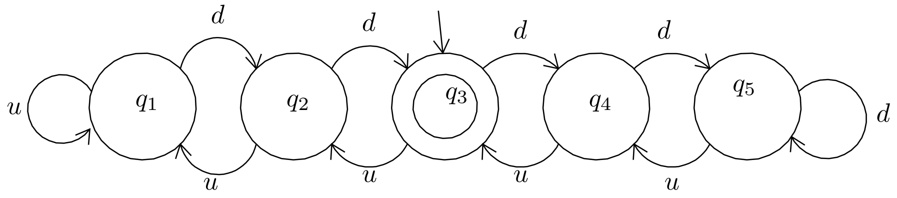
Some Theory of Computation Exercises – Week 1

Section 1 – Deterministic Finite Automata

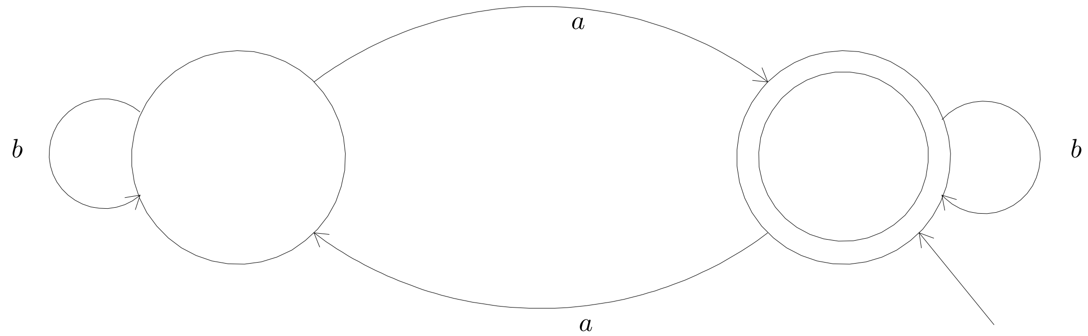
Question 1.3



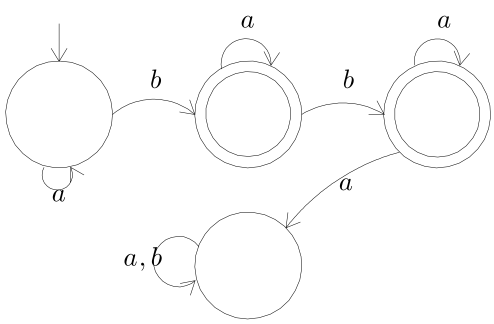
Question 1.4

Part c - $\{w\,|w\,\text{has even}\,a's\,\text{and one or two}\,b's\}$

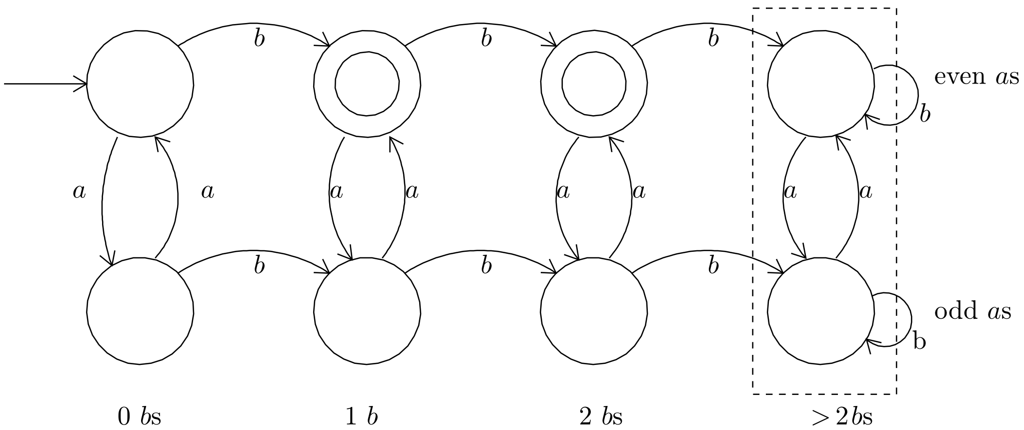
First we ask whether w has an even number of as, which requires two states – one to signify that an even number of as has been observed thus far, and one to signify that an odd number have been observed thus far:



Next we must determine whether w contains precisely one or two bs, which needs four states: initialization, a single b, two bs and the failure state



These two devices must now be combined to create a recogniser for their intersection. Every pair of states will be represented by a single state in the new device, so we can lay it out in a 4×2 grid thus:

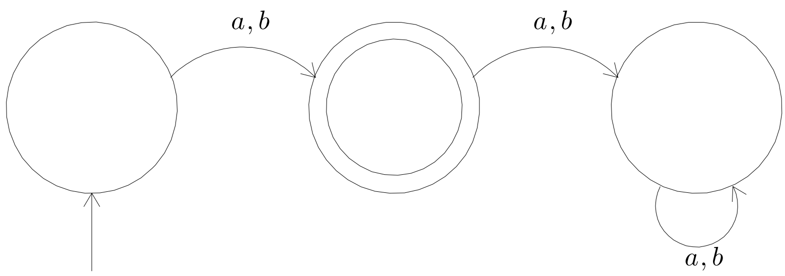


Notice that the two states in the dotted rectangle can be combined into a single state, as no sequence of characters will allow the device to escape from the box and neither state is an accepting state.

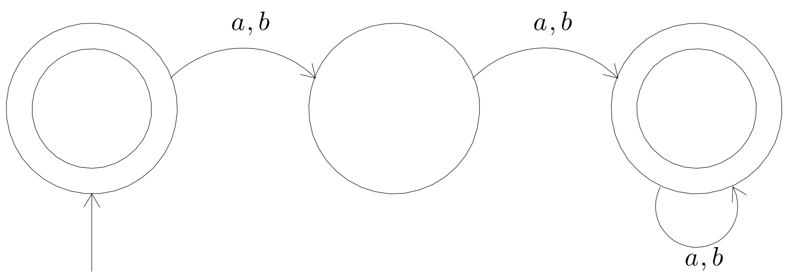
Question 1.5

Part h – recognise $\{w|w\,\textbf{is any string except}\,''a'',\,''b''\}$

As suggested in the question we can first construct a device which recognises the complement of this language, namely $\{a,\ b\}$, and then negate its output to find the solution which we are looking for.



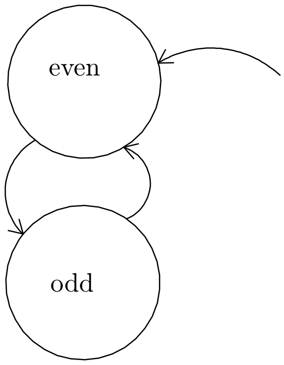
To negate its output we convert all the accepting states into rejecting states and vice-versa:



Question 1.6

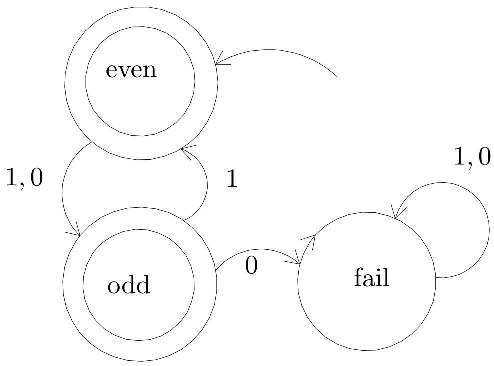
Part i $-\{w|\,\textbf{every odd position is 1}\}$

This is similar to the even/odd number of as required in question 1.4 above. Here we need a gadget to keep track of whether we are looking at an odd/even numbered position, which will head into an attracting failure state if the position is odd and is not 1.



Here's an odd/even detector, to which we just need to add the fail state

We can only fail by detecting a 0 when the position is odd, so we get something like this:



Adding that we accept if we don't fail

Question 1.33 - multiplying by 3

Here $\Sigma = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$ and we want to prove the regularity of the set of strings from Σ^* wherein the bottom row is three times the top row, so for example:

In L	Not in L
$\begin{pmatrix} 0 \\ 0 \end{pmatrix} - 0 \times 3 = 0$	$\begin{pmatrix} 0 \\ 1 \end{pmatrix} - 0 \times 3 \neq 1$
$\begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} - 1 \times 3 = 3$	etc

Table 1. Some example strings

Since we are given that regularity of the reverse of a language implies the regularity of the language, we prove that L is regular by constructing a DFA which recognises the reverse of L, in which the least significant bit comes first.

