CSE-233 : Section A

Summer 2020

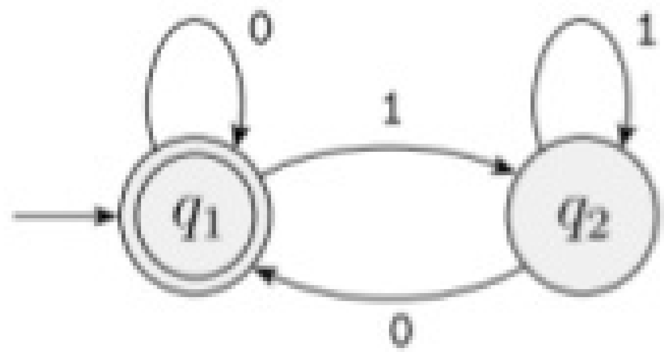
Introduction to Finite Automata

Reference:

Book2 Chapter 1.1

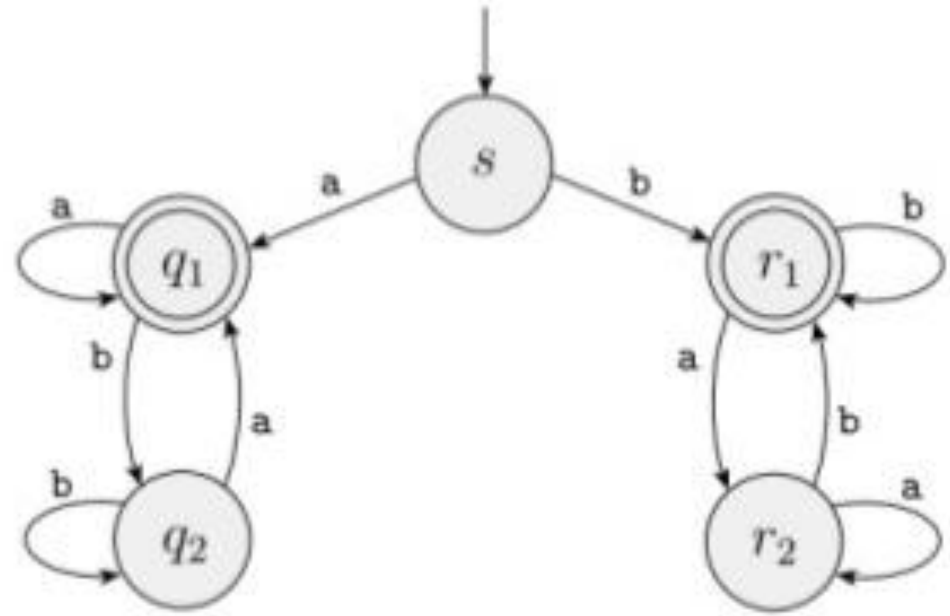
Md. Saidul Hoque Anik
anik@cse.uiu.ac.bd

Previous Class (1)



1. States = ?
2. Accepted Inputs (Alphabets)?
3. Initial/Starting State = ?
4. Transition Function = ?
5. Final State = ?
6. Language = ?

Previous Class (2)



1. States = ?
2. Accepted Inputs (Alphabets)?
3. Initial/Starting State = ?
4. Transition Function = ?
5. Final State = ?
6. Language = ?

