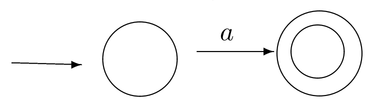
CS 420, Springl 2019
Homework 3 Solutions

1. Convert the regular expression $(ab \cup c)^*$ into an NFA using the method from class (which is the same as the method from the book and is different from the method in JFLAP).

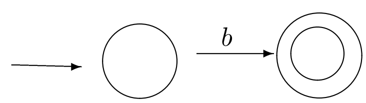
   Solution:

   First we have the following NFAs for a, b, and c.

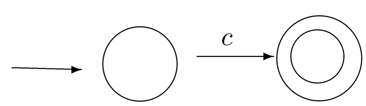
   a:

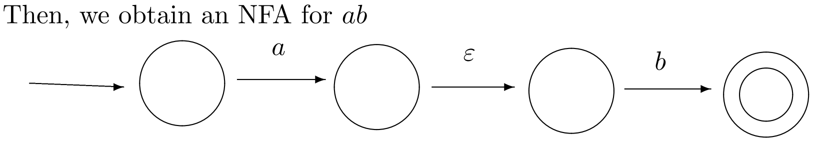
   b:

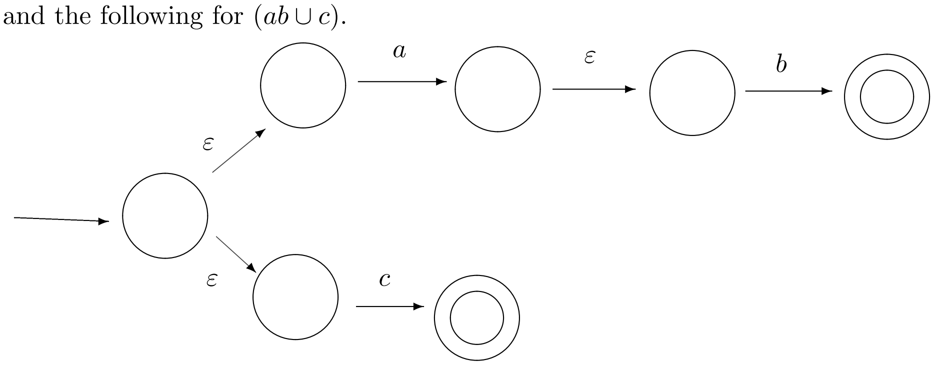
   c:

