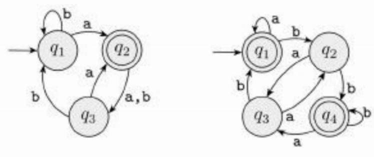
Automata theory A-1.1

1.

[A]**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?
b. What is the set of accept states?
c. What sequence of states does the machine go through on input aabb?
d. Does the machine accept the string aabb?
e. Does the machine accept the string ε?

		M_1	M_2
a.	Start States	q_1	q_1
b.	Accepted states	$\{q_2\}$	$\{q_1, q_4\}$

c.

M_1	M_2
$\delta(q_1, a) = q_2$	$\delta(q_1, a) = q_1$
$\delta(q_2, a) = q_3$	$\delta(q_1, a) = q_1$
$\delta(q_3, b) = q_1$	$\delta(q_1, b) = q_2$
$\delta(q_1, b) = q_1$	$\delta(q_2, b) = q_4$

d. M_2 accepts the string while M_1 doesn't.

e. No, DFA's do not accept ε.

2. [A]1.2 Give the formal description of the machines M_1 and M_2 pictured in Exercise 1.1.

$M_1 = (\{q_1, q_2, q_3\}, \{a, b\}, \delta_1, q_1, \{q_2\})$

where δ_1 is:

	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_1

$M_2 = (\{q_1, q_2, q_3, q_4\}, \{a, b\}, \delta_2, q_1, \{q_1, q_4\})$

where δ_2 is:

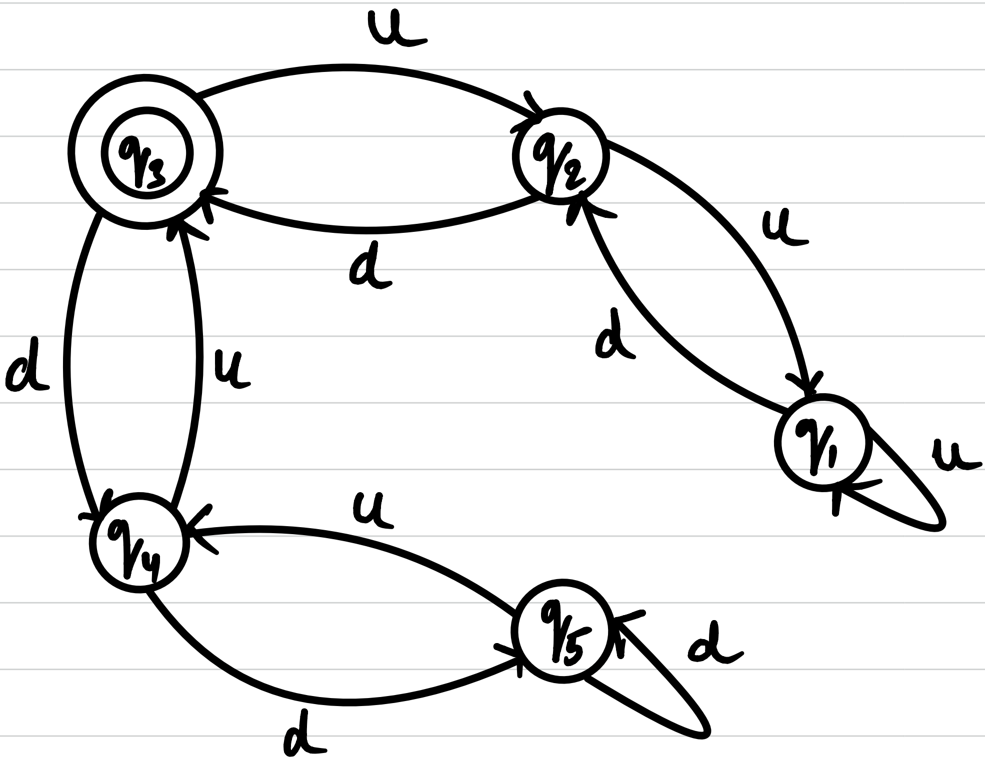
	a	b
q_1	q_1	q_2
q_2	q_3	q_4
q_3	q_2	q_1
q_4	q_3	q_4

3.

1.3 The formal description of a DFA M is $(\{q_1, q_2, q_3, q_4, q_5\}, \{\mathtt{u}, \mathtt{d}\}, \delta, q_3, \{q_3\})$, 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

State diagram:



4. [A]1.11 Prove that every NFA can be converted to an equivalent one that has a single accept state.

Let the initial NFA be N.

$$N = (Q, \Sigma, \delta, q_0, F)$$

To transform N such that F contains only one state, we need to create a new state q' that connect all $q \in F$ to it.

$\therefore$ we have the new NFA,

$$N' = (Q \cup \{q'\}, \Sigma, \delta', q_0, \{q'\})$$

where δ' is given by:

$$\delta'(q, a) = \begin{cases} \delta(q, a) & a = \epsilon,\ q \notin F \\ \delta(q, a) \cup \{q'\} & a = \varepsilon,\ q \in F \\ \emptyset & a \in \Sigma_\varepsilon,\ q = q' \end{cases}$$

5.

