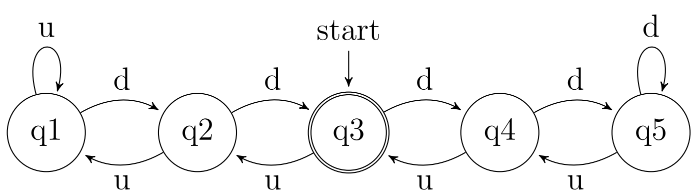
Jason Downing
Email: jason_downing@student.uml.edu
Foundations of Computer Science
Homework #2 - Chapter 1: DFA, NFA
10/6/2016

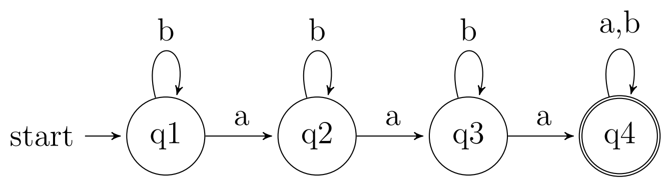
1.3
The formal description of a DFA M is {q1, q2, q3, q4, q5 },
{u, d}, δ, q3 , {q3}, where δ is given by the following table:

	u	d
q1	q1	q2
q2	q1	q3
q3	q2	q4
q4	q3	q5
q5	q4	q5

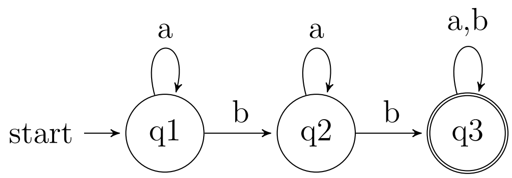
Initial state is q3, so that is where the machine will start.
We can use the table to create the nodes, and connect them as needed.
The accepted state is q3 so we will mark that with a double circle to show that as the accepted state of the machine.



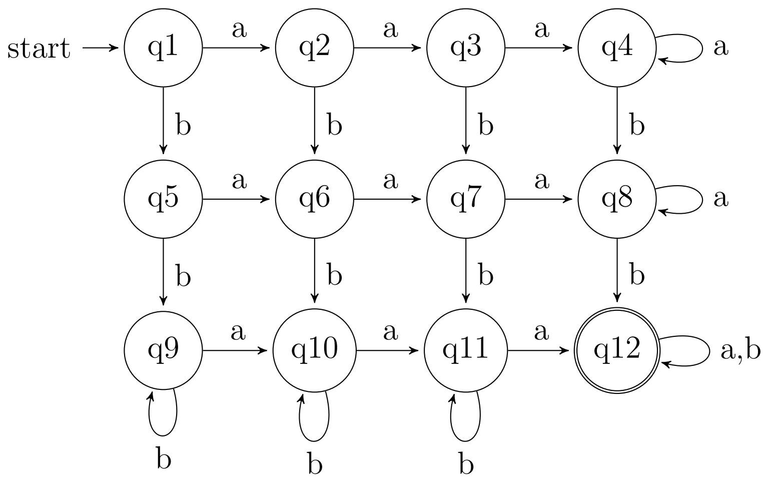
1.4: a, c, e, f, g. For all parts, $\sum = \{a, b\}$
a. { w| w has at least three a's }



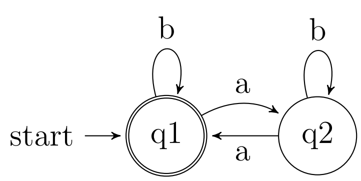
{ w| w has at least two b's }



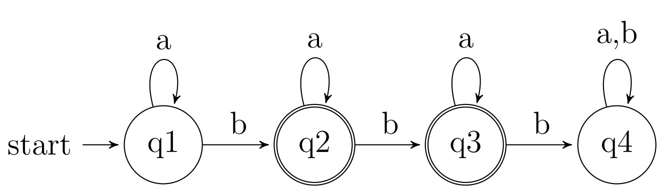
{w| w has at least three as and at least two bs}



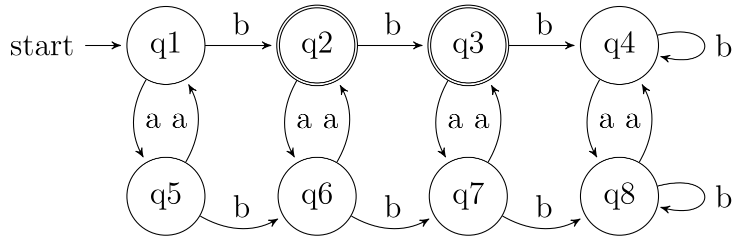
c. { w| w has an even number of a's }



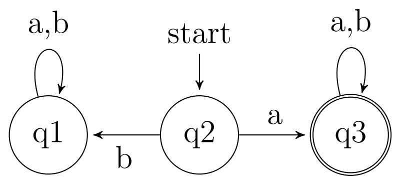
{ w| w has one or two b's }



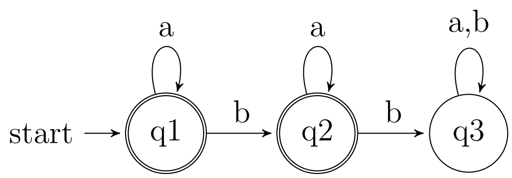
{ w| w has an even number of a's and one or two b's }



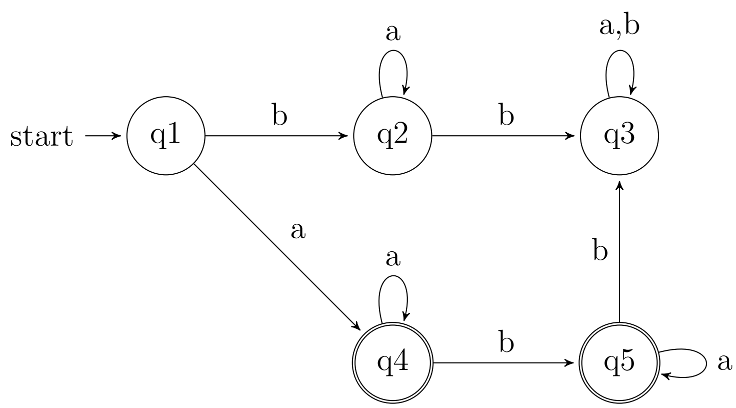
e. { w| w starts with an a }



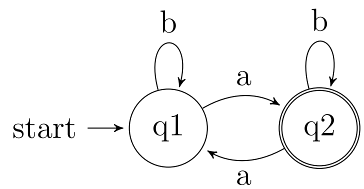
{ w| w has at most one b }



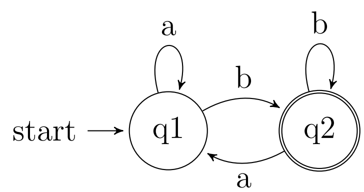
{ w| w starts with an a and has at most one b }



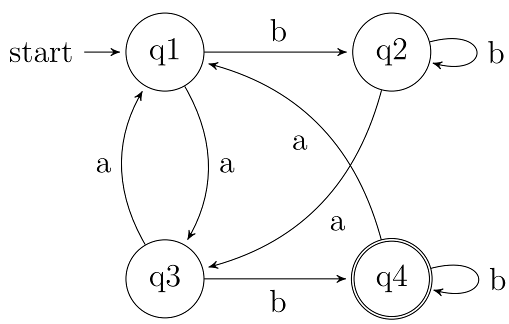
f. { w| w has an odd number of a's }



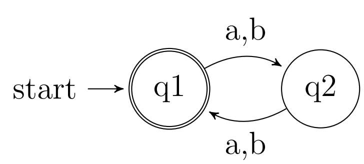
{ w| w ends with a b }



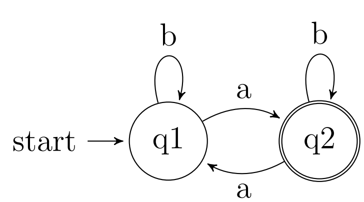
{ w| w has an odd number of a's and ends with a b }