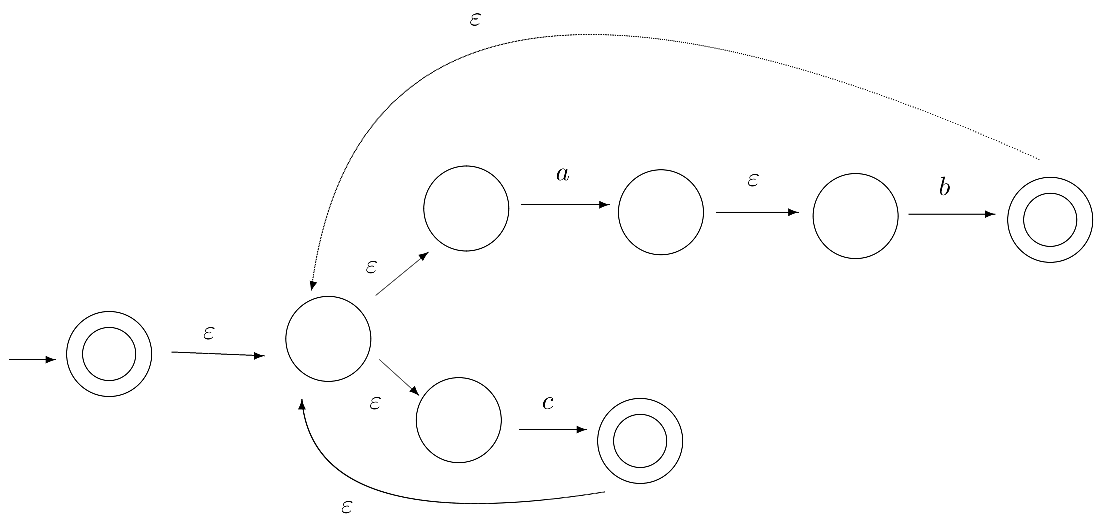
   Then, we obtain an NFA for ab

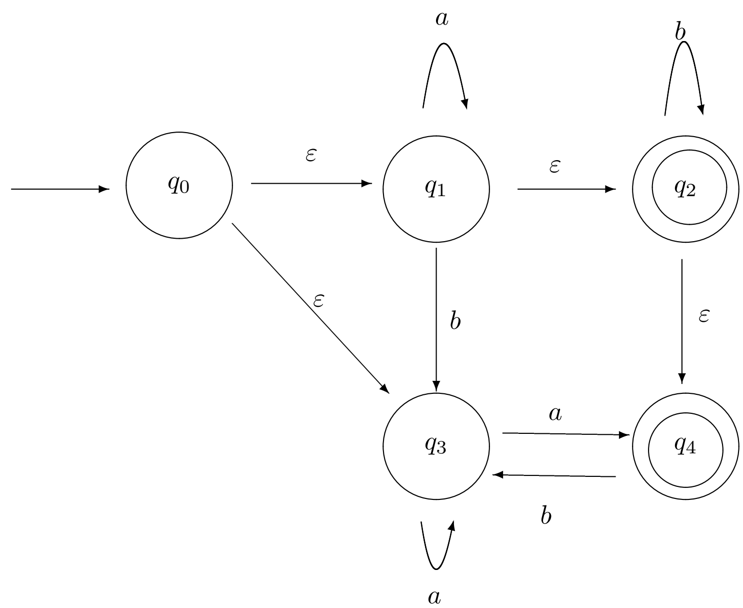
   and the following for $(ab \cup c)$.

Finally, we obtain the NFA below for $(ab \cup c)^*$.



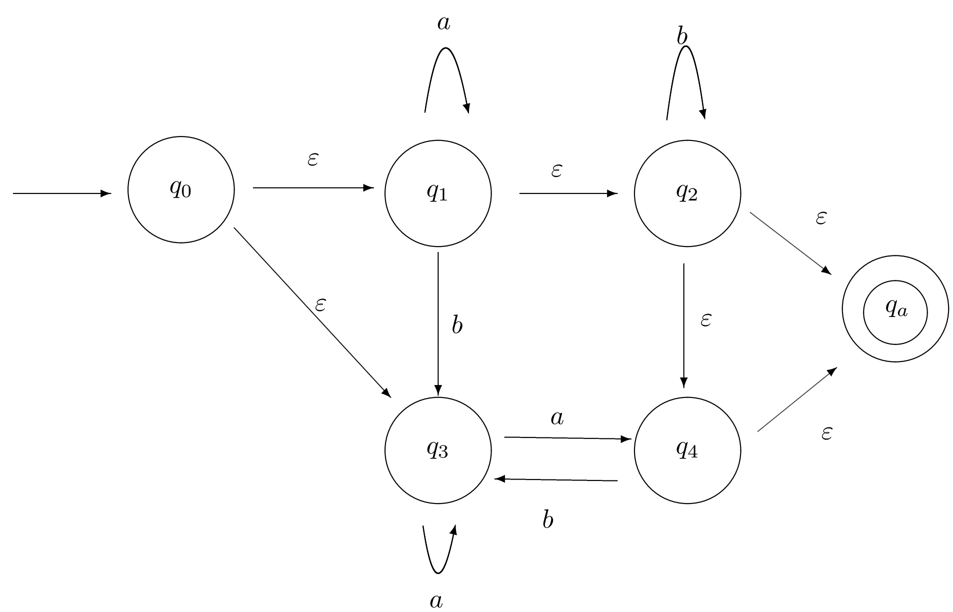
2. Let N be the following NFA:



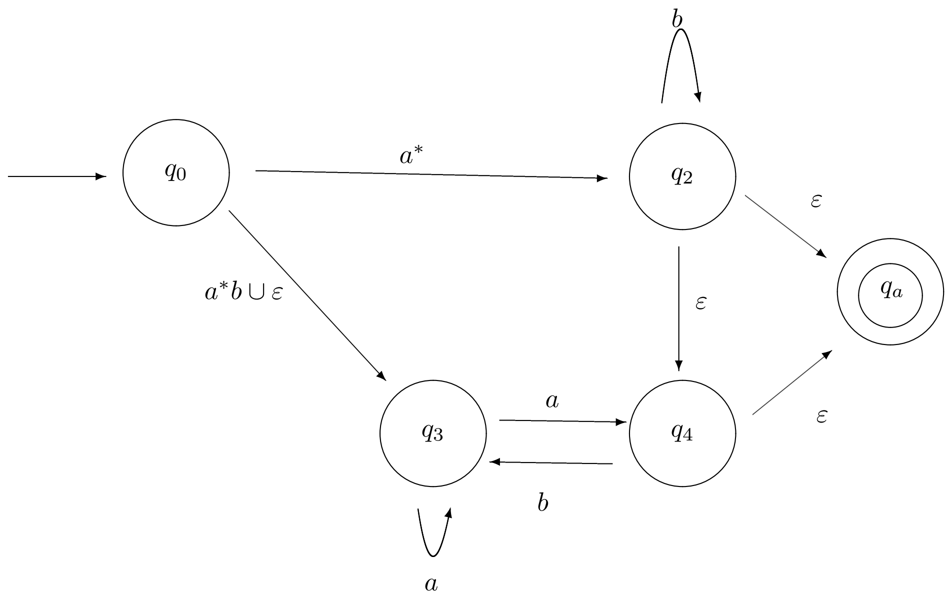
Convert N into a regular expression using the method from class (which is the same as the method in the book, and is not the same as the method in JFLAP).

Solution:

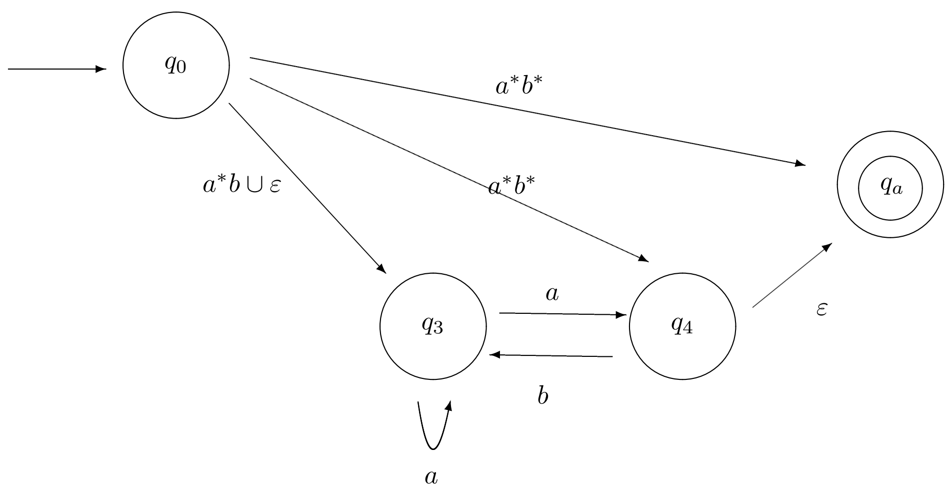
Normally, when we turn an NFA into a GNFA, we have to add a new start state with an ε-transition to the old start state, but in this case, the NFA has no transitions into the start state, so it is not necessary to add a new start state, so we get the following GNFA when we transform the NFA to a GNFA.