For the language to be regular it must be possible to perform the recognition using an amount of memory which doesn't increase with the length of the input (this is where we record our current state), and with access only to one symbol of input at a time. Multiplying by 3 definitely has this property, as we can work with 1 column at a time and the carry never exceeds 2. For example:

C_1	0	1	1	0	0
C_0	1	1	1	1	0
(7)	0	0	1	1	1
(3)	0	0	0	1	1
(7×3)	1	0	1	0	1

Reading this from right to left:

1. $1 \times 1 = 1$ + no carry = 1 (output), and

   $1 \times 1 = 1$ (new carry state is 01)

2. $1 \times 1 = 1$ + carry (1) = 10, so output is 0 and $C_0 = 1$

   $1 \times 1 = 1$, so we are now carrying 2, $C_1 = 1$ (1 from above and 1 from this)

3. $1 \times 1 = 1$ + carry (2) = 11 (3), so output is 1 and $C_0 = 1$

   $1 \times 1 = 1$, so we are still carrying 2

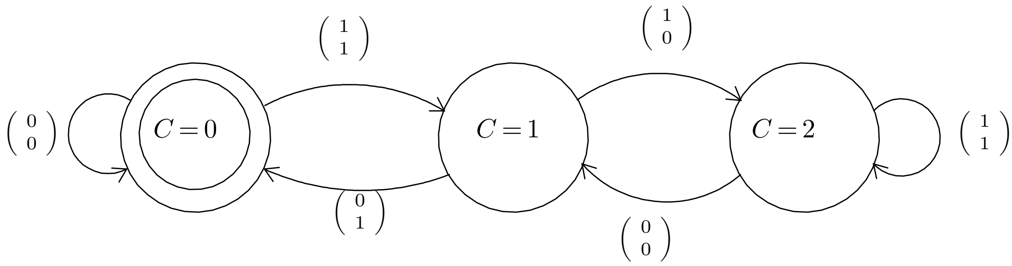
4. $1 \times 0 = 0$ + carry (2) = 10 (2), so output is 0 and $C_0 = 1$

   $1 \times 0 = 0$ = 0 so $C_1 = 0$

5. $1 \times 0 = 0$ + carry (1) = 1, so output is 1 and $C_0 = 0$

   $1 \times 0 = 0$ = 0 so $C_1 = 0$

So we need 3 states, one for no carry, one for a single carry and one for two carries, and then for any valid transition the result bit must be the least significant bit of the sum of the input bit and the current carry. The current carry is then updated to be the most significant bit of the sum of input and old carry + the input; that gives us the following DFA. Any missing transitions lead to an inescapable failure state which I haven't drawn in the interests of parsimony.



So for example in $C = 2$ when we see an input bit of 1 we are adding $1 \times 1 + 2 = 3 = 11$ to the total, so we expect to see an output bit of 1 (the low bit), and a carry of at least 1 (the high bit). We are also adding 1×1 in the next column, so the carry must remain at 2. Similar arguments hold for all the other valid transitions.

Theory of Computation: Problems Class 2

1.16a)

NFA

$Q=\{1,2\}$

$q_0=1$

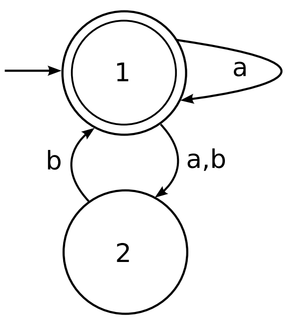
$F=\{1\}$

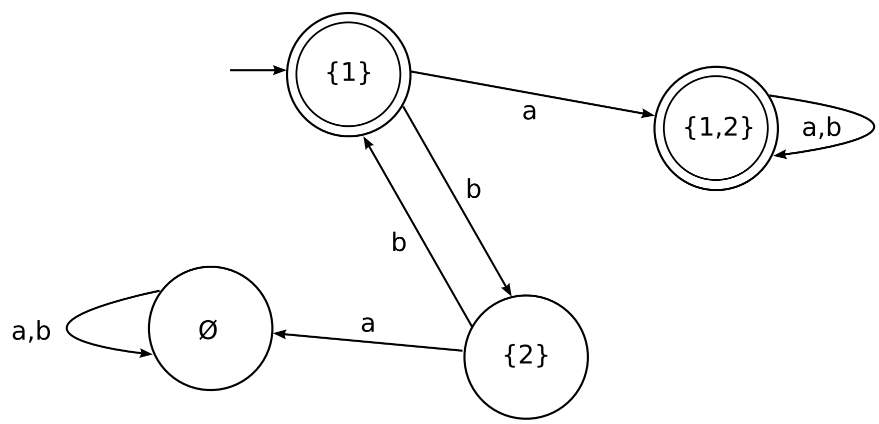
DFA

$Q'=\{\varnothing,\{1\},\{2\},\{1,2\}\}$

$q_0'=\{1\}$

$F'=\{\{1\},\{1,2\}\}$





1.16b)

NFA

$Q=\{1,2,3\}$

$q_0=1$

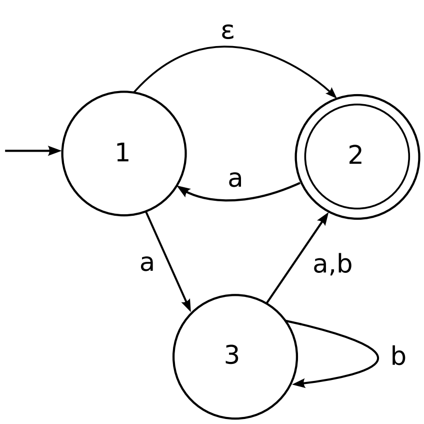
$F=\{2\}$

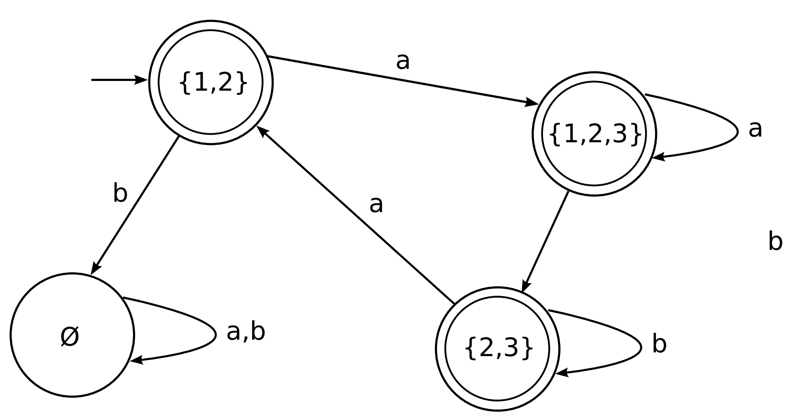
DFA

$Q'=\{\varnothing,\{1\},\{2\},\{1,2\},\{1,3\},\{2,3\},\{1,2,3\}\}$

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

$F'=\{\{2\},\{1,2\},\{2,3\},\{1,2,3\}\}$





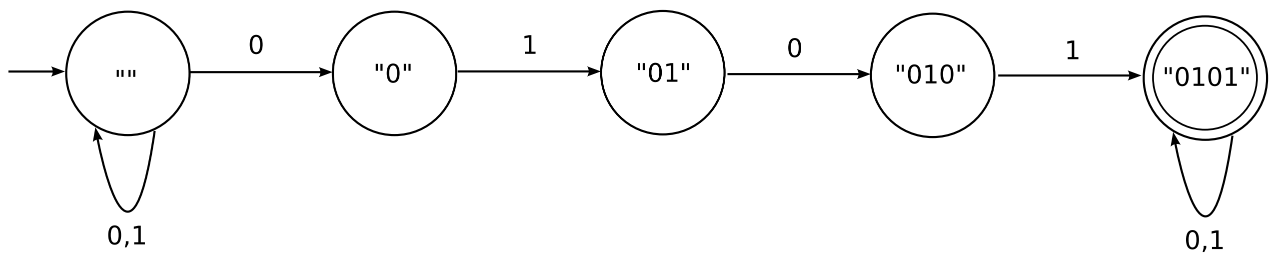
1.7b)

R={w | w contains the substring 0101}

Q={"","0","01","010","0101"}

q_0=""

F={"0101"}



1.34)

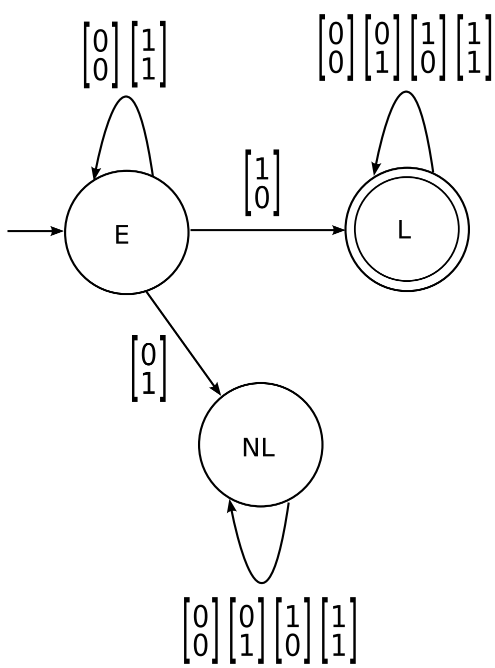
$$\Sigma=\left\{\begin{bmatrix}0\\0\end{bmatrix},\begin{bmatrix}0\\1\end{bmatrix},\begin{bmatrix}1\\0\end{bmatrix},\begin{bmatrix}1\\1\end{bmatrix}\right\}$$