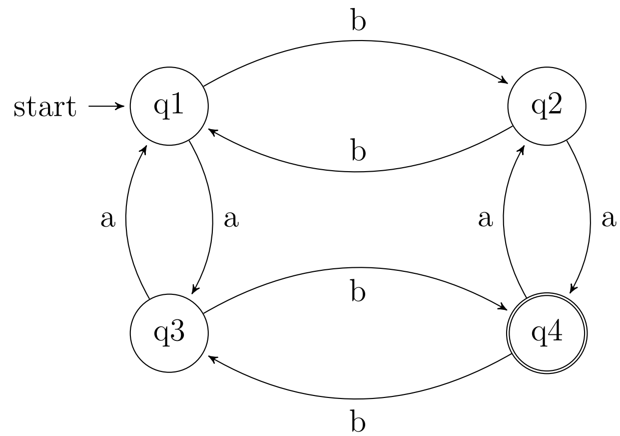
g. { w| w has even length }



{ w| w has an odd number of a's }



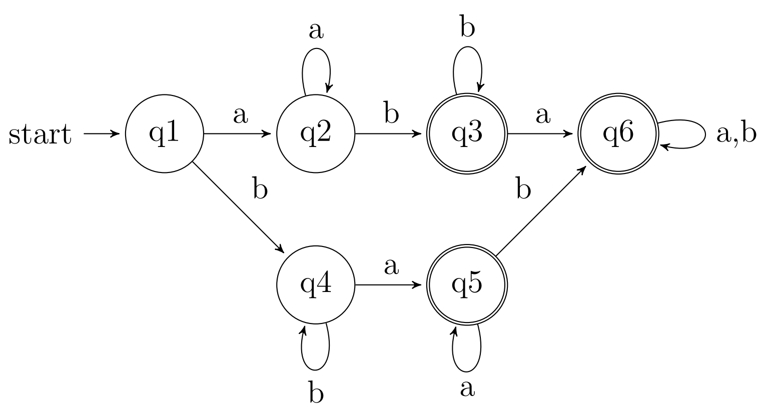
{ w| w has even length and an odd number of a's }



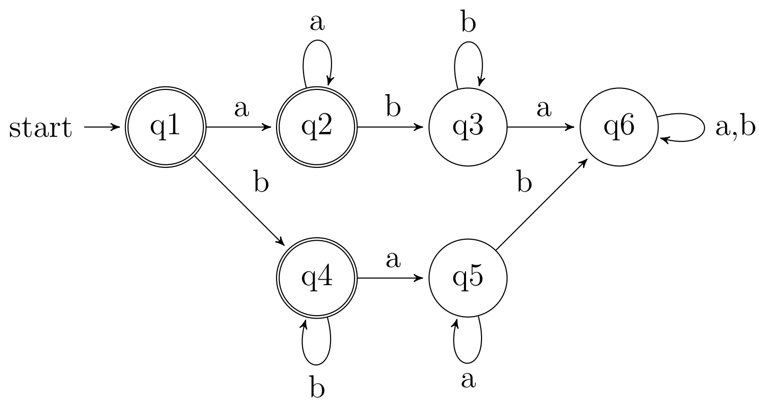
1.5: c, d, e, f, g, h. For all parts, $\sum = \{a, b\}$

c. { w| w contains neither the substrings ab nor ba }

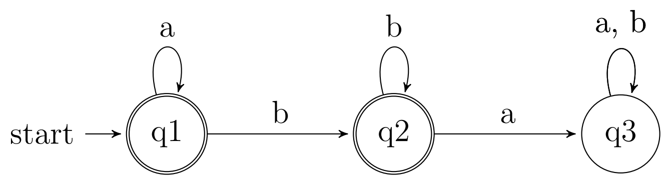
Simpler language



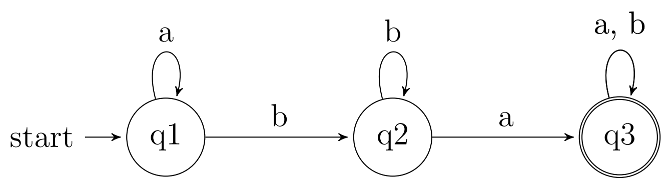
Language given.



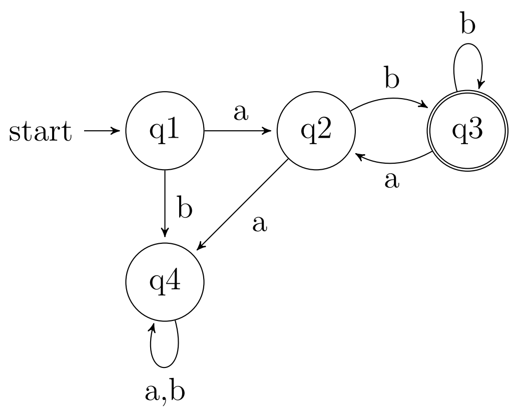
d. { w| w is any string not in a*b* }
Simpler language



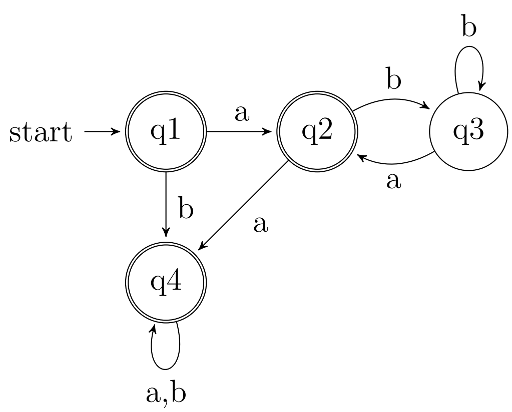
Language given



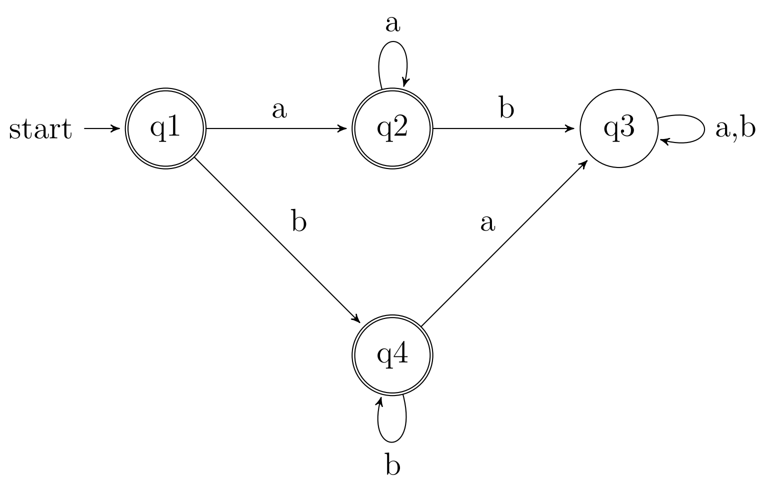
e. { w| w is any string not in $(ab^{+})^{*}$ }
Simpler language



