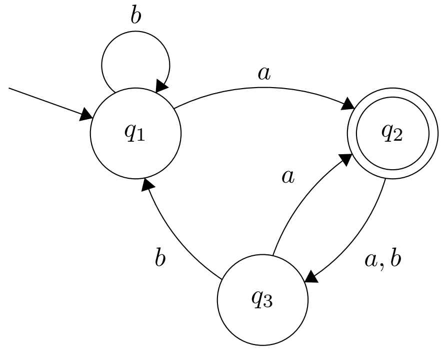
CS581 Theory of Computation: Homework #1

Due on January 20 2015 at 2:00pm

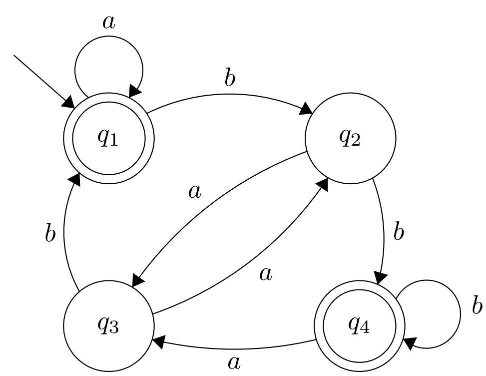
Harry H. Porter Winter 2016

Konstantin Macarenco

Problem 1



M_1



M_2

a. Start State: M_1 - q_1, M_2 - q_1
b. Set of accept states M_1 - $F = \{q_2\}$, M_2 - $F = \{q_1, q_4\}$,
c. $M_1 = \{q_1, q_2, q_3, q_1, q_1\}$, $M_2 = \{q_1, q_1, q_1, q_2, q_4\}$
d. M_1 No, M_2 Yes
e. M_1 No, M_2 Yes

Problem 2

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

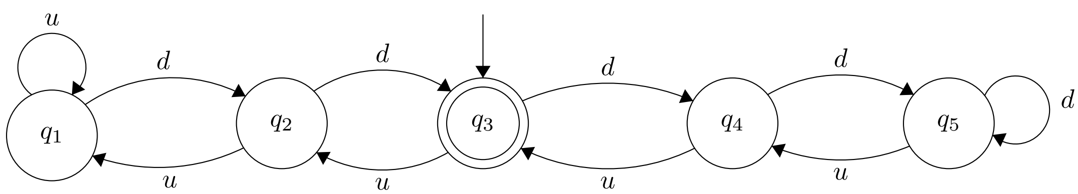
1. $Q = \{q_1, q_2, q_3, q_4\}$
2. $\sum = \{a, b\}$
3. δ described as

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

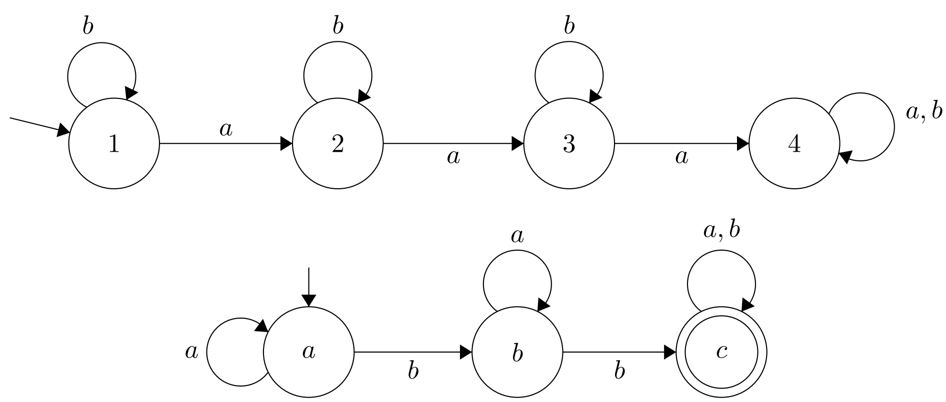
4. Start state $q_1 \in Q$
5. $F = \{q_1, q_4\} \subseteq Q$ Start state $q_1 \in Q$