1.14 **a.** Show that if M is a DFA that recognizes language B, swapping the accept and nonaccept states in M yields a new DFA recognizing the complement of B. Conclude that the class of regular languages is closed under complement.

b. Show by giving an example that if M is an NFA that recognizes language C, swapping the accept and nonaccept states in M doesn't necessarily yield a new NFA that recognizes the complement of C. Is the class of languages recognized by NFAs closed under complement? Explain your answer.

a. Let M' be the DFA upon swapping accepted and non-accepted states in M.
Consider a string 's' that is accepted by M' and hence belongs to language B'.
Clearly 's' will not be accepted by M since it'll end up at the non-accepted state. If we were to consider a string 's' that wasn't accepted by M' then ~~wkt~~ it is accepted by M.

From the two cases we've seen above,

	M	M'
$s \in B$	✓	x
$s \in B'$	x	✓

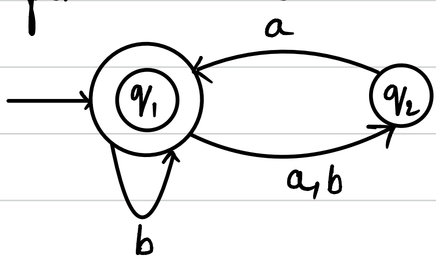
we can conclude that B' is the complement of set B.

B' is going to be a regular language since a complement of regular language is recognized by the automaton.
Hence, the class of regular languages is closed under complement.

b. Let us try with an example to see if we can prove via contradiction.

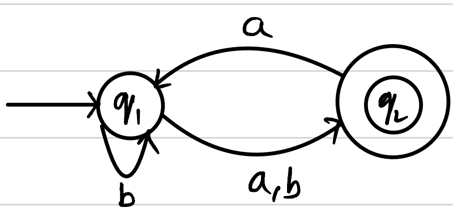
Consider a NFA that has the same stating state and final state:

N



s = b, baa, bba ...

N'



s = b, baa, ...

In case of N, N' there are a few common strings that are accepted by both.
$\therefore$ Swapping states doesnt lead to a new NFA that recognizes its complement.

However this doesn't mean that NFA is not closed under complement. Since the class of languages recognized by NFA's are the same ones that are recognized by DFA's, NFA's are closed under complement. Just that we can't generate a new NFA whose complent is recognized by swapping states.

6.

1.18 Give regular expressions generating the languages of Exercise 1.6.

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

a. $\{w \mid w$ begins with a 1 and ends with a 0$\}$
b. $\{w \mid w$ contains at least three 1s$\}$
c. $\{w \mid w$ contains the substring 0101 (i.e., $w = x0101y$ for some x and y)$\}$
d. $\{w \mid w$ has length at least 3 and its third symbol is a 0$\}$
e. $\{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$
f. $\{w \mid w$ doesn't contain the substring 110$\}$
g. $\{w \mid$ the length of w is at most 5$\}$
h. $\{w \mid w$ is any string except 11 and 111$\}$
i. $\{w \mid$ every odd position of w is a 1$\}$
j. $\{w \mid w$ contains at least two 0s and at most one 1$\}$
k. $\{\varepsilon, 0\}$
l. $\{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$
m. The empty set
n. All strings except the empty string

$M = \{Q, \Sigma, \delta, q_0, F\}$
where Σ is given to be $\{0,1\}$

a. $1\Sigma^*0$

b. $\Sigma^*1\Sigma^*1\Sigma^*1\Sigma^*$

c. $\Sigma^*0101\Sigma^*$

d. $\Sigma\Sigma0\Sigma^*$

e. $(0 \cup 1\Sigma)(\Sigma\Sigma)^*$

f. $(0 \cup (10)^*)^*1^*$

g. $(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)(\varepsilon \cup \Sigma)$

h. $(\Sigma^*0\Sigma^*) \cup (1111\Sigma^*) \cup 1 \cup \varepsilon$

i. $(1\Sigma)^*(1 \cup \varepsilon)$

j. $0^*(100 \cup 010 \cup 001)0^*$

k. $\varepsilon \cup 0$

l. $(1^*01^*01^*)^* \cup (0^*10^*10^*)$

m. $\emptyset$

n. $\Sigma\Sigma^*$

7.

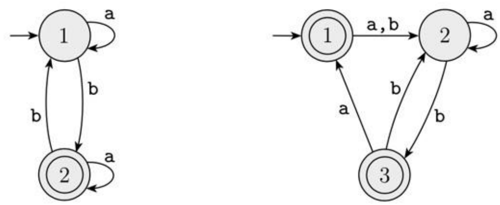
1.15 For each of the following languages, give two strings that are members and two strings that are *not* members—a total of four strings for each part. Assume the alphabet $\Sigma = \{a,b\}$ in all parts.

a. a^*b^*.
b. $a(ba)^*b$.
c. $a^* \cup b^*$.
d. $(aaa)^*$.
e. $\Sigma^*a\Sigma^*b\Sigma^*a\Sigma^*$.
f. $aba \cup bab$.
g. $(\varepsilon \cup a)b$.
h. $(a \cup ba \cup bb)\Sigma^*$.