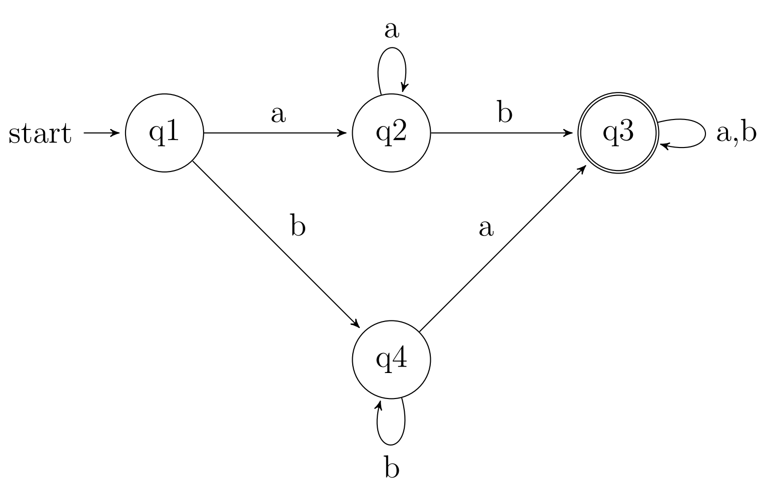
Language given



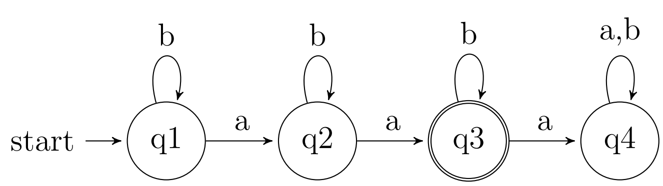
f. { w | w is any string not in a* $\bigcup$ b* }
Simpler language



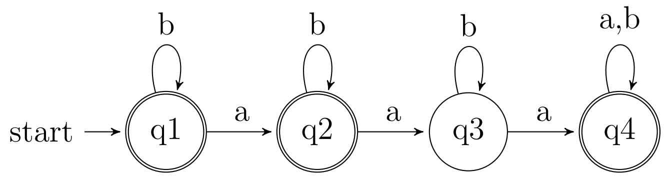
Language given



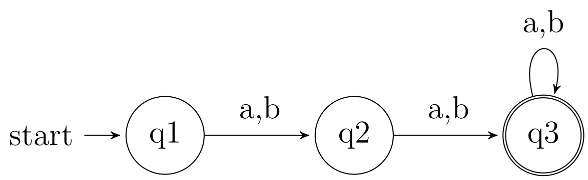
g. { w| w is any string that doesn't contain exactly two a's }
Simpler language



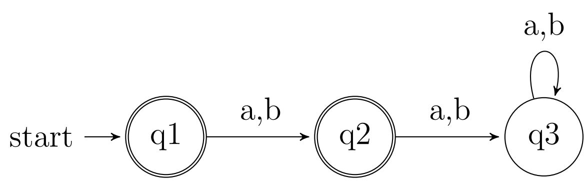
Language given



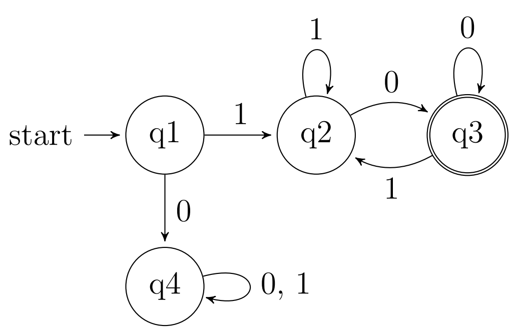
h. { w| w is any string except a and b }
Simpler language



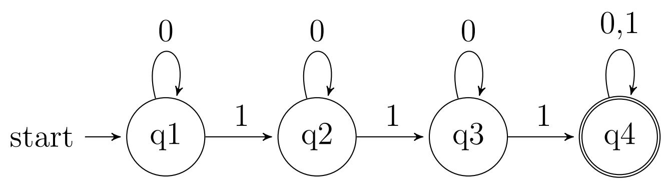
Language given



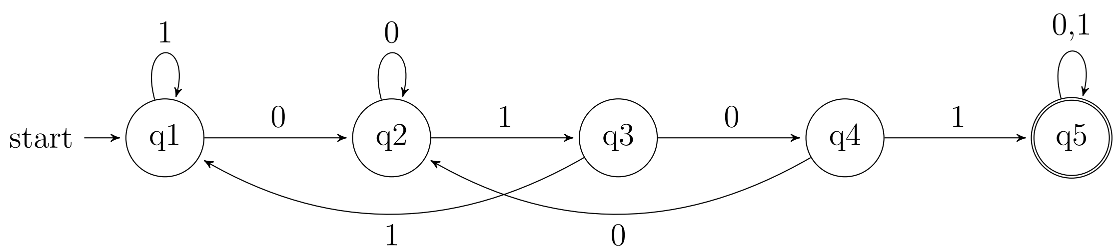
1.6: a, b, c, d, e, f, g, h, I, j, k, l, m, n
a. {w| w begins with a 1 and ends with a 0}



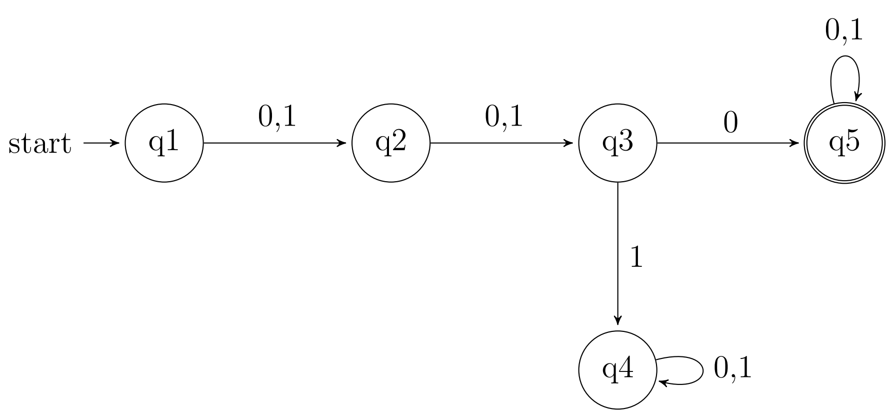
b. {w| w contains at least three 1s}



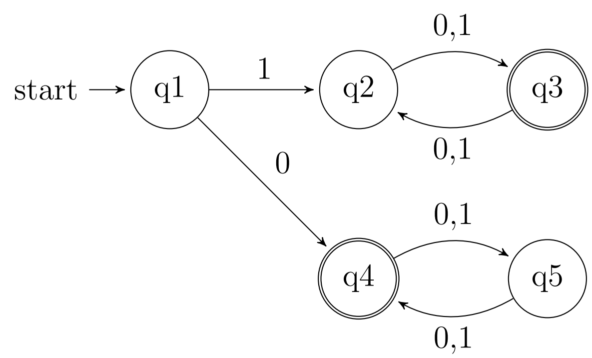
c. {w| w contains the substring 0101 (i.e., w = x0101y for some x and y)}



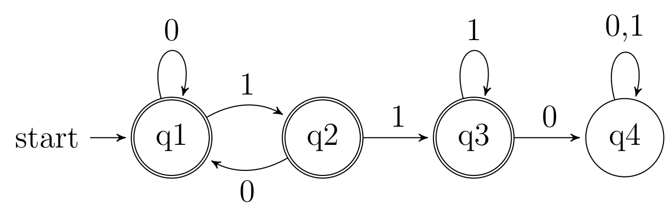
d. {w| w has length at least 3 and its third symbol is a 0}



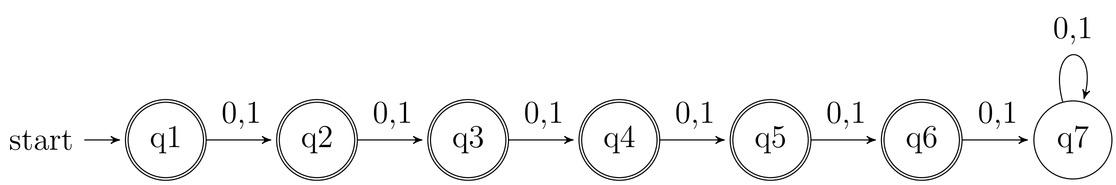
e. {w| w starts with 0 and has odd length, or starts with 1 and has even length}



