Getting a regular expression out of an NFA/DFA

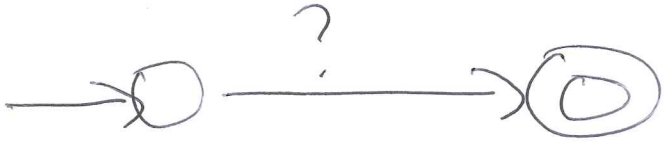
9/7/18

A <u>super-NFA</u> is an NFA where we allow arbitrary regular expressions to label the transitions.

One way of thinking of an NFA - for a transition, we have ~~a~~ to pay a toll given by the label out of the beginning of the string to use it - and a string is accepted if we can use up our string and get to a final state.

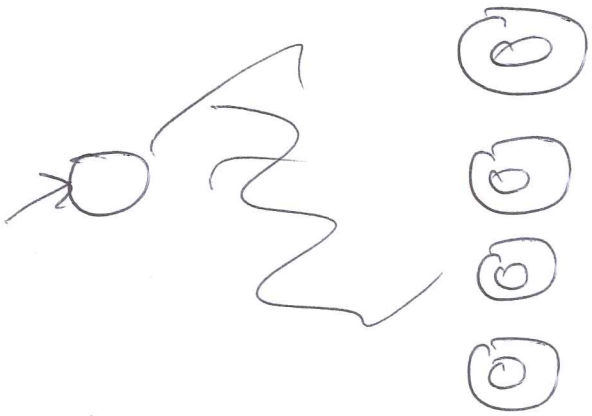
For a super-NFA, I can pay any toll that matches the r.e. on a transition.

Getting a r.e. out of an NFA is the same as converting an NFA to an equivalent super NFA of the form

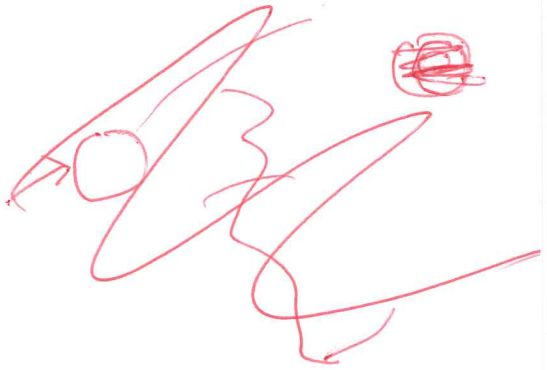


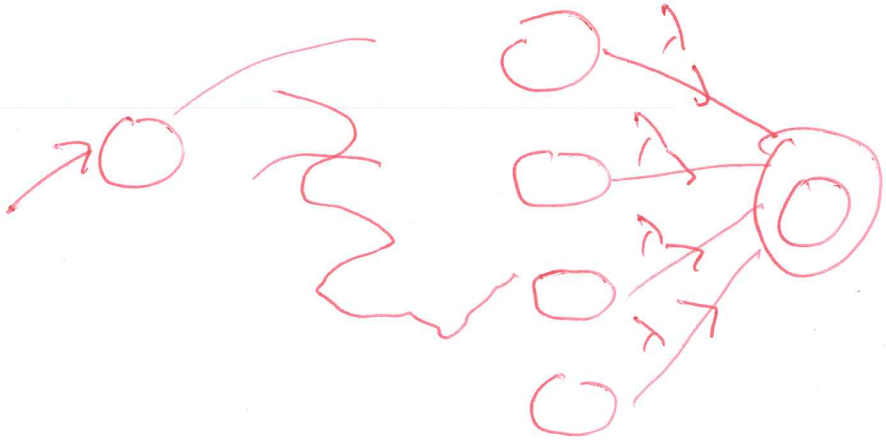
So we need to figure out how to get rid of states (in the process making transition labels more complicated) while keeping the accepted language the same.

0) Change my NFA to have only one final state.

