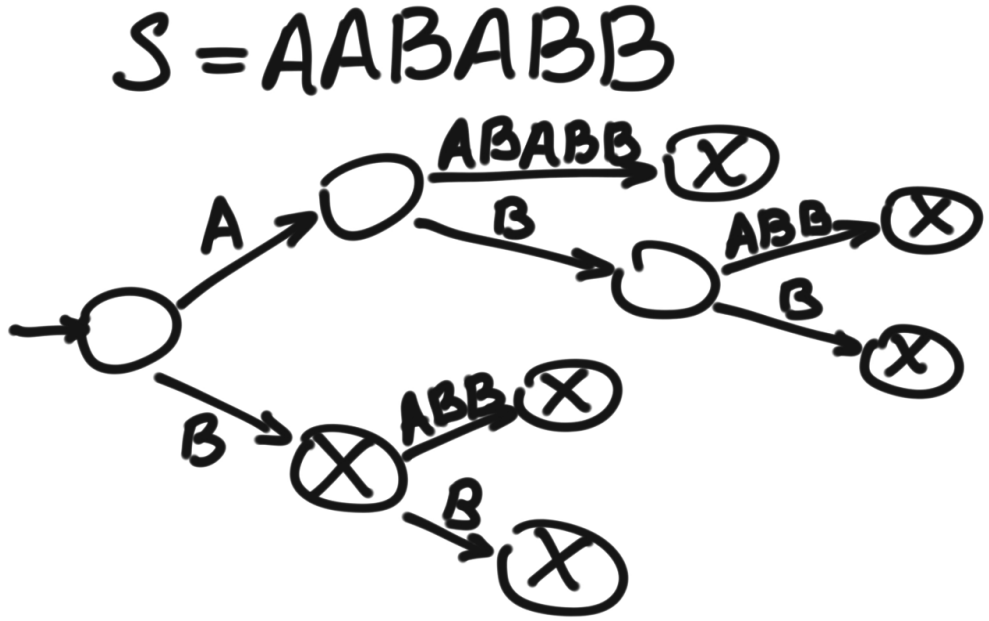
- $\forall v : deg_{out}(v) > 1$, кроме, возможно, корня и терминальных вершин.
- Ребра, исходящие из одной вершины, начинаются с разных букв.
- Число вершин = $\Theta(num\_leaves + |P|) = \Theta(|P|)$, $|P|$ - число строк в боре.
- Время построения $\Theta(\sum |P_i|)$

Позиция внутри ребра называется *неявной*. Позиция на вершине - *явной*.



Сжатый суффиксный бор

Сжатый суффиксный бор строки S - сжатый бор, построенный на множестве суффиксов S.



Из свойств сжатого бора следует, что сжатый суффиксный бор занимает $\Theta(|S|)$ памяти, а наивное построение занимает $\Theta(|S|^2)$

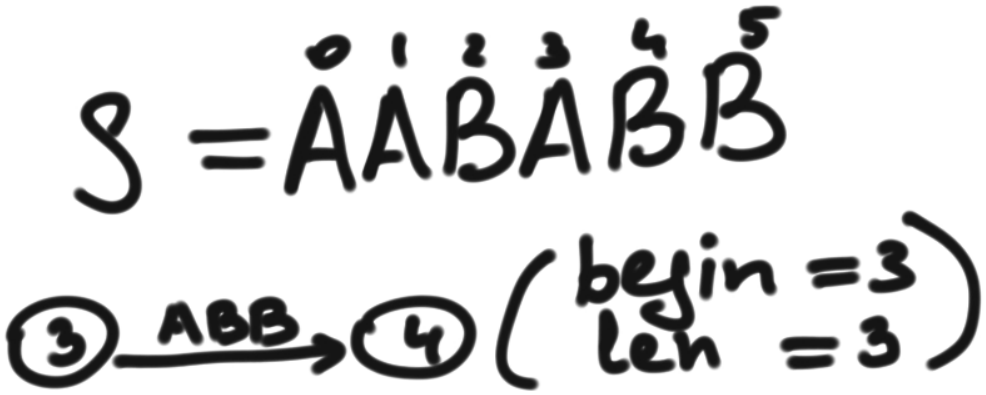
Сжатый суффиксный бор

Чтобы не хранить кучу подстрок на ребрах предлагается следующий трюк:

Храним строку целиком, а на ребрах пишем пару индексов - начало и длину подстроки, которая на них написана.