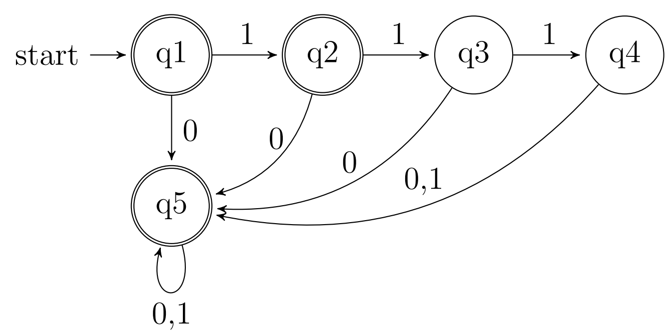
f. {w| w doesnt contain the substring 110}



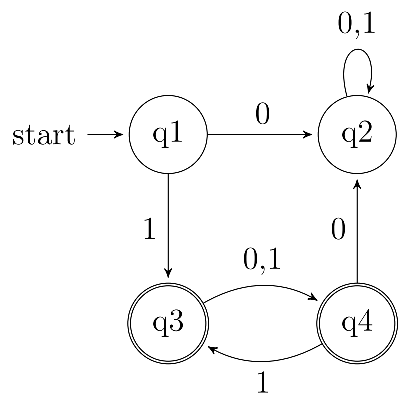
g. {w| the length of w is at most 5}



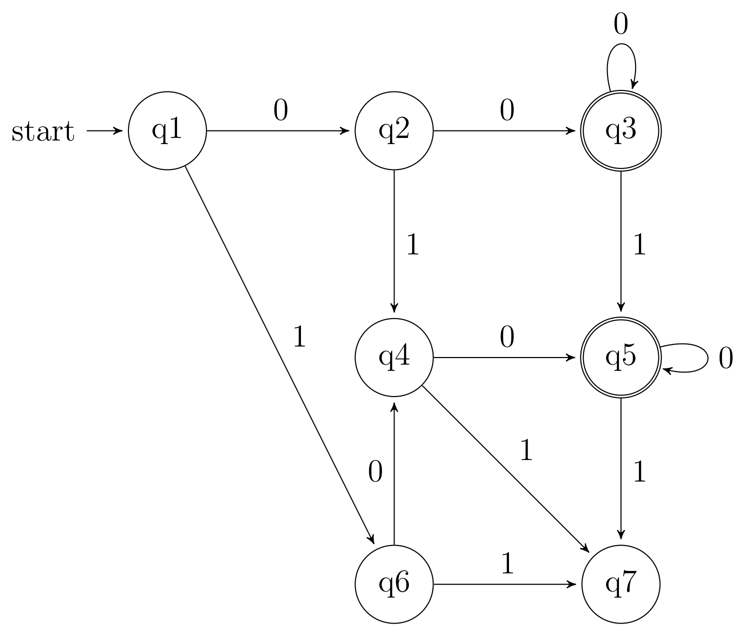
h. {w| w is any string except 11 and 111}



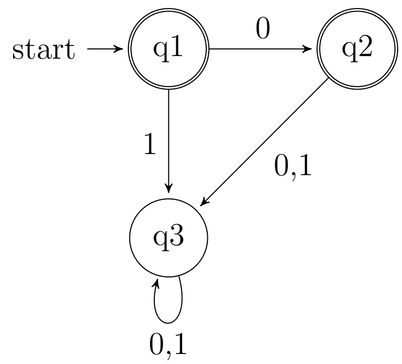
I. {w| every odd position of w is a 1}



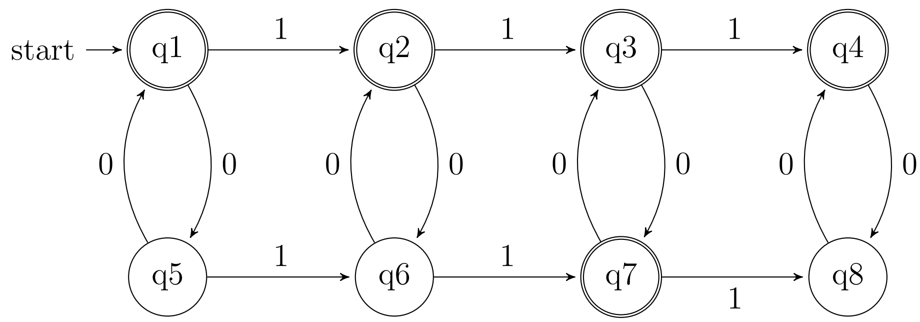
j. {w| w contains at least two 0s and at most one 1}



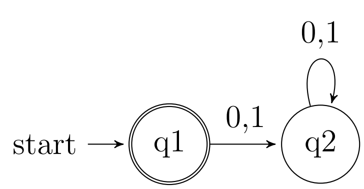
k. {e, 0}



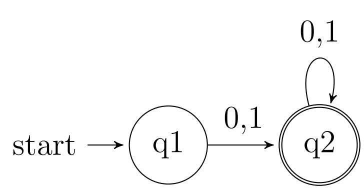
L. {w| w contains an even number of 0s, or contains exactly two 1s}



m. The empty set



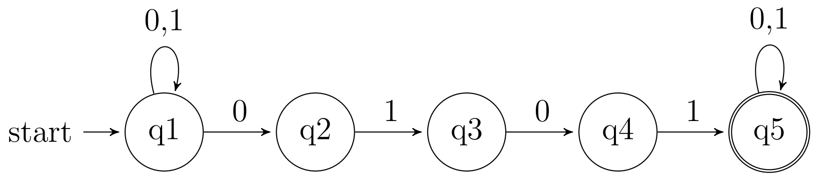
n. All strings except the empty string



1.7: b, c, d, e, g, h

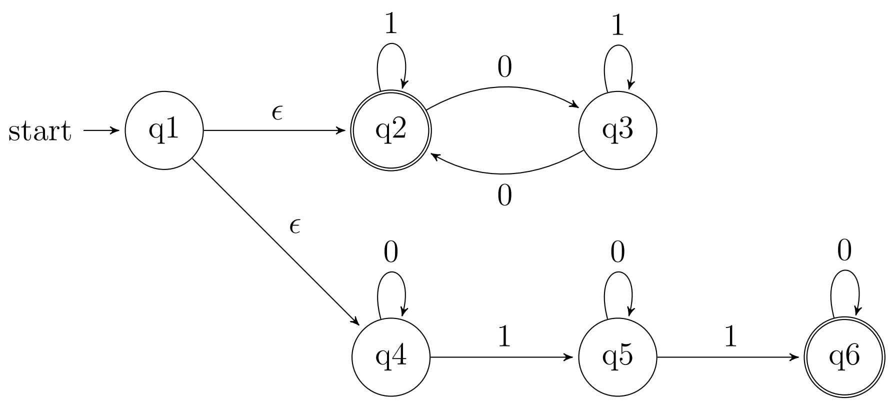
b. The language of Exercise 1.6l with six states

{w| w contains the substring 0101 (i.e., w = x0101y for some x and y)}

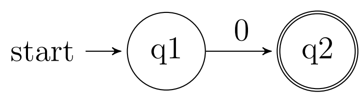


c. The language of Exercise 1.6l with six states

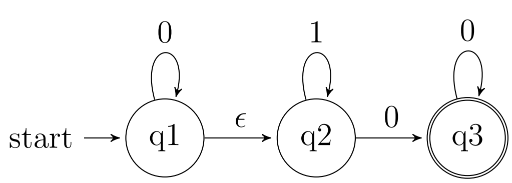
{w| w contains an even number of 0s, or contains exactly two 1s}



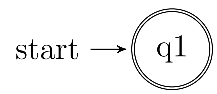
d. The language 0 with two states



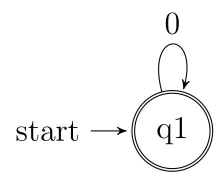
e. The language $0^*1^*0^+$ with three states