	Recognized	Not-recognized
a.	ε	ba
	ab	aba
b.	ab	ε
	abab	ba
c.	ε	ab
	a	aab
d.	ε	a
	aaa	aa
e.	aba	ε
	aaba	aaa
f.	aba	ε
	bab	a
g.	b	ε
	ab	bb
h.	a	ε
	bb	aa

8.

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



(a) (b)

a) Some recognized strings are:

b, bbb, bbbbb .... odd number of b's

ab, abbba ... adding 'a' before and after any number of times

aaabaaabaaabaaa

$\Rightarrow$ any number of a's inbetween odd number of b's.

$\therefore$ $a^*b(a \cup ba^*b)^*$

b) Some recognized strings are:

$s_0 = \varepsilon$, $s_1 = (a \cup b)a^*b$, $s_2 = s_1 a$, $s_3 = s_1 b a^* b$, $s_4 = s_1 b a^* b a$

$s_5 = s_2 a^* b$

$\therefore$ $\varepsilon \cup ((a \cup b)a^*b((b \cup a(a \cup b))a^* b)^* (\varepsilon \cup a))$

9.

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

[A]**a.** $A_1 = \{0^n 1^n 2^n \mid n \geq 0\}$

b. $A_2 = \{www \mid w \in \{\mathtt{a}, \mathtt{b}\}^*\}$

[A]**c.** $A_3 = \{\mathtt{a}^{2^n} \mid n \geq 0\}$ (Here, $\mathtt{a}^{2^n}$ means a string of 2^n a's.)

a. Proof by contradiction : Assume A_1 is regular

let $s = 0^p 1^p 2^p$ where $p = 0, 1, 2, 3, \ldots$

From pumping lemma we can say that s can be divided into 3 pieces, $s = xyz$ such that for any $i \geq 0$, $xy^iz \in A_1$

Consider a case where $y = 0^p$ or 1^p or 2^p only. then xy^2z will not have equal number of 0's, 1's and 2's which leads to a contradiction

b. Proof by contradiction : Assume A_2 is regular

let $s = a^p b a^p b$ where $p = 0, 1, 2, 3, \ldots$

From pumping lemma we can say that s can be divided into 3 pieces, $s = xyz$ such that for any $i \geq 0$, $xy^iz \in A_2$

If we take $y = a^p b$ then the condition $|xy| \leq p$ will fail hence it leads to a contradition.

c. Proof by contradiction : Assume A_3 is regular

let $s = a^{2P}$ where $P = 0, 1, 2, 3, \ldots$

From pumping lemma we can say that s can be divided into 3 pieces, $s = xyz$ such that for any $i \geq 0$, $xy^iz \in A_3$

Consider two consecutive strings, ie, a^{2P} and $a^{2(P+1)} = a^{2 \cdot 2P}$ which is twice the length of the previous string.

$\Rightarrow y = a^{2P}$

but then it violates $|xy| \leq P$, leading to a contradiction.

10. **1.30** Describe the error in the following "proof" that 0^*1^* is not a regular language. (An error must exist because 0^*1^* *is* regular.) The proof is by contradiction. Assume that 0^*1^* is regular. Let p be the pumping length for 0^*1^* given by the pumping lemma. Choose s to be the string 0^p1^p. You know that s is a member of 0^*1^*, but Example 1.73 shows that s cannot be pumped. Thus you have a contradiction. So 0^*1^* is not regular.

Let $s = 0^p 1^p$ where $s = xyz$ and $x = 0$, $y = 0$, $z = 0^{p-2}1^p$

$\therefore \quad xy^iz = 0\,0^i\,0^{p-2}1^p$, $|y| = 1$, $|xy| = 2$

We must check against the 3 conditions:

i) For any $i \geq 0$, $xy^iz \in 0^*1^*$

ii) $|y| > 0$

iii) $|xy| \leq p$

Since all conditions are satisfied, s can be pumped, hence 0^*1^* is regular.

11.

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

a. $\{w \mid w$ begins with a 1 and ends with a 0$\}$
b. $\{w \mid w$ contains at least three 1s$\}$
c. $\{w \mid w$ contains the substring 0101 (i.e., $w = x0101y$ for some x and y)$\}$
d. $\{w \mid w$ has length at least 3 and its third symbol is a 0$\}$
e. $\{w \mid w$ starts with 0 and has odd length, or starts with 1 and has even length$\}$
f. $\{w \mid w$ doesn't contain the substring 110$\}$
g. $\{w \mid$ the length of w is at most 5$\}$
h. $\{w \mid w$ is any string except 11 and 111$\}$
i. $\{w \mid$ every odd position of w is a 1$\}$
j. $\{w \mid w$ contains at least two 0s and at most one 1$\}$
k. $\{\varepsilon, 0\}$
l. $\{w \mid w$ contains an even number of 0s, or contains exactly two 1s$\}$
m. The empty set
n. All strings except the empty string

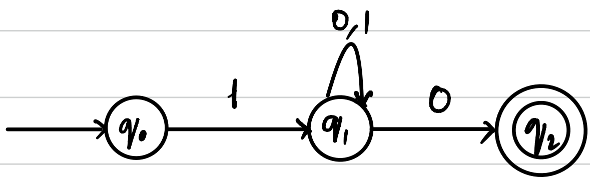
1.8 Use the construction in the proof of Theorem 1.45 to give the state diagrams of NFAs recognizing the union of the languages described in

a. Exercises 1.6a and 1.6b.
b. Exercises 1.6c and 1.6f.

