

Homework 3

Problem 1: [10 points]

For each of the language below, construct a regular expression that represents this language. All languages are over alphabet $\Sigma = a, b$. In addition to the regular expression itself, provide a brief explanation (1-3 sentences) of the idea behind your construction.

Hint: Sometimes you might need to deal with ϵ and single-character strings (a and b) as special cases...

(a) The language is $\{w \mid w \text{ does NOT end with } aa\}$. For example, ϵ , a , and $aaba$ are in the language, while baa and $babaa$ are not.

Solution:

$$r_a = \epsilon + a + b + ((a + b)^*b)$$

Explanation:

For this problem we had to craft a regular expression that represents the language of all strings that don't end with aa which also means single characters such as a , b , and the empty string ϵ is in the language. In order to do this I explicitly handled the single-character and empty string cases then added the statements that handled all strings in the language with a length greater than one. In order to do this I do a kleene star to get any combination of a 's or b 's and then concatenate a b at the end of the string which ensures that the string can only end with bb or ab .

(b) The language is $\{w \mid \text{the first and last characters of } w \text{ are different}\}$. For example, $abab$, and $bbbaa$ are in the language, while ϵ , a and $bbbab$ are not.

Solution:

$$r_b = (a(a + b)^*b) + (b(a + b)^*a)$$

Explanation:

For this problem we had to craft a regular expression that represents the language of all strings where the first and last characters are different which also means that there must be at least two characters meaning ϵ , a , and b aren't in the language. In order to do this I just handled the two cases one where the string starts with a and the other where the string starts with b . In both cases I do a kleene star of all combinations of a and/or b strings then concatenate the opposite character at the end.

Problem 2: [5 points]

Let $L = \{a^m b a^n \mid m \geq n\}$. For example, b , $aaba$, and $aaabaaa$ are in the language, while ϵ , $aabba$, and $abaaa$ are not. Use the pumping lemma to prove that L is not a regular language over alphabet $\Sigma = \{a, b\}$.

Hint: If you get stuck, reviewing the last example in Lec 13 might help.

Solution:

In order to prove the language is not regular using a pumping lemma we will play a game against our adversary that we will call A:

1. A chooses $p \geq 0$
2. I choose $w = a^p b a^p$
 - $|w| = 2p + 1 \geq p$
 - $w \in L$ since $m = n = p$ which means $m \geq n$
3. A splits w into $w = xyz$ where $|xy| \leq p$ and $y \neq \epsilon$
 - x and y will be only a 's in order to hold the condition that $|xy| \leq p$
 - This means:
 - $y = a^j$ for some j such that $0 < j \leq p$ (can't be 0 since y can't be empty)
 - $x = a^k$ for some k such that $0 \leq k \leq p - j$
 - $z = a^{p-k-j} b a^p$ (the remaining a 's minus xy on the left side plus the b and the right side a 's)
4. I choose $i = 0$
 - $xy^i z = xy^0 z = xz = a^{p-j} b a^p$
 - So $m = p - j$ and $n = p$ and since $p - j < p$ or $m < n$ then $w = xy^i z \notin L$.

I WIN.

Therefore L is not a regular language.

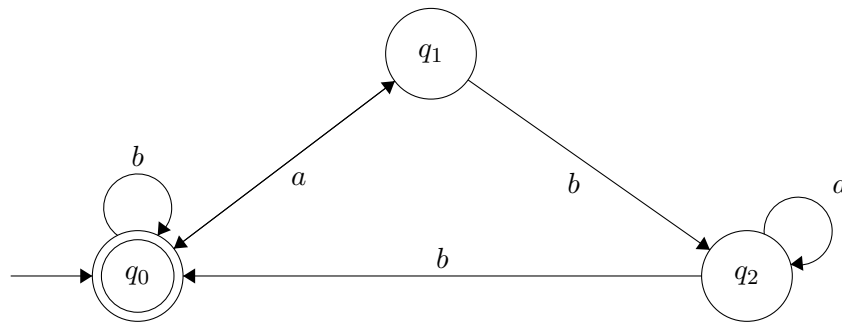
QED

Problem 3: [5 points]