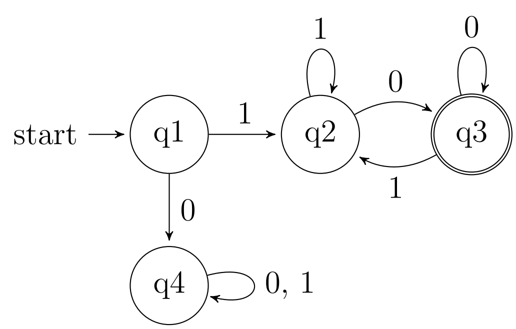
g. The language $\{\epsilon\}$ with one state



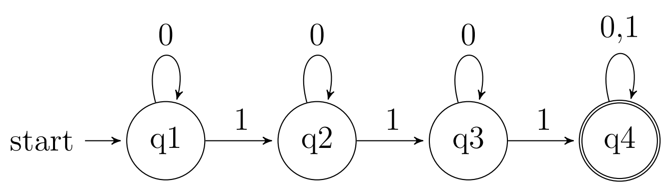
h. The language 0* with one state



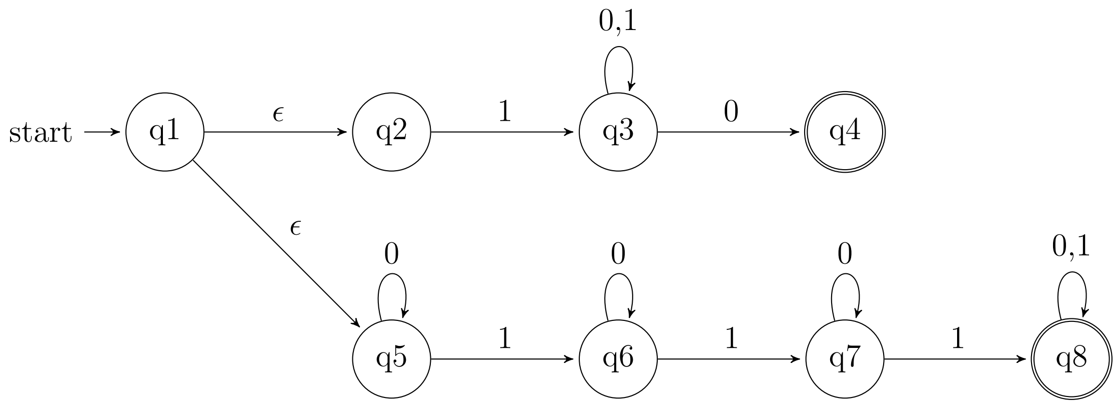
1.8: a, b
Use the construction in the proof of Theorem 1.45 to give the state diagrams of NFAs recognizing the union of the languages described in 1.8 a. Exercise 1.6a {w| w begins with a 1 and ends with a 0}



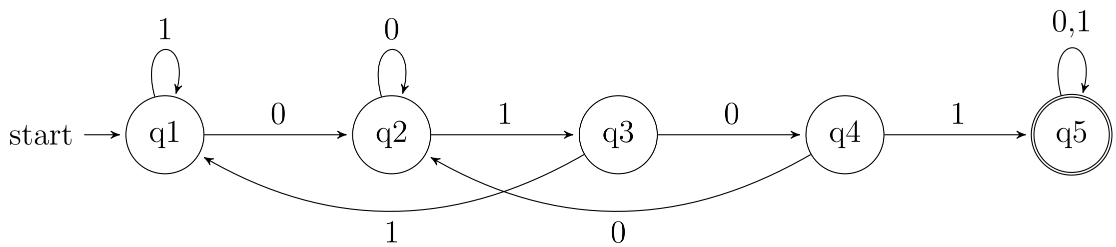
Exercise 1.6b
{w| w contains at least three 1s}



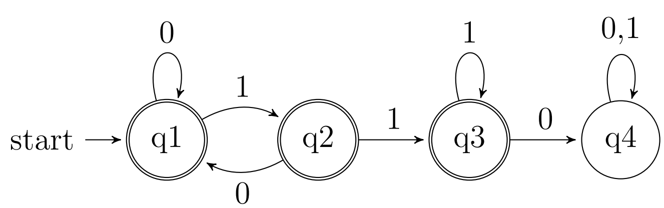
Exercise 1.6a and Exercise 1.b joined



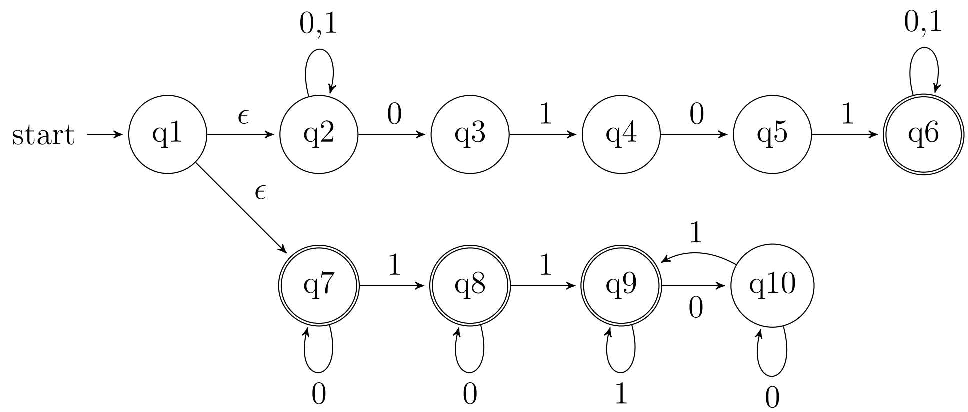
1.8 b. Exercises 1.6c and 1.6f
Exercises 1.6c



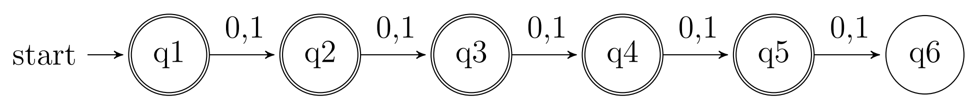
Exercises 1.6f



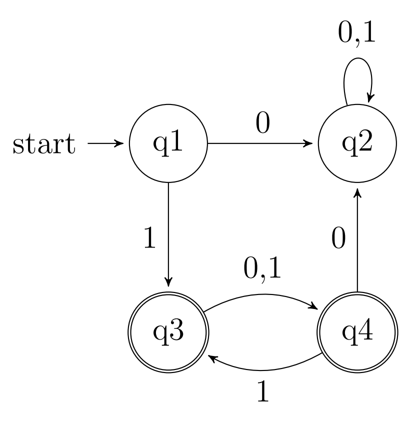
Exercises 1.6c and Exercises 1.6f joined



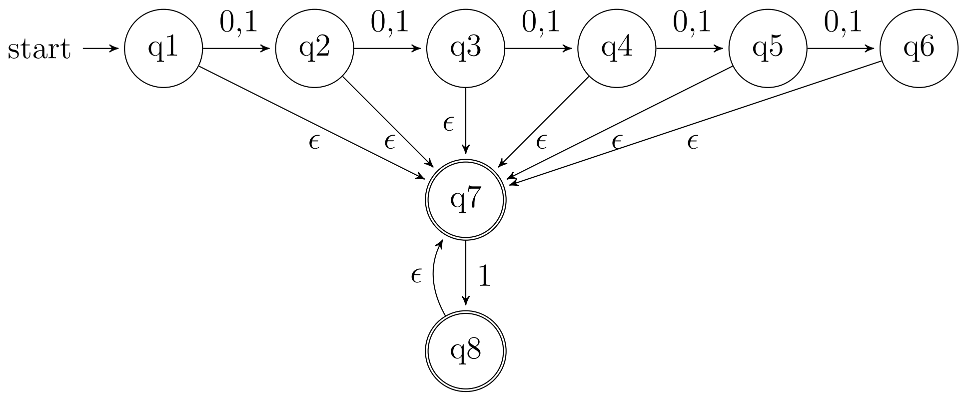
1.9: a, b
Use the construction in the proof of Theorem 1.47 to give the state diagrams of NFAs recognizing the concatenation of the languages described in
a. Exercises 1.6g and 1.6i
Exercise 1.6g