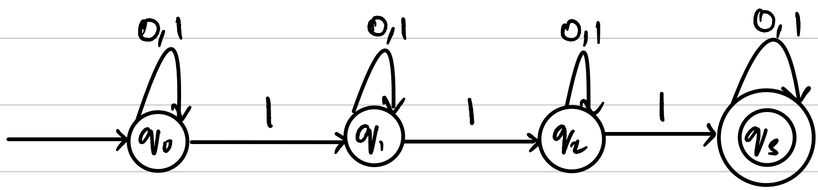
a. $L_1 = 1\Sigma^*0$

$L_2 = \Sigma^*1\Sigma^*1\Sigma^*1\Sigma^*$

L_1

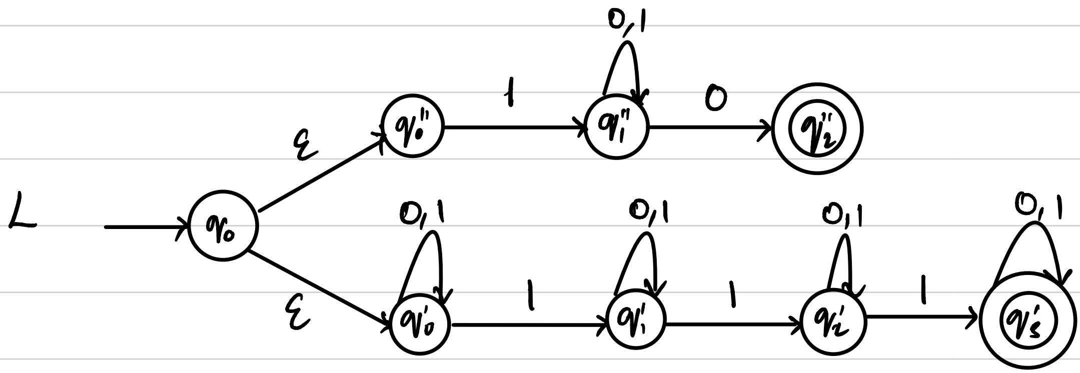


L_2

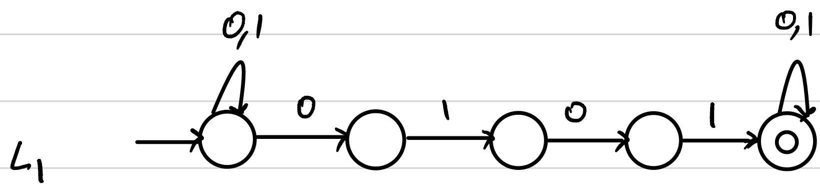


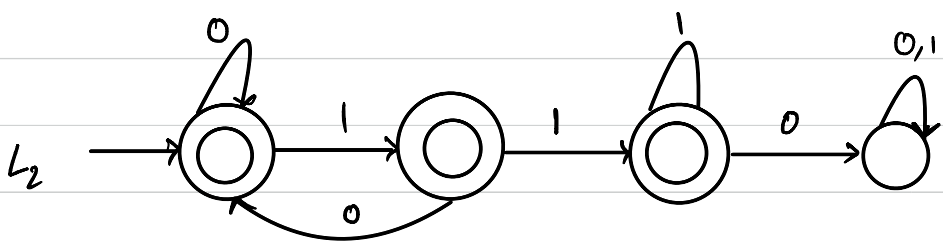
We need to find their union,

let $L = L_1 \cup L_2$

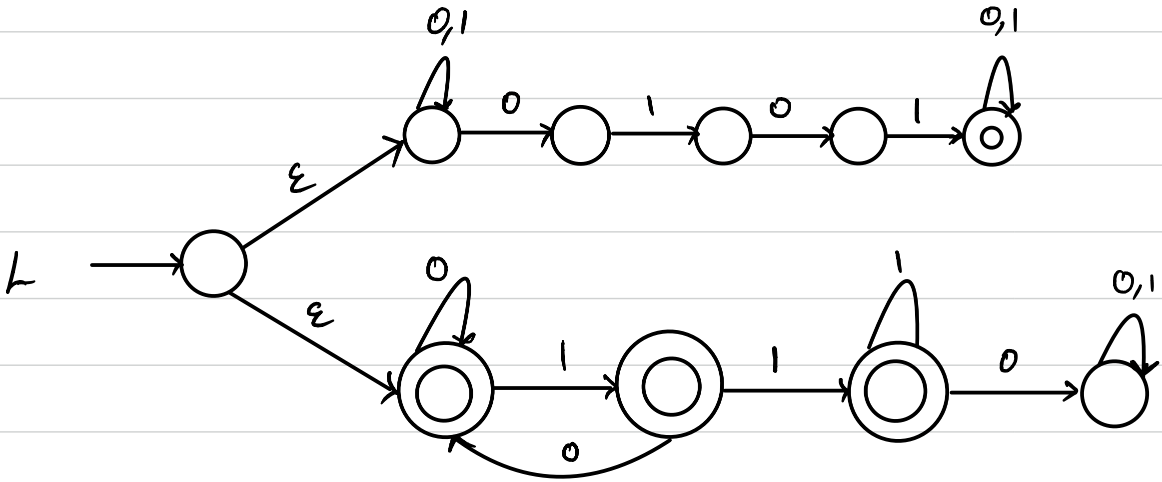


b) $L_1 = \Sigma^* 0101 \Sigma^*$





We need to find $L = L_1 \cup L_2$



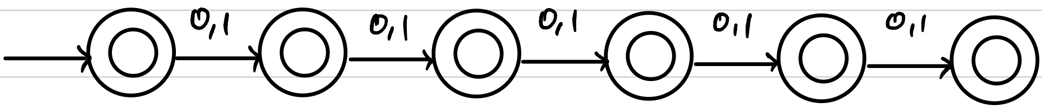
12.

