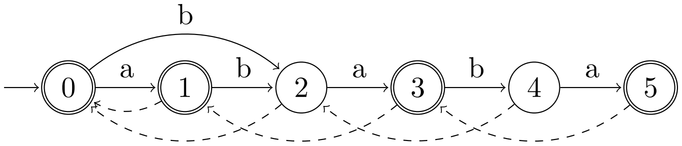
Tutorial Week 9

Definition 1. *The* suffix automaton *corresponding to some text y, can be regarded as a minimal deterministic final state machine which is a compressed version of a suffix trie. In it, the states are associated to classes representing factors of the string, and their number is upper bounded by $2n-1$, for a text of length n. The number of arcs, in their turn, are at most $3n-4$. One obtains a suffix automaton with the maximum number of states for a string of the form ab^k, while for ab^kc we get an automaton with the maximum number of arcs. The* suffix links *for a suffix automaton, must not be confused with those of a suffix tree (although they bare the same name). For a state p, its suffix link points to a different state associated to its longest suffix occurring in a different right context. As for the suffix trees, in the intend to save on space usage, we can merge together all "non-forks", to obtain a* compact suffix automaton*.*

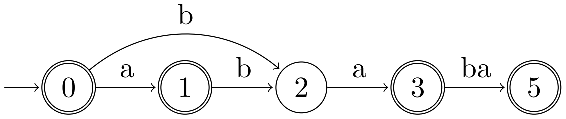
Exercise 1. *Consider the following list of sequences: ababa, abcacabb, abcacababc, and abacabacab. For each of them, construct the corresponding suffix automaton, including the corresponding suffix links. Finally, construct the associated compact suffix automaton.*

What is the complexity of the algorithm? Think about the space usage in each of the cases.

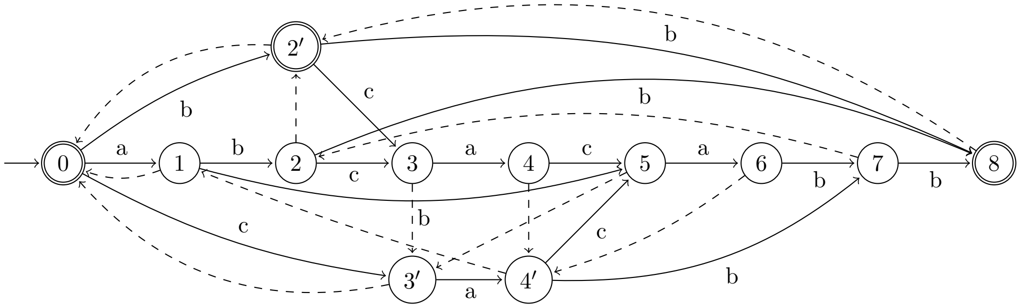
Solution: The suffix automaton associated to *ababa* is



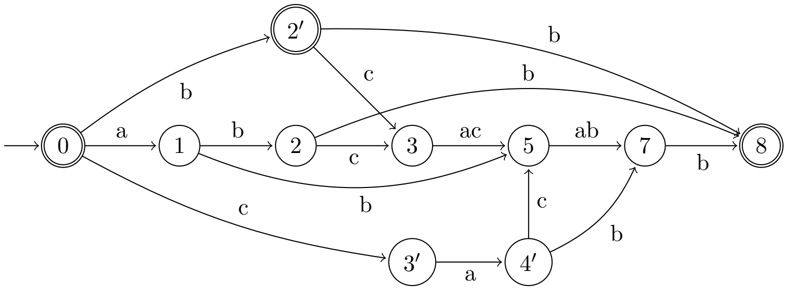
The CSA corresponding to this automaton is



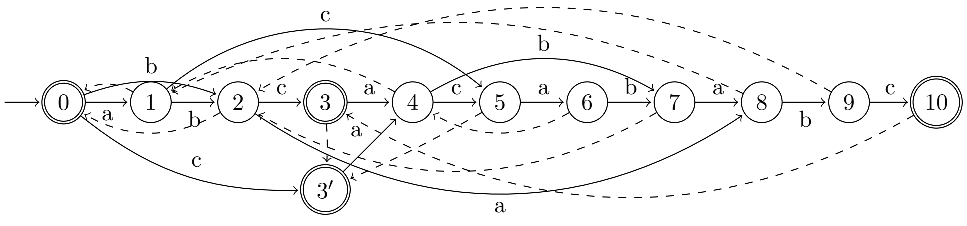
The suffix automaton associated to *abcacabb* is



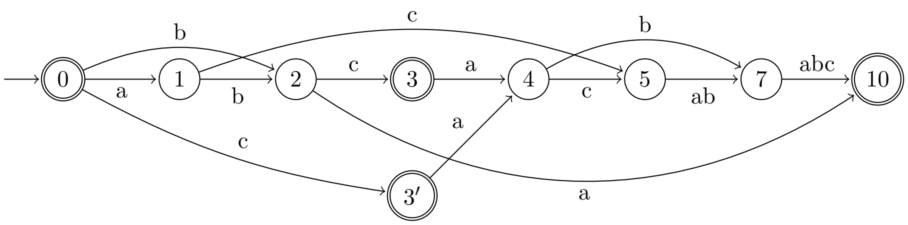
The *CSA* corresponding to this automaton is



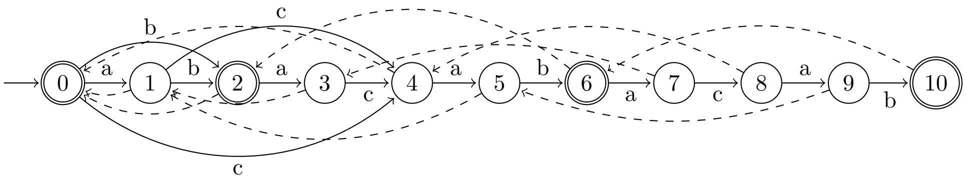
The suffix automaton associated to *abcacababc* is



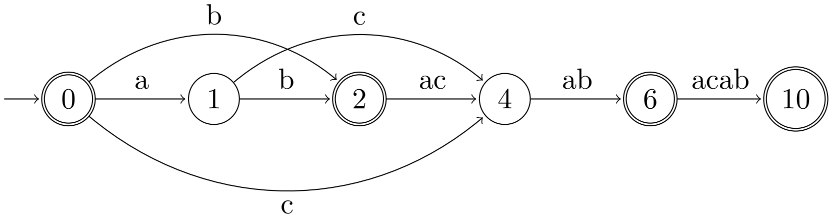
The *CSA* corresponding to this automaton is



The suffix automaton associated to *abacabacab* is



The CSA corresponding to this automaton is



■