CS581 Theory of Computation: Homework #1

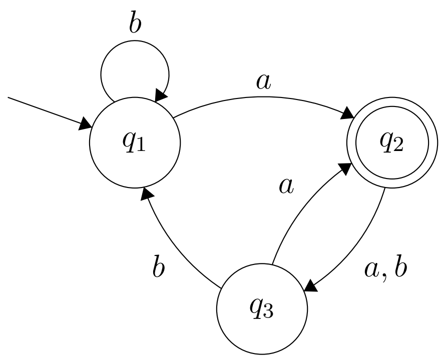
Due on January 20 2015 at 2:00pm

Harry H. Porter Winter 2016

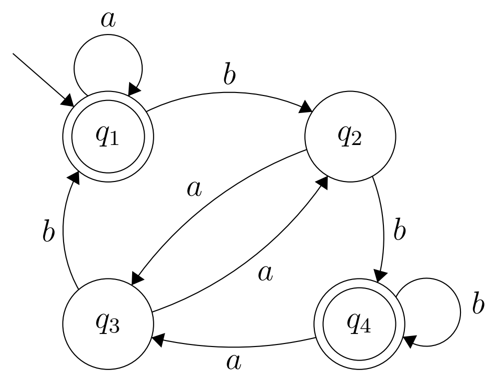
Konstantin Macarenco

Problem 1.1

The following are the state diagrams of two DFAs, M_1 and M_2. Answer the following questions about each of these machines.



M_1



M_2

a. What is the start state?
   Answ: M_1 - q_1, M_2 - q_1
b. What is the set of accept states?
   Answ: M_1 - $F = \{q_2\}$, M_2 - $F = \{q_1, q_4\}$
c. What sequence of states dows the machine go through on input aabb??
   Answ: $M_1 = \{q_1, q_2, q_3, q_1, q_1\}$, $M_2 = \{q_1, q_1, q_1, q_2, q_4\}$
d. Does the machine accept the string aabb?
   Answ: M_1 No, M_2 Yes
e. Does the machine accept the string ϵ?
   Answ: M_1 No, M_2 Yes

Problem 1.2 Give the formal description of the machines M_1 and M_2 from exercise 1.1.
M_1

1. $Q = \{q_1, q_2, q_3\}$
2. $\sum = \{a, b\}$
3. δ described as

Table 1: M_1 Transition function

	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_1

4. Start state $q_1 \in Q$
5. $F = \{q_3\} \subseteq Q$ Start state $q_1 \in Q$

M_2

1. $Q = \{q_1, q_2, q_3, q_4\}$
2. $\sum = \{a, b\}$
3. δ described as

Table 2: M_2 Transition function

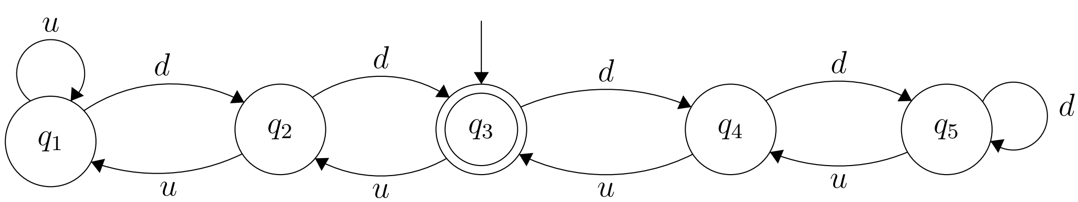
	a	b
q_1	q_1	q_2
q_2	q_3	q_4
q_3	q_2	q_1
q_4	q_3	q_4

4. Start state $q_1 \in Q$
5. $F = \{q_1, q_4\} \subseteq Q$ Start state $q_1 \in Q$

Problem 1.3

The formal description of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$, where δ is given by the following table. Give the state diagram of this machine.

	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5



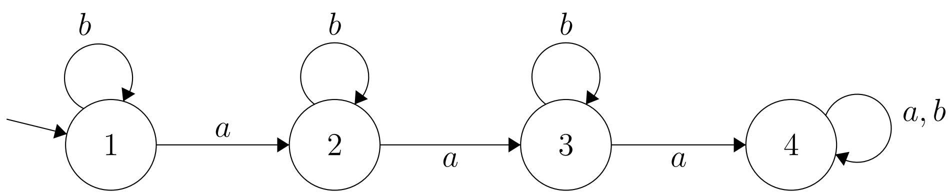
Problem 1.4

Each of the following languages is the intersection of two simpler languages. In each part, construct DFAs for the simpler languages, then combine them using the construction discussed in footnote 3 (page 46) to give the state diagram of a DFA for the language given. In all parts, $\sum = \{a, b\}$