Note

How a DFA Processes Strings

Let us build an automaton that accepts the words that contain 01 as a subword

$\Sigma = \{0, 1\}$

$L = \{x01y \mid x, y \in \Sigma^*\}$

We use the following states

A: start

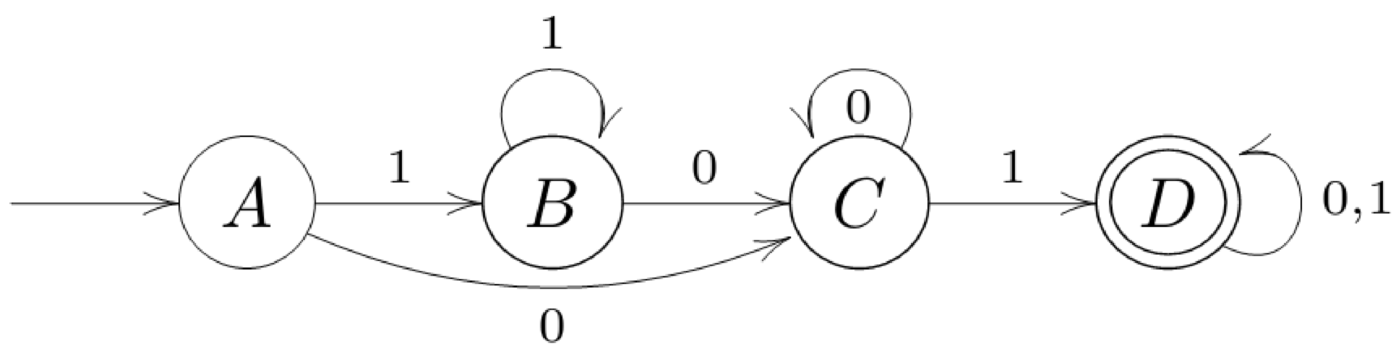
B: the most recent input was 1 (but not 01 yet)

C: the most recent input was 0 (so if we get a 1 next we should go to the accepting state D)

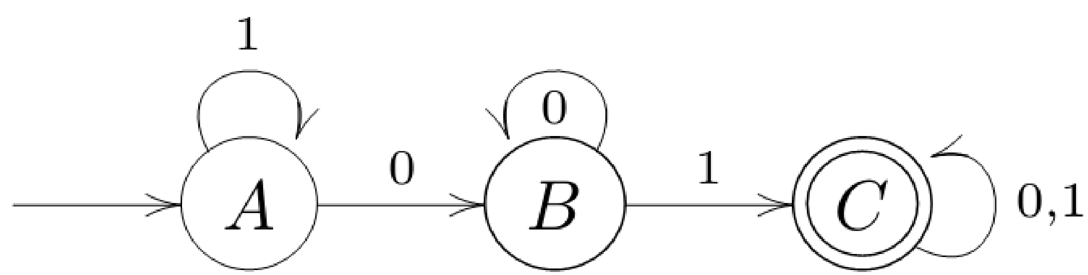
D: we have encountered 01 (accepting state)

Minimization

The same language may be represented by different DFA



and



Making a useful machine

EXAMPLE 1.13

Figure 1.14 shows machine M_5, which has a four-symbol input alphabet, $\Sigma = \{\langle\text{RESET}\rangle, 0, 1, 2\}$. We treat $\langle\text{RESET}\rangle$ as a single symbol.