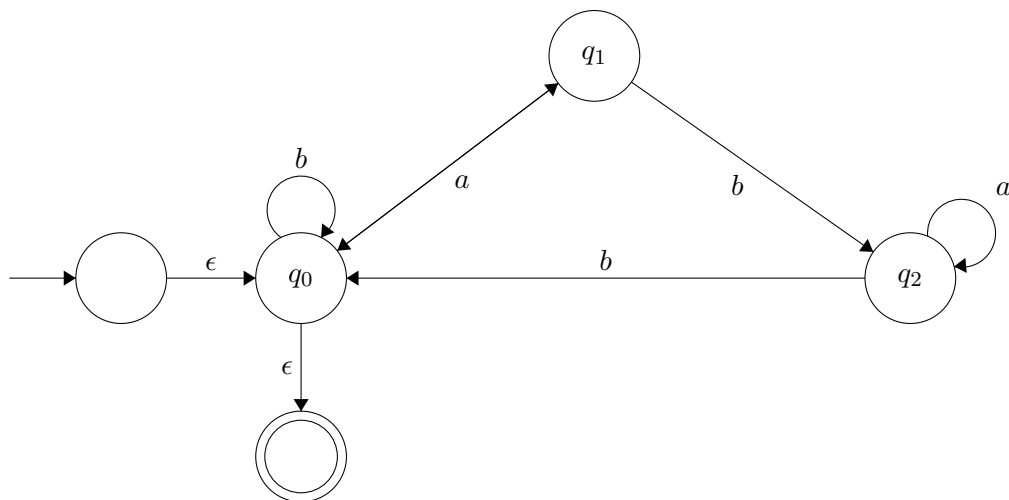
Give a regular expression equivalent to the following NFA (in fact it is a DFA).

I encourage you to write down the intermediate steps in addition to your final answer, so that you might get partial credits even if the final answer is wrong. (If your submission only includes the final answer, then it's either 5 points or 0 points.)

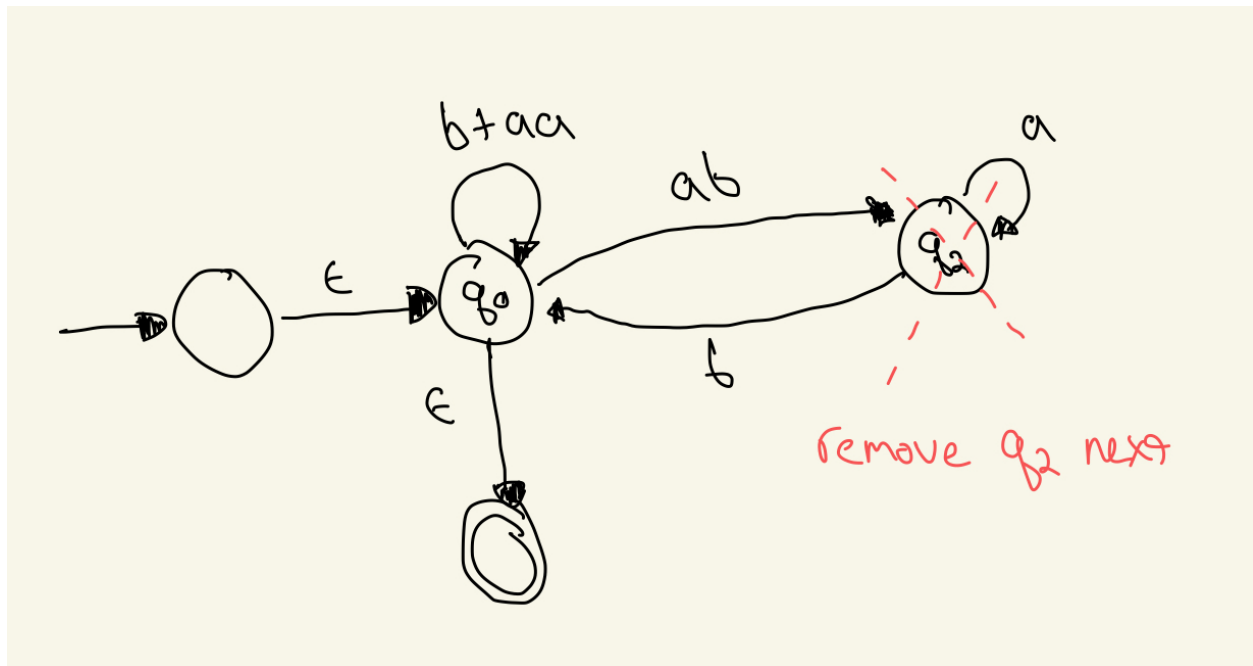
Hint: As discussed in class, the very first step should be adding a "dummy" starting state and a separate "dummy" accepting state; this is not strictly necessary but it's error-prone if you don't do this.