Problem 3

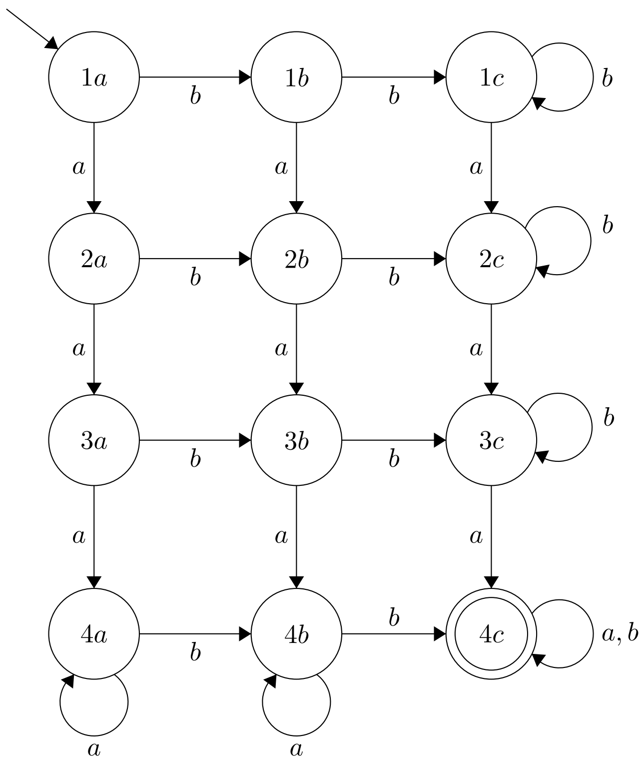


Problem 4

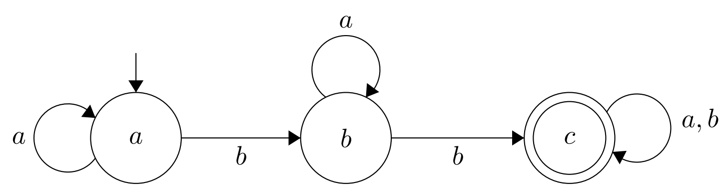
a

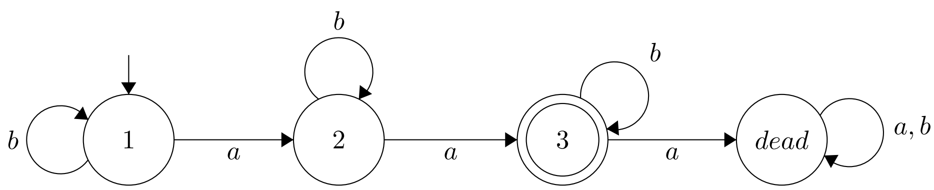


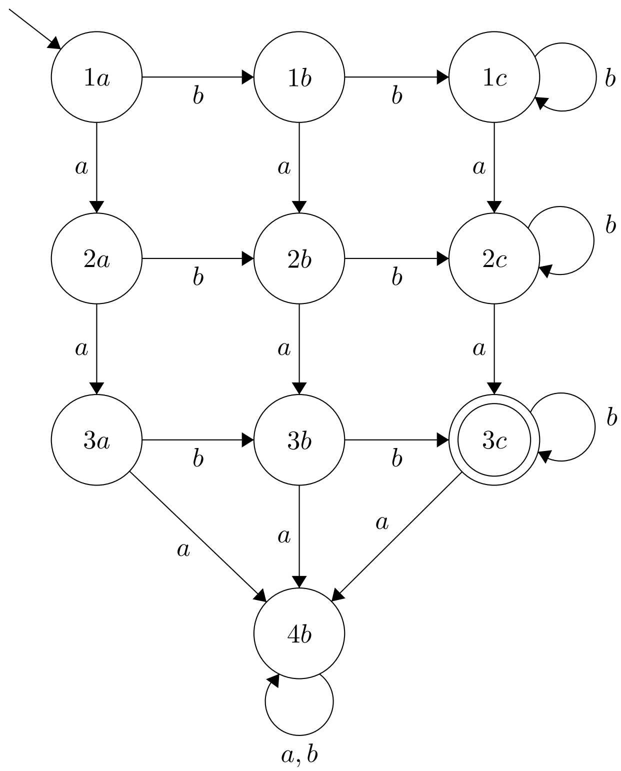
Problem 4 continued on next page...