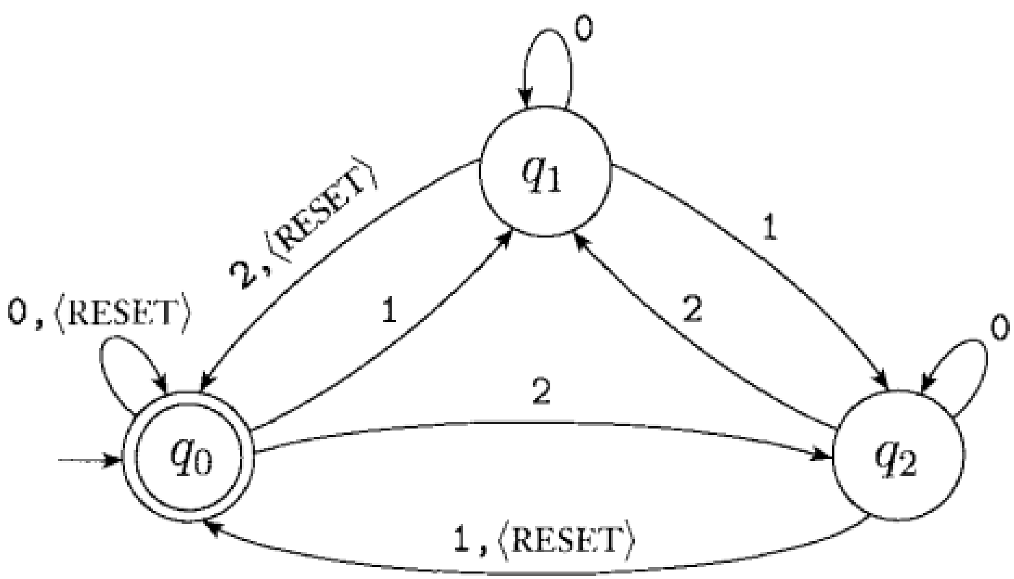
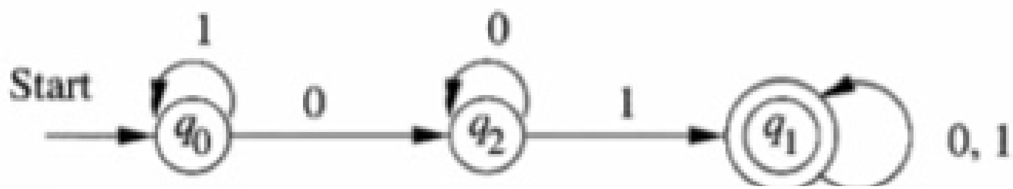
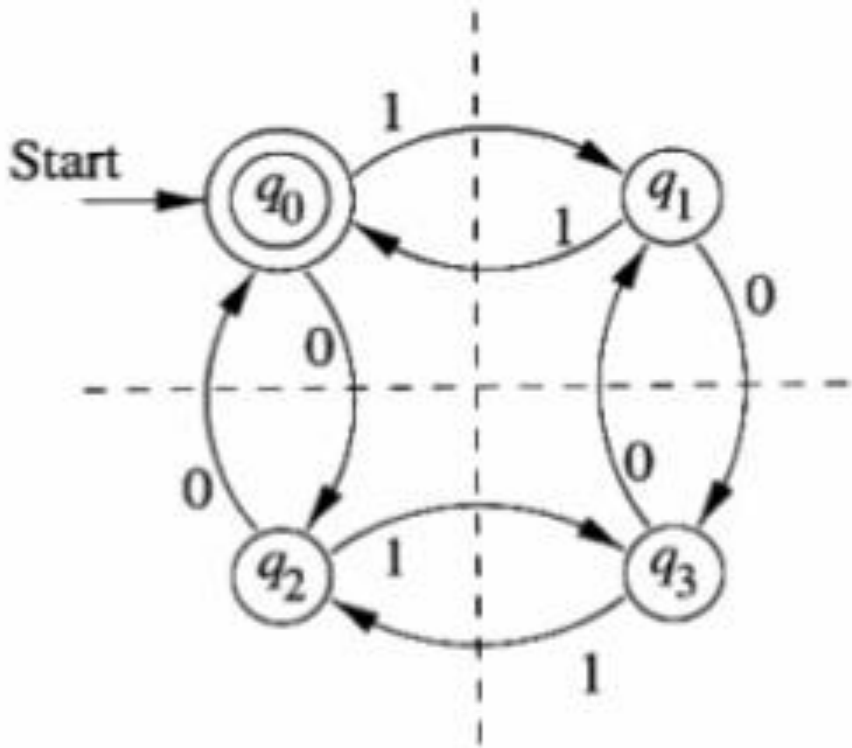


FIGURE 1.14
Finite automaton M_5

Can you tell what this machine does?

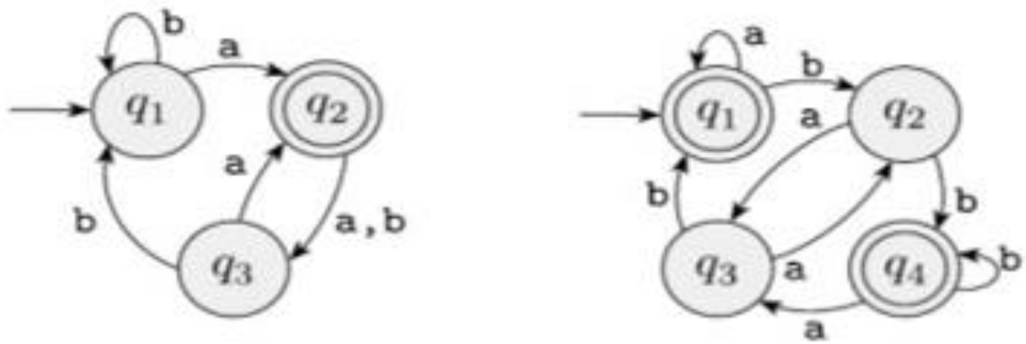
Some more examples





Sample Question

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?