Причем ребро храним в вершине, в которую оно ведет.

```
struct Node {
  dict[char, NodeId] transitions;
  size_t begin, length;
}
```



Связь сжатого суффиксного бора и суффиксного автомата

Обозначение: $S' = reversed(S)$

Пусть $\overline{SA}$ - автомат, состоящий из инвертированных суффиксных ссылок суффиксного автомата, а CT - сжатый суффиксный бор.

Утверждение. $\overline{SA}(S) = CT(S')$, причем на ребрах написана перевернутая разница наибольших строк, соответствующих концам ребра.

Связь CST и SA

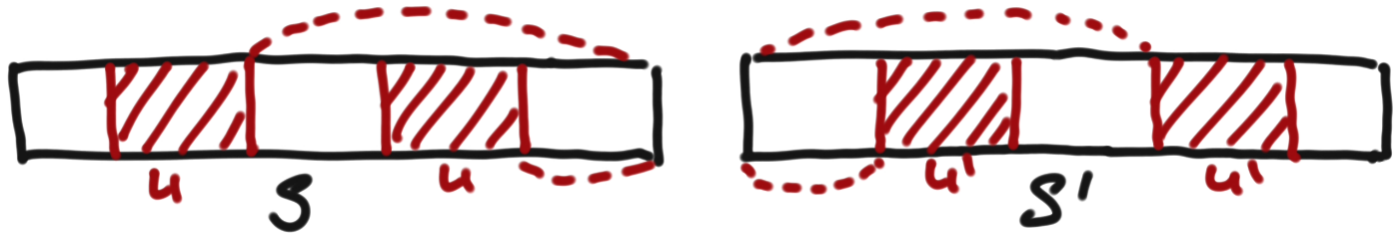
Утверждение. $\overline{SA}(S) = CT(S')$, причем на ребрах написана перевернутая разница наибольших строк, соответствующих концам ребра.

Доказательство.

1. Введем понятие левого контекста, аналогично правому:

$$L_S(u) = \{w | wu - \text{префикс строки } S\}$$

Легко заметить, что существует биекция между правыми контекстами S и левыми контекстами S'. То есть продолжения строки u до суффикса $S \Leftrightarrow$ продолжения строки u' до префикса S'.

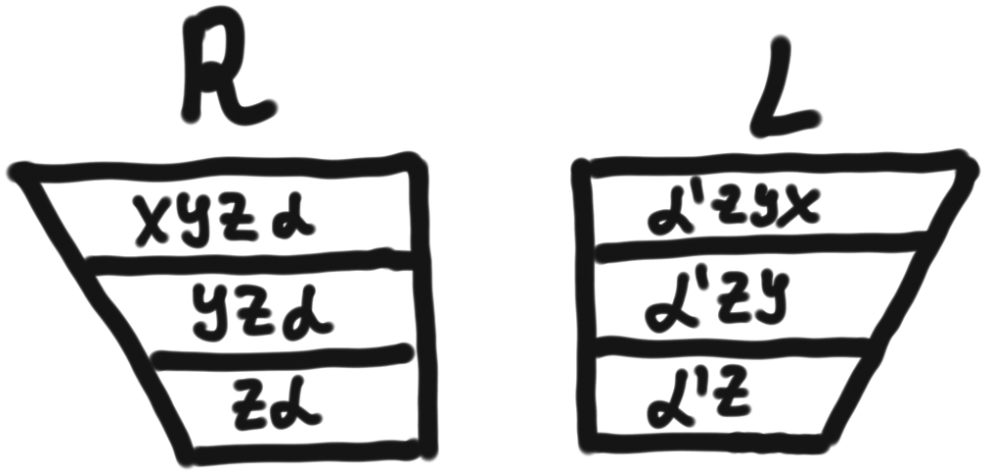


Связь CST и SA

Утверждение 2. $\overline{SA}(S) = CT(S')$, причем на ребрах написана перевернутая разница наибольших строк, соответствующих концам ребра.