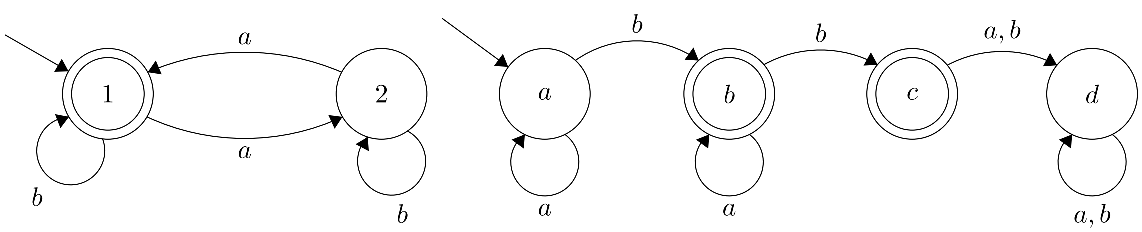
b.

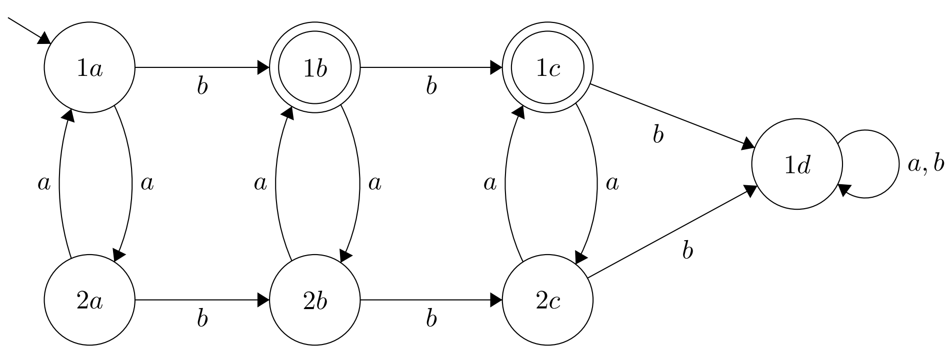






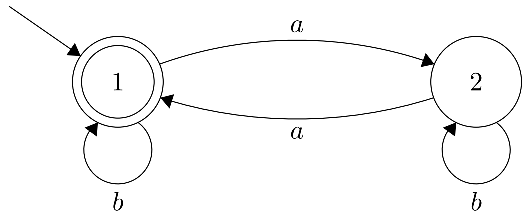
c

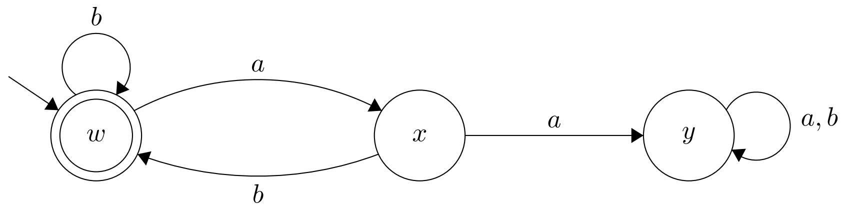


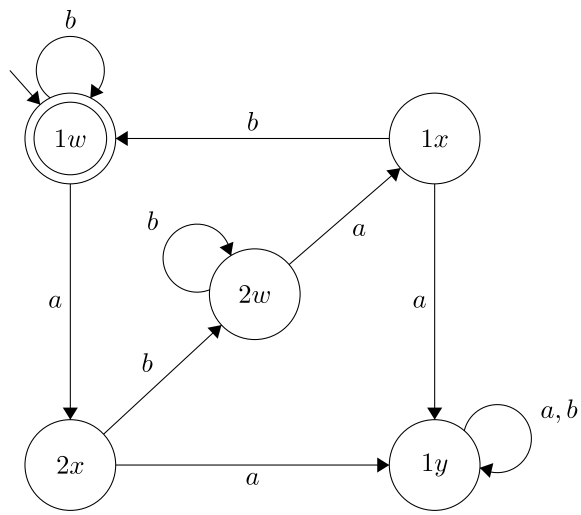


Problem 4 continued on next page. . .