1.9 Use the construction in the proof of Theorem 1.47 to give the state diagrams of NFAs recognizing the concatenation of the languages described in

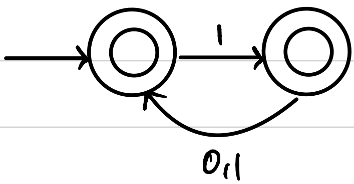
a. Exercises 1.6g and 1.6i.
b. Exercises 1.6b and 1.6m.

a.

L_1

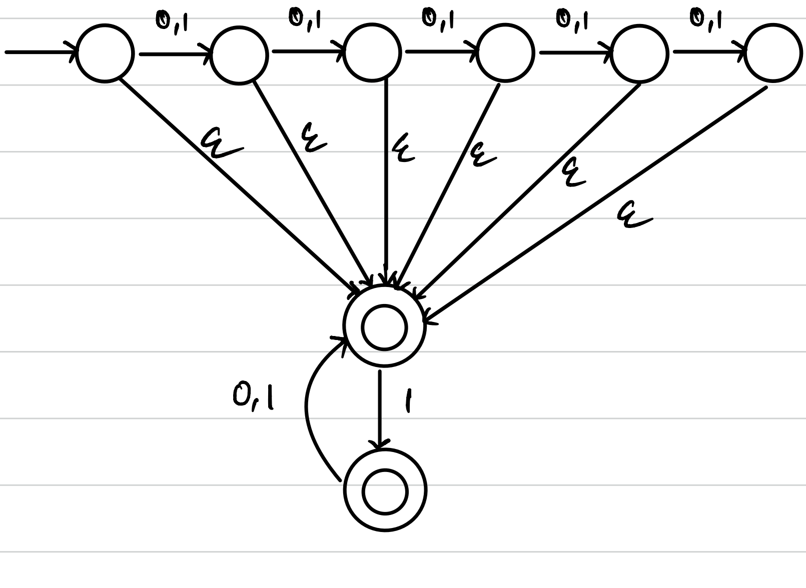


L_2



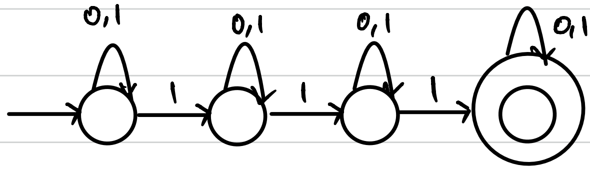
$(1\Sigma)^*$

$L_1 \circ L_2$

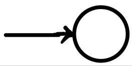


b.

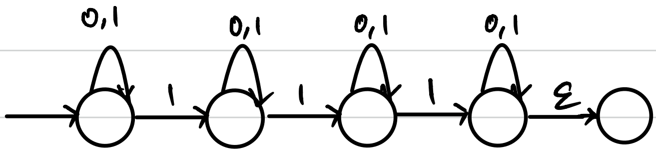
L_1



L_2



$L_1 \circ L_2$



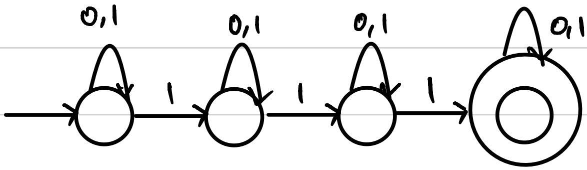
13.