Доказательство.

2. Аналогично правым контекстам (Утверждение 1.) для любой пары строк u, v с одинаковыми левыми контекстами верно, что одна из них является префиксом другой.



Связь CST и SA

Утверждение 2. $\overline{SA}(S) = CT(S')$, причем на ребрах написана перевернутая разница наибольших строк, соответствующих концам ребра.

Доказательство.
3. Покажем, что состояния в CST соответствуют классам левой эквивалетности.

Без ограничения общности $|u| < |v|$

- a) Если строки u и v лежат в одном классе, то значит u является префиксом v, причем на пути из u до v нет развилок (так как u встречается там же, где и v). То есть концы u и v лежат на одном ребре, а значит ведут в одну вершину.
- b) Если u и v лежат в разных классах, то существует $\alpha \in L_S(u), \notin L_S(v)$. Значит существует путь в u, который не ведет в $v \Rightarrow$ они в разных вершинах.

Таким образом, нашли биекцию между вершинами $SA(S)$ и $CST(S')$.

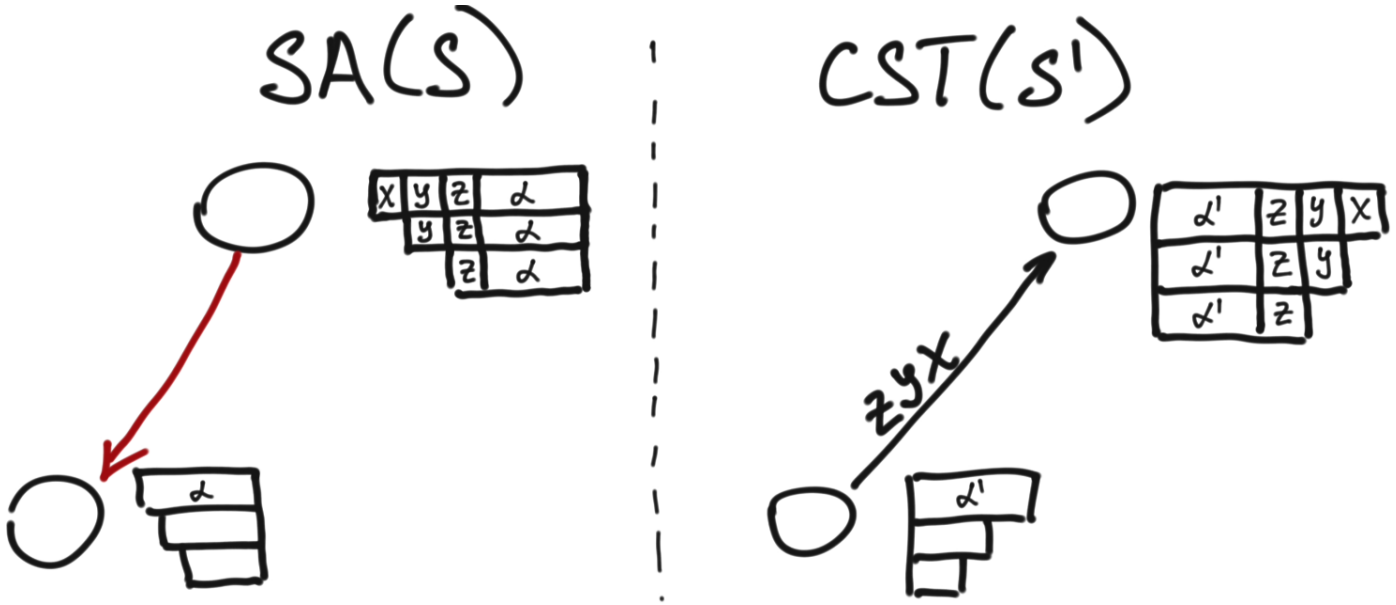
Связь CST и SA

Утверждение 2. $\overline{SA}(S) = CT(S')$, причем на ребрах написана перевернутая разница наибольших строк, соответствующих концам ребра.