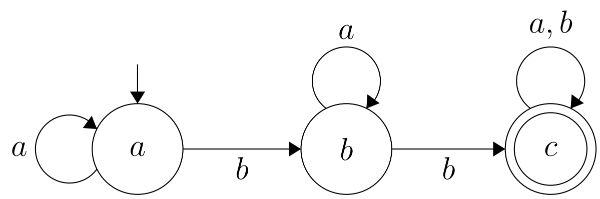
Problem 1.4 a

$\{w|w$ has at least three a's and at least two b's$\}$

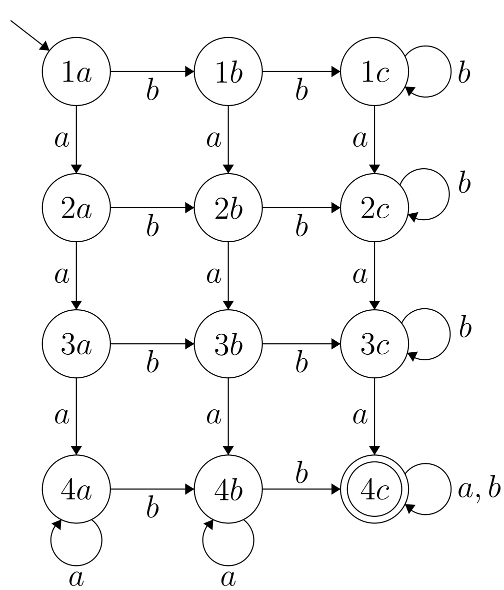
1. $\{w|w$ has at least three a's$\}$



2. $\{w|w$ has at least two b's$\}$



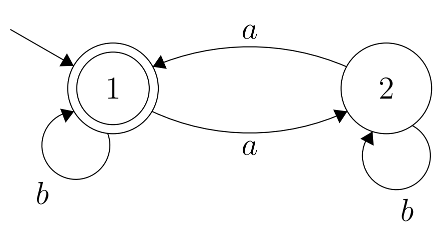
3.$\{w|w$ has at least three a's and at least two b's$\}$

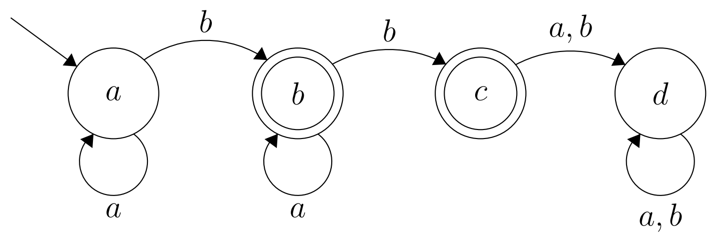


Problem 1.4 c

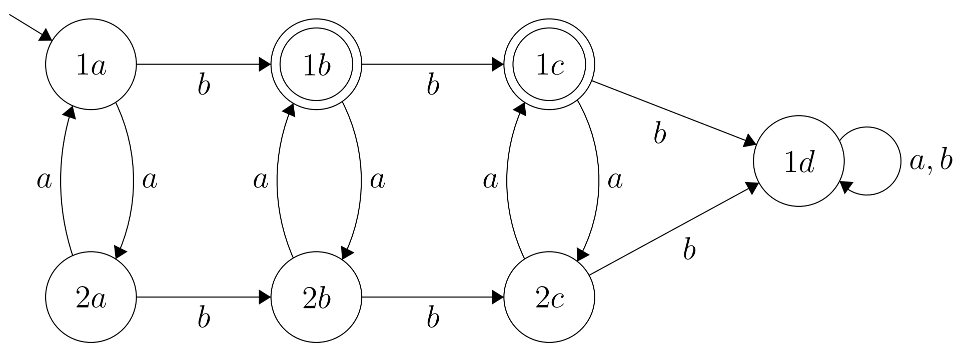
$\{w|w$ has an even number of a's and one ore two b's$\}$

1. $\{w|w$ has an even number of a's$\}$ 2. $\{w|w$ has one or two b's$\}$





3. $\{w|w$ has an even number of a's and one ore two b's$\}$

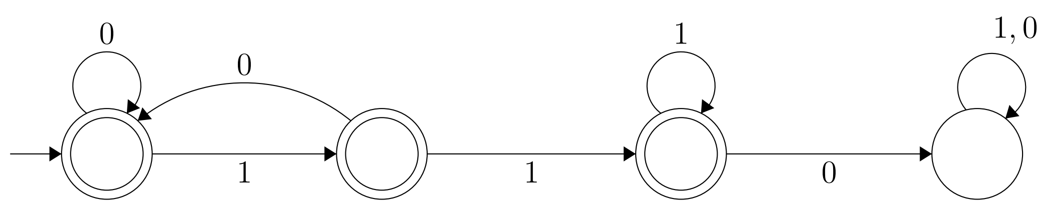


Problem 1.6

Give state diagrams of DFAs recognizing the following languages. In all parts, the alphabet is $\{0, 1\}$