R={w =Σ^*| the top row of w is a larger number than the bottom row}

Q={E, L, NL}

q_0=E

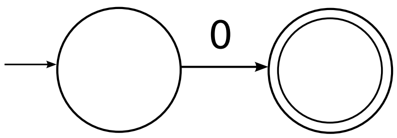
F={L}



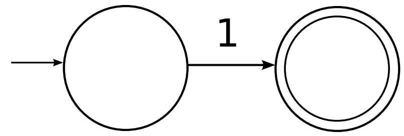
Theory of Computation: Problems Class 3

1.18a)

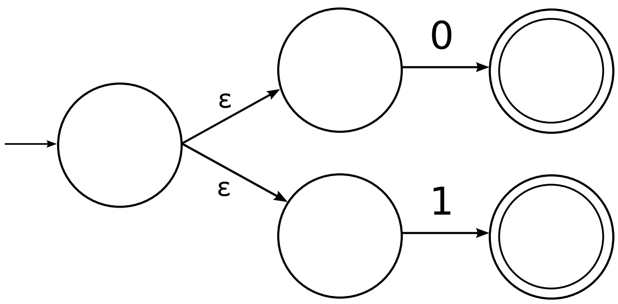
0



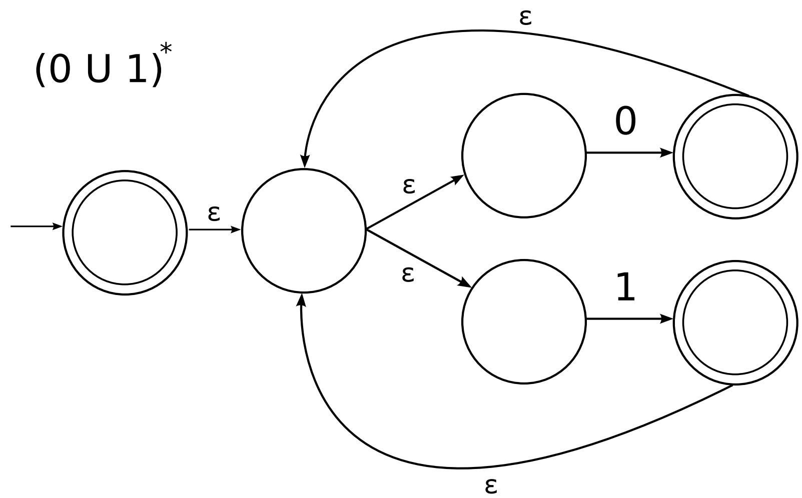
1



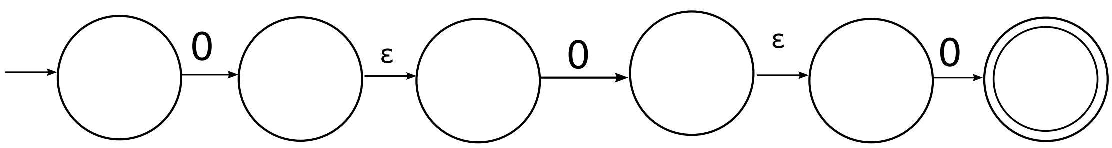
0 U 1



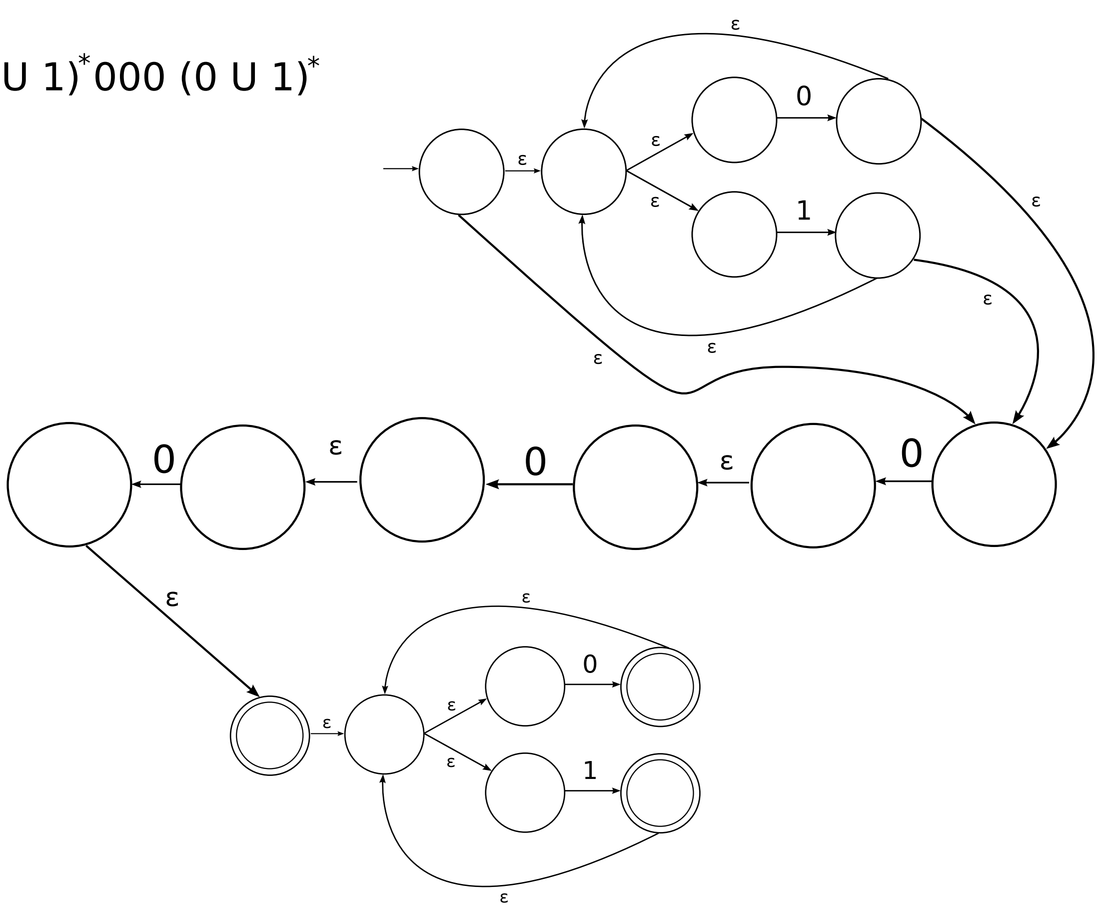
(0 U 1)*



000

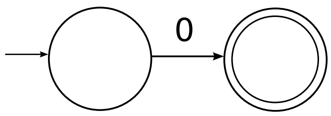


R=(0 U 1)* 000 (0 U 1)*

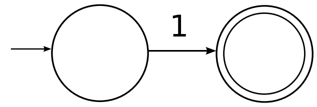


1.18b)

