

# Suffix Trees

# What for? And how?

Given a piece of text  $T$ , to process it so that it may be searched for different patterns.

Given a text  $T$ , say of size  $n$ . Make a list  $L$  of all suffixes of  $T$ . Thus,  $L$  will have  $n$  words.

Example:  $T = \text{axbbabac}$  of length 8 . Now make a trie by adding these 8 words, one at a time, to get a structure  $Q$ , with potentially  $n^2$  nodes. Supposing i want to look for “ba”. Since there are 2 suffixes which begin with aac, these are aacaacdbaa and aacdbaa, these would have been paths in  $Q$ , with a common initial segment, viz. aac.

# The Trie

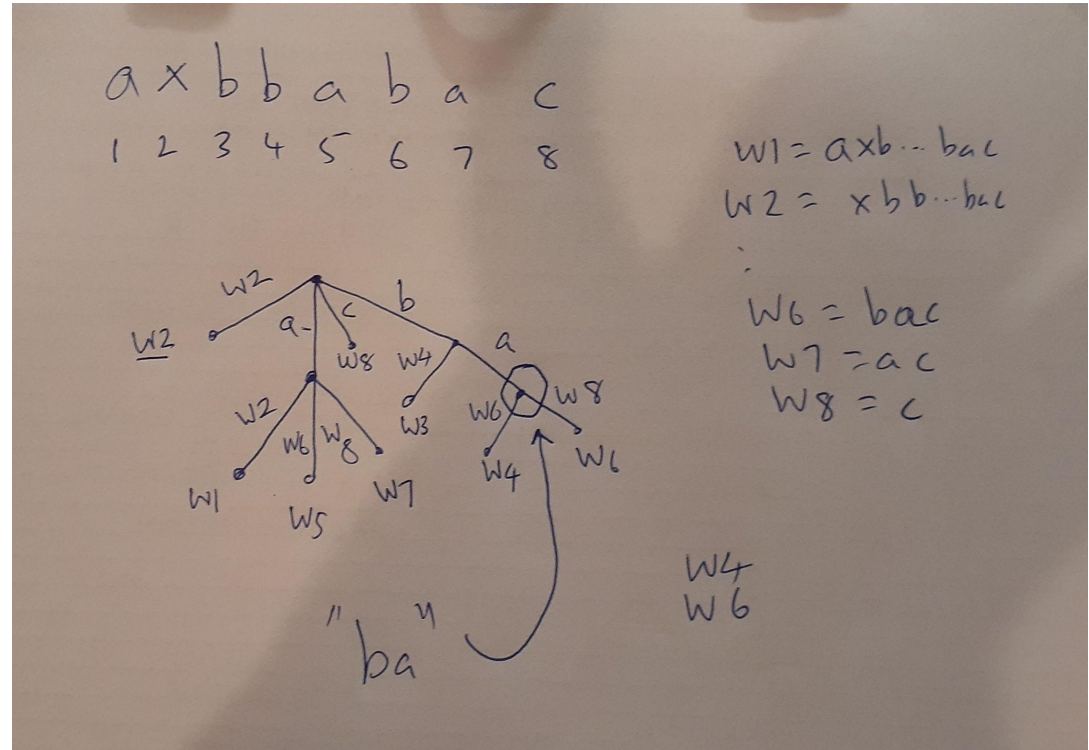
$T = axbbabac$ . Let  $W(i)$  be the word  $T(i), T(i+1), \dots, T(n)$ . Thus

$W_1 = axbbabac$

$W_4 = babac$

$W_7 = ac$

The trie formed by adding all of these is shown alongside.



# Properties

1. Each edge is labeled with a substring of T.
2. For every  $W(i)$ , there is a path from the root to a terminal node which reads  $W(i)$ .
3. For any string, in this case "ba", the node located in the trie for "ba" should be located. Now all terminal nodes, in this case  $W_4$  and  $W_6$  give the position of the string, viz., 4 and 6