Problem 1.6 f

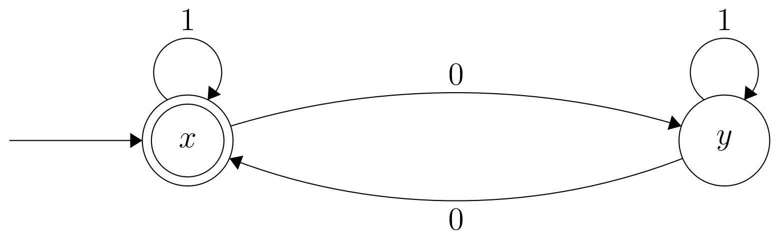
$\{w|w$ doesn't contain the substring 110$\}$



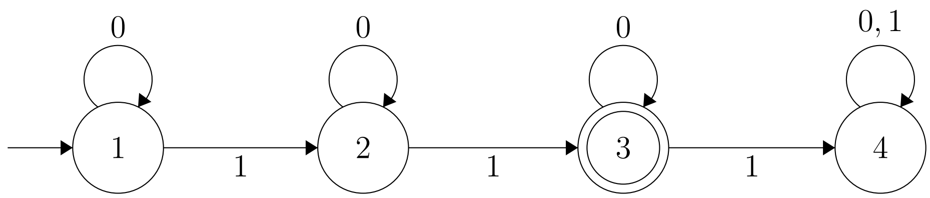
Problem 1.6 l

$\{w|w$ contains an even number os 0s, or contains exactly two 1s $\}$

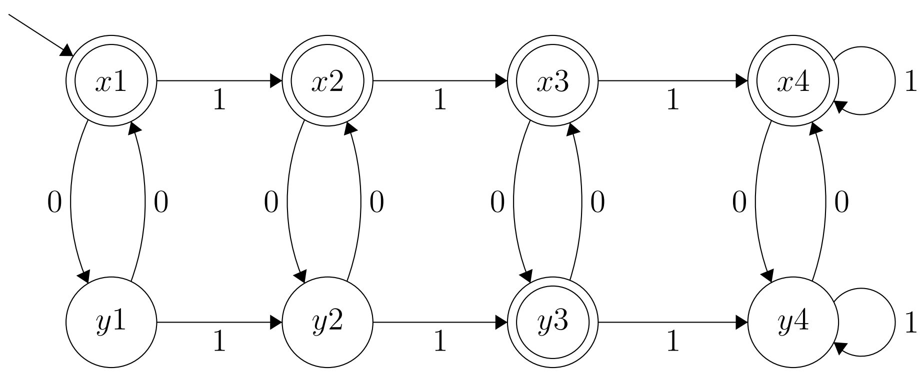
1. $\{w|w$ contains an even number os 0s$\}$



2. $\{w|w$ contains exactly two 1s $\}$



3. $\{w|w$ contains an even number os 0s, or contains exactly two 1s $\}$



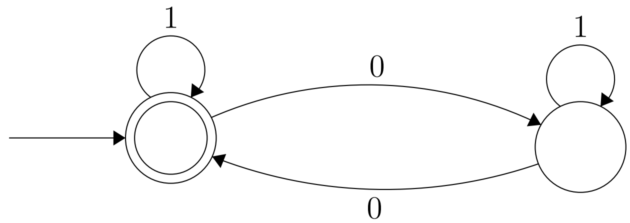
Problem 1.7

Give state diagrams of NFAs with the specified number of states recognizing each of the following languages. In all parts, the alphabet is $\{0, 1\}$.

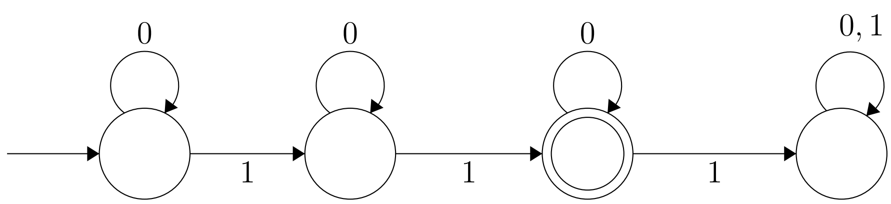
Problem 1.7 c

$\{w|w$ contains an even number os 0s, or contains exactly two 1s $\}$ With six states.

