DFA

CSE 2233

Sabrina Jahan Maisha
Lecturer, Department of Computer Science & Engineering
International Islamic University Chittagong.

Regular Languages

Language:

A set of strings all of which are chosen from some Σ^*, where Σ is a particular alphabet, is called a language.

Regular Language:

These languages are exactly the ones that can be *described/recognized by Finite Automata*. This language can be processed by computers having a very small amount of memory.

Here,

- Computer is meant by a *computational model* that varies depending on the feature we want to focus on.
- Finite automata are good models for computers with an extremely limited amount of memory.

Finite Automata (Deterministic)

Finite Automata:

A *finite automaton (FA)* is a simple idealized machine used to *recognize patterns* within input taken from some character set (or alphabet) $\sum$. The job of an FA is to *accept* or *reject* an input depending on whether the pattern defined by the FA occurs in the input.

Graphical Representation:

- Nodes (for states)
- Arcs (for transitions)

We execute our FA on an input sequence as follows:

1) Begin in the start state
2) If the next input char matches the label on a transition from the current state to a new state, go to that new state
3) Continue making transitions on each input char
   i. If no move is possible, then stop
   ii. If in accepting state, then accept

Finite Automata (Deterministic)

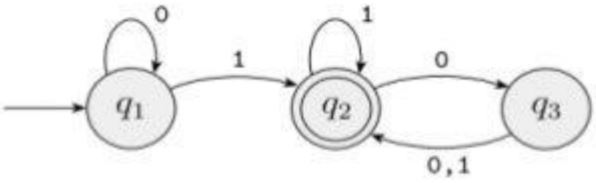
A finite automaton is a 5 tuple $(Q, \Sigma, \delta, q_0, F)$ where,

- Q is a finite set called the states
- Σ is a finite set called the alphabet
- $\delta : Q \times \Sigma \rightarrow Q$ is the transition function
- $q_0 \in Q$ is the start state and
- $F \subseteq Q$ is the set of accept states

Types of Finite Automata

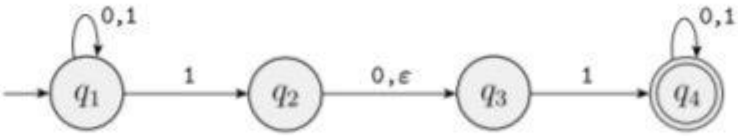
DFA:

The term “deterministic finite automata” refers to the fact that on each input there is one and only one state to which the automaton can transition from its current state.

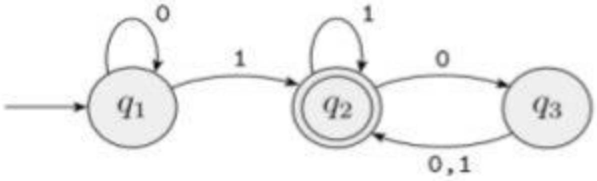


NFA:

“Nondeterministic finite automata” refers to the fact that on each input the automaton can transit to several states at once.



FA - Machine M



We can describe M_1 formally by writing $M_1 = (Q, \Sigma, \delta, q_1, F)$, where

1. $Q = \{q_1, q_2, q_3\}$,
2. $\Sigma = \{0,1\}$,
3. δ is described as

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2,

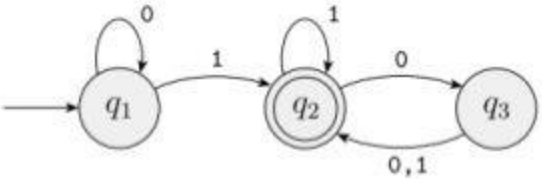
4. q_1 is the start state, and
5. $F = \{q_2\}$.

FA - Machine M

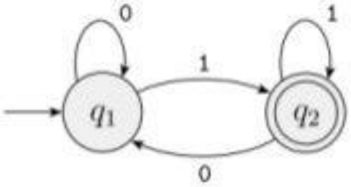
- If A is the set of all strings that machine M accepts, we say that A is the language of machine M.

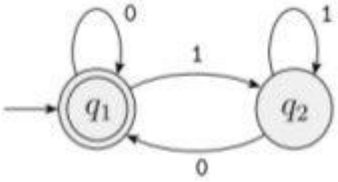
  L(M) = A *i.e.* M recognizes A
- A machine may accept several strings, but it always recognizes only one language.
- If the machine accepts no strings, it still recognizes one language- the empty language
- For this example, if L(M) = A, then

  A = { *w* | *w* contains at least one 1 and an even number of 0s follow the last 1 }