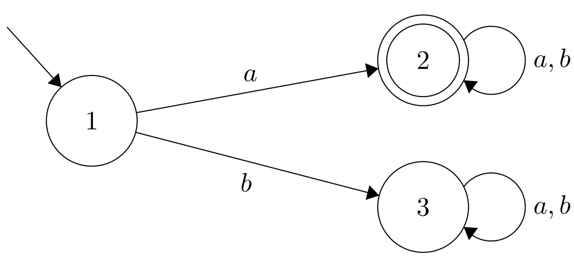
d

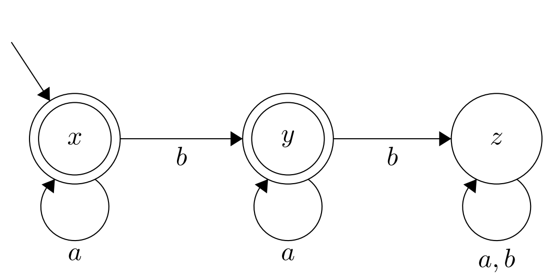

1
2
a
a
b
b

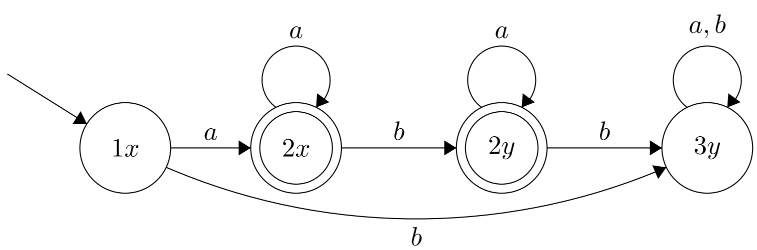

w
x
y
a
b
a
b
a, b


1w
1x
2w
2x
1y
b
b
a
b
a
a
b
a
a, b

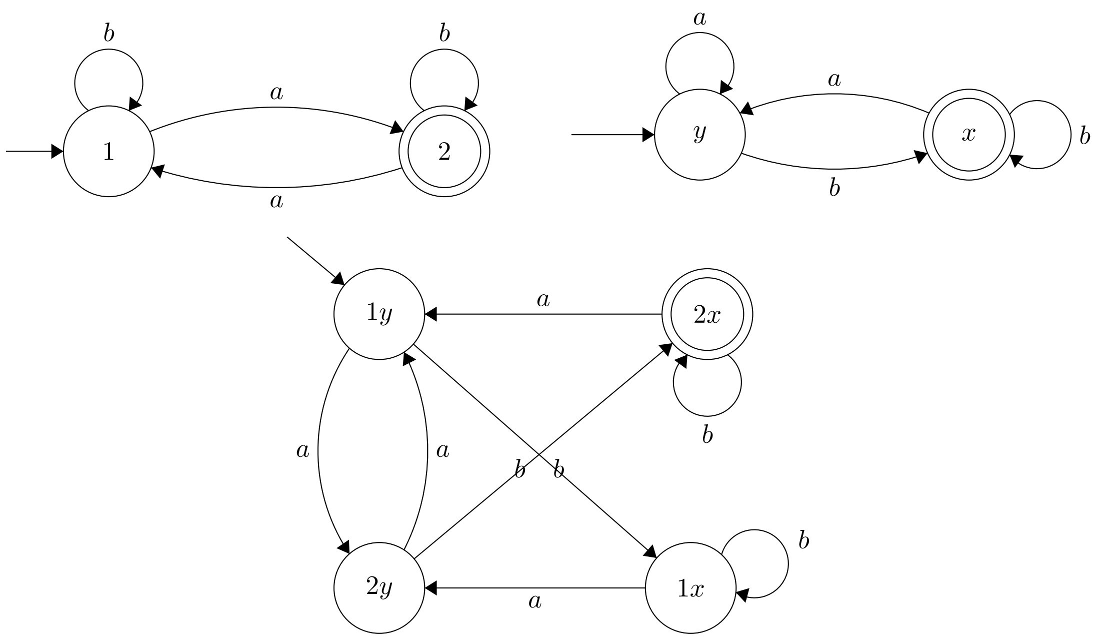
e


1
2
3
a
b
a, b
a, b


x
y
z
b
b
a
a
a, b

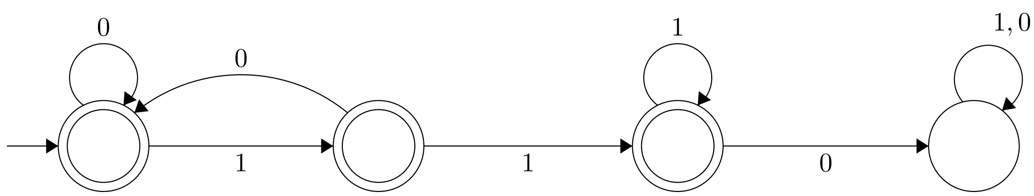

1x
2x
2y
3y
a
b
b
b
a
a
a, b

f



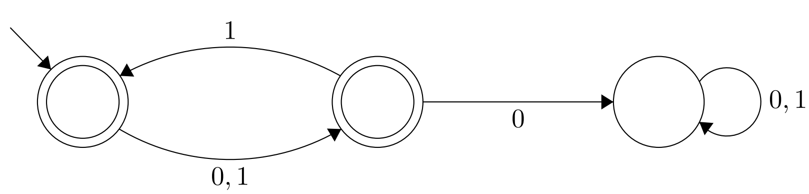
Problem 5

1.6f



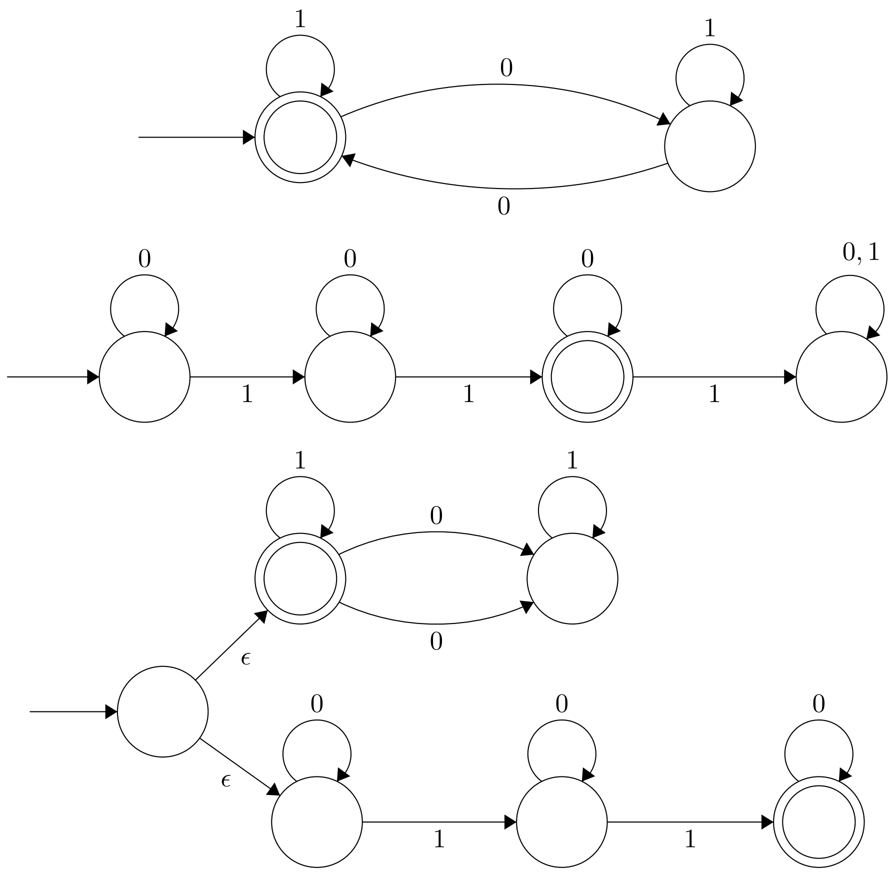
Problem 6

1.6i

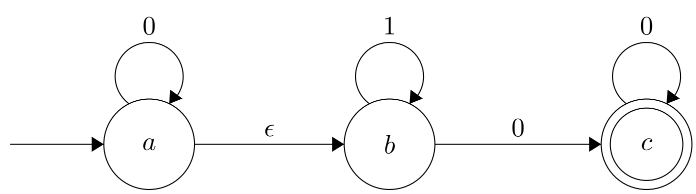


Problem 7

1.7c

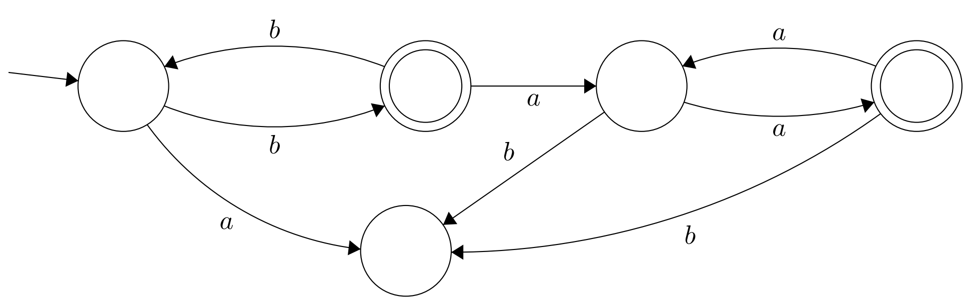


1.7e



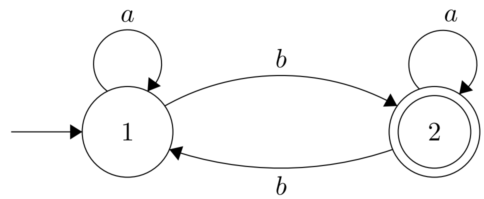