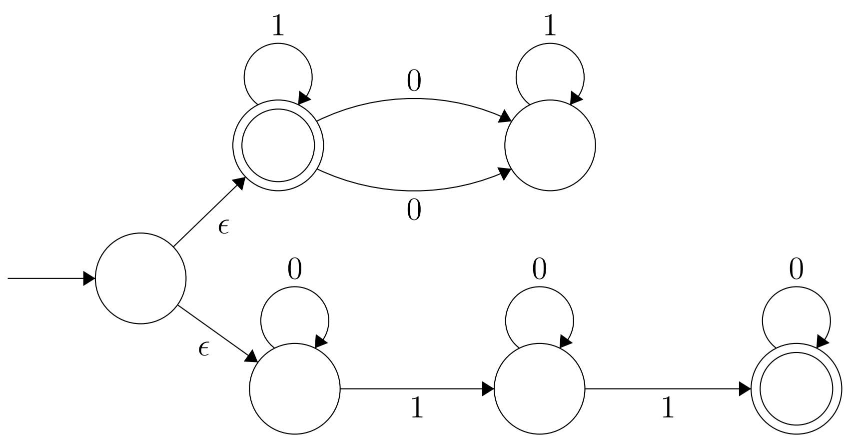
1. $\{w|w$ contains an even number os 0s$\}$



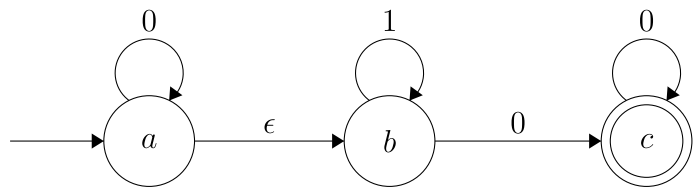
2. $\{w|w$ contains exactly two 1s $\}$



3. $\{w|w$ contains an even number os 0s, or contains exactly two 1s $\}$

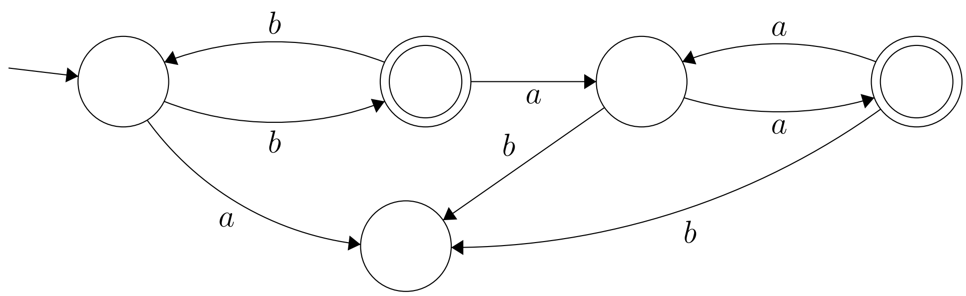


Problem 1.7 e
The language $0^*1^*0^+$ with three states



Problem 1.12
Let $D = \{w|w$ contains an even number of a's and an odd number of b's and does not contain the substring $ab\}$. Give a DFA with five state that recognizes D and a regular expression that generates D. (Suggestion: Describe D more simply.)

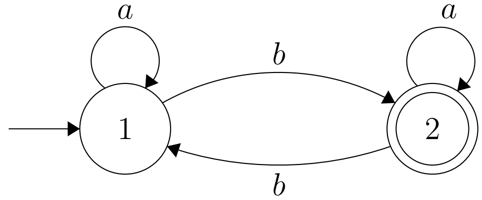
We can exclude all strings that starts with a from the DFA, more simply $D = \{b^{2k}a^{2k+1}\}$. D is recognized by the following DFA:



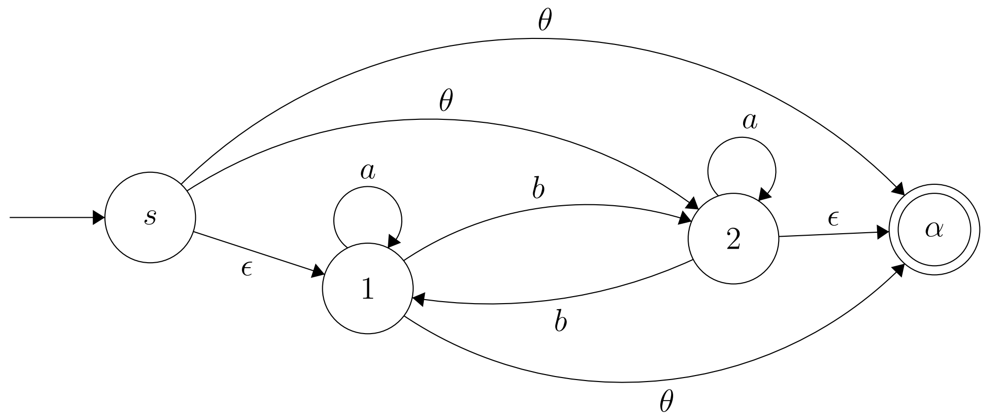
Regular expression:
$R_D = b(bb)^*(aa)^*$

Problem 1.21
Use the procedure described in Lemma 1.60 to convert the following finite automata to regular expression.