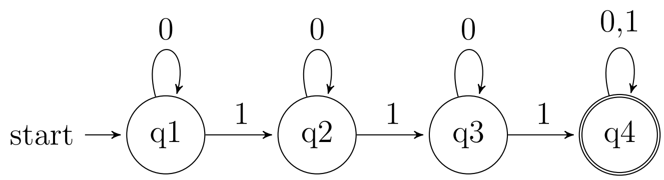
Exercise 1.6i
{w| every odd position of w is a 1}



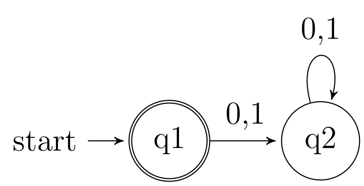
Exercise 1.6g and Exercise 1.6i joined



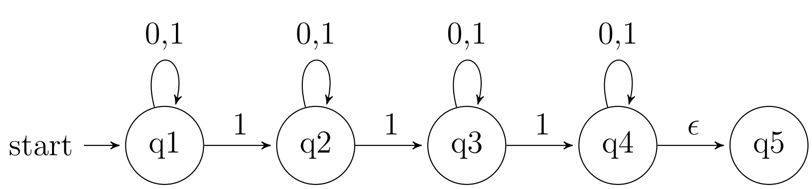
b. Exercises 1.6b and 1.6m
Exercise 1.6b
{w| w contains at least three 1s}



Exercise 1.6m - The empty set

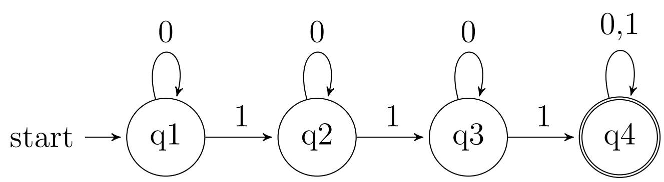


Exercises 1.6b and 1.6m joined

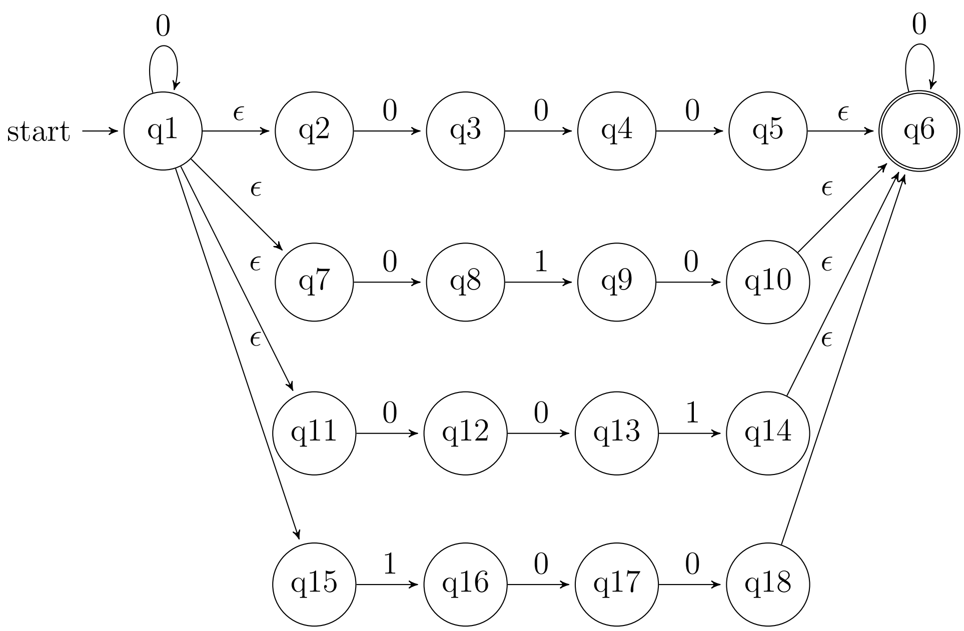


1.10: a, b, c
Use the construction in the proof of Theorem 1.49 to give the state diagrams of NFAs recognizing the star of the languages described in

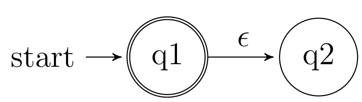
a. Exercise 1.6b
{w| w contains at least three 1s}



b. Exercise 1.6j
{w| w contains at least two 0s and at most one 1}

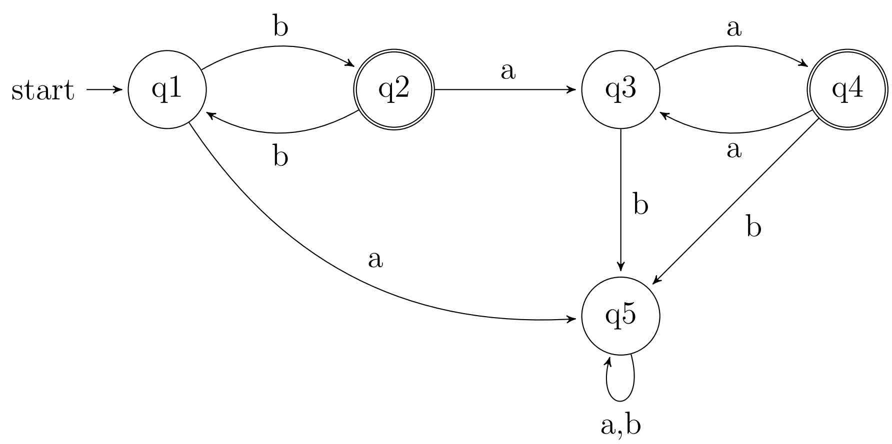


c. Exercise 1.6m



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 states that recognizes D and a regular expression that generates D.
(Suggestion: Describe D more simply.)



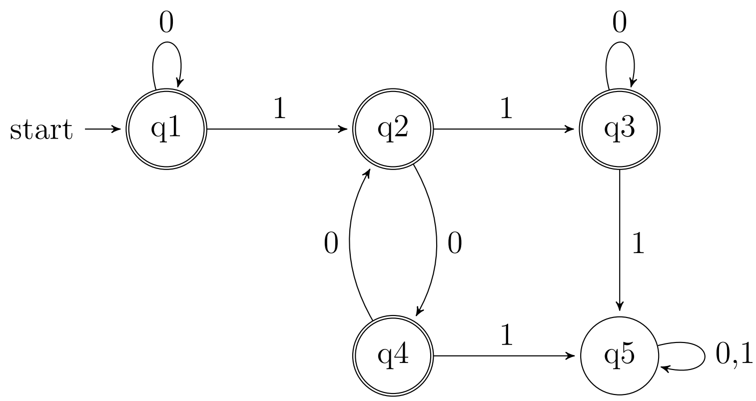
R1 = {w| w contains an even number of a's}
R1 = (aa)*

R2 = {w| w contains an odd number of b's and does not contain the substring ab }
R2 = b(bb)*

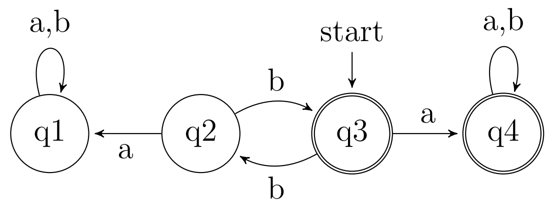
R = b(bb)*(aa)*

1.13

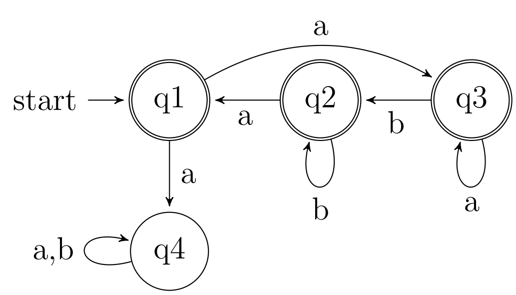
Let F be the language of all strings over 0,1 that do not contain a pair of 1s that are separated by an odd number of symbols. Give the state diagram of a DFA with five states that recognizes F.



