Finite Automata (FA)

FA:

- Finite automata is a mathematical model which has no memory.

Consists of:

- states

- transition among states

Types:

1. Deterministic Finete Automata (DFA)
2. Non-Deterministic Finite Automata (NFA/NDFA)

DFA:

- For each input, only one transition is possible from the current state.

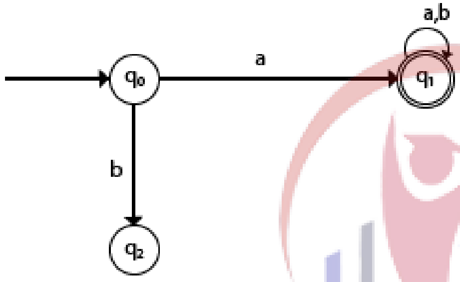


Fig:- DFA

- A DFA can be represented by a 5-tuple (Q, ∑, δ, q_0, F) where:

Q is a finite set of states.

∑ is a finite set of symbols called the alphabet.

δ is the transition function where $\delta: Q \times \sum \rightarrow Q$

$\mathbf{q_0}$ is the initial state from where any input is processed ($q_0 \in Q$).

F is a set of final state/states of Q ($F \subseteq Q$).

Eg: Draw a DFA for the language accepting strings ending with ‘0011’ over input alphabets ∑ = {0, 1}

Solution,

Regular expression for the given language = (0 + 1)*0011

We will construct DFA for the following strings-

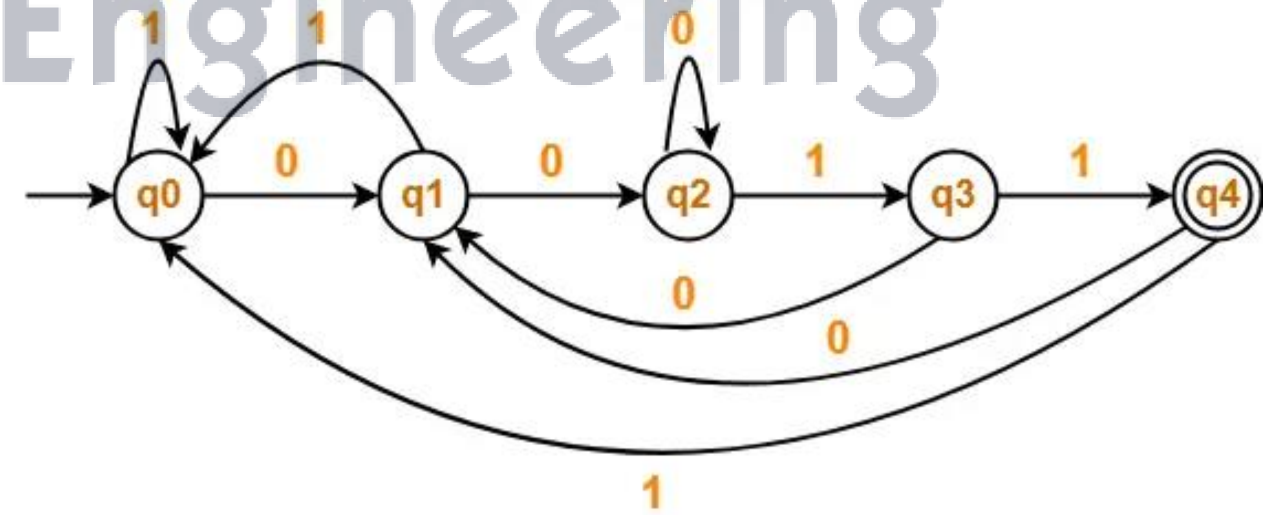
0011

00011

000011

0010011

00110011



DFA

NFA:

- Similar to DFA, For each input, several transitions is possible from the current state
- An NFA can be represented by a 5-tuple M = (Q, Σ, δ, q_0, F) where:

Q = finite set of states

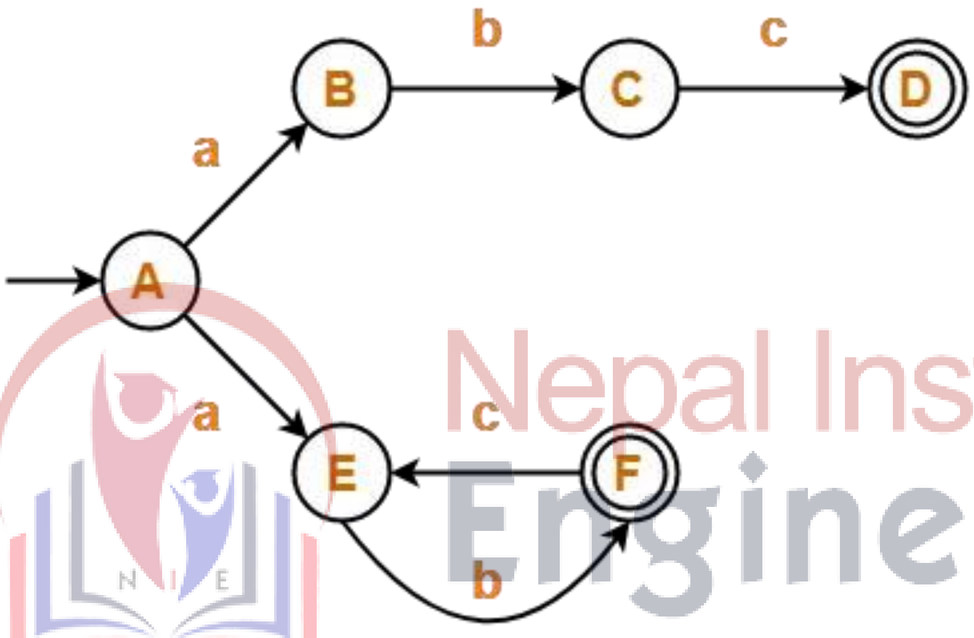
Σ = non-empty finite set of symbols called as input alphabets

$\delta : Q \times \Sigma \rightarrow 2^Q$ is a total function called as transition function

$q0 \in Q$ is the initial state

$F \subseteq Q$ is a set of final states

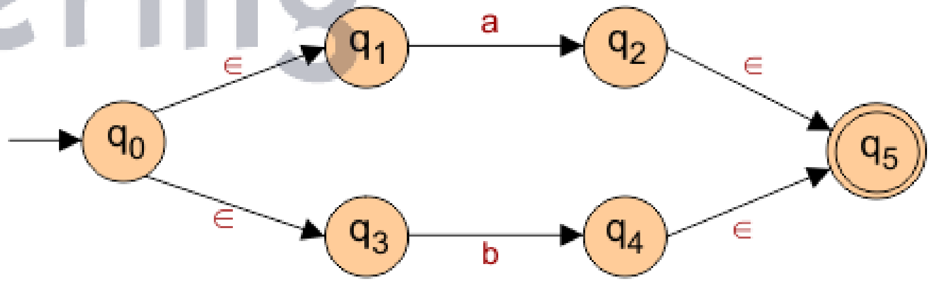
Eg:



Example of Non-Deterministic Finite Automata

(Without Epsilon)

- **e-NFA:** epsilon means empty string; NFA can change its state from one to another without reading any input string is called e-NFA.
- same as NFA but input string contains e(epsilon).
- transition function: $\boldsymbol{\delta : Q \times (\sum \cup \in) \rightarrow 2^Q}$
- **Eg:**



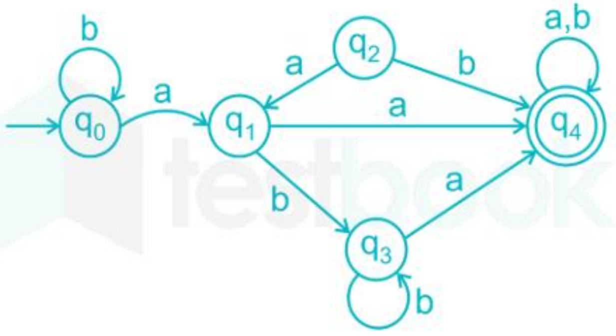
∈-NFA

Minimization of FSM:

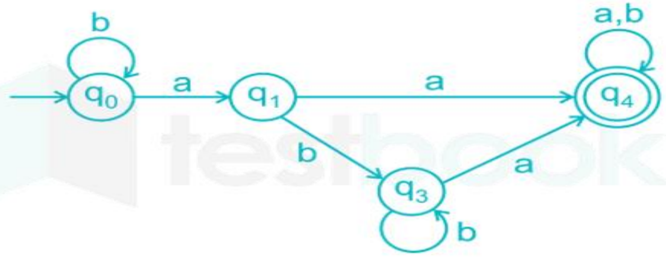
- The process of reducing a given DFA to its minimal form is called as minimization of DFA.
- It contains the minimum number of states.
- The DFA in its minimal form is called as a **Minimal DFA**.
- The two popular methods for minimizing a DFA are
  - Equivalence Theorem
  - Table Filling Method

Q1. What is the number of states obtained

after minimizing the given DFA



solution :q2 is not reachable from any states so it is dead state (discard it)



- 0 equi sets= [q0,q1,q3], [q4]
- 1 equi sets =[q0], [q1,q3], [q4]
- 2 equi sets=. [q0],[q1,q3],[q4]

Therefore no of states is 3.

Equivalence of NFA and DFA:

- NFA and DFA are equal in Power though DFA has more number of transitions than NFA.
- NFA is very simple to construct than DFA.
- Convert DFA-> NFA (every DFA is also an NFA).
- Convert NFA->DFA (every NFA is not a DFA because NFA one state has multiple next states but DFA not, though we can convert NFA-> to DFA using Subset construction method.

Regular Expressions:

- Are algebraic descriptions of a language. It is used to:
  - Specify or search text strings.
  - Languages defined by NFA , DFA can be also defined by regular expression.
- A language is defined by NFA, DFA and regular expression is called regular languages.

Operators:

- L1={01,101,11} and L2={101, e}
  - Union: L1UL2={01,101,11,e}
  - Concatenation: L1L2 = {01101,01,101101,101,11101,11} e-identity element.
  - Closure: L*; if L={0,1} then L*={e,0,1,00,01,10,11,…}

RE Defination:

Basis:

1. RE=e.; {e}
2. RE=a; {a}

Induction:

1. If R1 & R2 are REs then

   RE=R1+R2; L(R1)UL(R2)

2. RE=R1R2; L(R1)L(R2)
3. RE=R1*; (L(R1))*

Q1. Construct a RE for the language accepting all strings which have bab as a substring over input={a,b}.

Solution:

RE=(a+b)*bab(a+b)*

Q2. write a regular expression for input {0,1} accepting string starting and ending with different symbols.

0 (0+1)* 1 + 1 (0+1)* 0

Pumping Lemma for Regular Languages:

- For showing certain languages not to be regular.
- Let L be a regular language. Then there exists a constant n such that for every string w in L, $|w| \geq n$.
- We can break w into three strings, w=xyz such that

  1. $y \neq e$ or $|y|>0$
  2. $|xy| \leq n$
  3. for all $k \geq 0$, the string xy^kz is also in L

Find suitable integer k such that $xy^kz \notin L$ hence L is not regular.

Eg: assume L is regular and n be a constant

let $L = o^n1^{n;\ n>0}$

split w = xyz such that

1. $y \neq e$ or $|y|>0$
2. $|xy| \leq n$
3. for all $k \geq 0$, the string xy^kz is also in L

$W = o^n1^n = 0011$ (when n=2)

Xy = 00 i.e. x=0, y=0 and z=11

K=2 then $xy^kz = 00^211 = 00011 \notin L$ so contradiction, hence we can say that $L=\{o^n1^n;\ n \geq 1\}$ is not regular

Equivalence of RE and FA:

1. an algorithm that, given a regular expression R, produces an FA A such that L(A) == L(R).
2. an algorithm that, given an FA A, produces a regular expression R such that L(R) == L(A).

MCQ Links

https://www.studocu.com/sg/document/lovely-professional-university/automata/finite-automata-unitn-1/9257389

https://www.studocu.com/sg/document/thapar-institute-of-engineering-and-technology/data-structures-and-algorithms/engineeringinterviewquestions-com-mcqs-on-regular-language-expression-2-answers/16127192