a
b
ε
ε
q_0
q_1
q_2
ε
q_a
ε
b
ε
a
q_3
q_4
ε
b
a

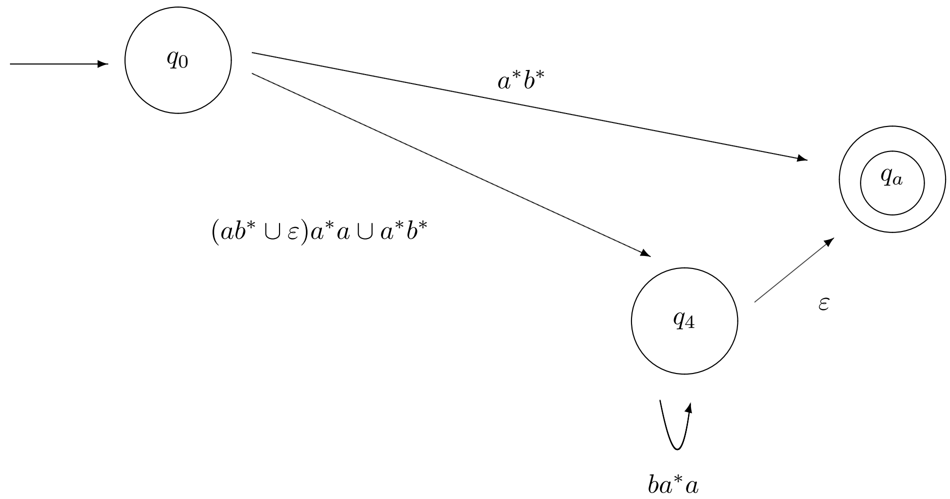
Eliminating state q_1 gives



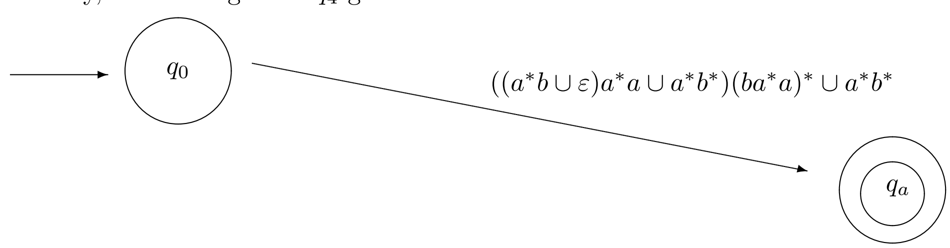
Eliminating state q_2 gives



Eliminating state q_3 gives



Finally, eliminating state q_4 gives



and the regular expression is $((a^*b \cup \varepsilon)a^*a \cup a^*b^*)(ba^*a)^* \cup a^*b^*$.

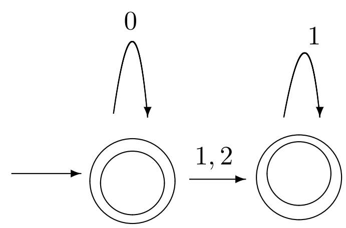
3. **Problem 1.31** Show that if A is regular so is A^R.
Proof: Since A is regular,there is some NFA $M = (Q, \Sigma, \delta, q_0, F)$ that recognizes A. We construct an NFA M' such that $L(M') = A^R$. The idea is that the start state of M' is the accept state of M, the accept state of M' is the start state of M, and for every transition in M, there is a transition with the same label in M', but going in the reverse direction. The problem with this is that M may have more than one accept state. Since M' can have only one start state, we add a new start state to M'

and ε transitions from the new start state to all the accept states of M. The formal construction is:
$M' = (Q \cup \{q_s\}, \Sigma, \delta', q_s, \{q_0\})$ where q_s is a new state and

$$\delta'(q, a) = \begin{cases} \{p : q \in \delta(p, a)\} & \text{if } q \neq q_s \\ F & \text{if } q = q_s, a = \varepsilon \\ \emptyset & \text{if } q = q_s, a \neq \varepsilon \end{cases}$$

4. (a) Let M be the NFA given in the solution to Problem 2(b) on Homework 2. Give an NFA N with two states and no ε-transitions that recognizes the same language. (Your NFA can have more than one accept state.)