1.16 a.

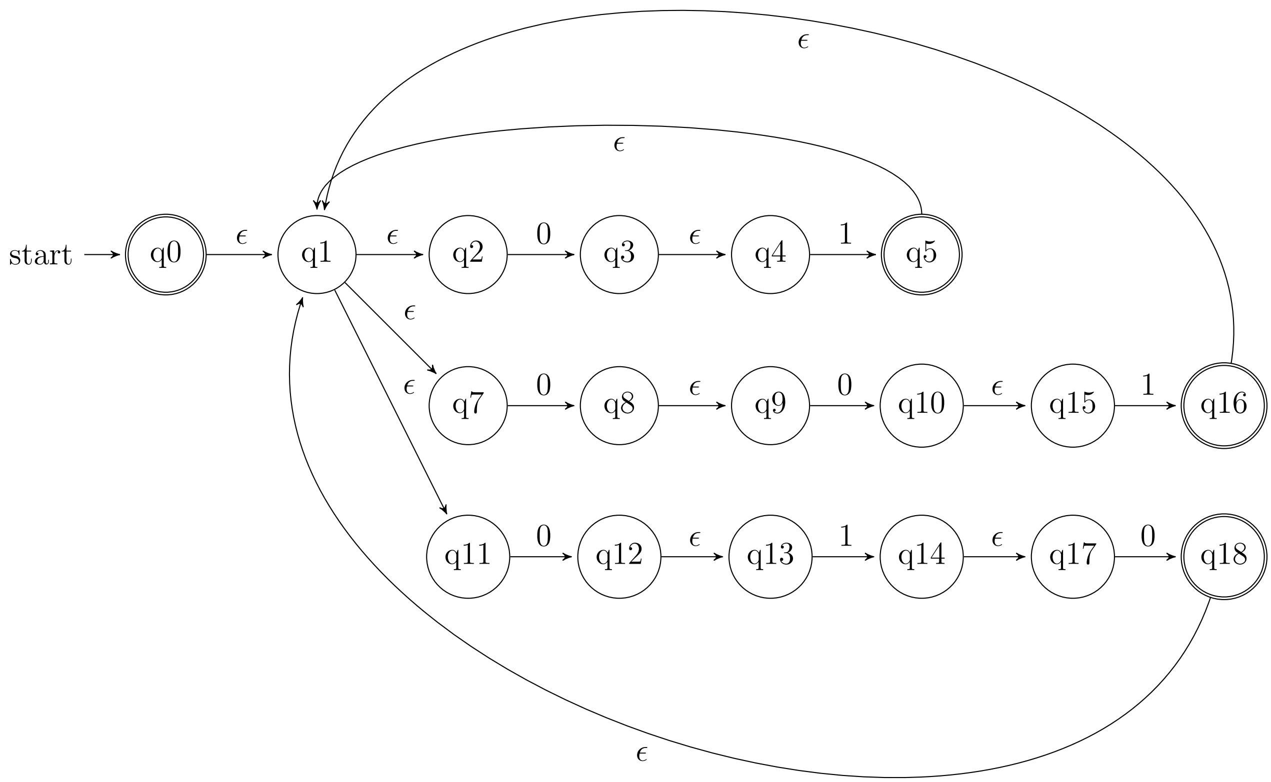


b.

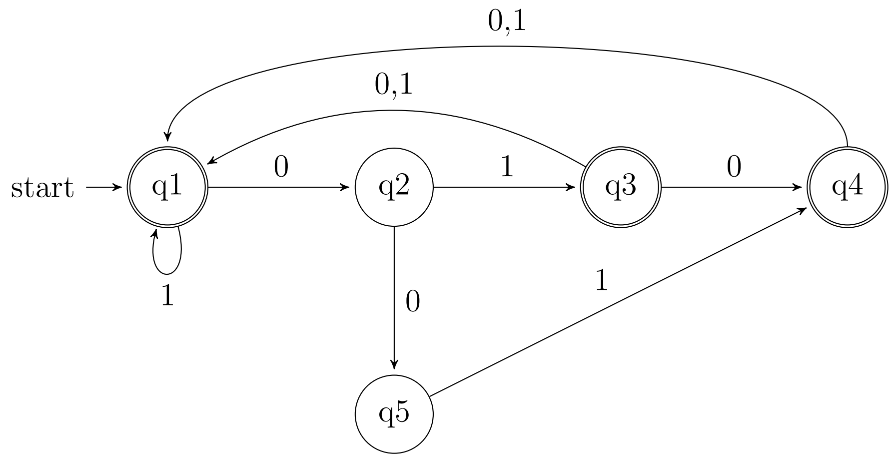


1.17: a, b

a. Give an NFA recognizing the language $(01 \bigcup 001 \bigcup 010)^*$.



b. Convert this NFA to an equivalent DFA. Give only the portion of the DFA that is reachable from the start state.



1.18 a. {w| w begins with a 1 and ends with a 0}

$$1\sum^{*}0$$

b. {w| w contains at least three 1s}

$$\sum^{*}1\sum^{*}1\sum^{*}1\sum^{*}$$

c. {w| w contains the substring 0101 (i.e., w = x0101y for some x and y)}

$$\sum^{*}0101\sum^{*}$$

d. {w| w has length at least 3 and its third symbol is a 0}

$$\sum\sum 0\sum^{*}$$

e. {w| w starts with 0 and has odd length, or starts with 1 and has even length}

$$(0\bigcup 1\sum)(\sum\sum)^{*}$$

f. {w| w doesnt contain the substring 110}

$$0^{*}(10^{+})^{*}1^{*}$$

g. {w| the length of w is at most 5}

$$(\epsilon \bigcup \sum)^5$$

h. {w| w is any string except 11 and 111}

$$\epsilon \bigcup \sum \bigcup 0 \sum \bigcup 10 \bigcup 0 \sum \sum \bigcup 10 \sum \bigcup 110 \bigcup \sum{}^3 \sum{}^+$$

I. {w| every odd position of w is a 1}

$$(1 \sum)^*(\epsilon \bigcup 1)$$

j. {w| w contains at least two 0s and at most one 1}

$$00^+ \bigcup 100^+ \bigcup 0^+10^+ \bigcup 00^+1$$

k. {", 0}

$$0 \bigcup \epsilon$$

L. {w| w contains an even number of 0s, or contains exactly two 1s}

$$1^*(01^*01^*) \bigcup 0^*10^*10^*$$

m. The empty set

$$\emptyset$$

n. All strings except the empty string

$$\sum{}^+$$

1.20: a, b, c, d, e, f, g, h
a. Members: ab, abb
Not members: ba, bba

b. Members: ab, ababab
Not members: aba, bab

c. Members: aaa, bbb
Not members: aabb, bbaa

d. Members: aaa, aaaaaa
Not members: a, aaaa

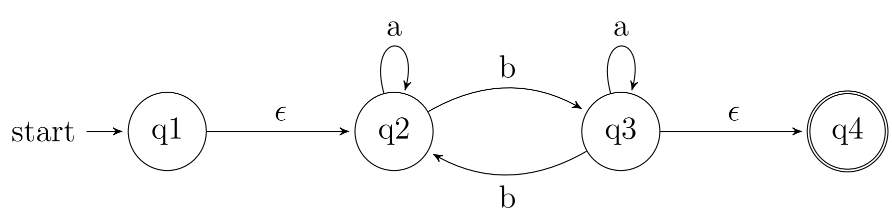
e. Members: aba, bbaaabaabb
Not members: a, b

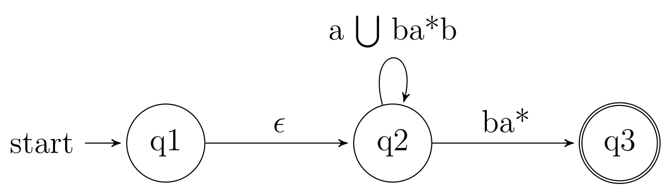
f. Members: aba, bab
Not members: ababab, ba