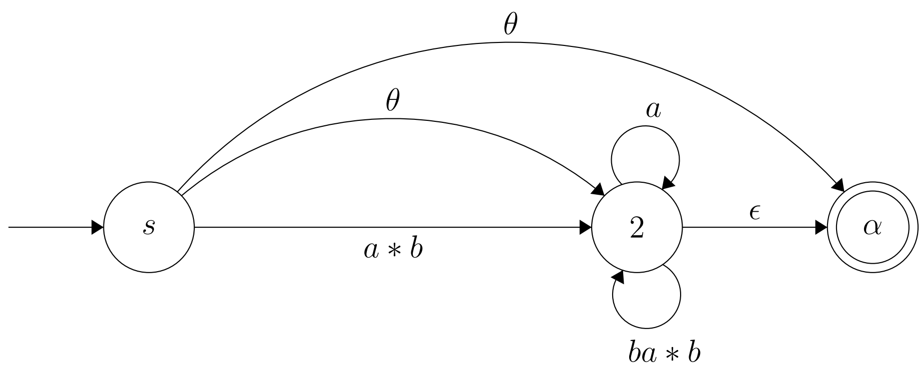
Problem 1.21 a

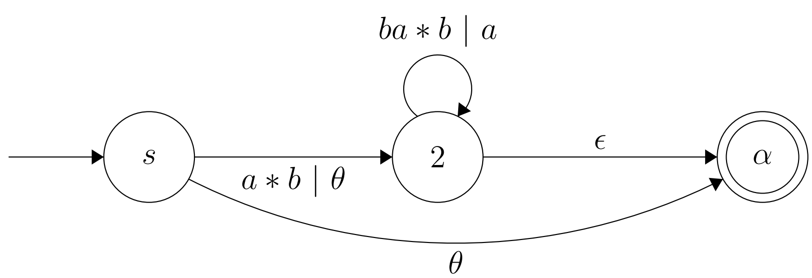


1. Turn it into GNFA

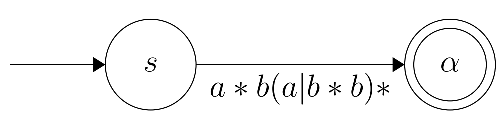


2. Rip state 1 out, and repair the connections.



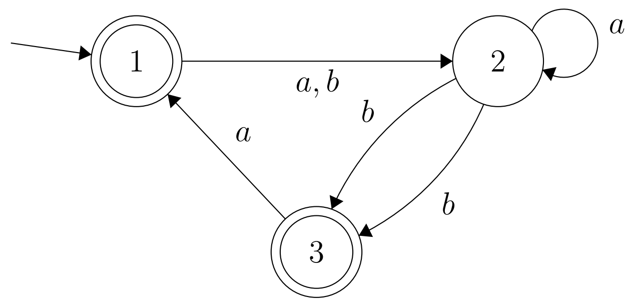


3. Rip state 2 out, and repair the connections.

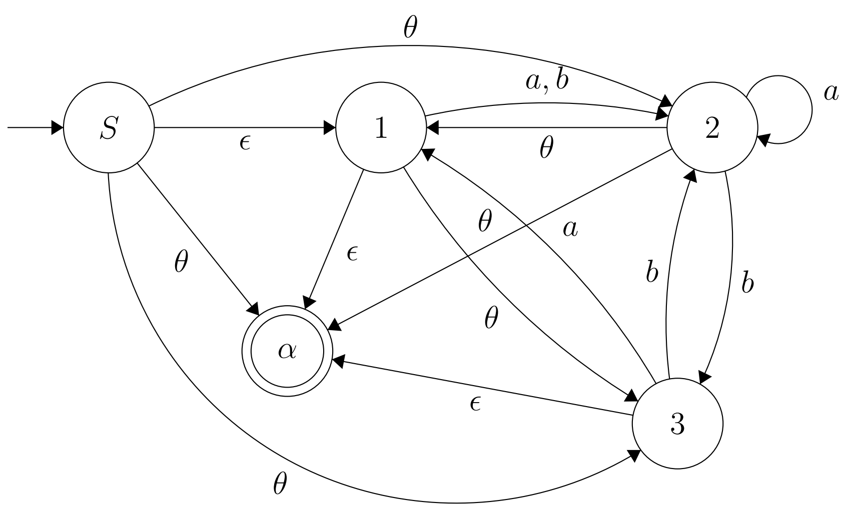


Answ: $a^*b(a|b^bb)^*$

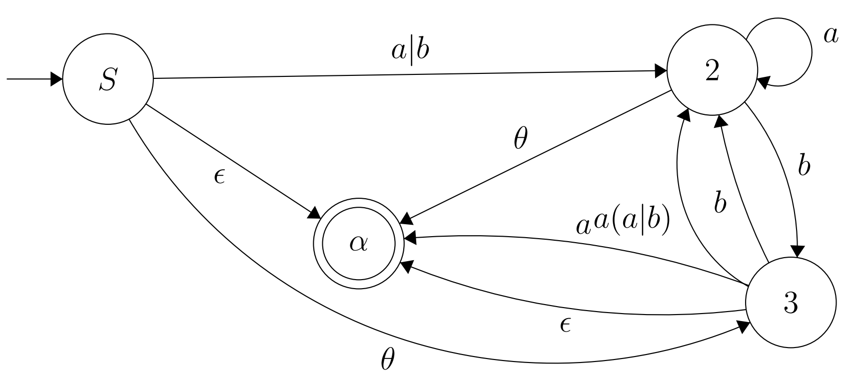
Problem 1.21 b



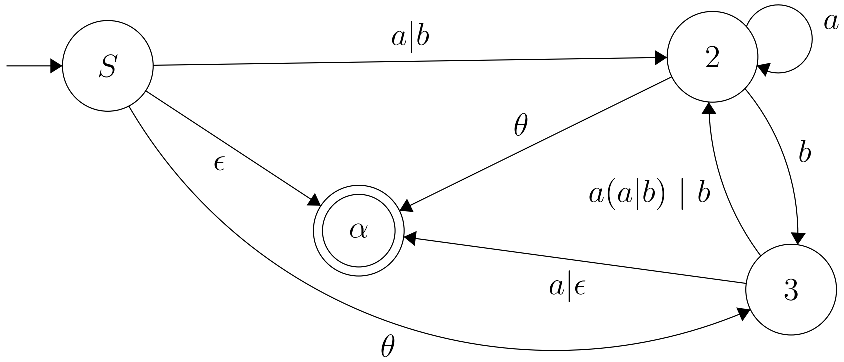
1. Convert DFA to GNFA.



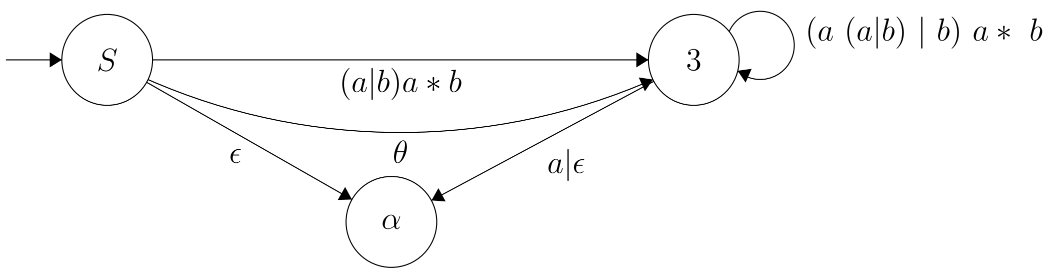