Solution:



(b) Generalize what you did in Part (a) of this problem by proving the following theorem:

Theorem: If M is an NFA, then there is an NFA N with the following properties

(1) N has the same number of states as M.
(2) N has no ε-transitions.
(3) $L(N) = L(M)$.

Solution: We obtain N from M by doing three things:

1. We remove all ε-transitions.
2. If in M there is a series of ε-transitions that leads from a state p to an accept state q, then in N, p is also an accept state.
3. If in M there is a series of ε-transitions from a state p to s state q, and M can go from q to a state r reading a symbol a, then N can go directly from p to r reading a.

Formally, let $M = (Q, \Sigma, \delta, q_0, F)$. We define $N = (Q, \Sigma, \delta', q_0, F')$. If q is a state in Q, we use the notation $E_M(q)$ for the set of states that can be reached from q using 0 or more ε-transitions in M. The definition of N is now given by

$$F' = \{q \in Q | E_M(q) \cap F \neq \emptyset\}$$

and

$$\delta'(q,a) = \begin{cases} \emptyset & \text{if } a = \varepsilon \\ \bigcup\{\delta(p,a) | p \in E_M(q)\} & \text{if } a \neq \varepsilon \end{cases}$$

5. Use the Pumping Lemma to show that the following languages are not regular:

   (a) $\{a^n b^m c^r | n, m \geq 0 \text{ and } r = n + m\}$;

   Solution: Given $p \geq 1$, choose $s = a^p b^p c^{2p}$. Then, s is in the language and $|s| = 4p \geq p$. Given x, y, z with $s = xyz$, $|xy| \leq p$ and $|y| > 0$, we choose $i = 2$. Then, since $|xy| \leq p$, y consists only of a's, so $xy^2z = a^{p+|y|}b^p c^{2p}$ which is not in the language since $|y| > 0$, so $p + |y| + p \neq 2p$.

   (b) $\{0^n 1 0^m | n \leq m\}$.

   Solution: Given $p \geq 1$, choose $s = 0^p 1 0^p$. Then, s is in the language and $|s| = 2p + 1 \geq p$. Given x, y, z with $s = xyz$, $|xy| \leq p$ and $|y| > 0$, we choose $i = 2$. Then, since $|xy| \leq p$, y consists only of 0's, so $xyyz = 0^{p+|y|}10^p$ which is not in the language since $|y| > 0$, so $p + |y| \not\leq p$.

   (c) $\{c^4 a^n b^m | n \geq m\}$.

   Solution: Given $p \geq 1$, choose $s = c^4 a^p b^p$. Then, s is in the language and $|s| = 2p+4 \geq p$. Given x, y, z with $s = xyz$, $|xy| \leq p$ and $|y| > 0$, we choose $i = 0$.To see that xz is not in the language, we consider two cases.

   Case 1: y contains at least one c. Then, xz contains fewer than 4 c's so is not in the language.

   Case 2: y does not contain any c's. Then, since $|xy| \leq p$, y must contain only a's, so $xz = c^4 a^{p-|y|} b^p$ which is not in the language, since $|y| > 0$, so $p - |y| \not\geq p$.

   (d) $\{a^n b^m c^{2m} | n, m \geq 0\}$.

   Solution: Given $p \geq 1$, choose $s = b^p c^{2p}$. [Note that you must choose s to contain no a's since if there are any a's in s, then s can be pumped by letting $y = a$.] Then, s is in the language and $|s| = 3p \geq p$. Given x, y, z with $s = xyz$, $|xy| \leq p$ and $|y| > 0$, we choose $i = 2$. Since $|xy| \leq p$, y must consist only of b's, so $xy^2z = b^{p+|y|}c^{2p}$ which is not in the language since $|y| > 0$, so $2(p + |y|) \neq 2p$.