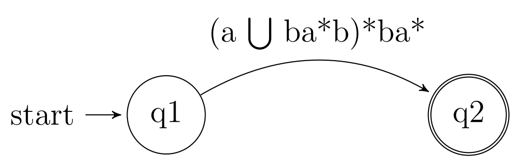
g. Members: b, ab
Not members: a, ba

h. Members: a, bbab
Not members: b, ϵ

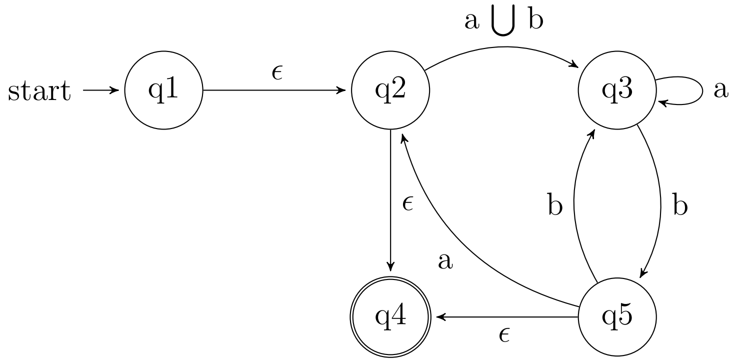
1.21 a.

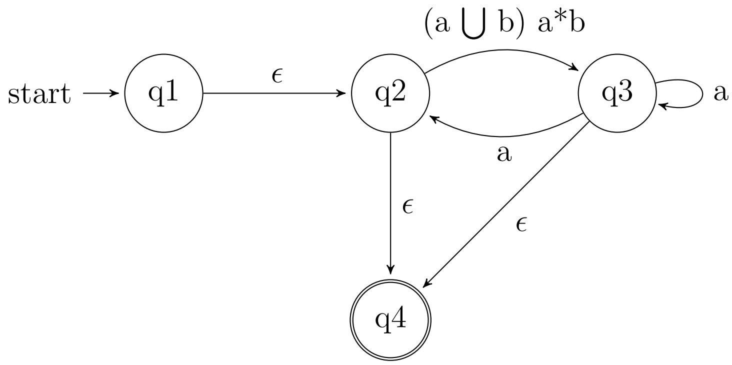


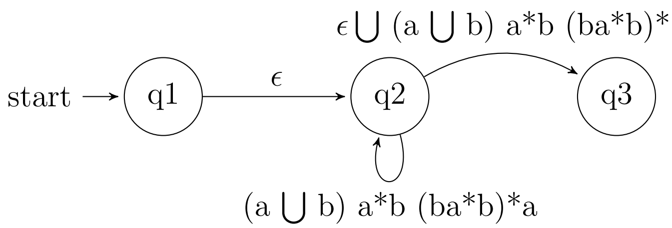


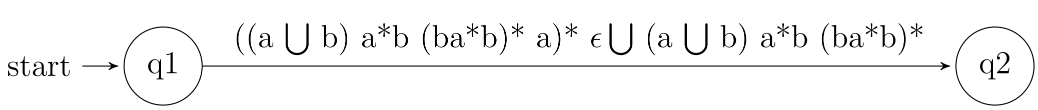


b.


start
q1
ϵ
q2
a ⋃ b
q3
a
ϵ
b
b
a
q4
q5
ϵ

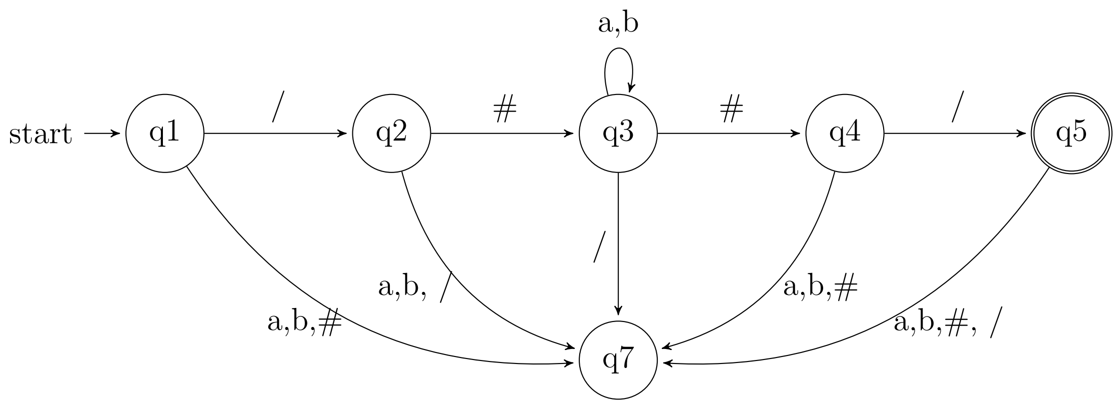

(a ⋃ b) a*b
start
q1
ϵ
q2
q3
a
a
ϵ
ϵ
q4


ϵ ⋃ (a ⋃ b) a*b (ba*b)*
start
q1
ϵ
q2
q3
(a ⋃ b) a*b (ba*b)*a

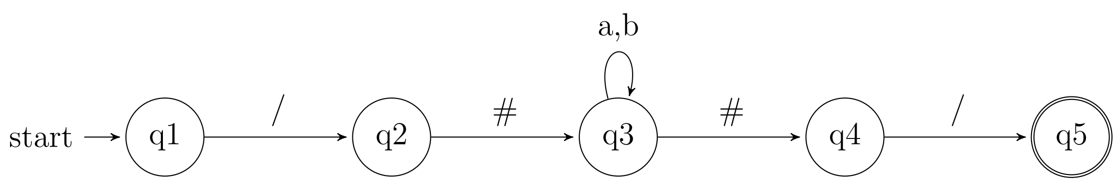

start
q1
((a ⋃ b) a*b (ba*b)* a)* ϵ ⋃ (a ⋃ b) a*b (ba*b)*
q2

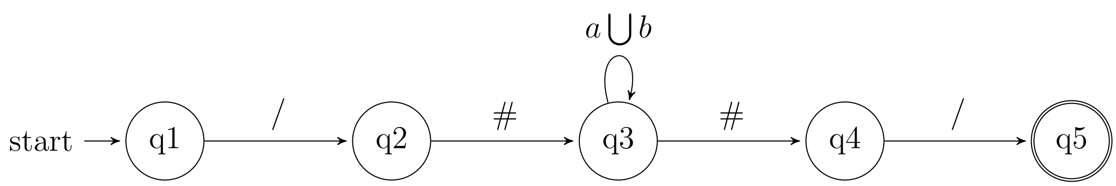
1.22 In certain programming languages, comments appear between delimiters such as /# and #/. Let C be the language of all valid delimited comment strings. A member of C must begin with /# and end with #/ but have no intervening #/. For simplicity, assume that the alphabet for C is $\sum = \{a, b, /, \#\}$.

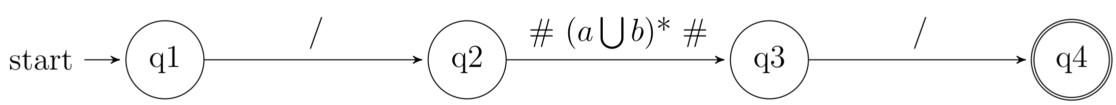
a. Give a DFA that recognizes C.

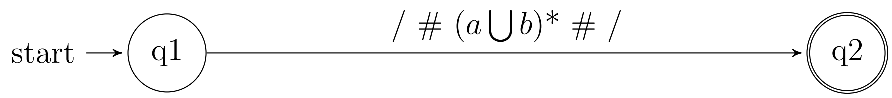


b. Give a regular expression that generates C.









The regular expression is: / # $(a \bigcup b)^*$ # /