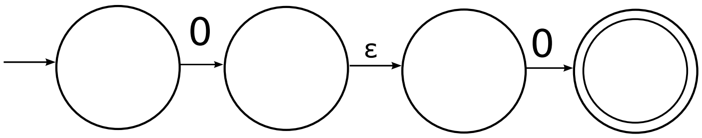
0



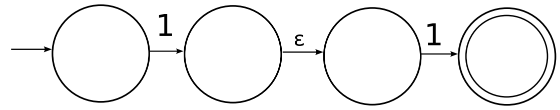
1



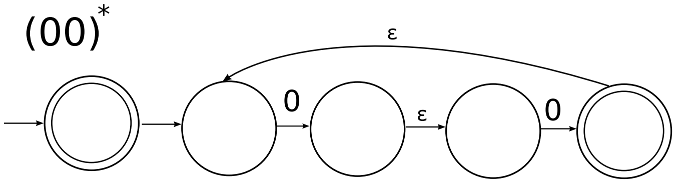
00



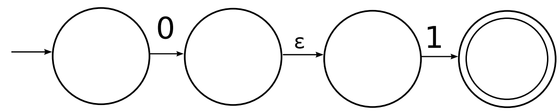
11



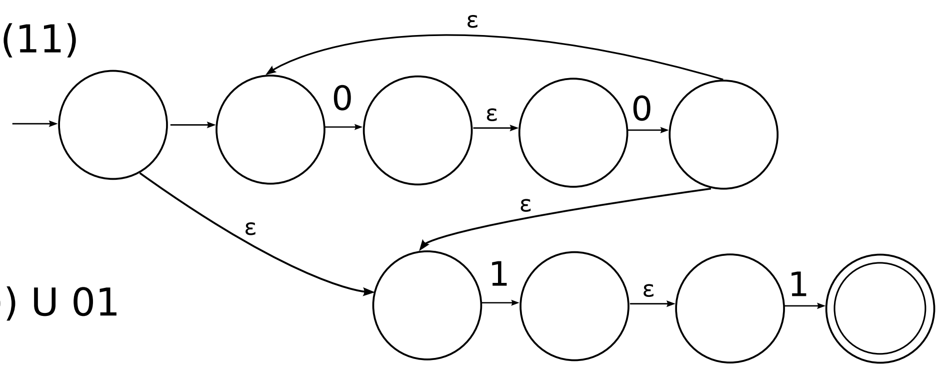
$(00)^*$



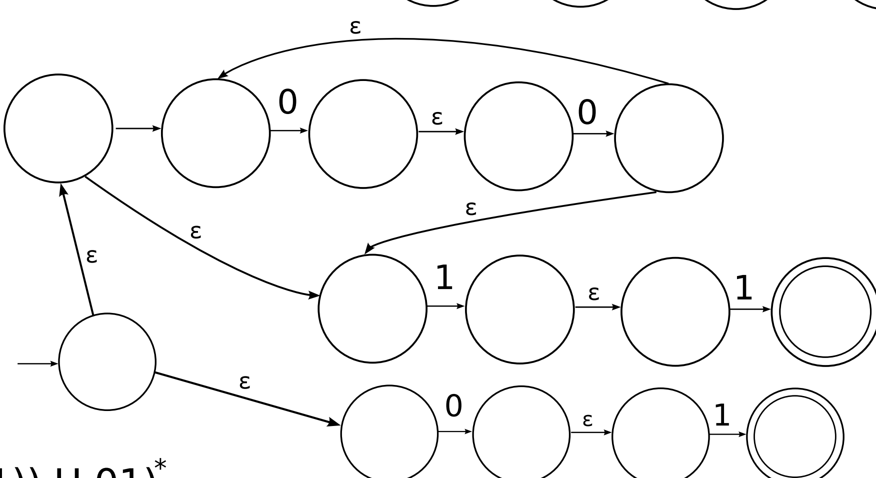
01



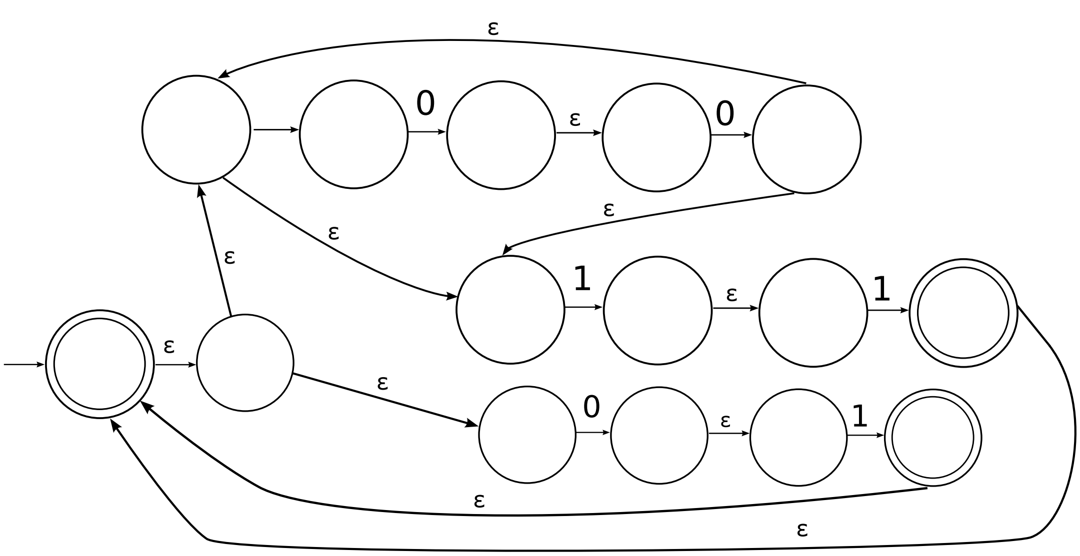
$(00)^*(11)$



$((00)^*(11)) \cup 01$

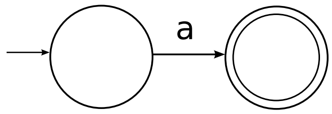


$(((00)^*(11)) \cup 01)^*$

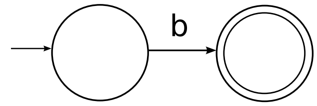


1.28a)

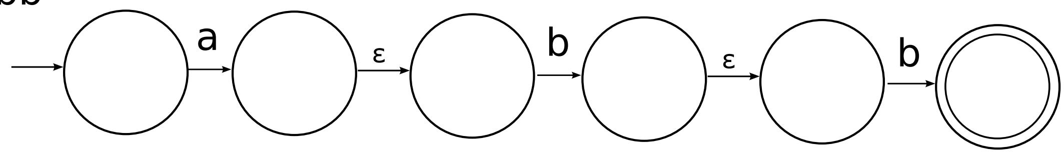
a



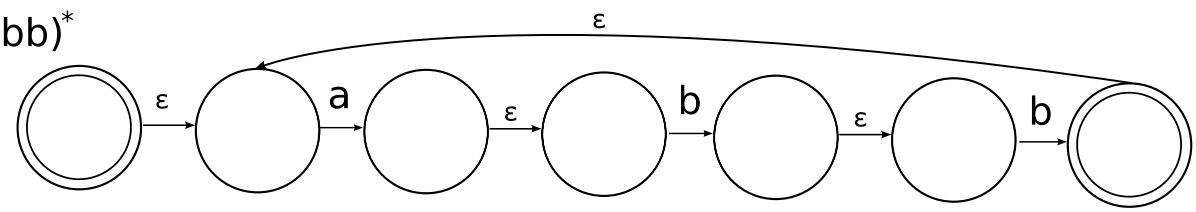
b



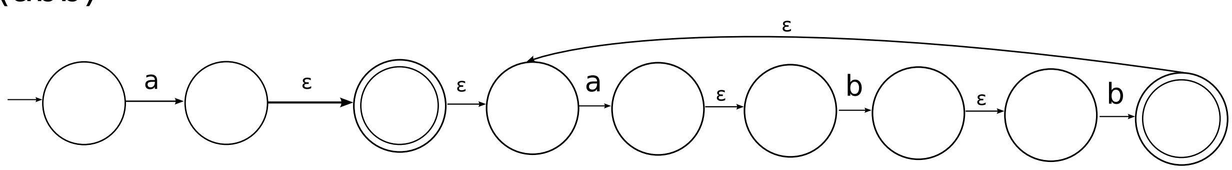
abb



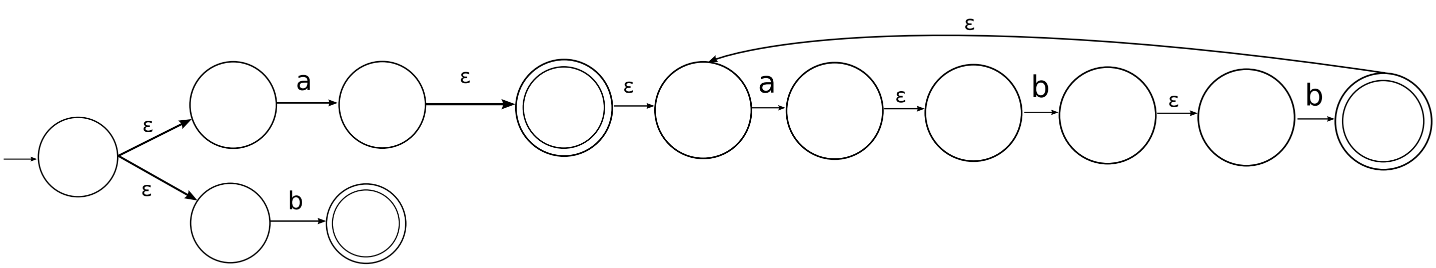
$(abb)^*$



$a(abb)^*$



$a(abb)^* \cup b$



Deterministic Finite Automata

1. Exercises in pages 83-84.
   Exercise 1.1
   a. The start state of each machine is q_1.
   b. The sets of accept states of M_1 and M_1 are respectively $F=\{q_2\}$ and $F=\{q_1,q_4\}$.
   c. M_1 goes through states q_1, q_2, q_3, q_1 and q_1. M_2 goes through states q_1, q_1, q_1, q_2 and q_4.
   d. M_1 does not accept the string aabb. M_1 does.
   e. M_1 does not accept ε. M_2 does.