Доказательство.

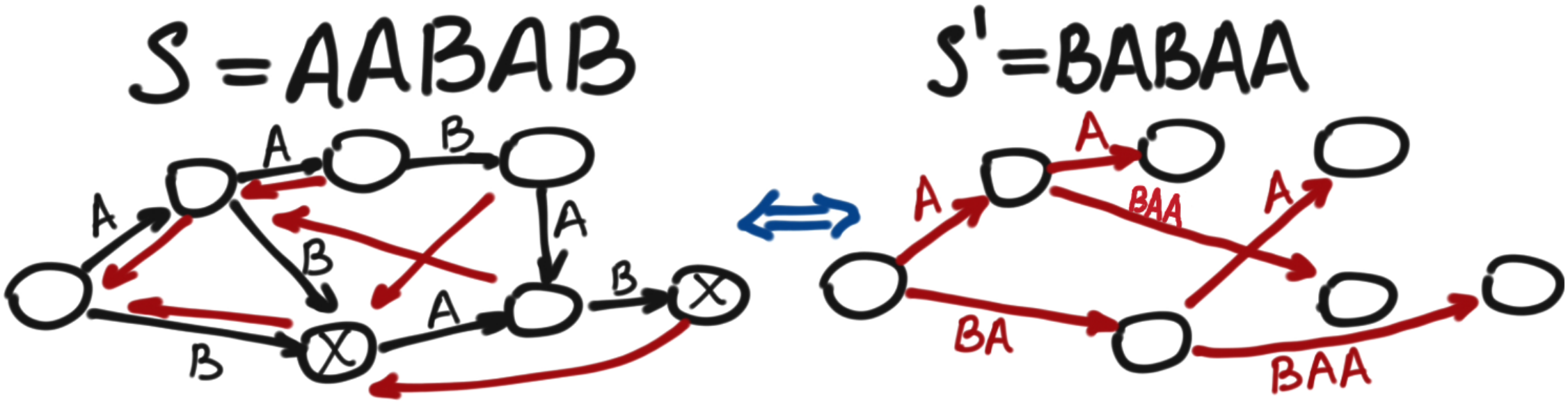
4. Осталось показать, что ребра в CST(S') - это суффссылки SA(S).

Картинка красноречива (?):



■

Связь сжатого суффиксного бора и суффиксного автомата: пример



Связь сжатого суффиксного бора и суффиксного автомата: алгоритм

Таким образом, чтобы построить сжатый суффиксный бор строки S, нужно:

1. Построить суффиксный автомат строки S'.
2. Извлечь из суффиксного автомата вершины и суффиксные ссылки.

```
def ExtractTransitions(automaton, node_id):
  suffix_id = automaton.nodes[node_id].suffix
  nodes[node_id].begin =
    str.Size() - 1 - (automaton.EndPos(node_id) - automaton.Length(suffix_id))
  node.length = automaton.Length(node_id) - automaton.Length(suffix_id)
  nodes[suffix_id].transitions[str[node.begin]] = node_id

def CompressedTrie(automaton):
  nodes = CreateNodes(automaton.Size())
  str = automaton.Str().Reverse()
  nodes[0].begin = nodes_[0].length = None
  for node_id from 1 to automaton.Size() - 1:
    ExtractTransitions(automaton, node_id)
```

Упражнение

Пусть α - строка соответствующая некоторой вершине n в сжатом суффиксном боре, а β наибольший суффикс α, у которого тоже есть своя вершина (m). Тогда ссылку из n в m назовем *суффиксной ссылкой*.

Упражнение. Докажите, что инвертированные сплошные переходы в $\overline{SA}(S)$ будут суффиксными ссылками в $CT(S')$.

Суффиксное дерево

Суффиксное дерево строки S - сжатый суффиксный бор строки S, в котором каждый суффикс оканчивается в листовой вершине. Последнего можно добиться, если к строке S приписать в конец символ '#' (символ, которого нет в алфавите).