1.2 Give the formal description of the machines M_1 and M_2 pictured in Exercise 1.1.

Sample Question

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

Designing an DFA

Typical Problem

Problem
Given a language L, design a DFA M that accepts L, i.e., $\mathbf{L}(M) = L$.

Methodology

- Imagine yourself in the place of the machine, reading symbols of the input, and trying to determine if it should be accepted.
- Remember at any point you have only seen a part of the input, and you don't know when it ends.
- *Figure out what to keep in memory.* It cannot be all the symbols seen so far: it must fit into a finite number of bits.

Designing an DFA

Strings containing 0

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that contain at least one 0.

Designing an DFA

Strings containing 0

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that contain at least one 0.

Solution
What do you need to remember? Whether you have seen a 0 so far or not!

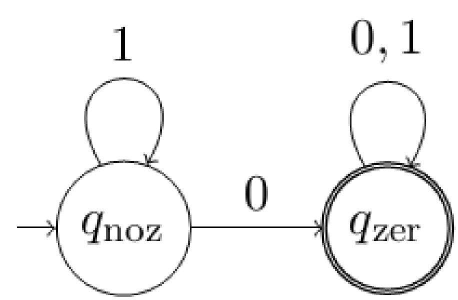


Figure 10: Automaton accepting strings with at least one 0.

Designing an DFA (2)

Even length strings

Problem

Design an automaton that accepts all strings over $\{0, 1\}$ that have an even length.

Designing an DFA (2)

Even length strings

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that have an even length.

Solution
What do you need to remember? Whether you have seen an odd or an even number of symbols.

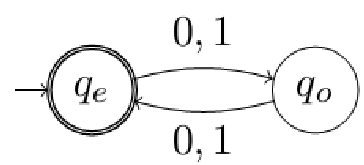


Figure 11: Automaton accepting strings of even length.

Designing an DFA (3)

Pattern Recognition

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that have 001 as a substring, where u is a substring of w if there are w_1 and w_2 such that $w = w_1 u w_2$.

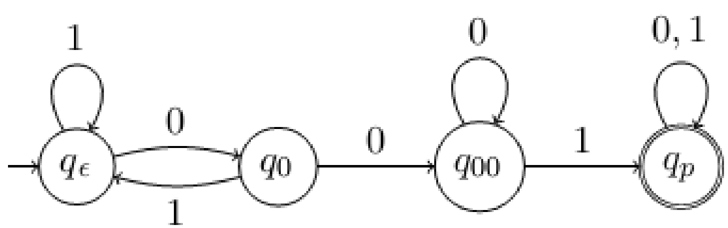
Designing an DFA (3)

Pattern Recognition

Problem
Design an automaton that accepts all strings over $\{0, 1\}$ that have 001 as a substring, where u is a substring of w if there are w_1 and w_2 such that $w = w_1 u w_2$.

Solution
What do you need to remember? Whether you

- haven't seen any symbols of the pattern
- have just seen 0
- have just seen 00
- have seen the entire pattern 001



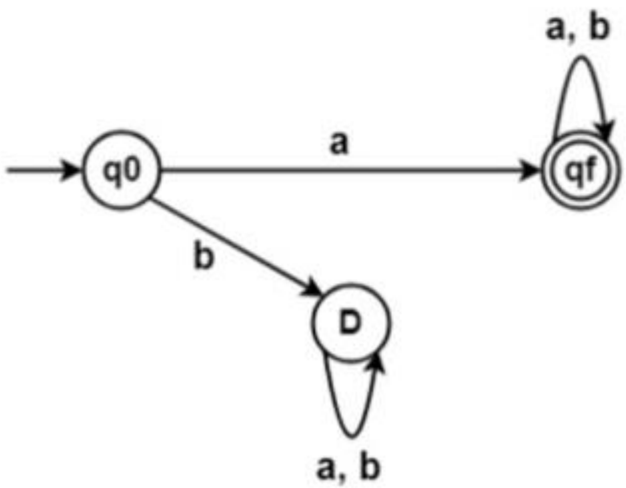
Automaton accepting strings having 001 as substring.