2. Rip state 1 out, and repair connections.



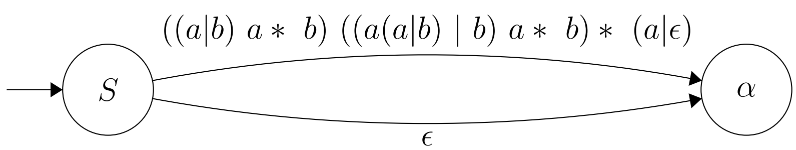
3. Unite multiple connections.



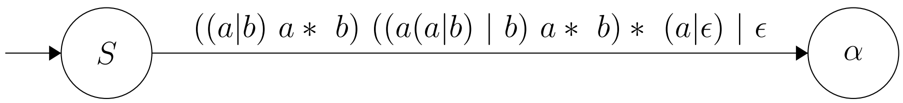
4. Rip out state 2, and repair broken edges.



5. Rip out state 3 and repair broken edges.



6. Unite the edges.



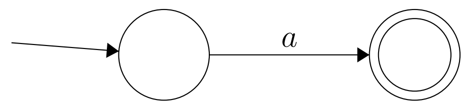
Answ: $((a|b)a^*b)((a(a|b)|b)a^*b)^*(a|\epsilon)|\epsilon$

Problem 1.28
Convert the following regular expression to NFA using the procedure given in Theorem 1.54. In all parts, $\sum = \{a, b\}$.