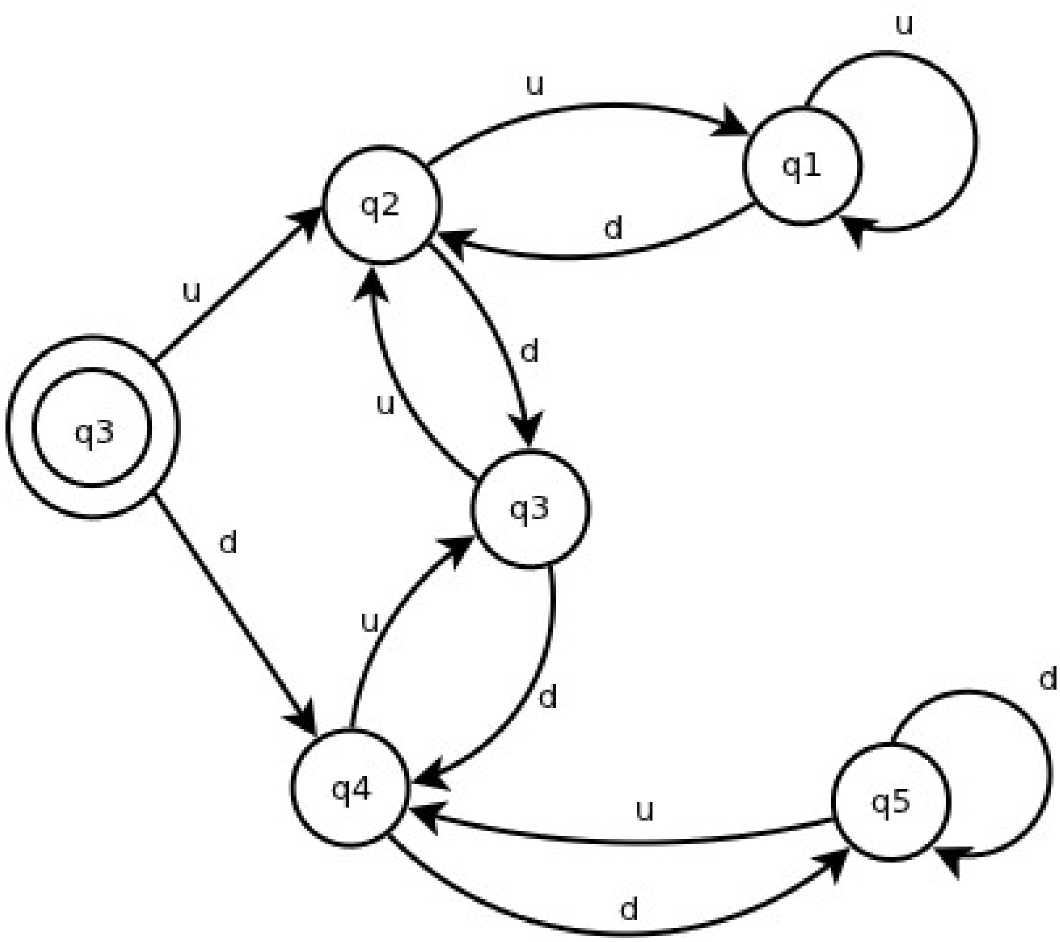
   Exercise 1.2
   a. Formal description of M_1.
      $\Sigma=\{a,b\}$
      $Q=\{q_1,q_2,q_3\}$
      $F=\{q_2\}$
      δ is given by

	a	b
q_1	q_2	q_1
q_2	q_3	q_3
q_3	q_2	q_1

   b. Formal description of M_2.
      $\Sigma=\{a,b\}$
      $Q=\{q_1,q_2,q_3,q_4\}$
      $F=\{q_3,q_4\}$
      δ is given by

	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

Exercise 1.3



Exercise 1.4

a. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w \mid w$ has at least three a's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2,s_3\}$

$q_{01}= s_0$

$F_1=\{s_3\}$

δ is given by

	a	b
s_0	s_1	s_0
s_1	s_2	s_1
s_2	s_3	s_2
s_3	s_3	s_3

The formal definition of a machine M_2 accepting the language $L(M_2)=\{w \mid w$ has at least two b's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_2=\{t_0,t_1,t_2\}$

$q_{02}= t_0$

$F_2=\{t_2\}$

δ is given by

	a	b
t_0	t_0	t_1
t_1	t_1	t_2
t_2	t_2	t_2

The formal definition of a machine M accepting the language L(M)={w|w has at least three a's and at least two b's} is as follows:

$\Sigma=\{a,b\}$
$Q=Q_1\times Q_2$
$q_0=(s_0,t_0)$
$F=F_1 x F_2=\{(s_3,t_2)\}$
δ is given by

	a	b
(s_0,t_0)	(s_1,t_0)	(s_0,t_1)
(s_0,t_1)	(s_1,t_1)	(s_0,t_2)
(s_0,t_2)	(s_1,t_2)	(s_0,t_2)
(s_1,t_0)	(s_2,t_0)	(s_1,t_1)
(s_1,t_1)	(s_2,t_1)	(s_1,t_2)
(s_1,t_2)	(s_2,t_2)	(s_1,t_2)
(s_2,t_0)	(s_3,t_0)	(s_2,t_1)
(s_2,t_1)	(s_3,t_1)	(s_2,t_2)
(s_2,t_2)	(s_3,t_2)	(s_2,t_2)
(s_3,t_0)	(s_3,t_0)	(s_3,t_1)
(s_3,t_1)	(s_3,t_1)	(s_3,t_2)
(s_3,t_2)	(s_3,t_2)	(s_3,t_2)

b. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ has exactly two a's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2,s_3\}$

$q_{01}= s_0$

$F_1 =\{s_2\}$

δ is given by

	a	b
s_0	s_1	s_0
s_1	s_2	s_1
s_2	s_3	s_2
s_3	s_3	s_3

The formal definition of a machine M_2 accepting the language $L(M_2)=\{w|w$ has at least two b's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_2=\{t_0,t_1,t_2\}$

$q_{02}= t_0$

$F_2 =\{t_2\}$

δ is given by

	a	b
t_0	t_0	t_1
t_1	t_1	t_2
t_2	t_2	t_2

The formal definition of a machine M accepting the language $L(M_2)=\{w|w$ has at exactly two a's and at least two b's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q=Q_1\times Q_2$

$q_0=(s_0,t_0)$

$F=F_1 x F_2=\{(s_2,t_2)\}$

δ is given by

	a	b
(s_0,t_0)	(s_1,t_0)	(s_0,t_1)
(s_0,t_1)	(s_1,t_1)	(s_0,t_2)
(s_0,t_2)	(s_1,t_2)	(s_0,t_2)
(s_1,t_0)	(s_2,t_0)	(s_1,t_1)
(s_1,t_1)	(s_2,t_1)	(s_1,t_2)
(s_1,t_2)	(s_2,t_2)	(s_1,t_2)
(s_2,t_0)	(s_3,t_0)	(s_2,t_1)
(s_2,t_1)	(s_3,t_1)	(s_2,t_2)
(s_2,t_2)	(s_3,t_2)	(s_2,t_2)
(s_3,t_0)	(s_3,t_0)	(s_3,t_1)
(s_3,t_1)	(s_3,t_1)	(s_3,t_2)
(s_3,t_2)	(s_3,t_2)	(s_3,t_2)

c. The formal definition of a machine M_1 accepting the language L(M_1)={w|w has an even number of a's} is as follows:

Σ={a,b}

Q_1={s_0,s_1,s_2}

q_{01}= s_0

F_1 ={s_2}

δ is given by

	a	b
s_0	s_1	s_0
s_1	s_2	s_1
s_2	s_1	s_2

The formal definition of a machine M_2 accepting the language L(M_2)={w|w has one or two b's} is as follows:

Σ={a,b}

Q_2={t_0,t_1,t_2,t_3}

q_{02}= t_0

F_2 ={t_1,t_2}

δ is given by

	a	b
t_0	t_0	t_1
t_1	t_1	t_2
t_2	t_2	t_3
t_3	t_3	t_3

The formal definition of a machine M accepting the language L(M)={w|w has an even number of a's and one or two b's} is as follows:

$\Sigma=\{a,b\}$

$Q=Q_1\times Q_2$

$q_0=(s_0,t_0)$

$F=F_1 x F_2=\{(s_2,t_1),(s_2,t_2)\}$

δ is given by

	a	b
(s_0,t_0)	(s_1,t_0)	(s_0,t_1)
(s_0,t_1)	(s_1,t_1)	(s_0,t_2)
(s_0,t_2)	(s_1,t_2)	(s_0,t_3)
(s_0,t_3)	(s_1,t_3)	(s_0,t_3)
(s_1,t_0)	(s_2,t_0)	(s_1,t_1)
(s_1,t_1)	(s_2,t_1)	(s_1,t_2)
(s_1,t_2)	(s_2,t_2)	(s_1,t_3)
(s_1,t_3)	(s_2,t_3)	(s_1,t_3)
(s_2,t_0)	(s_1,t_0)	(s_2,t_1)
(s_2,t_1)	(s_1,t_1)	(s_2,t_2)
(s_2,t_2)	(s_1,t_2)	(s_2,t_3)
(s_2,t_3)	(s_1,t_3)	(s_2,t_3)

d. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w \mid w \text{ has an even number of a's}\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2\}$

$q_{01}= s_0$

$F_1 =\{s_2\}$

δ is given by

	a	b
s_0	s_1	s_0
s_1	s_2	s_1
s_2	s_1	s_2

The formal definition of a machine M_2 accepting the language $L(M_2)=\{w \mid \text{in } w \text{ each a is followed by at least one b}\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_2=\{t_0,t_1,t_2,t_3\}$

$q_{02}= t_0$

$F_2 =\{t_0,t_2\}$

δ is given by

	a	b
t_0	t_1	t_0
t_1	t_3	t_2
t_2	t_1	t_2
t_3	t_3	t_3

The formal definition of a machine M accepting the language L(M)={w|w has an even number of a's and each is followed by at least one b} is as follows:

$\Sigma=\{a,b\}$

$Q=Q_1\times Q_2$

$q_0=(s_0,t_0)$

$F=F_1 x F_2=\{(s_0,t_0),(s_0,t_2)\}$

δ is given by

	a	b
(s_0,t_0)	(s_1,t_1)	(s_0,t_0)
(s_0,t_1)	(s_1,t_3)	(s_0,t_2)
(s_0,t_2)	(s_1,t_1)	(s_0,t_2)
(s_0,t_3)	(s_1,t_3)	(s_0,t_3)
(s_1,t_0)	(s_2,t_1)	(s_1,t_0)
(s_1,t_1)	(s_2,t_3)	(s_1,t_2)
(s_1,t_2)	(s_2,t_1)	(s_1,t_2)
(s_1,t_3)	(s_2,t_3)	(s_1,t_3)
(s_2,t_0)	(s_1,t_1)	(s_2,t_0)
(s_2,t_1)	(s_1,t_3)	(s_2,t_2)
(s_2,t_2)	(s_1,t_1)	(s_2,t_2)
(s_2,t_3)	(s_1,t_3)	(s_2,t_3)

2. Not included in this solution sheet.

a. The formal definition of a machine M_1 accepting the language L(M_1)={w|w has an odd number of a's} is as follows:

Σ={a,b}
Q_1={s_0,s_1,s_2}
q_{01}= s_0
F_1 ={s_1}
δ is given by

	a	b
s_0	s_1	s_0
s_1	s_2	s_1
s_2	s_1	s_2

The formal definition of a machine M_2 accepting the language L(M_2)={w|w ends with a b} is as follows:

Σ={a,b}
Q_2={t_0,t_1}
q_{02}= t_0
F_2 ={t_1}
δ is given by

	a	b
t_0	t_0	t_1
t_1	t_0	t_1

The formal definition of a machine M accepting the language $L(M)=\{w|w$ has an odd number of a's and ends with a b$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q=Q_1\times Q_2$

$q_0=(s_0,t_0)$

$F=F_1 x F_2=\{(s_1,t_1)\}$

δ is given by

	a	b
(s_0,t_0)	(s_1,t_0)	(s_0,t_1)
(s_0,t_1)	(s_1,t_0)	(s_0,t_1)
(s_1,t_0)	(s_2,t_0)	(s_1,t_1)
(s_1,t_1)	(s_2,t_0)	(s_1,t_1)
(s_2,t_0)	(s_1,t_0)	(s_2,t_1)
(s_2,t_1)	(s_1,t_0)	(s_2,t_1)

b. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ has an even length$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1\}$

$q_{01}=s_0$

$F_1=\{s_0\}$

δ is given by

	a	b
s_0	s_1	s_1
s_1	s_0	s_0

The formal definition of a machine M_2 accepting the language $L(M_2)=\{w|w$ has an odd number of a's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_2=\{t_0,t_1,t_2\}$

$q_{02}= t_0$

$F_2 =\{t_1\}$

δ is given by

	a	b
t_0	t_1	t_0
t_1	t_2	t_1
t_2	t_1	t_2

The formal definition of a machine M accepting the language $L(M)=\{w|w$ has an even length and an odd number of a's$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q=Q_1\times Q_2$

$q_0= (s_0,t_0)$

$F =F_1 x F_2=\{(s_0,t_1)\}$

δ is given by

	a	b
(s_0,t_0)	(s_1,t_1)	(s_1,t_0)
(s_0,t_1)	(s_1,t_2)	(s_1,t_1)
(s_0,t_2)	(s_1,t_1)	(s_1,t_2)
(s_1,t_0)	(s_0,t_1)	(s_0,t_0)
(s_1,t_1)	(s_0,t_2)	(s_0,t_1)
(s_1,t_2)	(s_0,t_1)	(s_0,t_2)

Exercise 1.5

a. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ contains the substring ab$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2\}$

$q_{01}= s_0$

$F_1 =\{s_2\}$

δ is given by

	a	b
s_0	s_1	s_0
s_1	s_1	s_2
s_2	s_2	s_2

A machine M_2 accepting the language $L(M_2)=\{w|w$ does not contain the substring ab$\}$ is defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_0,s_1\}$.

b. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ contains the substring baba$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2,s_3,s_4\}$

$q_{01}= s_0$

$F_1 =\{s_4\}$

δ is given by

	a	b
s_0	s_0	s_1
s_1	s_2	s_1
s_2	s_0	s_3
s_3	s_4	s_1
s_4	s_4	s_4

A machine M_2 accepting the language $L(M_2)=\{w|w$ does not contain the substring baba$\}$ is defined by the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_0,s_1,s_2,s_3\}$.

c. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ contains either the substring ab or ba$\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2,s_3\}$

$q_{01}= s_0$

$F_1 =\{s_3\}$

δ is given by

	a	b
s_0	s_1	s_2
s_1	s_1	s_3
s_2	s_3	s_2
s_3	s_3	s_3

A machine M_2 accepting the language $L(M_2)=\{w|w$ contains neither the substring ab or ba$\}$ is defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_0,s_1,s_2\}$.

d. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ is any string in $a^*b^*\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2,s_3,s_4\}$

$q_{01}= s_0$

$F_1=\{s_0,s_1,s_2,s_3\}$

δ is given by

	a	b
s_0	s_1	s_2
s_1	s_1	s_3
s_2	s_4	s_2
s_3	s_4	s_3
s_4	s_4	s_4

A machine M_2 accepting the language $L(M_2)=\{w|w$ is any string not in $a^*b^*\}$ is defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_4\}$.

e. The formal definition of a machine M_1 accepting the language L(M_1)={w|w is any string in (ab+)*} is as follows:

$\Sigma=\{a,b\}$
$Q_1=\{s_0,s_1,s_2,s_3\}$
$q_{01}= s_0$
$F_1 =\{s_0,s_3\}$
δ is given by

	a	b
s_0	s_2	s_1
s_1	s_1	s_1
s_2	s_1	s_3
s_3	s_2	s_3

A machine M_1 accepting the language L(M_1)={w|w is any string not in (ab+)*} is defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_1,s_2\}$.

f. The formal definition of a machine M_1 accepting the language L(M_1)={w|w is any string in a*Ub*} is as follows:

$\Sigma=\{a,b\}$
$Q_1=\{s_0,s_1,s_2,s_3\}$
$q_{01}= s_0$
$F_1 =\{s_0,s_1,s_2\}$
δ is given by

	a	b
s_0	s_1	s_2
s_1	s_1	s_3
s_2	s_3	s_2
s_3	s_3	s_3

A machine M_2 accepting the language $L(M_2)=\{w|w \text{ is any string not in } a^*Ub^*\}$ defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_3\}$.

g. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w \text{ is any string that contains exactly two a's}\}$ is as follows:

$\Sigma=\{a,b\}$

$Q_1=\{s_0,s_1,s_2,s_3\}$

$q_{01}= s_0$

$F_1 =\{s_2\}$

δ is given by

	a	b
s_0	s_1	s_0
s_1	s_2	s_1
s_2	s_3	s_2
s_3	s_3	s_3

A machine machine M_2 accepting the language $L(M_2)=\{w|w \text{ is any string that doesn't contain exactly two a's}\}$ defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_0,s_1,s_3\}$.

h. The formal definition of a machine M_1 accepting the language $L(M_1)=\{w|w$ is any string a or b$\}$ is as follows:

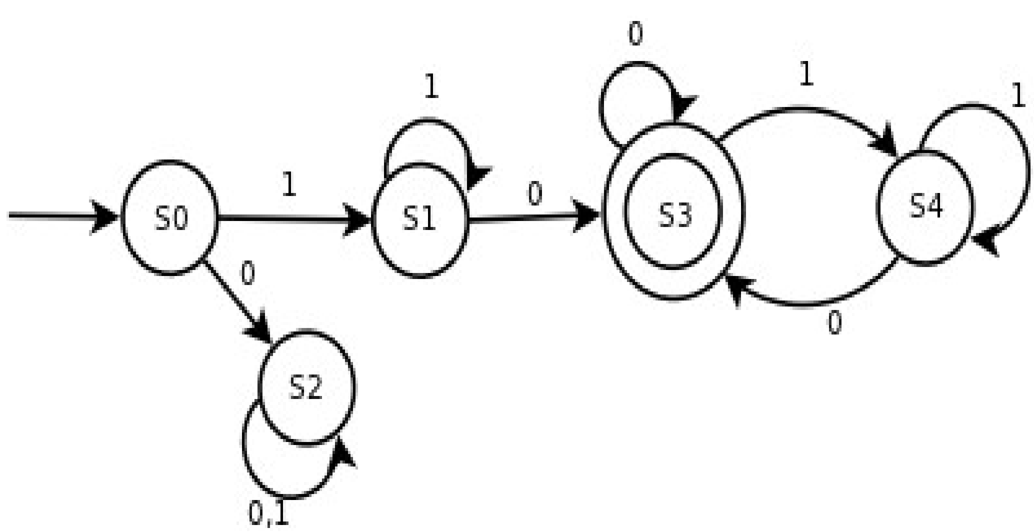
$\Sigma=\{a,b\}$
$Q_1=\{s_0,s_1,s_2\}$
$q_{01}= s_0$
$F_1 =\{s_1\}$
δ is given by

	a	b
s_0	s_1	s_1
s_1	s_2	s_2
s_2	s_2	s_2

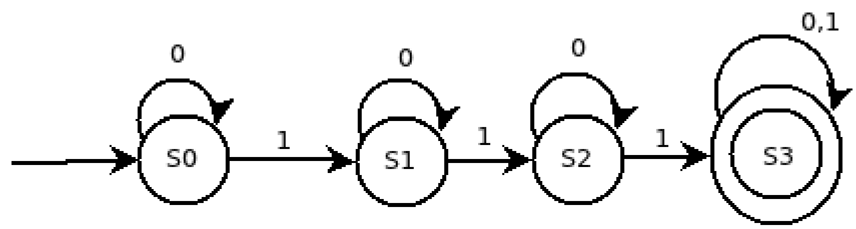
A machine M_2 accepting the language $L(M_2)=\{w|w$ is any string except a and b$\}$ is defined by the same alphabet, the same states, the same transition function, the same start state and a set of accept states $F_2=Q-F_1=\{s_0,s_2\}$.