FA - Machine M1, M2

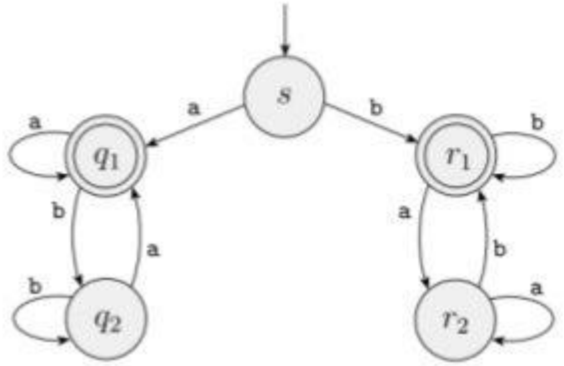


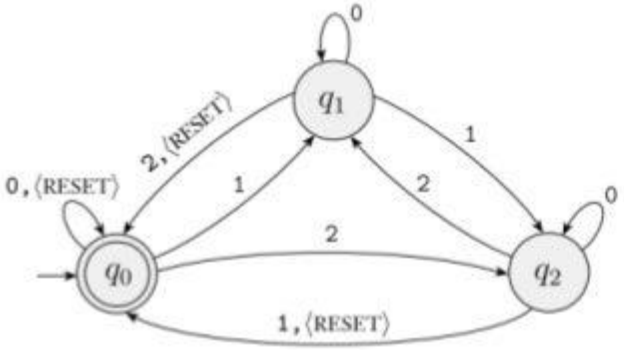


- Q = ?
- Σ =?
- δ = ?
- Start state = ?
- Final state, F= ?
- L(M1) = ?
- Acceptability check: 01011

- Q = ?
- Σ =?
- δ = ?
- Start state = ?
- Final state, F= ?
- L(M2) = ?
- Acceptability check: 01100

FA - Machine M3, M4

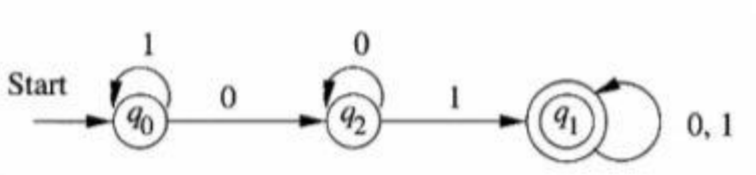




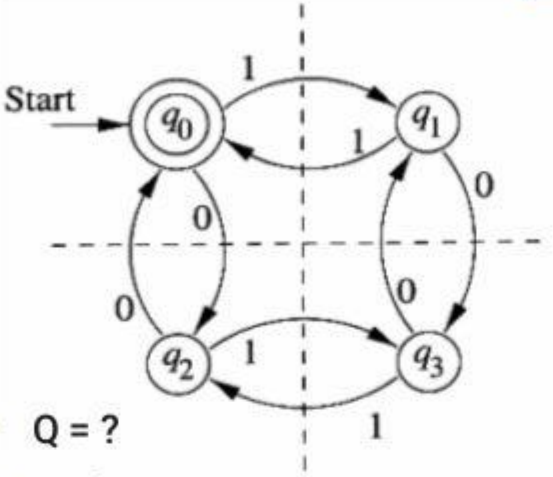
- Q = ?
- Σ =?
- δ = ?
- Start state = ?
- Final state, F= ?
- L(M3) = ?
- Acceptability check: abba, baba

- Q = ?
- Σ =?
- δ = ?
- Start state = ?
- Final state, F= ?
- L(M4) = ?
- Acceptability check: 120120

FA - Example M5, M6



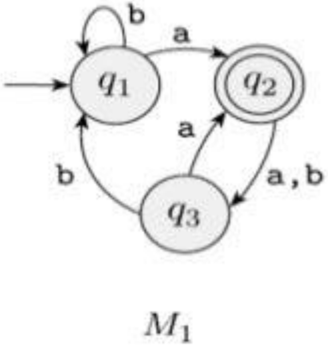
- Q = ?
- Σ =?
- δ = ?
- Start state = ?
- Final state, F= ?
- L(M5) = ?
- Acceptability check: 10100

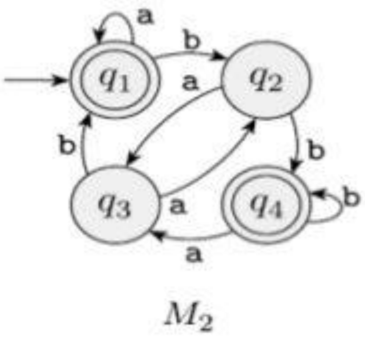


- Q = ?
- Σ =?
- δ = ?
- Start state = ?
- Final state, F= ?
- L(M6) = ?
- Acceptability check: 10011010

FA - Practice 1

1. Consider the following two DFA's, M1 and M2. Now answer the following questions about each of the machines:





- What is the start state?
- What is the set of accept states?
- What sequence of states does the machine go through on input 'aabb' ?
- Does the machine accept the string 'aabb' ?
- Does the machine accept the string ε ?
- Give the formal description of the machines M1 and M2.

FA - Practice 2

2. The formal description of a DFA is $(\{q_1, q_2, q_3, q_4, q_5\}, \{u, d\}, \delta, q_3, \{q_3\})$
Where δ is represented by the following transition table:

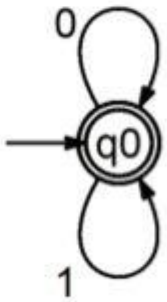
	u	d
q_1	q_1	q_2
q_2	q_1	q_3
q_3	q_2	q_4
q_4	q_3	q_5
q_5	q_4	q_5

- Draw the State diagram of this machine

DFA Machine Design

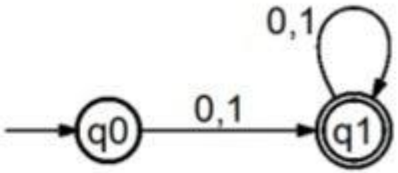
Machine - 1.1

- Give a DFA for Σ = { 0, 1 } that accepts any string containing any number of 0's, 1's or empty string.