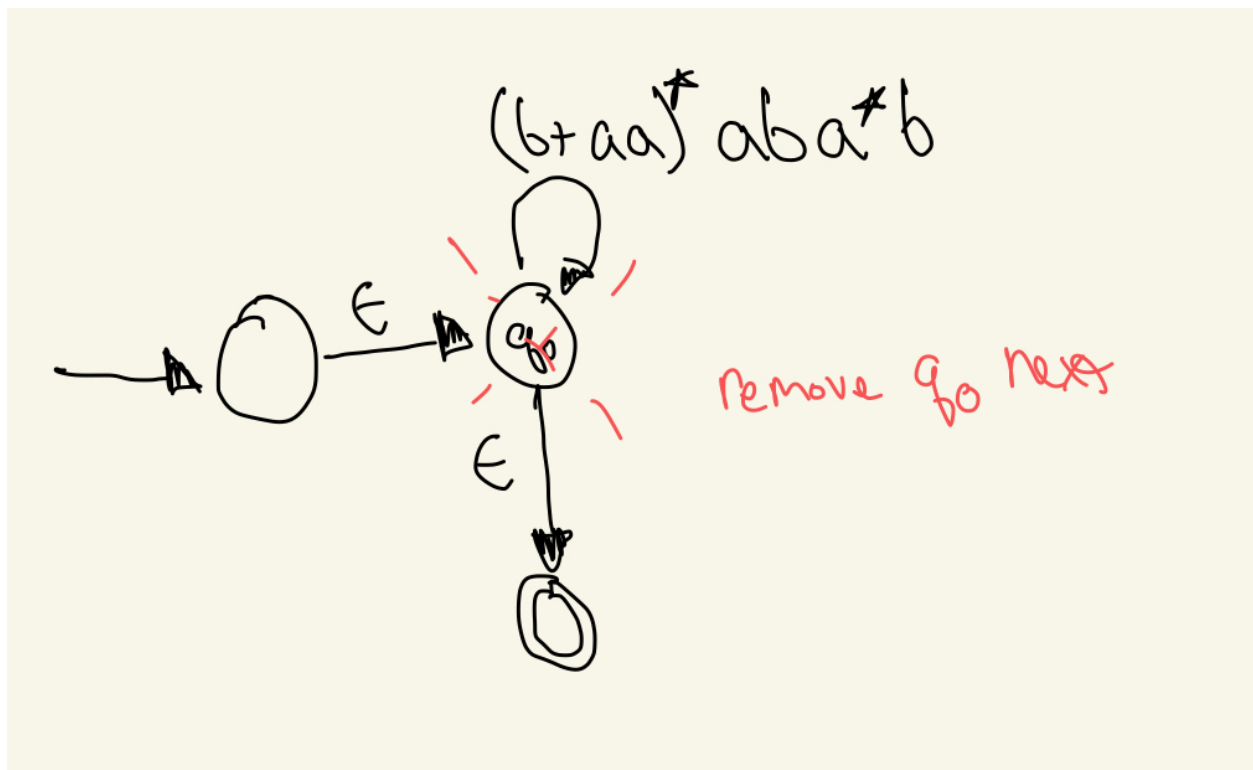
Solution:



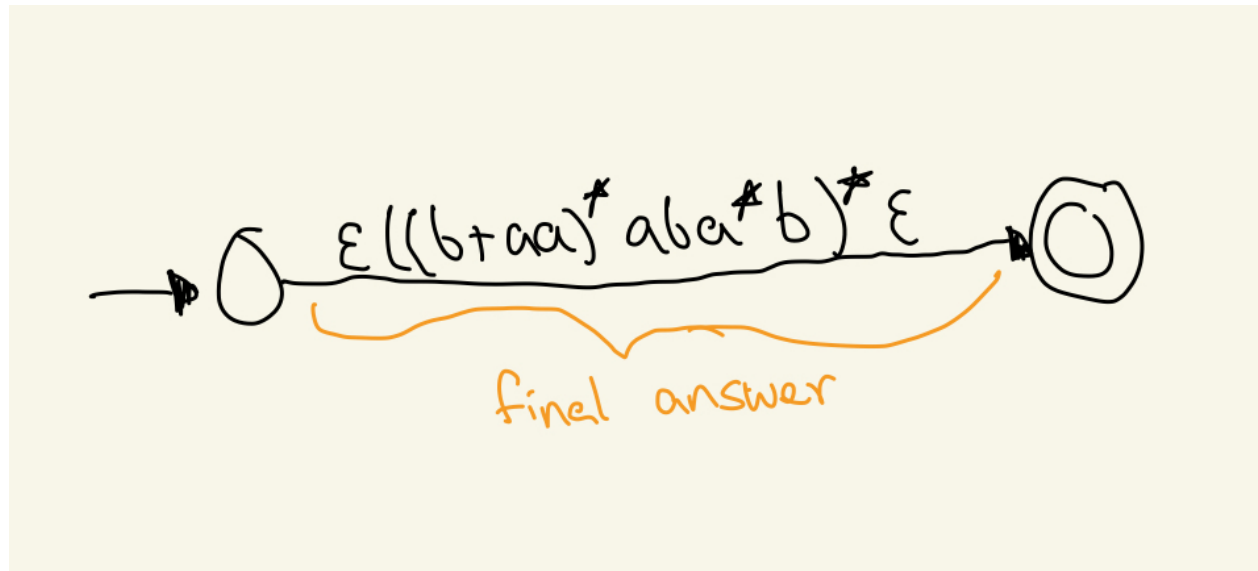
First we'll add a dummy starting and accepting state, and now we'll start by removing q_1 .



After removing q_1 we have the following NFA. Now to remove q_2 .



After removing q_2 we have the following NFA. Now to remove q_0 .



Now all the states have been removed, besides the dummy ones, meaning that we are at our final answer with the corresponding regular expression for the NFA being:

$$\epsilon((b + aa)^*aba^*b)^*\epsilon$$