Problem 1

Find all strings in $L((a+bb)^*))$ of length 5

- zero instances of bb
  1. aaaaa
- one instance of bb
  2. aaabb
  3. aabba
  4. abbaa
  5. bbaaa
- two instances of bb
  6. bbbba
  7. bbabb
  8. abbbb

Problem 3

Find an NFA that accepts the language $L(aa^*(ab+b))$.

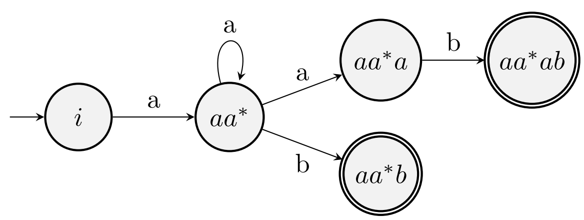


Figure 1: Accepts $L(aa^*(ab+b))$

Problem 7

Find a regular expression for the set $\{a^nb^m : n \geq 3, m \text{ is odd}\}$.

Answer: $(aaaa^*)(b(bb)^*)$

Problem 18

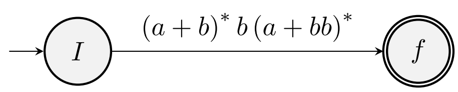
Find a regular expression for:

$$L = \left\{ w \in \{0,1\}^* \; : \; w \text{ has exactly one pair of consecutive zeroes} \right\}$$

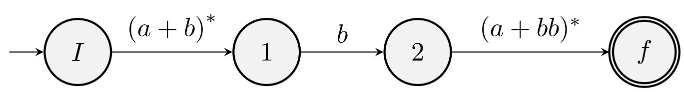
Answer: $(0 + \lambda)\left((1 + 101)^*\right)(00)\left((1 + 101)^*\right)(0 + \lambda)$

Problem 5

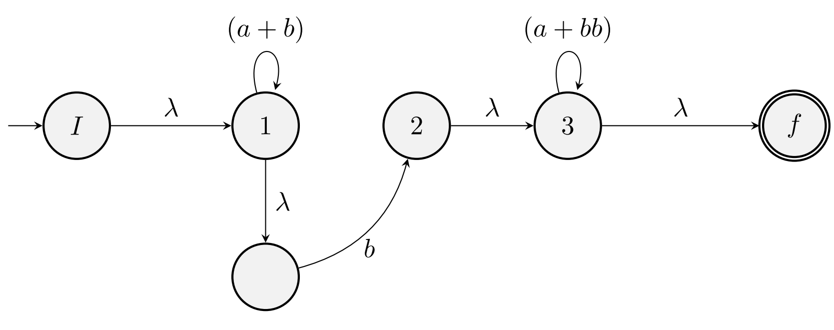
Give an nfa that accepts the language $L\left((a + b)^* \, b \, (a + bb)^*\right)$



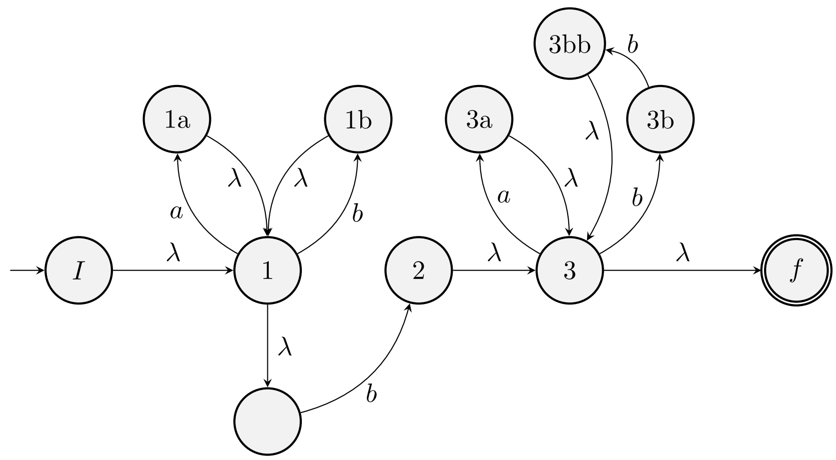
Step 1, make a trivial NFA



Step 2, split off the concats



Step 3, expand kleene stars

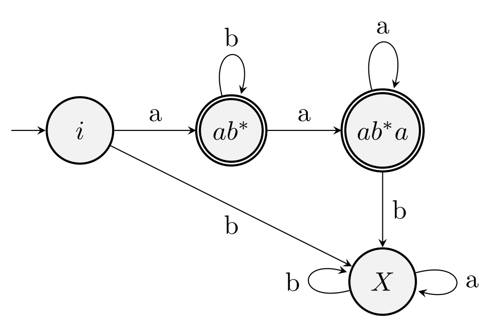


Step 4, break off unions.