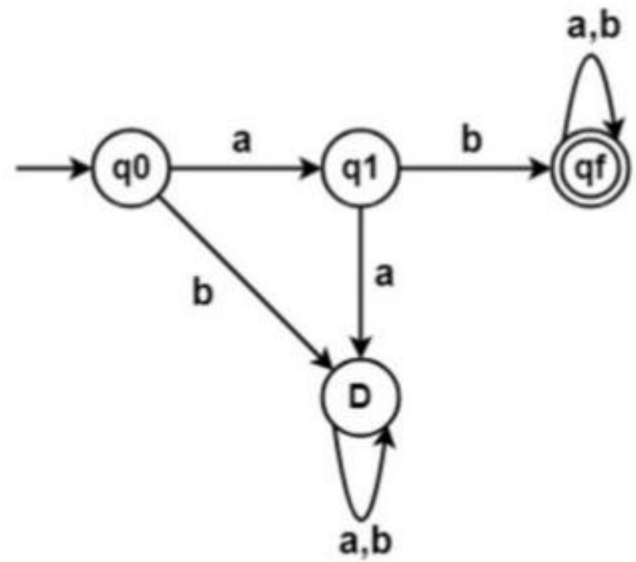
More Design Examples (Set – A)

Draw a DFA for the language accepting strings-

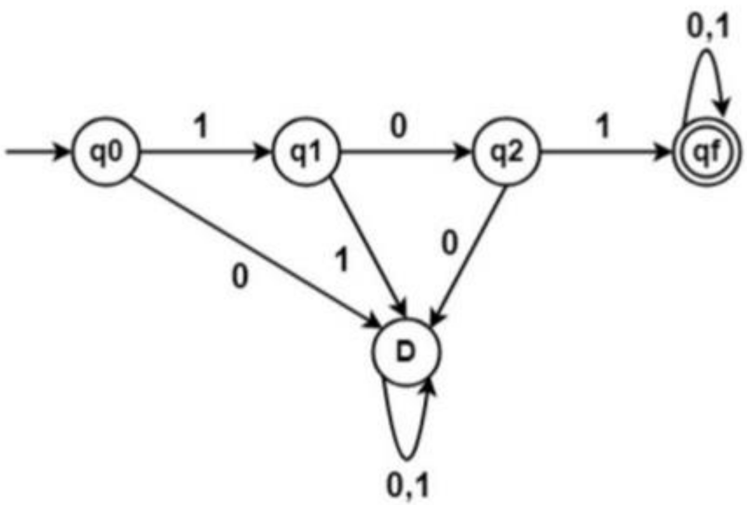
1. Starting with ‘a’ over input alphabet = {a, b}
2. Starting with ‘ab’ over input alphabet = {a, b}
3. Starting with ‘101’ over input alphabet = {0, 1}
4. Starting with ‘00’ over input alphabet = {0, 1}
5. Starting with ‘aba’ over input alphabet = {a, b}

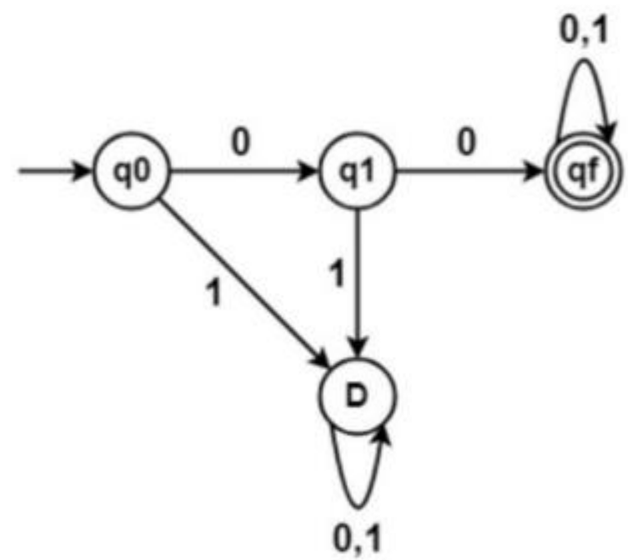
Solution (Set A)