1.10 Use the construction in the proof of Theorem 1.49 to give the state diagrams of NFAs recognizing the star of the languages described in

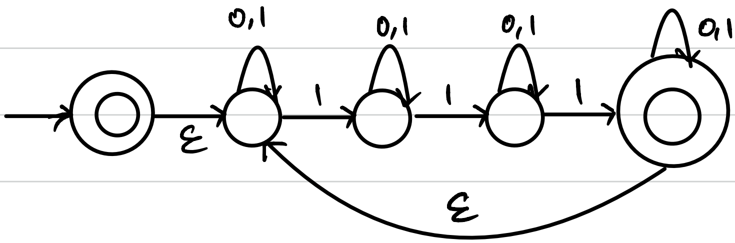
a. Exercise 1.6b.
b. Exercise 1.6j.
c. Exercise 1.6m.

a. $L_1 = \Sigma^* 1 \Sigma^* 1 \Sigma^* 1 \Sigma^*$

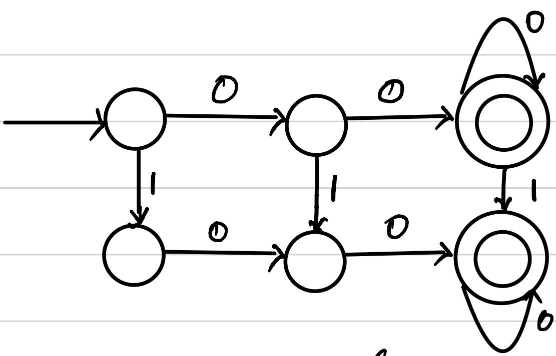
L_1



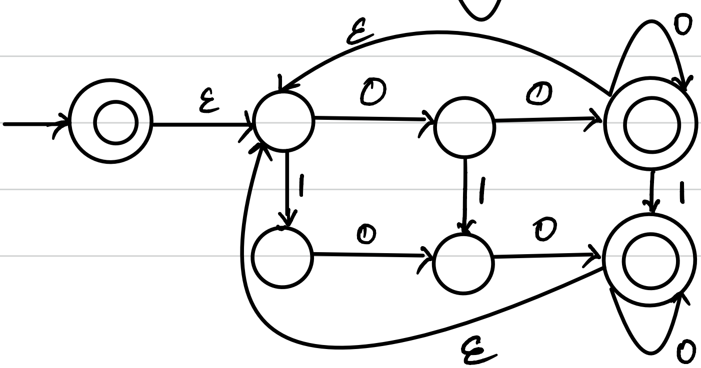
L_1^*



b. L_1



L_1^*

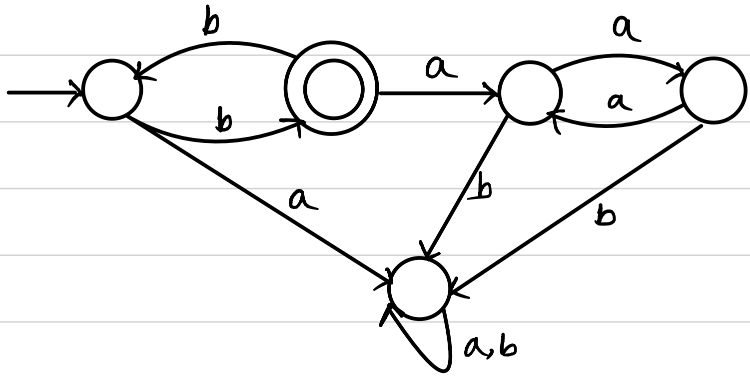


14\.

1.12 Let $D = \{w \mid w$ 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 states that recognizes D and a regular expression that generates D. (Suggestion: Describe D more simply.)

The language of D can be said to contain odd number of b's followed by even number of a's

DFA with 5 states that recognizes strings like b, baa, baaaa ... $b(aa)^*$



Constructing a regular exprⁿ the generates D:

let $D_1 = \{w \mid w$ contains odd number of b's$\}$

$D_2 = \{w \mid w$ contains even number of a's$\}$

R_1 (generates D_1) $= b(bb)^*$

R_2 (generates D_2) $= (aa)^*$

$\therefore$ $R = R_1 \circ R_2$

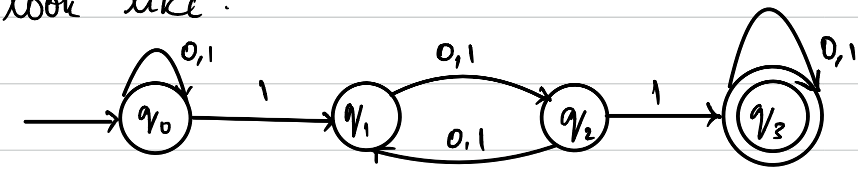
$R = b(bb)^*(aa)^*$

15.

1.13 Let F be the language of all strings over $\{0,1\}$ that do not contain a pair of 1s that are separated by an odd number of symbols. Give the state diagram of a DFA with five states that recognizes F. (You may find it helpful first to find a 4-state NFA for the complement of F.)

The NFA for the complement of the language will look like:

$\overline{F}$



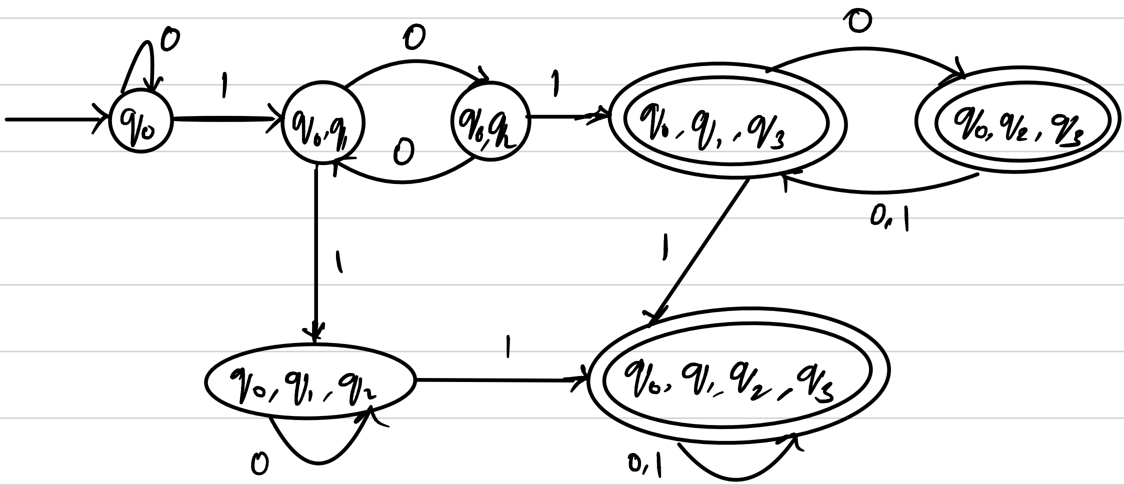
$\overline{F}$ accepts strings that contains a pair of 1's that are separated by odd number of symbols.

Transition table for $\overline{F}$:

	0	1
q_0	q_0	$\{q_0, q_1\}$
q_1	q_2	q_2
q_2	q_1	$\{q_1, q_3\}$
q_3	q_3	q_3

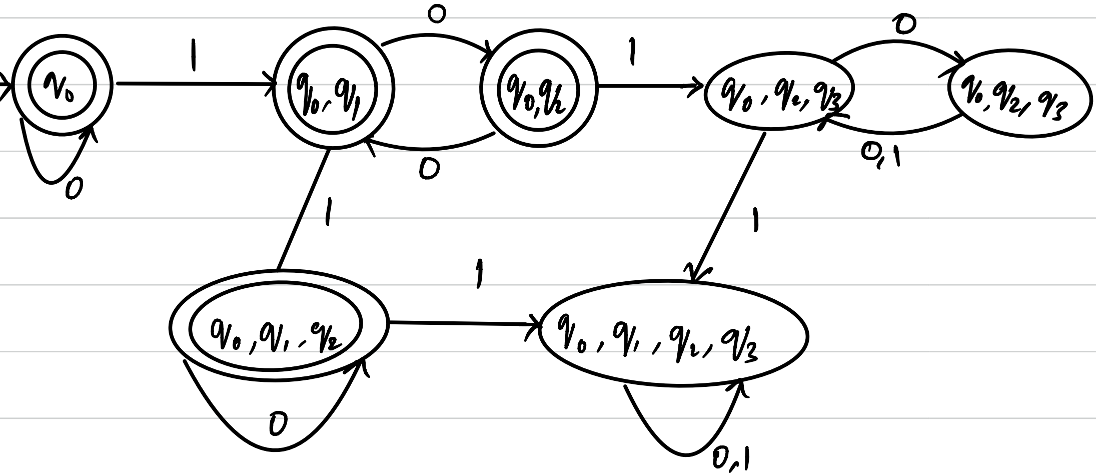
The transition function for the converted DFA $\bar{D}$ will be:

	0	1
q_0	q_0	$\{q_0, q_1\}$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_1, q_3\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_1, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_3\}$	$\{q_0, q_1, q_3\}$

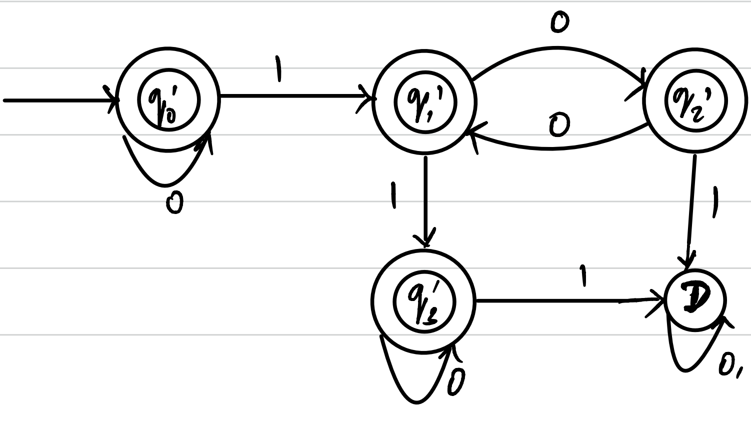


If we take the compliment of $\overline{D}$:

D:

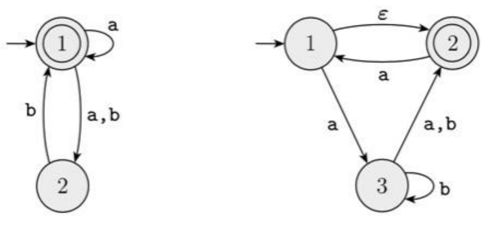


We need to find a 5-state DFA which can be obtained from D by combining the dead states - $\{q_0, q_1, q_3\}$, $\{q_0, q_2, q_3\}$, $\{q_0, q_1, q_2, q_3\}$



16.