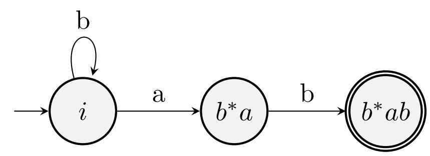
All labels are now single symbols, so the NFA is complete.
Leaving node 1 we have $(a+b)^*$.
Leaving node 2 we have $(a+b)^* b$
And leaving node 3 we finish with $(a+b)^* b (a+bb)^*$

Problem 7(b)

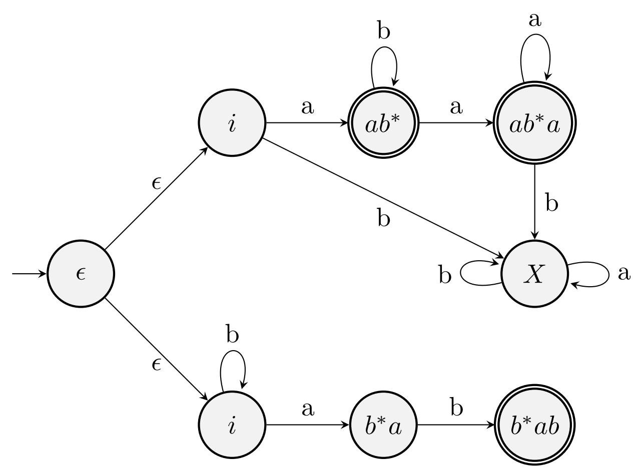
Find a DFA to accept $L = L(ab^*a^*) \cap L(b^*ab)$



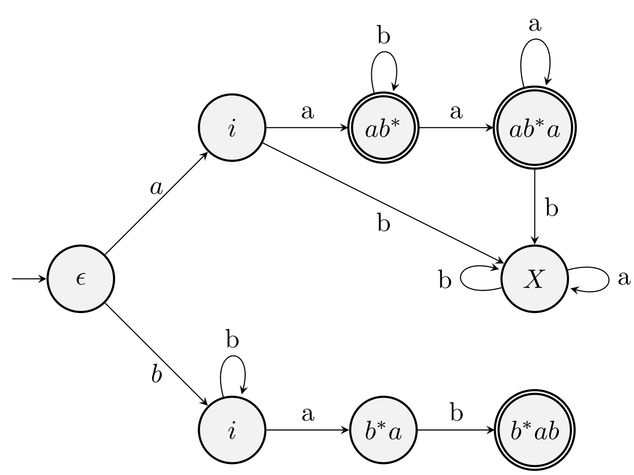
Step 1, design DFA for $L(ab^*a^*)$



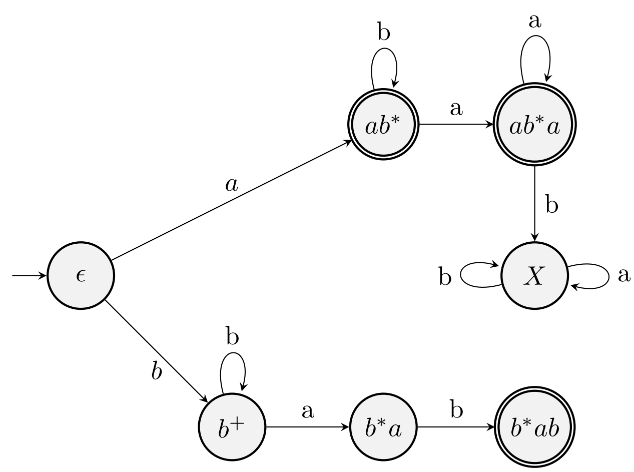
Step 2, design a partial DFA for b^*ab



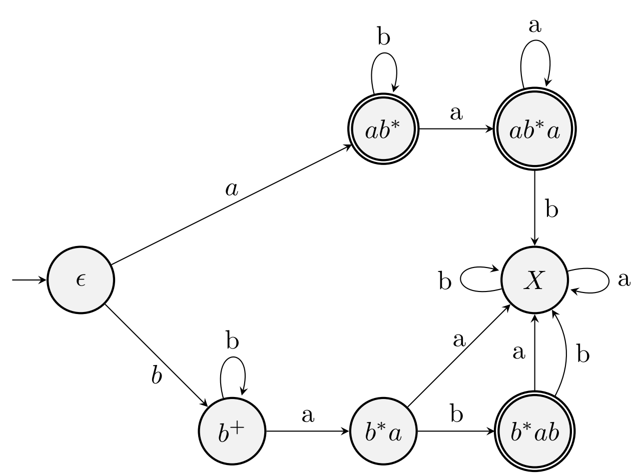
Step 3, combine them into an NFA



Step 4, observe that the top branch wants to start with a, bottom branch wants to start with b



Step 5, remove state i from top branch to make the DFA work properly, rename state i on the bottom branch (also observe that b^+ is sufficient, because if there is no leading b for the regex b^*ab, then ab will be accepted by the top branch)



Step 6, observe you forgot to give every node in the bottom branch an edge for every symbol, fix that.