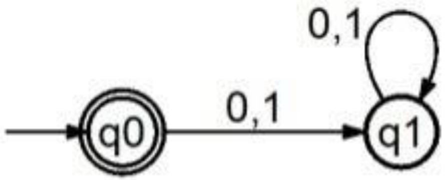
Machine - 1.2

- Give a DFA for Σ = { 0, 1 } that accepts any string containing any number of 0's, 1's except empty string.



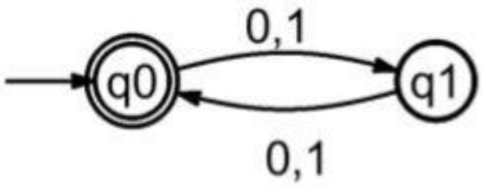
Machine – 1.3

- Give a DFA for $\Sigma = \{ 0, 1 \}$ that accepts only the empty string.



Machine – 2.1

- Give a DFA for Σ = { 0, 1 } that accepts only even length strings (including empty string).

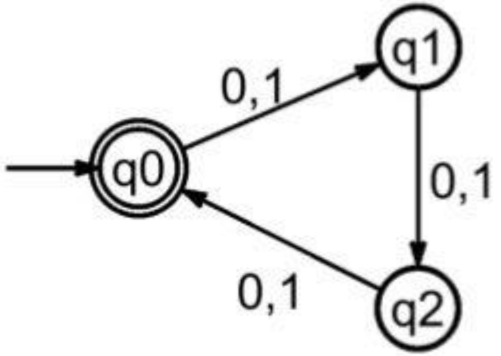


Machine – 2.2

L(M) = { w | w has odd length }

Machine – 2.3

– Give a DFA for Σ = { 0, 1 } that accepts only those strings(including empty string) whose length is a multiple of 3.

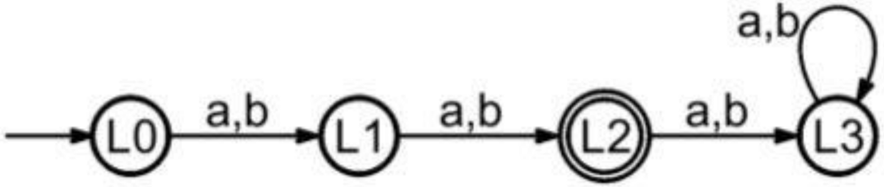


Machine – 2.4
L(M) = { w | the length of w *mod* 3 = 1 }

Machine – 2.5
L(M) = { w | the length of w *mod* 4 = 0 }

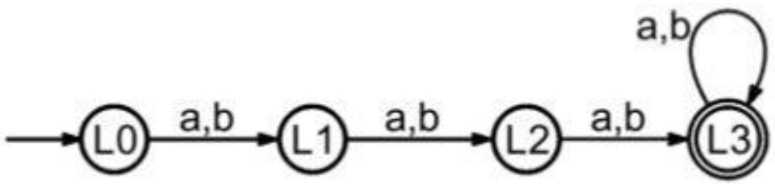
Machine – 3.1

- Give a DFA for Σ = { a, b } and strings that have exact length 2



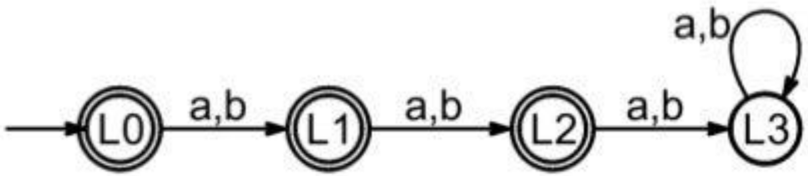
Machine – 3.2

- Give a DFA for $\Sigma = \{ a, b \}$ and strings that have length at least 3



Machine – 3.3

– Give a DFA for Σ = { a, b } and strings that have length at most 2

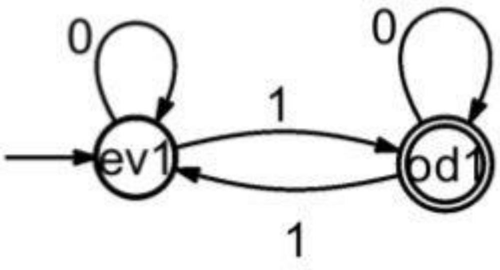


Machine – 3.4
What about length at least 2

Machine – 3.5
What about length at most 5

Machine – 4.1

- Give a DFA for Σ = { 0, 1 } and strings that have an odd number of 1's and any number of 0's