1.16 Use the construction given in Theorem 1.39 to convert the following two nondeterministic finite automata to equivalent deterministic finite automata.



(a) (b)

a. Step by step process:

1. $Q' = P(Q) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$

2. $\delta'(\emptyset, a) = \delta(\emptyset, a) = \emptyset$

$\delta'(\emptyset, b) = \delta(\emptyset, b) = \emptyset$

$\delta'(\{1\}, a) = \delta(1, a) = \{1, 2\}$

$\delta'(\{1\}, b) = \delta(1, b) = \{2\}$

$\delta'(\{2\}, a) = \delta(2, a) = \emptyset$

$\delta'(\{2\}, b) = \delta(2, b) = \{1\}$

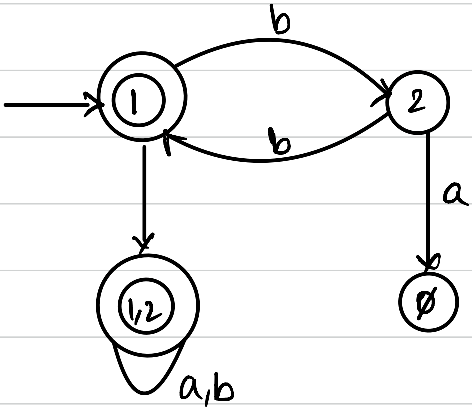
$\delta'(\{1,2\}, a) = \delta(1, a) \cup \delta(2, a) = \{1, 2\}$

$\delta'(\{1,2\}, b) = \delta(1, b) \cup \delta(2, b) = \{1, 2\}$

3. $q_0' = \{q_0\} = 1$ is the start state of the DFA

4. $F' = \{R \in Q' \mid R$ contains an accept state of $N\}$

DFA



b. Step by step process:

1. $Q' = P(Q) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

2. $\delta'(\emptyset, a) = \emptyset$

$\delta'(\emptyset, b) = \emptyset$

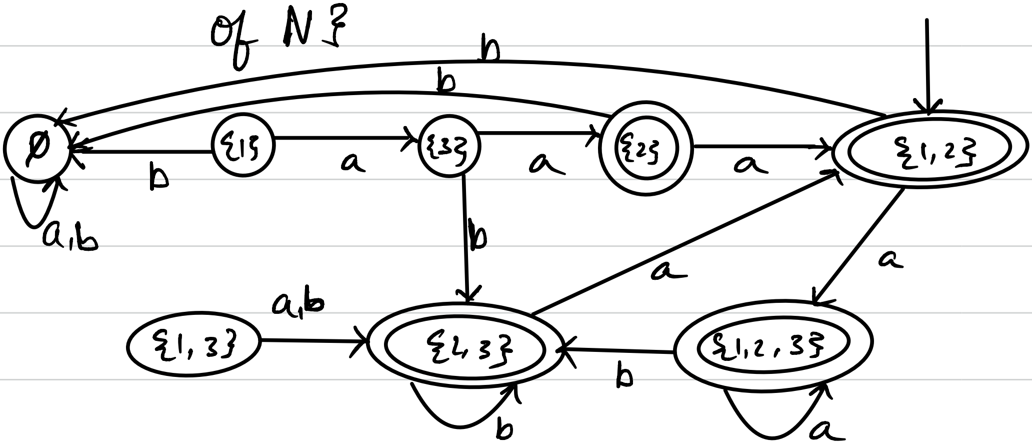
$\delta'(\{1\}, a) = \delta(1, a) = \{3\}$

$\delta'(\{1\}, b) = \delta(1, b) = \emptyset$

$$\delta'(\{2\}, a) = \delta(2, a) = \{1,2\}$$
$$\delta'(\{2\}, b) = \delta(2, b) = \emptyset$$
$$\delta'(\{3\}, a) = \delta(3, a) = \{2\}$$
$$\delta'(\{3\}, b) = \delta(3, b) = \{2,3\}$$
$$\delta'(\{1,2\}, a) = \delta(1,a) \cup \delta(2,a) = \{1,2,3\}$$
$$\delta'(\{1,2\}, b) = \delta(1,b) \cup \delta(2,b) = \emptyset$$
$$\delta'(\{1,3\}, a) = \delta(1,a) \cup \delta(3,a) = \{2,3\}$$
$$\delta'(\{1,3\}, b) = \delta(1,b) \cup \delta(3,b) = \{2,3\}$$
$$\delta'(\{2,3\}, a) = \delta(2,a) \cup \delta(3,a) = \{1,2\}$$
$$\delta'(\{2,3\}, b) = \delta(2,b) \cup \delta(3,b) = \{2,3\}$$
$$\delta'(\{1,2,3\}, a) = \delta(1,a) \cup \delta(2,a) \cup \delta(3,a) = \{1,2,3\}$$
$$\delta'(\{1,2,3\}, b) = \delta(1,b) \cup \delta(2,b) \cup \delta(3,b) = \{2,3\}$$

3. $q_0' = \{1,2\}$

4. $F' = \{R \in Q' \mid R$ contains an accept state of $N\}$



We can remove the nodes : {1}, {2}, {3}, {1,3} since they dont have any incoming arrows.