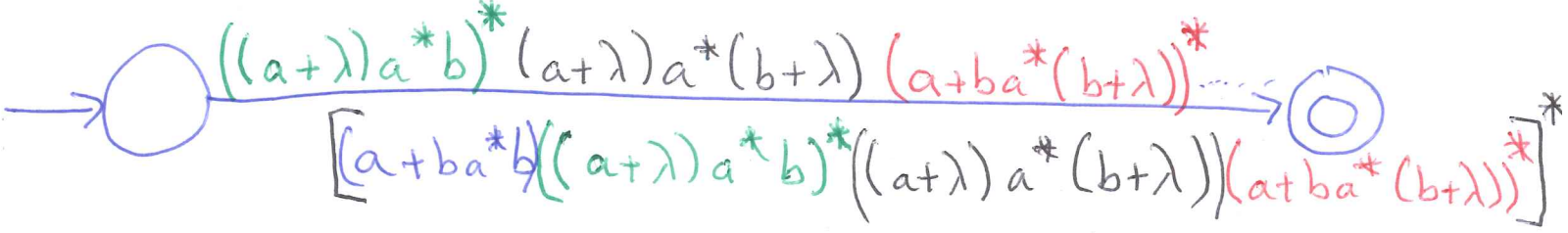


becomes

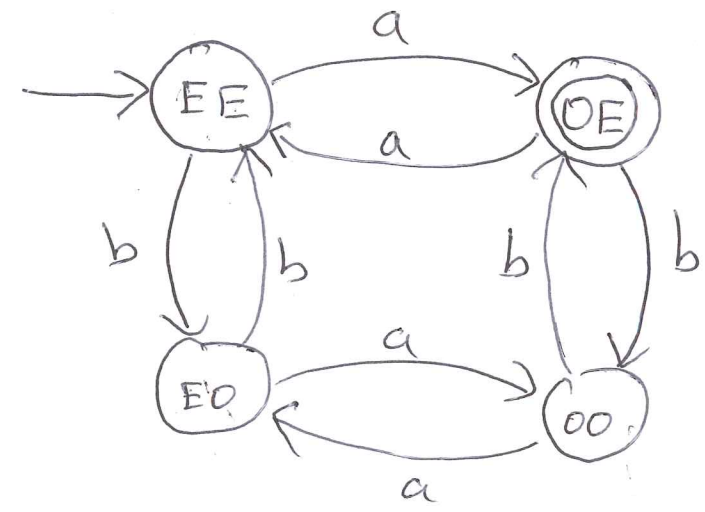




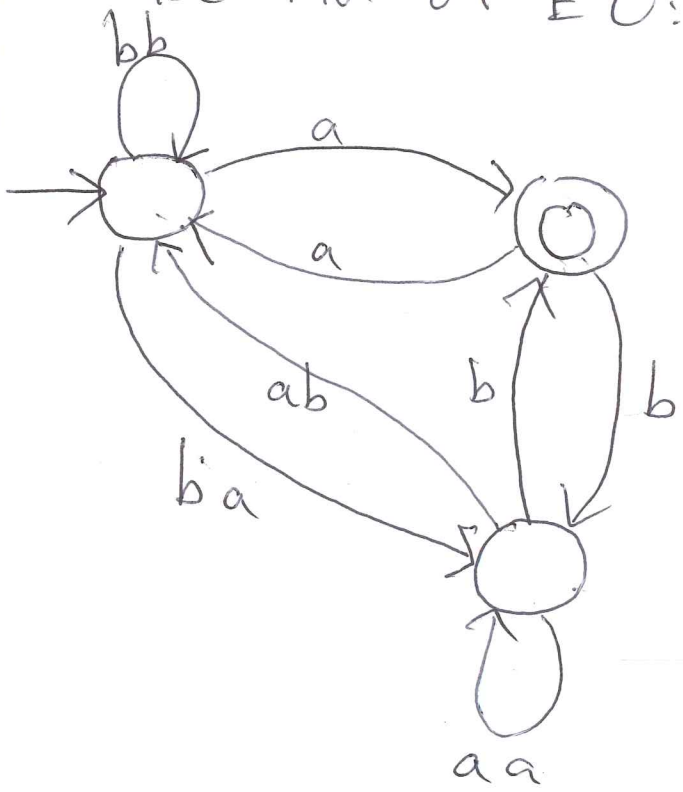
When we are down to one final and one initial state:



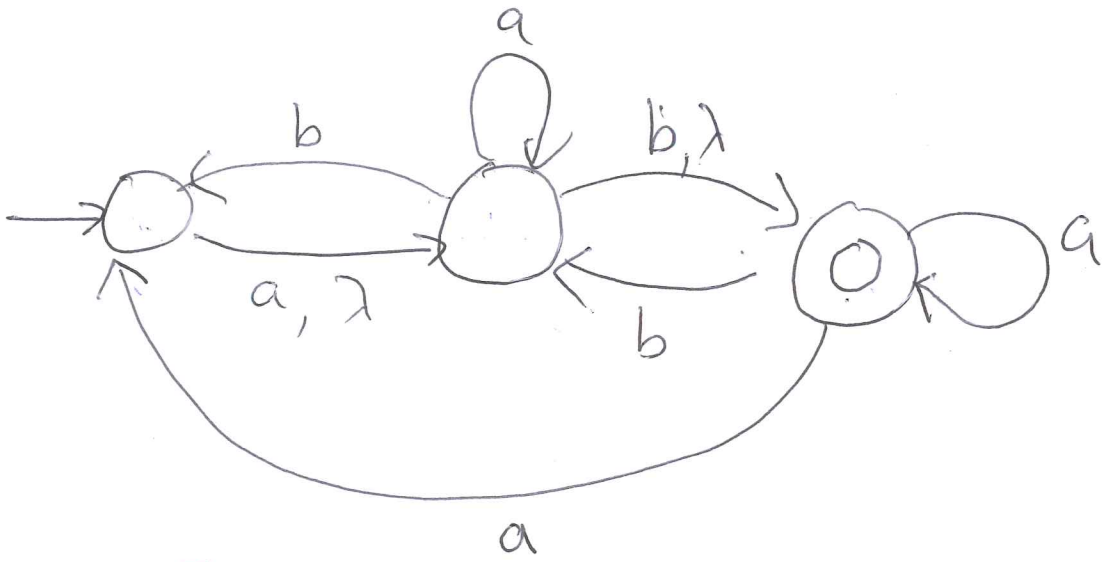
Find a regular expression for the language of strings with an odd # of a's and an even # of b's:

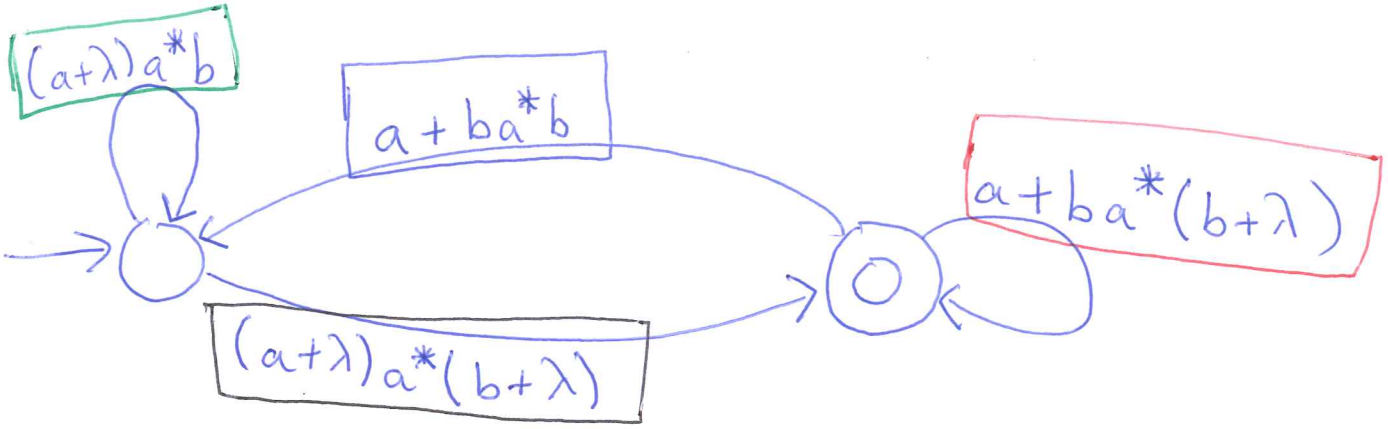


Get rid of EO:

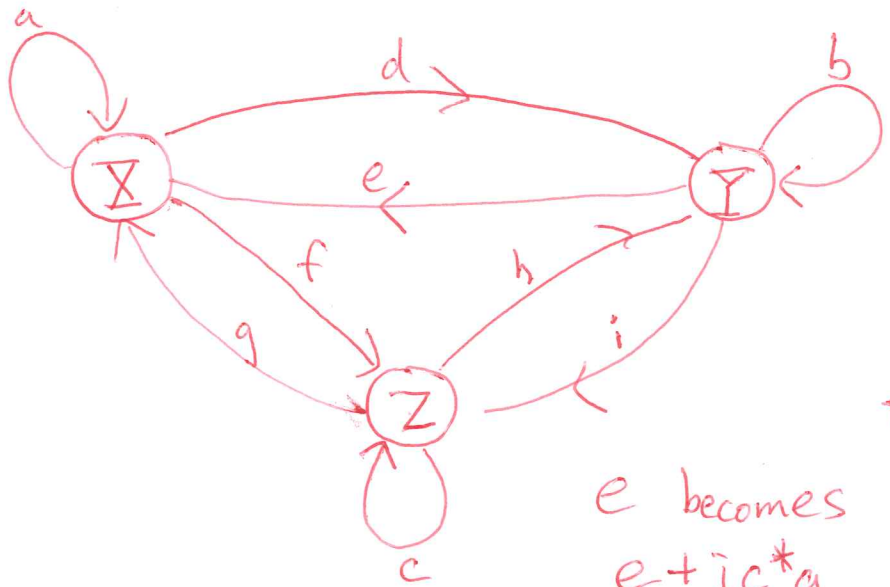


1) Get rid of non-final states as follows:





Formula



To get rid of Z, between any states X and Y

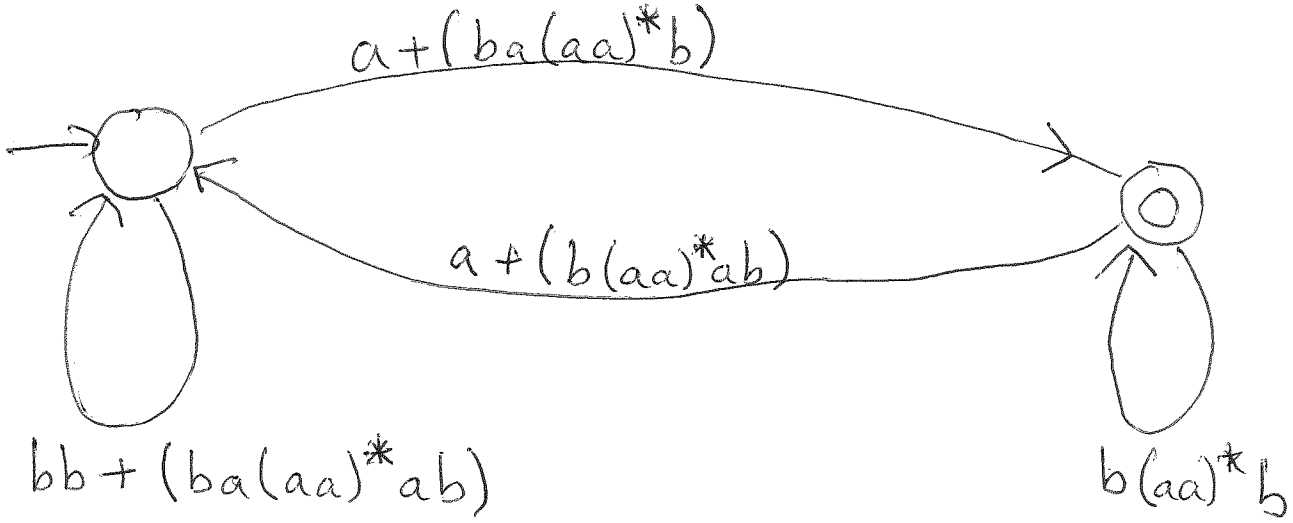
a becomes $a+fc^*g$

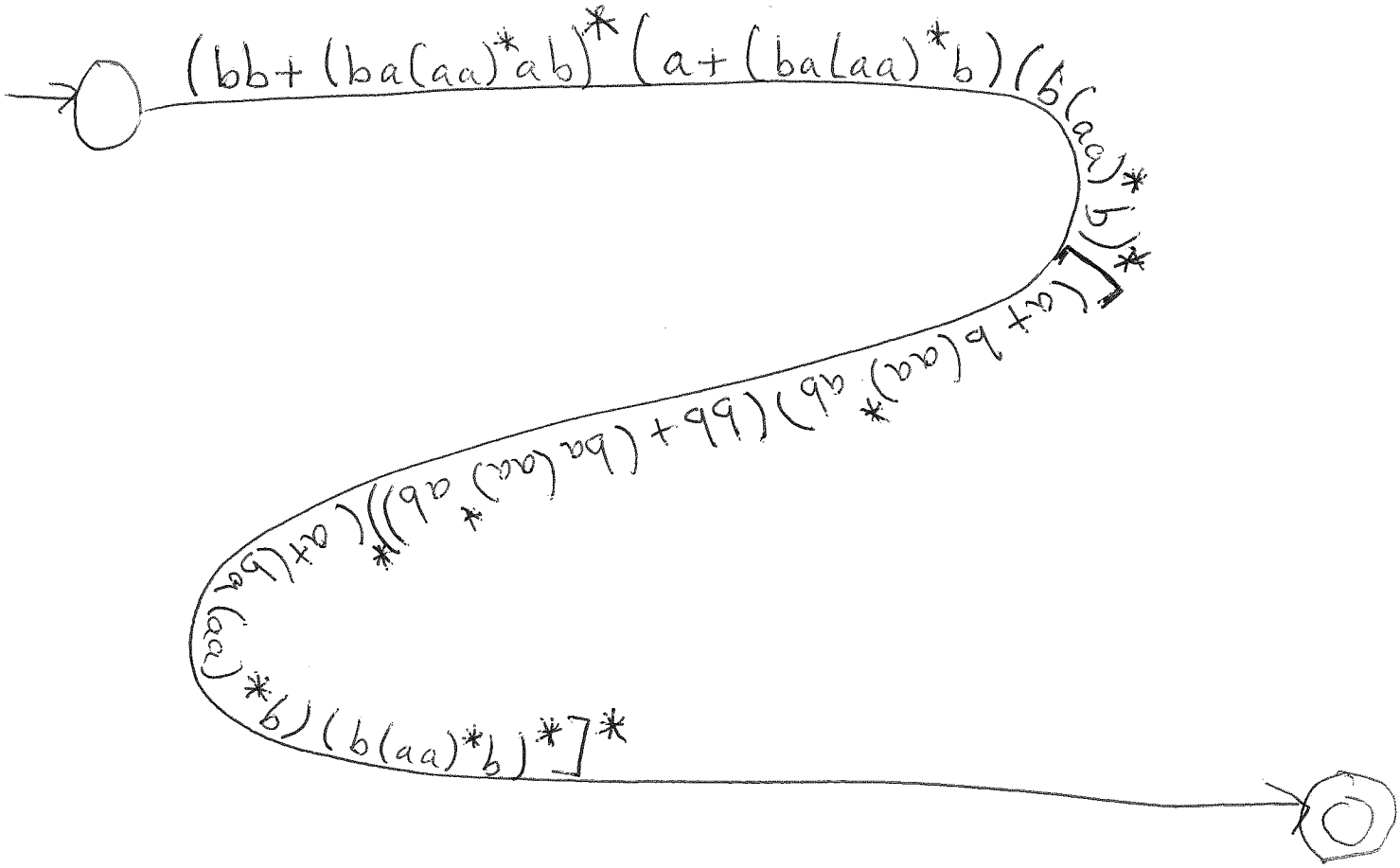
d becomes $d+fc^*h$

e becomes $e+ic^*g$

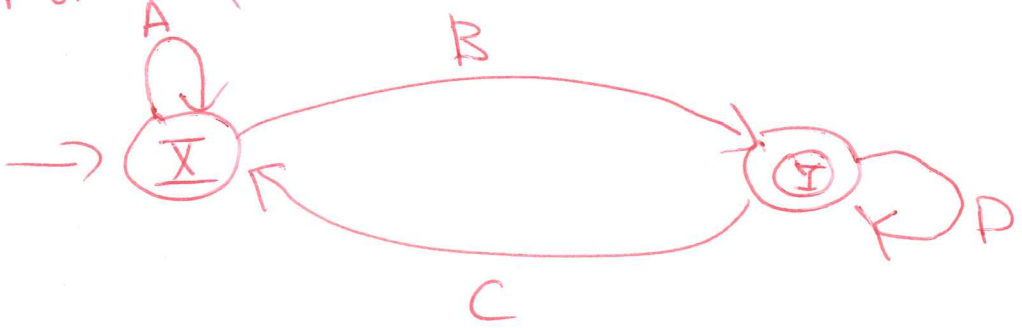
b becomes $b+ic^*h$

Get rid the non-init, non-final state.





Formula



is equiv to:

$$A^*BD^*(CA^*BD^*)^*$$