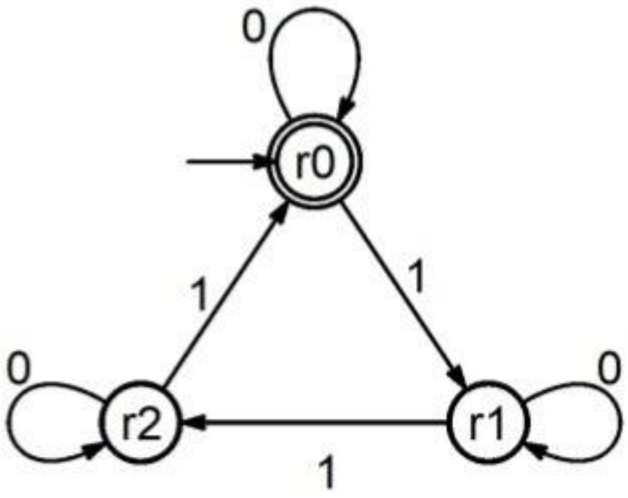


Machine – 4.2
What about empty string or, even no of ones and any number of 0's?

Machine – 4.3
What about odd no of 0's and any number of 1's?

Machine – 4.4

– Give a DFA for Σ = { 0, 1 } and strings that contain any number of 0's and the total number of 1's is a multiple of 3.

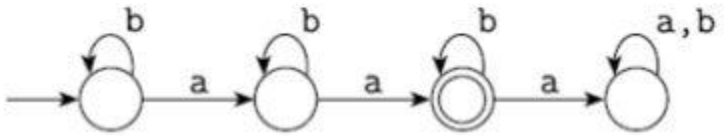


Machine – 4.5

What about strings with number of 0's is a multiple of 4 and any number of 1's

Machine – 5.1

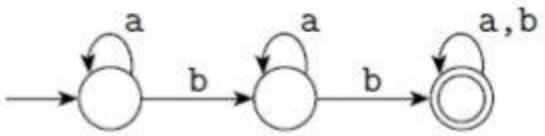
- Give a DFA for Σ = { a, b } and strings that have exactly 2 a's



Machine – 5.2
What about exactly 3 b's

Machine – 5.3

- Give a DFA for Σ = { a, b } and strings that have at least 2 b's



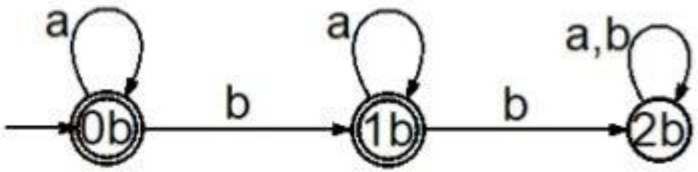
Machine – 5.4

What about at least 3 a's

Machine – 5.5

– Give a DFA for Σ = { a, b } and strings that have at most 1 b

- That is the complement of at least 2 b's.

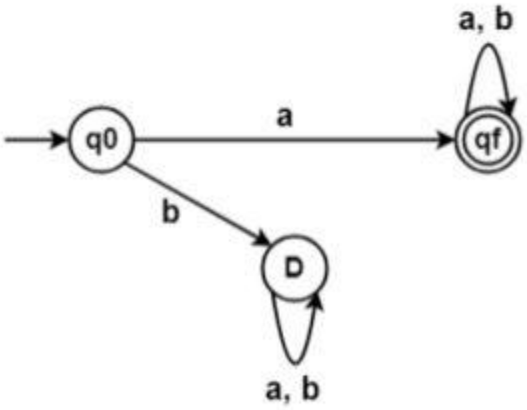


Machine – 5.6

What about at most 2 a's

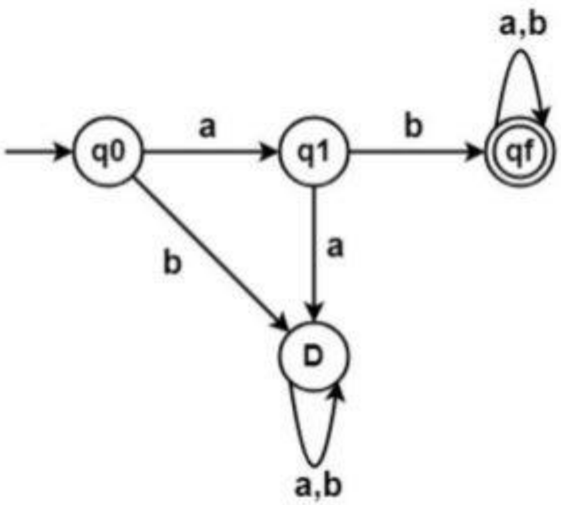
Machine – 6.1

- Draw a DFA for the language accepting strings starting with ‘a’ over input alphabets Σ = { a, b }