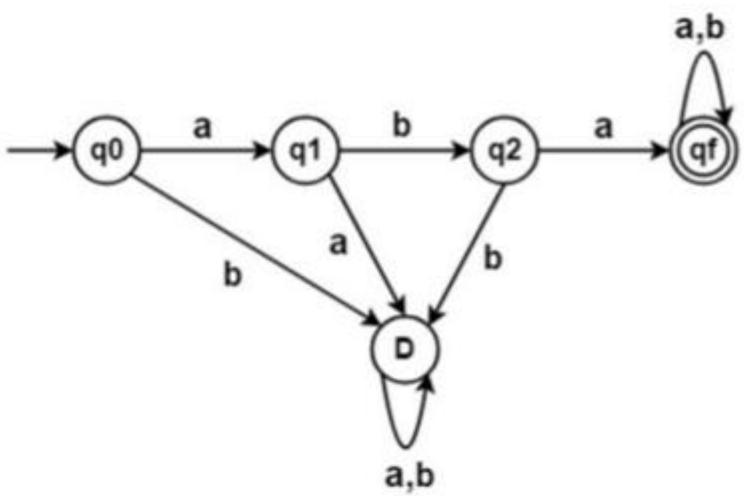


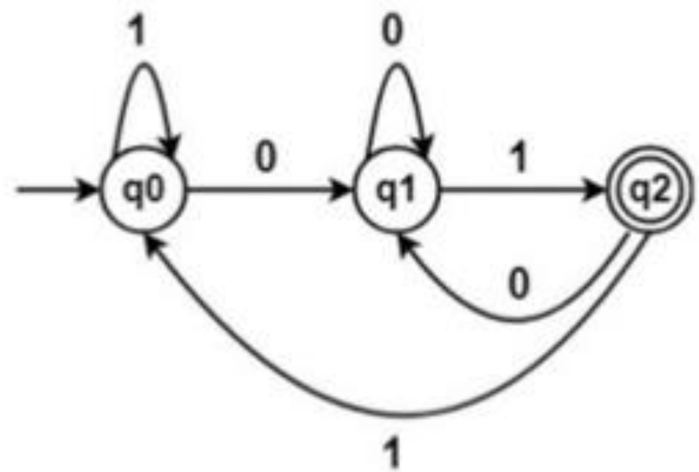
Solution (Set A)





Solution (Set A)



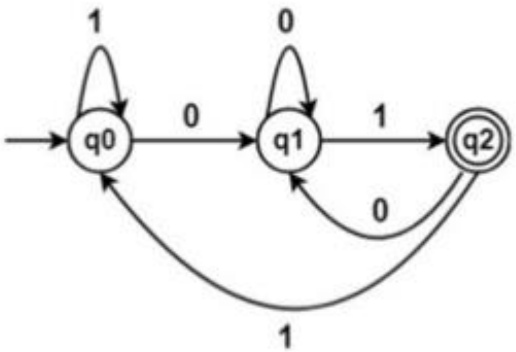


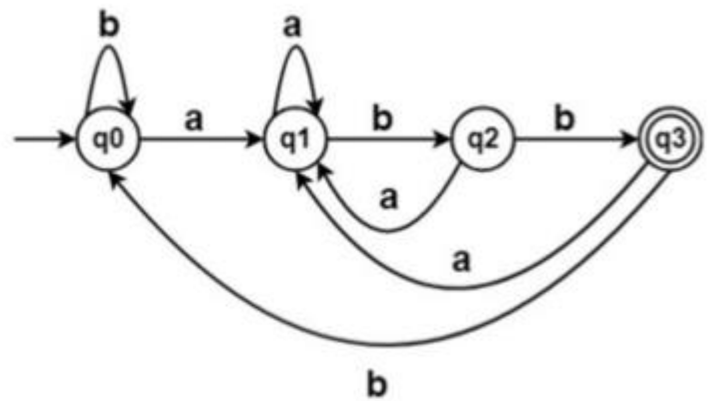
More Design Examples (Set – B)