Exercise 1.6

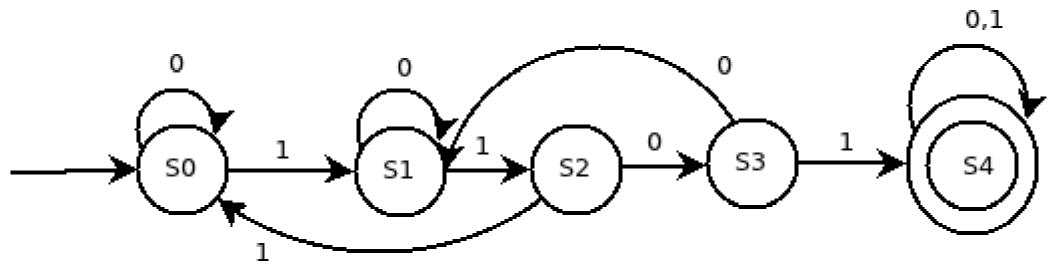
a) {w|w begins with a 1 and ends with a zero}.



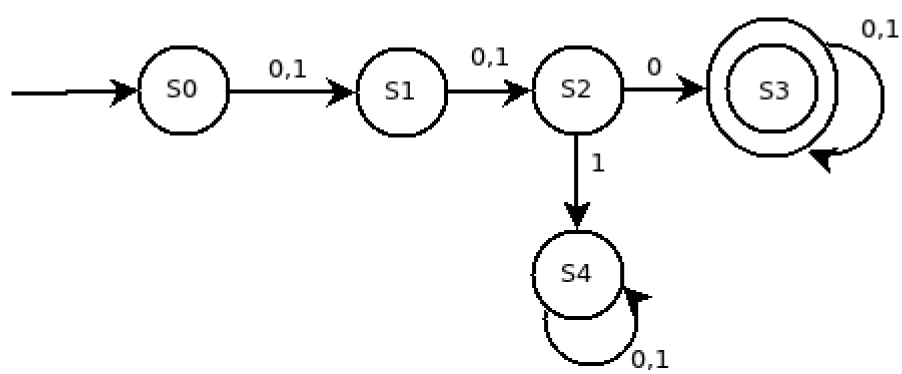
b) {w|w contains at least three 1s}



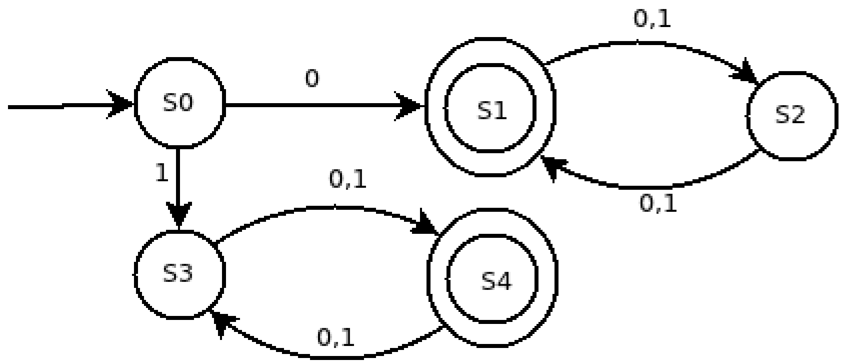
c) {w|w contains the substring 0101}



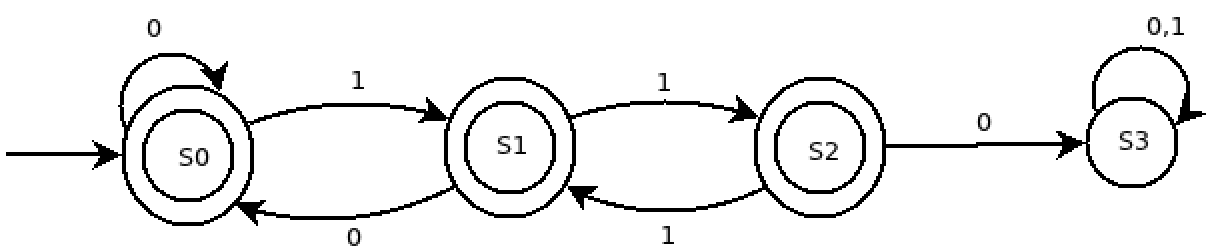
d) {w|w has length at least three and its third symbol is a 0}



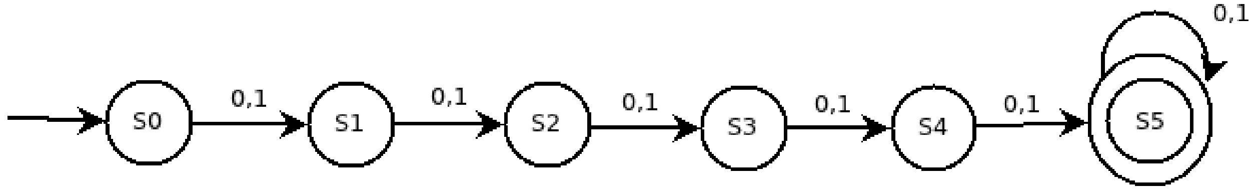
e) {w|w starts with 0 and has odd length, or starts with 1 and has even length}



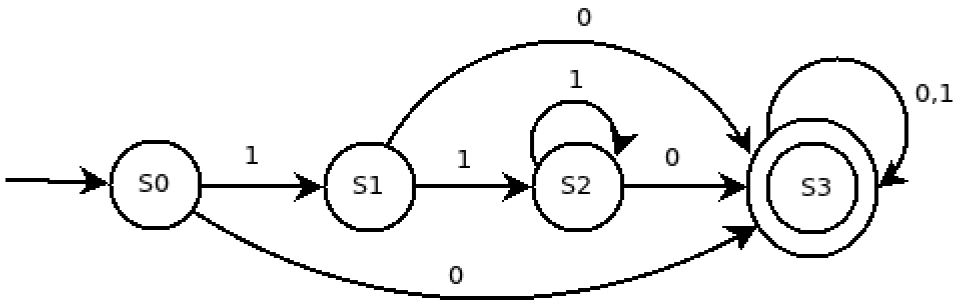
f) {w|w doesn't contain the substring 110}



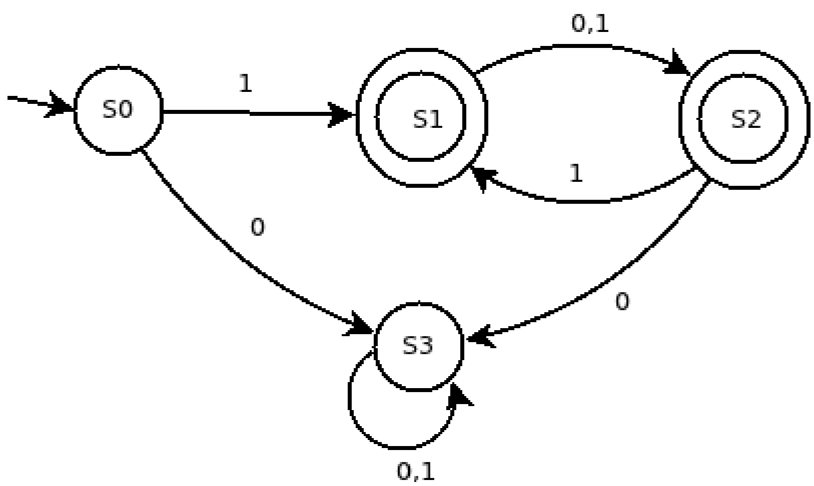
g) {w|the length of w is at most 5}



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

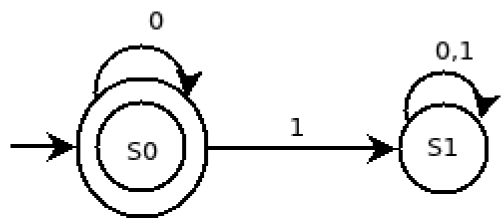


i) {w|every odd position of w is a 1}



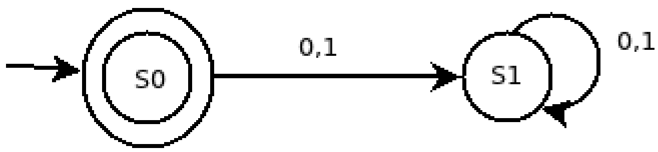
j) Not included in this solution sheet.

k) {empty,0}

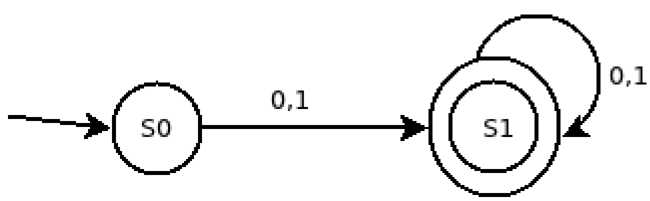


l) Not included in this solution sheet.

m) The empty string.