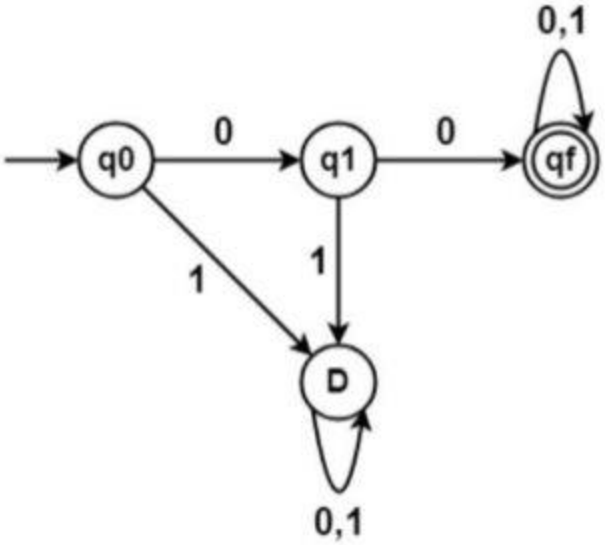
Machine - 6.2

- Draw a DFA for the language accepting strings starting with ‘ab’ over input alphabets Σ = { a, b }



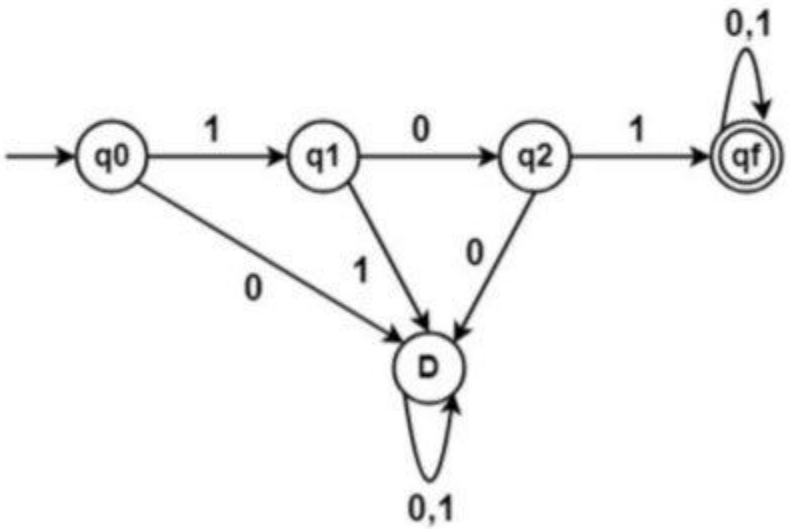
Machine – 6.3

– Draw a DFA that accepts a language L over input alphabets Σ = { 0, 1 } such that L is the set of all strings starting with '00'



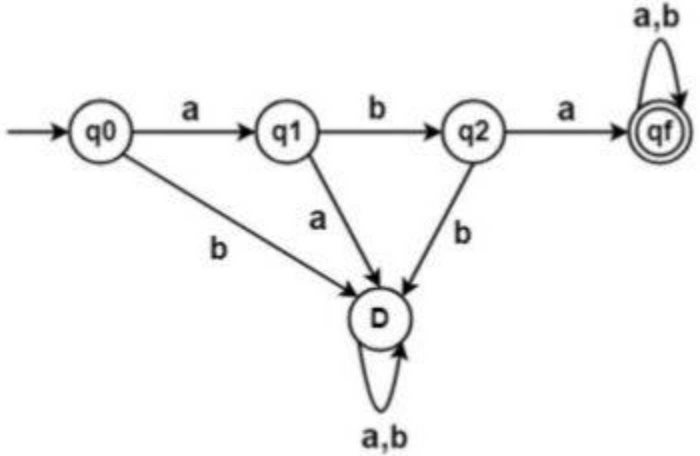
Machine - 6.4

- Draw a DFA for the language accepting strings starting with '101' over input alphabets $\Sigma = \{ 0, 1 \}$



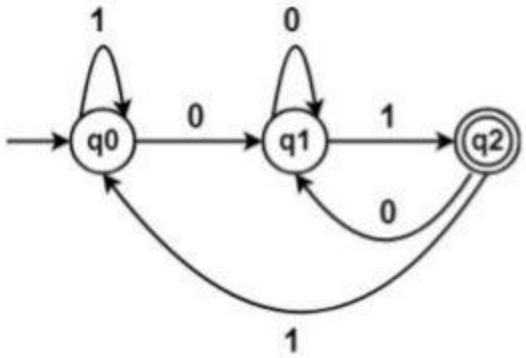
Machine – 6.5

– Construct a DFA that accepts a language L over input alphabets Σ = { a, b } such that L is the set of all strings starting with 'aba'



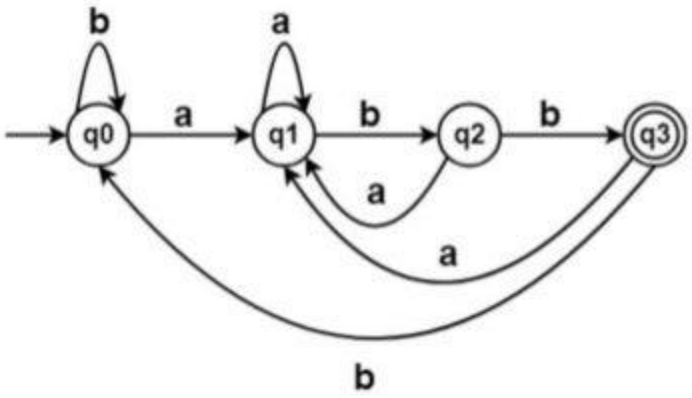
Machine – 7.1

– Draw a DFA for the language accepting strings ending with ‘01’ over input alphabets Σ = { 0, 1 }