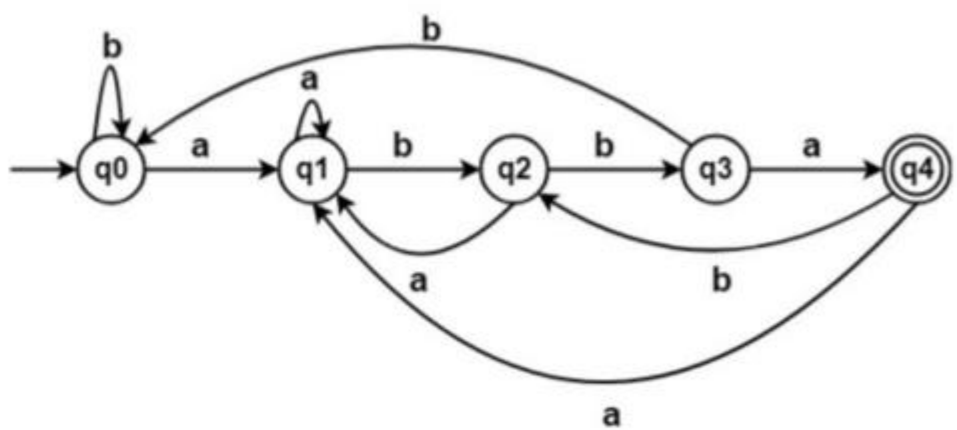
Draw a DFA for the language accepting strings-

1. Ending with ‘01’ over input alphabet = {0, 1}
2. Ending with ‘abb’ over input alphabet = {a, b}
3. Ending with ‘abba’ over input alphabet = {a, b}
4. Starting with ‘0011’ over input alphabet = {0, 1}

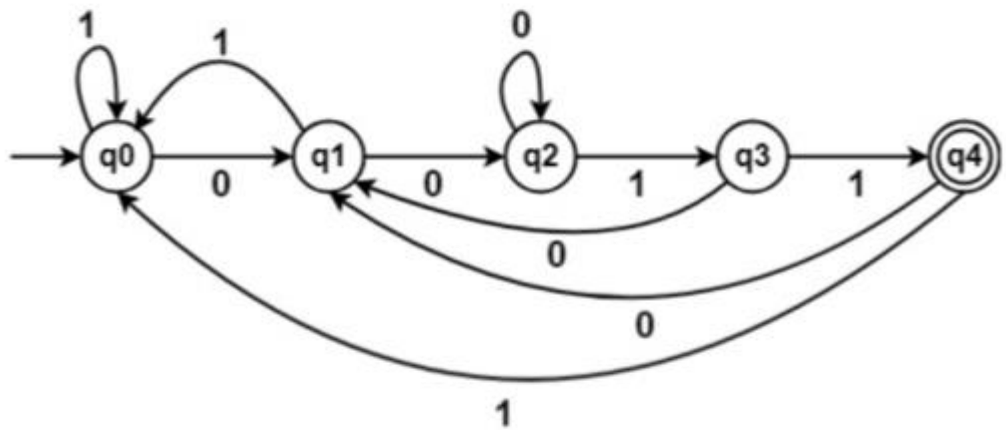
Solution (Set B)







Solution (Set B)



More Design Examples (Set – C)

Draw a DFA for the language accepting strings-

1. With substring ‘aab’ over input alphabet = {a, b, c}
2. With substring ‘aababb’ over input alphabet = {a, b, c}

More Design Examples (Set – D)

Draw a DFA for the language accepting strings-

1. Containing any number of 0’s, 1’s or empty string over alphabet {0, 1}