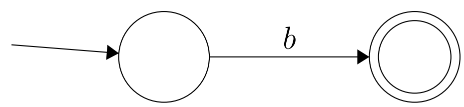
Problem 1.28 b

$a^+|(ab)^+$

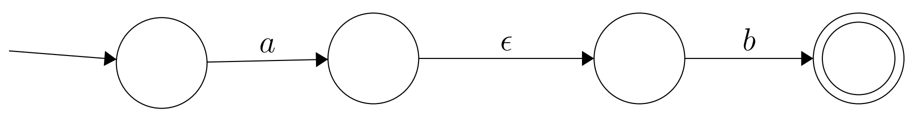
1. NFA that recognizes symbol a.



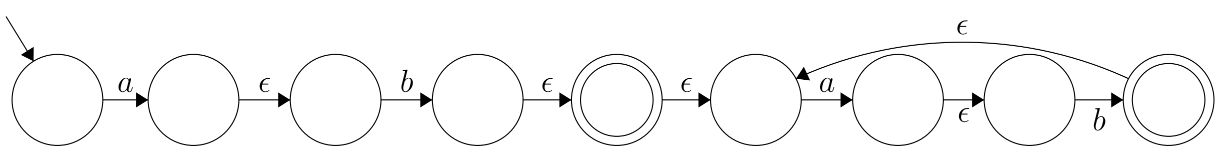
2. NFA that recognizes symbol b.



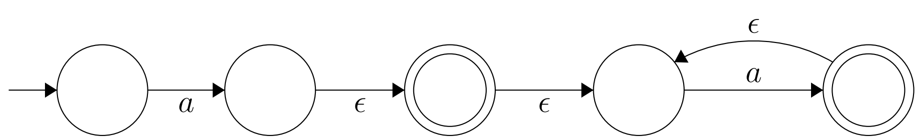
3. NFA that recognizes concatenation of 1 and 2, i.e string ab.



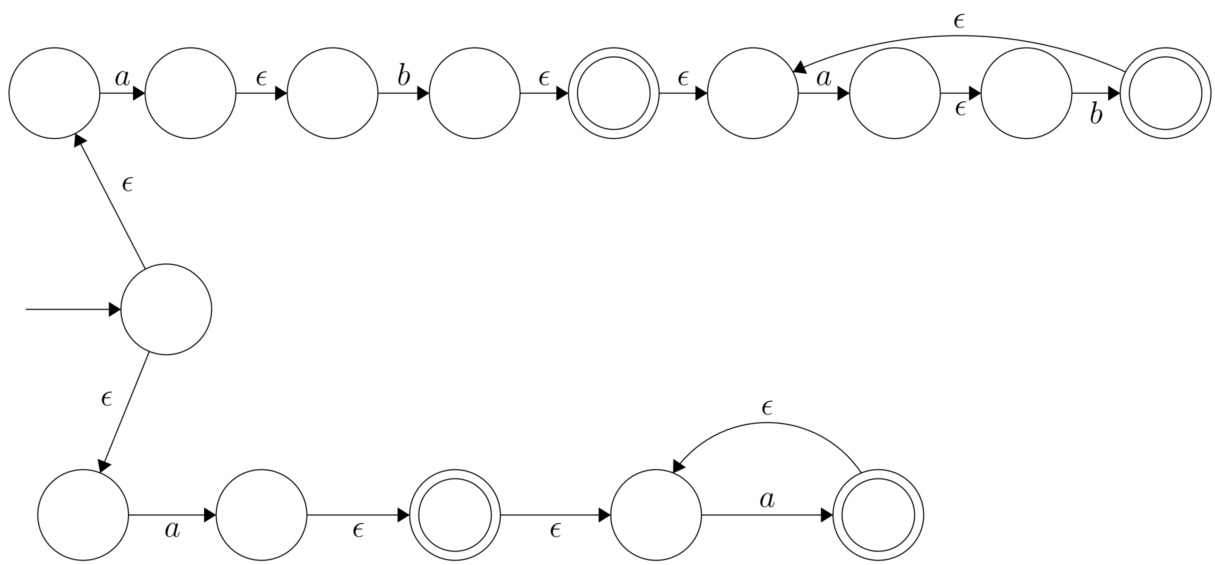
4. NFA that recognizes regular expression $(ab)^+$.



5. NFA that recognizes regular expression $(a)^+$.



Answ: Union of 4 and five for regular expression $R = a^+|(ab)^+$



Problem 1.29
Use the pumping lemma to show that the following languages are not regular.