Problem 8

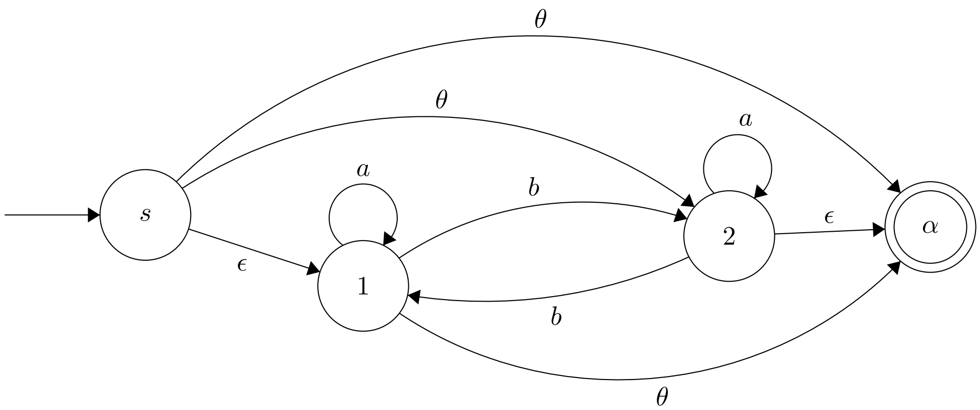
1.12

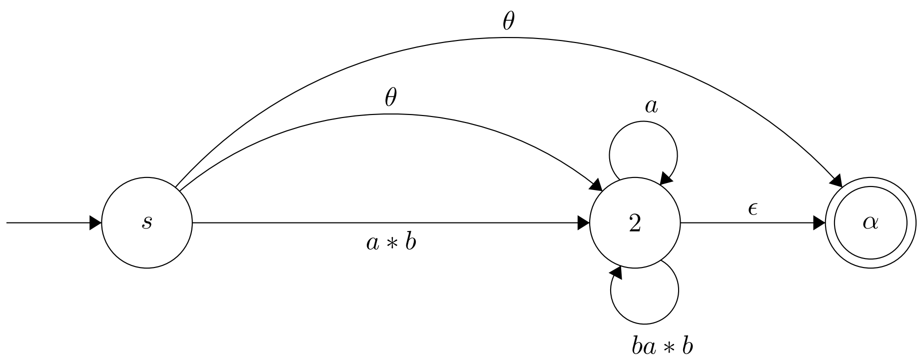


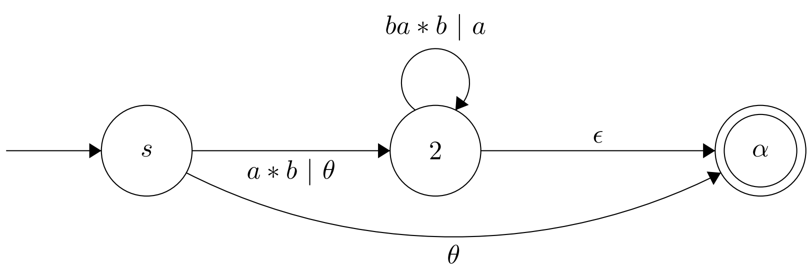
Problem 9

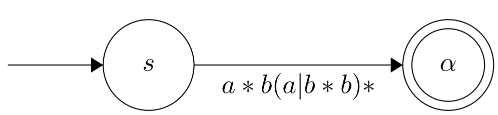
1.21









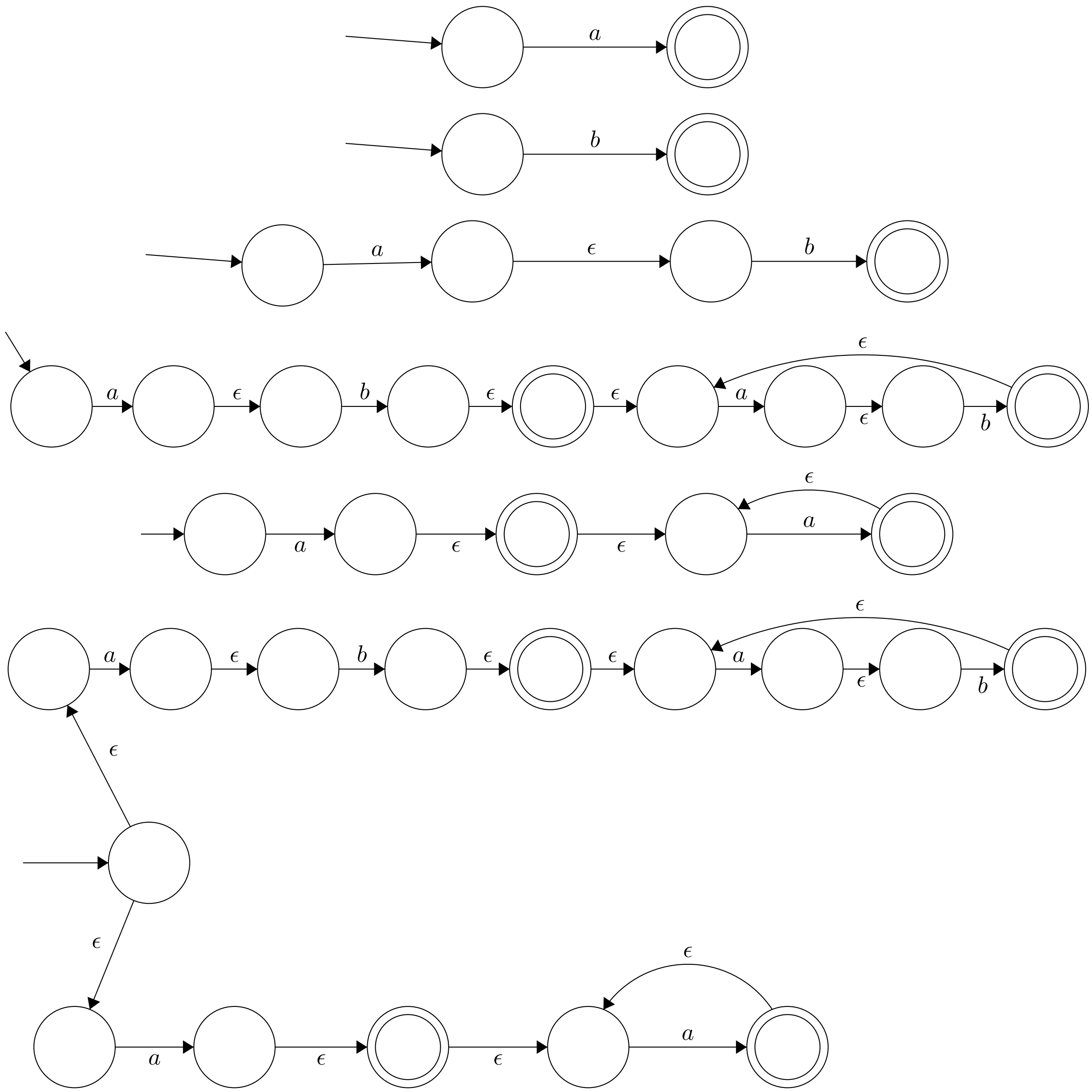


Problem 10

1.28b

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

$a^{+}|(ab)^{+}$



Problem 11

1.29b
Use the pumping lemma to show that the following languages are not regular.
$A_2 = \{www|w \in \{a, b\}^*\}$

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 $a^p b a^p b a^p b$.

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 12