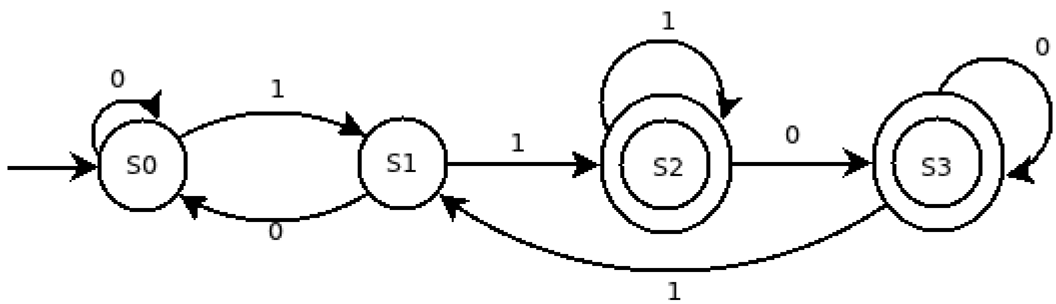


n) Any string except the empty string.

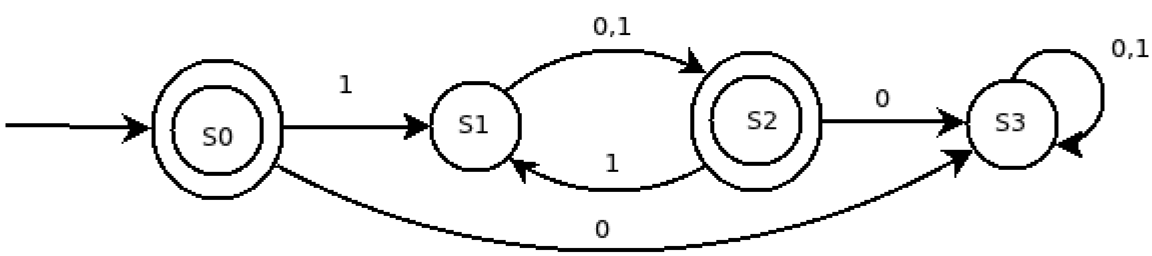


2. For each of the following regular expressions, draw a DFA recognizing the corresponding language.

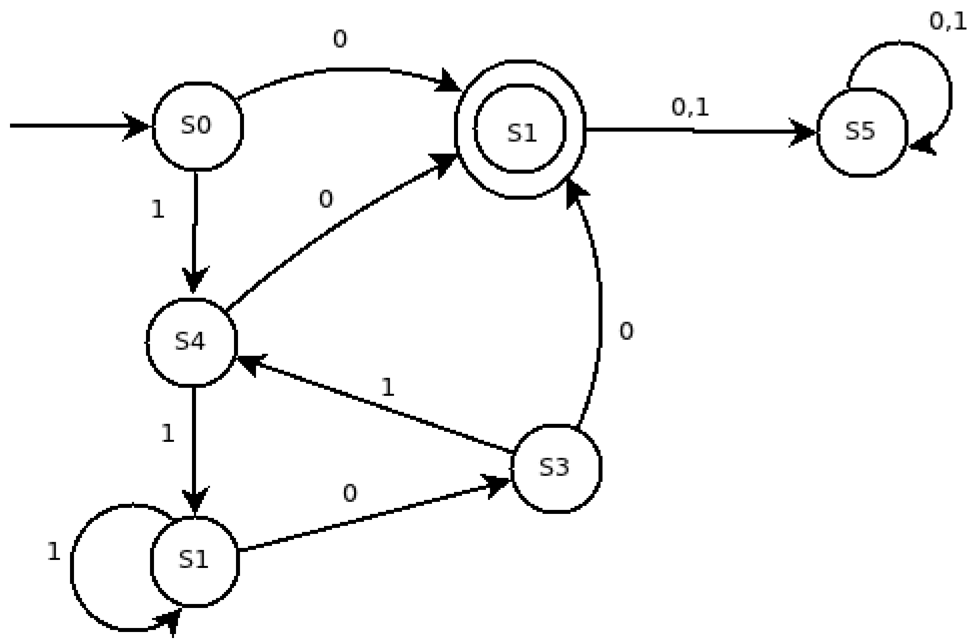
   a) (0 U 1)*110*



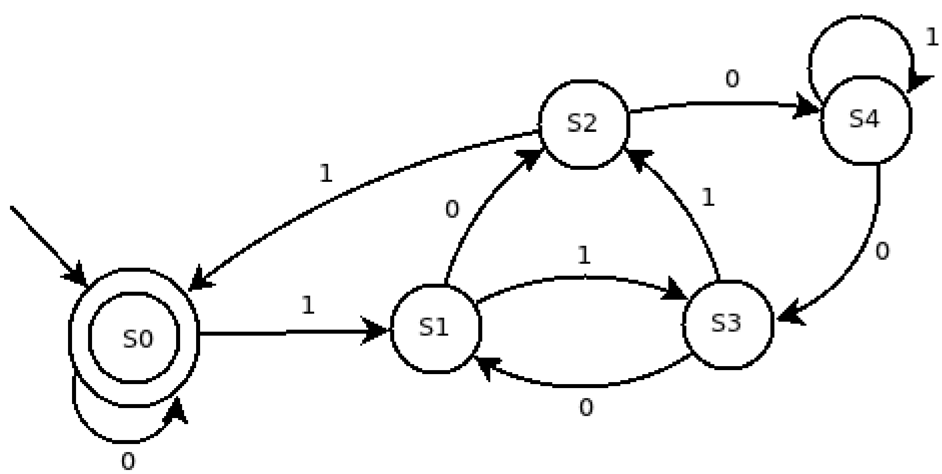
   b) (11 U 10)*



   c) (1 U 110)*0



3. Not included in this solution sheet.
4.



Nondeterministic Finite Automata

Exercise 1.11

Given a NFA $M_1=(Q_1, \Sigma_1, S_{01}, F_1, \delta_1)$ construct a NFA $M_2=(Q_2, \Sigma_2, S_{02}, F_2, \delta_2)$ as follows (given that $F_1=\{f_1, f_2, f_3, \ldots, f_n\}$).

$Q_2=Q_1$

$\Sigma_2=\Sigma_1$

$S_{02}= S_{01}$

$F_2=\{f_k\}$, $1\leq k\leq n$

$\delta_2=\delta_1$ plus the mappings $\delta(f, \boldsymbol{\varepsilon})=\{f_k\}$, for every $f\in F_1$ and $f\notin F_2$.

Exercise 1.14

a) Given a NFA $M_1=(Q_1, \Sigma_1, S_{01}, F_1, \delta_1)$ if the accept states are swapped the result is a new NFA $M_2=(Q_2, \Sigma_2, S_{02}, F_2, \delta_2)$ with the following definitions:

$Q_2=Q_1$

$\Sigma_2=\Sigma_1$

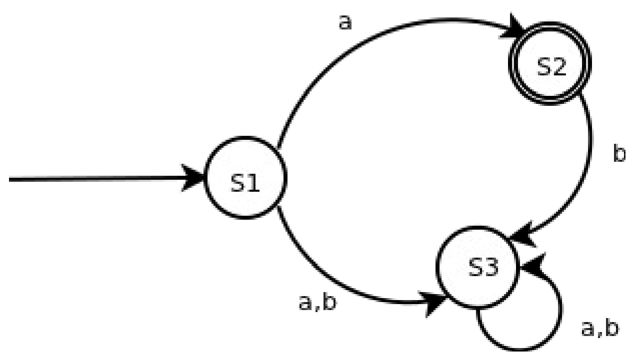
$S_{02}= S_{01}$

$F_2=Q_1-F_1$

$\delta_2=\delta_1$

The fact that the set of accept states of M_2 (F_2) is the complement of F_1 it guarantees that M_2 accepts only those words that are not accepted by M_1, i.e., the complement language of M_1.

b) The following NFA recognises the language $L=\{w|w \text{ is } a\}$.



The following machine was constructed by swapping the accept states of the previous but it doesn't accept the language `L={w|w is not a}`.