Problem 1.29 b
$A_2 = \{www|w \in \{a,b\}^*\}$

Answ: Assume that $A_2 = \{www|w \in \{a,b\}^*\}$ is regular. Let p be the pumping length given by the pumping lemma. Choose s to be the string $a^pba^pba^pb$. Because s is a member of A_2 and s is longer than p, the pumping lemma guarantees that s can be split into three pieces, $s = xyz$, where $|xy| \leq p$, hence y can only be contained in first a^p. Since $y \geq 1$, let $y = a^i$, $i > 0$. However, $xy^2z = a^{p+k}ba^pba^pb$, where $p + k > P$, is not in A_2. That is s cannot be pumped. This is a contradiction. Thus, A_2 is not regular.

Problem 1.31
For any string $w = w_qw_2 \cdot w_n$, the reverse of w, written w^R, is the string w in reverse order, $w_n \cdots w_2w_1$. For any language A, let $A^R = \{w^R|w \in A\}$. Show that if A is regular, so is A^R.

Answ: By theorem if A is regular, then there is a DFA $M = (Q, \sum, \delta, q_0, F)$ such that M recognizes A. We will show that it is possible to construct an NFA $N = (Q', \sum', \delta', q_0', F')$ that will recognize A^R.

Informally:
1. Reverse all the connections in the automaton.
2. Add a new state q_f
3. Draw ϵ connection from state q_f to every final state.
4. Make all the final states normal states.

5. Make start state final state.
6. Make q_f the start state.

Formally

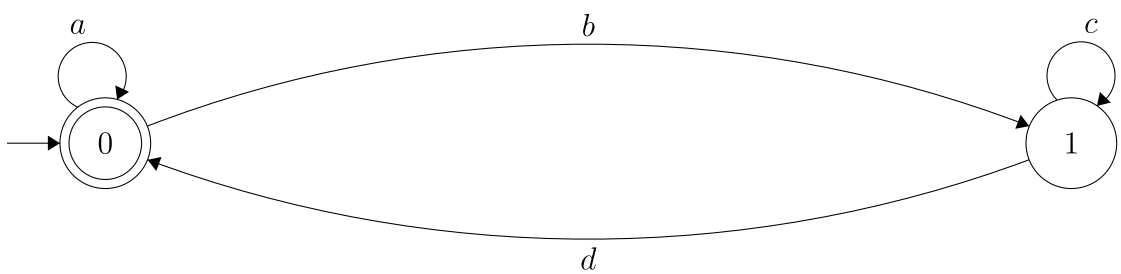
- New set of states $Q' = Q \cup \{q_0'\}$, where q_0 is the new start state
- New set of final states $F' = \{q_0\}$ i.e. we will accept only in the start state of the original DFA.
- New transition function δ' is definde as follows
  $\delta(q, a)' = F$, if $q = q_0'$ and $a = \epsilon$
  $\delta(q, a)' = \delta^{-1}(q, a)$, if $q \neq q_0'$ and $a \neq \epsilon$.
  $\delta(q, a)' = \emptyset$, otherwise.

δ $q_0, q_1, \cdots, q_n$ is in accepting computation of M on input $w_1 w_2 \cdots w_n$ if and only if δ' $q_n, \cdots, q_1, q_0$ is an accepting computation of N on input $w_n \cdots w_2 w_1$. Thus, N recognizes A^R. Therefore for any regular language A there exists an NFA that recognizes A^R.

Problem 1.32

Let w^R denote the reverse of the string w. For any language A, let $A^R = \{w^R | w \in A\}$. Then if A is regular, so is A^R (as in previous exercise).
Construction for the automata for the language A^R comes down to tracking of carry bit. If there is no carry bit, then language is in accept state, if there is a carry bit, then in fail.

$$a = \begin{bmatrix}0\\0\\0\end{bmatrix} \begin{bmatrix}1\\0\\1\end{bmatrix} \begin{bmatrix}0\\1\\1\end{bmatrix} \quad b = \begin{bmatrix}1\\1\\0\end{bmatrix} \quad c = \begin{bmatrix}0\\1\\0\end{bmatrix} \begin{bmatrix}1\\0\\0\end{bmatrix} \begin{bmatrix}1\\1\\1\end{bmatrix} \quad d = \begin{bmatrix}0\\0\\1\end{bmatrix}$$



Problem 1.46
Prove tat the following languages are not regular. You may use the pumping lemma and the closure of the class of regular languages under union, intersection and compliment.

Problem 1.46 c

$L = \{w | w \in \{0, 1\}^*$ is not a palindrome $\}$

We can show that L is not regular, by showing that it's compliment $L' = \{w | w \in \{0, 1\}^*$ is a palindrome $\}$ is not regular.
Choose $s = 0^p 1 0^p = xyz$, a palindrome in L'. Here xy contains only 0's, since $xy \leq p$. Let $y = 0^k$, should be in the language for all $k \geq 0$, however $xy^0z = 0^{p-k}10^p$ is not in the language. Thus, L' is not regular, therefore L is not regular as well.