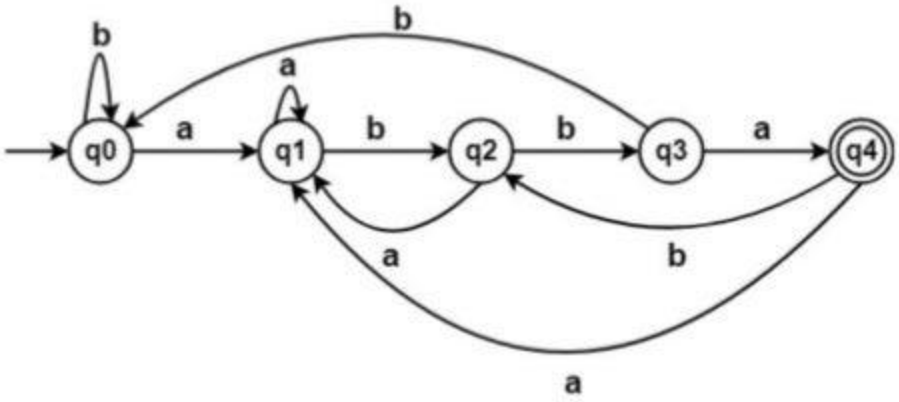
Machine - 7.2

- Draw a DFA for the language accepting strings ending with ‘abb’ over input alphabets Σ = { a, b }



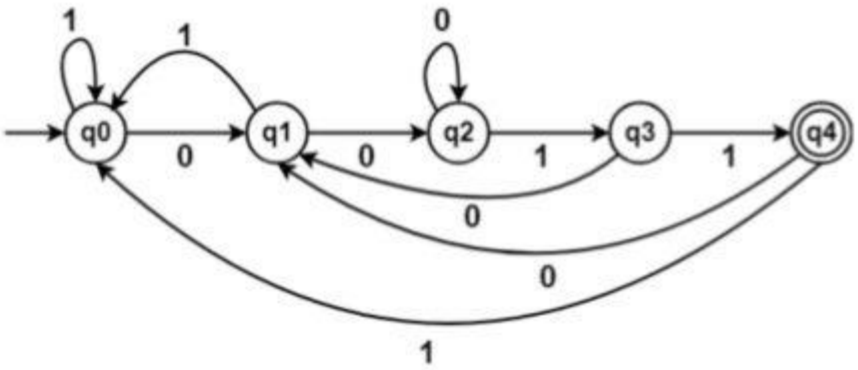
Machine – 7.3

- Draw a DFA for the language accepting strings ending with ‘abba’ over input alphabets Σ = { a, b }



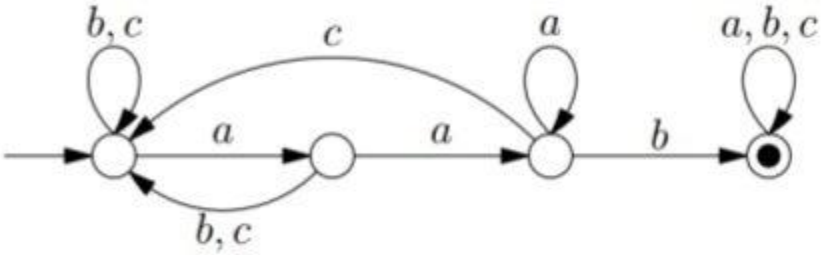
Machine – 7.4

– Draw a DFA for the language accepting strings ending with '0011' over input alphabets Σ = { 0, 1 }



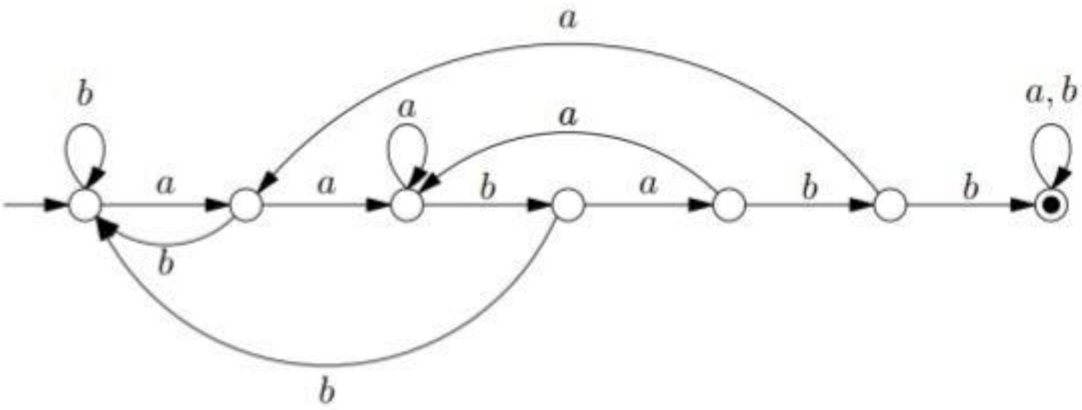
Machine – 8.1

- Give a DFA for Σ = { a, b, c } that accepts any string with 'aab' as a substring.