1.31

For any string $w = w_q w_2 \cdot w_n$, the reverse of w, written w^R, is the string w in reverse order, $w_n \cdots w_2 w_1$. For any language A, let $A^R = \{w^R | w \in A\}$. Show that if A is regular, so is A^R.

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

- $Q' = Q \cup \{q_0'\}$, where q_0 is the new start state
- $F' = \{q_0\}$, is the new final state. We accept only in the start state of the original DFA.
- Define transition functions δ'.

$$\delta(q, a)' = \begin{cases} F, & \text{if } q = q_0' \text{ and } a = \epsilon \\ \delta^{-1}(q, a), & \text{if } q \neq q_0' \text{ and } a \neq \epsilon. \\ \emptyset, & \text{otherwise.} \end{cases} \tag{1}$$

By how we dfined $\delta', q_0, q_1, \cdot, q_n$ is an accepting computation of M on input $w_1 w_2 \cdot w_n$ if and only if $q_0', q_n, \cdot, q_1, q_0$ is an accepting computation of N on input $w_n \cdot w_2 w_1$ (since $\delta(q_i, w_{i+1} = q_{i+i} iff q_i \in \delta'(q_{i+1}, w_{i+1})$ for $0 \leq i < n$ and $q_n \in \delta'(q_0', \epsilon)$). Thus, N recognizes A^R. Since for any regular language A there exists an NFA that recognizes A^R, we conclude that the class of regular languages is closed under reverse.

Problem 13

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.

Now, the construction of the automata for the language B^R is simple. We get columns of size 3 as our alphabets at each stage, we will need to keep track of two things, wheter there is a carry or not. Depending on wheter there was a carry or not, we just need to verify that the provided 3-column is consistent.

Figure 4 shows that NFA for B^R.

Problem 14

1.46c
$L = \{w | w \in \{0,1\}^*$ 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. Let $y = 0^k, k > 0$. Thus $xy^0 z = 0^{p-k} 1 0^p$ is not in the language. Thus, it's not regular.