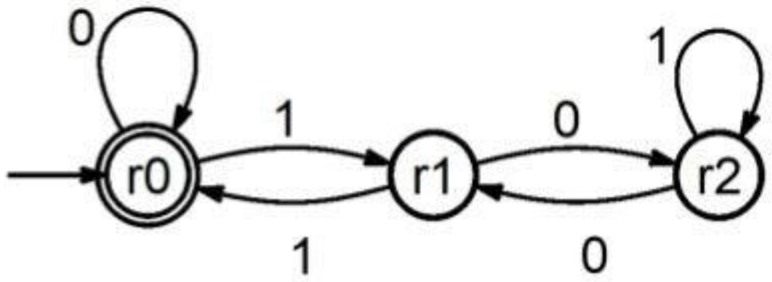
Machine - 8.2

- Give a DFA for Σ = { a, b } that accepts any string with ‘aababb’ as a substring.



Machine – 9.1

– Give a DFA for Σ = { 0, 1 } and only accepts binary strings those are a multiple of 3

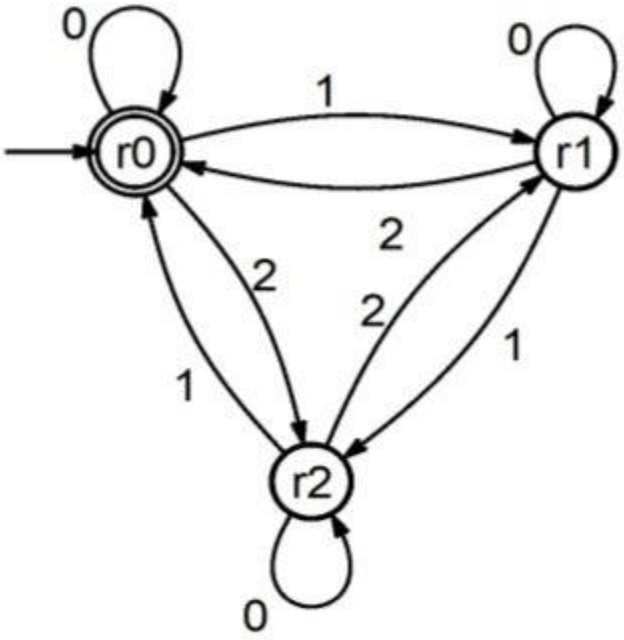


Machine – 9.2

What about multiple of 4

Machine – 10.1

– Give a DFA for Σ = { 0, 1, 2 } and sum of the numeric digits(symbols) of the strings is a multiple of 3.

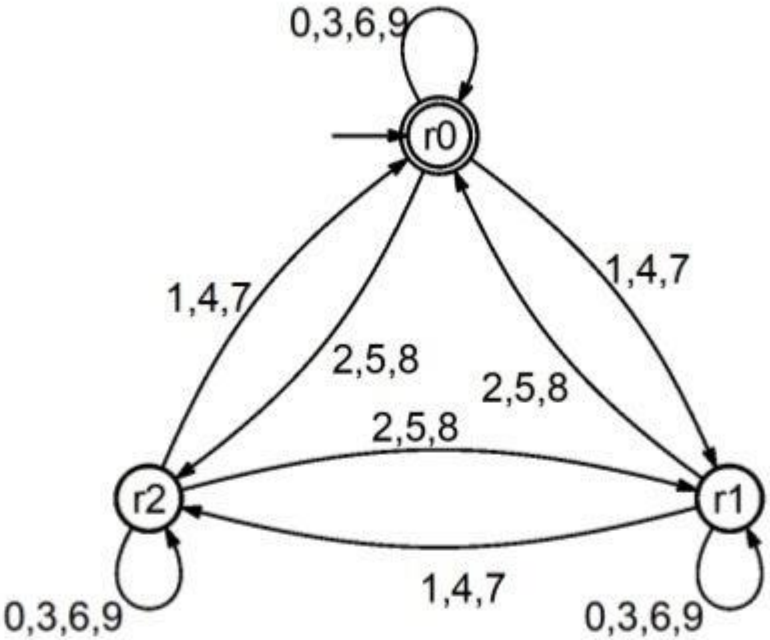


Machine – 10.2

What about multiple of 4

Machine – 11.1

– Give a DFA for Σ = { 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 } that accepts strings whose decimal equivalent is a multiple of 3

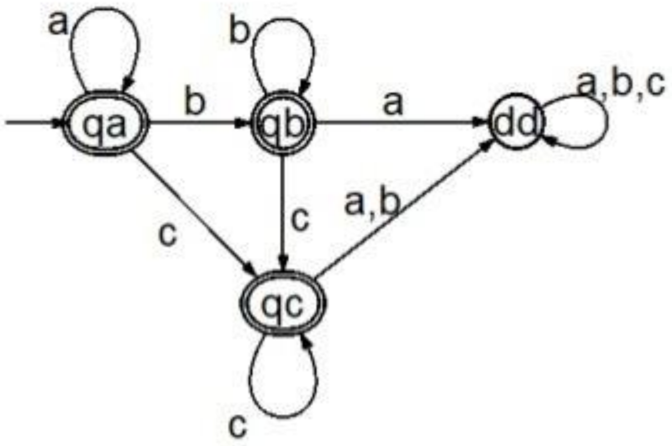


Machine – 11.2

What about multiple of 2

Machine – 12.1

- Give a DFA for Σ = { a, b, c } and recognizes strings having patterns = { $a^n b^m c^l$ | n,m,l >= 0 }

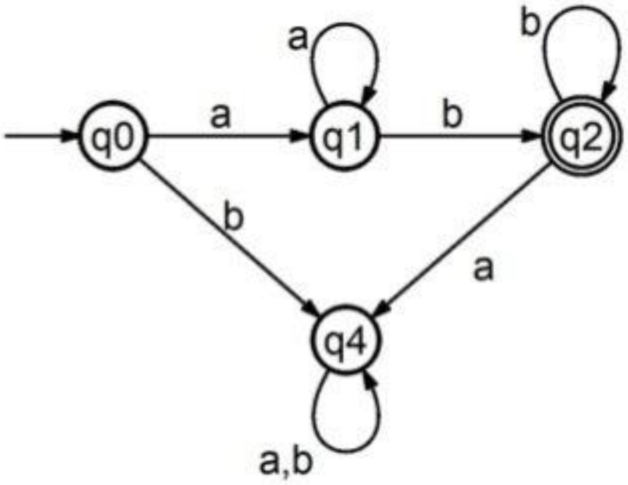


Machine – 12.2

What about { $a^n b^m$ | n,m >= 0 } over {a,b}

Machine – 12.3

- Give a DFA for Σ = { a, b, c } and recognizes strings having patterns = { $a^n b^m$ | n,m > 0 }

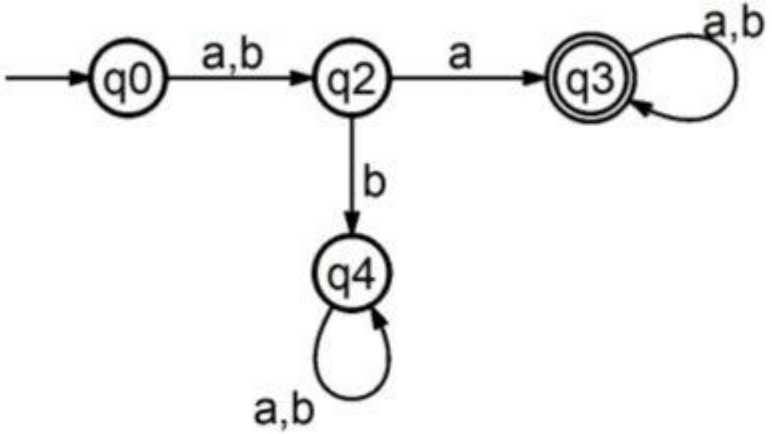


Machine – 12.4

What about { $a^n b^m c^l$ | n,m,l > 0 }

Machine - 13.1

- Give a DFA for Σ = { a, b } and the second symbol from the left side is ‘a’

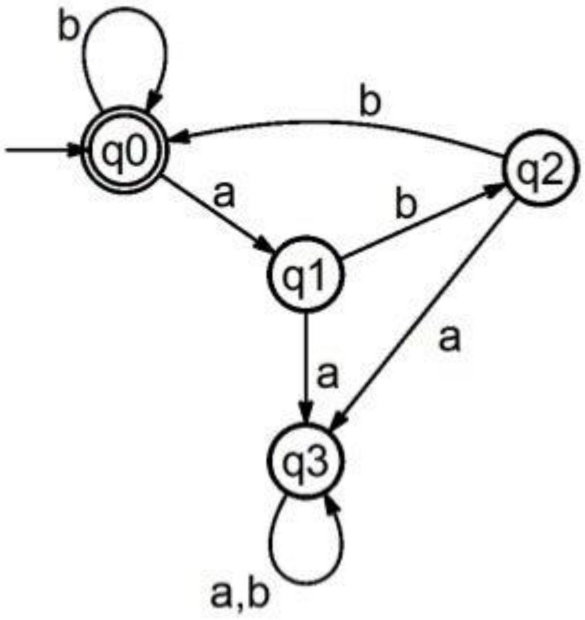


Machine – 13.2

What about third symbol from the left is ‘a’

Machine – 14.1

- Give a DFA for Σ = { a, b } and contains strings where each ‘a’ is followed by ‘bb’



Machine – 14.2

What about each ‘b’ followed by a ‘a’

Machine – 15.1

– Give a DFA for Σ = { a, b } and recognizes strings having patterns = { $a^n b^m$ | n+m = even }

Sol^n >> Class Lecture

Machine – 15.2

What about { $a^n b^m$ | n+m=odd } over {a,b}

THANKS!

Any Question ??

References:
Chapter 1(section 1.1), Introduction to the Theory of Computation, 3rd Edition by Michael Sipser

Acknowledgments ....

Imam Hossain , Lecturer, UIU

References:

Chapter 1(section 1.1), Introduction to the Theory of Computation, 3rd Edition by Michael Sipser