Finite Automata

Finite State Machine (Prerequisites)

①

Symbol - a, b, c.. / 0, 1, 2, 3, ...

Alphabet - Σ - Collection of Symbols.

Eg. {a, b}, {d, e, f, g}

{0, 1, 2} ....

String - Sequence of Symbols. Eg. a, b, 0, 1, aa, bb, ab, 01..

Language - Set of Strings. Eg. $\Sigma = \{0, 1\}$

Suppose:

L_1 = set of all Strings of length 3

= {00, 01, 10, 11} → Finite Set

L_2 = set of all strings that begin with 0

= {0, 00, 01, 000, 001, 010, 011, 0000, ...} → Infinite set

(2)

Finite State Machine (Prerequisite)

Powers of Σ:

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

Σ^0 = set of all strings of length 0 : $\Sigma^0 = \{\epsilon\}$

Σ^1 = set of all strings of length 1 : $\Sigma^1 = \{0,1\}$

Σ^2 = " " " " " length 2 : $\Sigma^2 = \{00, 01, 10,$

⋮

Σ^n = set of all strings of length n

Cardinality – Number of elements in a set

$$\boxed{\Sigma^n = 2^n}$$

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \Sigma^3$$

$= \{\epsilon\} \cup \{0,1\} \cup \{00,01,10,11\} \cup \ldots$

= Set of all possible strings of all lengths over $\{0,1\}$ $\rightarrow$ Infinite set.

③

Finite State machine

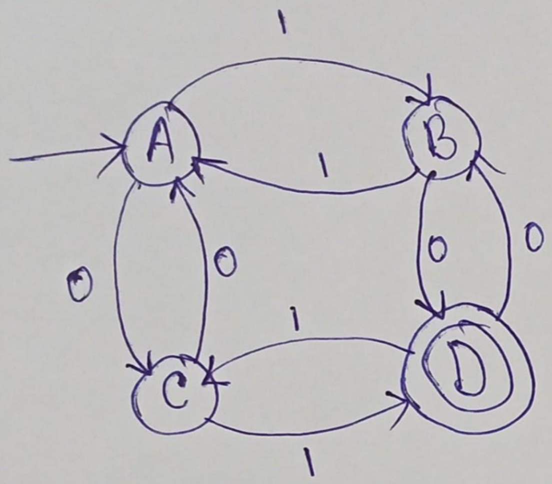
- Finite Automata (FA)
  - FA With Output
    - Moore machine
    - Mealy Machine
  - FA Without Output
    - DFA ✓
    - NFA
    - ϵ-NFA

DFA – Deterministic Finite Automata

For Finite state machine (properties)

- It is a simplest model of Computation
- It has a very limited memory

DFA



Structure of a DFA

⇓

$Q = \{A, B, C, D\}$

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

$q_0 = A$

$F = \{D\}$

⇒ DFA can be defined using 5 tuples:-

$(Q, \Sigma, q_0, F, \delta)$

Q = set of all states

Σ = inputs

q_0 = Start state/initial state

F = Set of final states

δ = transition function from $Q \times \Sigma \rightarrow Q$

	0	1
A	C	B
B	D	A
C	A	D
D	B	C

⇓

⇒ $\delta(A, 0) = C$

$\delta(A, 1) = B$

$\delta(B, 0) = D$

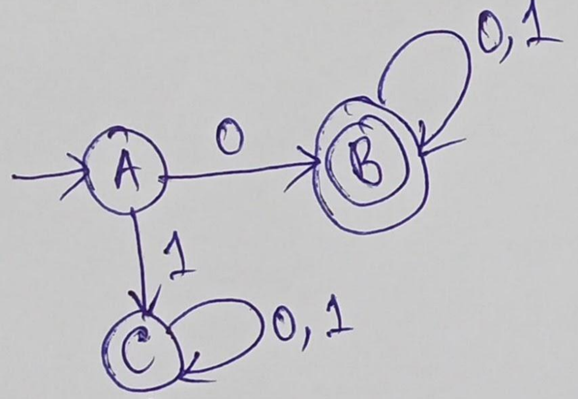
⋮

$\delta(D, 1) = C$.

Deterministic Finite Automata (Eample-1)

L1 = Set of all strings that starts with '0'

= {0, 00, 01, 000, 010, 011, 0000, ... }



Dead state
or
Trap state

Eg. 001

Initial state $A \xrightarrow{0} B \xrightarrow{0} B \xrightarrow{1} B$

Final state

Eg. 101

Initial state $A \xrightarrow{1} C \xrightarrow{0} C \xrightarrow{1} C$

Not a Final state

⑥

DFA (Example-2)

⇒ Construct a DFA that accepts sets of all strings over $\{0, 1\}$ of length 2.

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

$L = \{00, 01, 10, 11\}$

$$\begin{tikzcd} \arrow[r] & A \arrow[r, "0{,}1"] & B \arrow[r, "0{,}1"] & \circledcirc C \arrow[r, "0{,}1"] & D \arrow[loop above, "0{,}1"] \end{tikzcd}$$

Dead State
or
Trap state

Eg. 00

$A \xrightarrow{0} B \xrightarrow{0} \circledcirc C$ – Final State

Eg. 10

$A \xrightarrow{1} B \xrightarrow{0} \circledcirc C$ – Final State

Eg. 001

$A \xrightarrow{0} B \xrightarrow{0} \circledcirc C \xrightarrow{1} D$ – Not Final State

Eg. 1

$A \xrightarrow{1} B$ – Not Final State

7

DFA (Example-3)

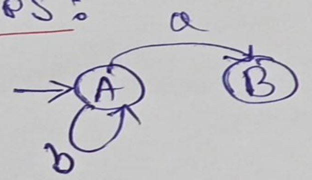
⇒ Construct a DFA that accepts any strings over {a, b} that does not contain the string aabb in it.

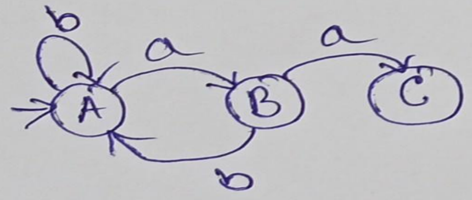
$$\Sigma = \{a, b\}$$

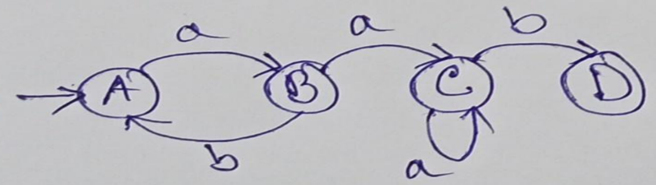
Try to design a simpler problem

⇒ Let us construct a DFA that accepts all strings over {a, b} that contains the string aabb in it.

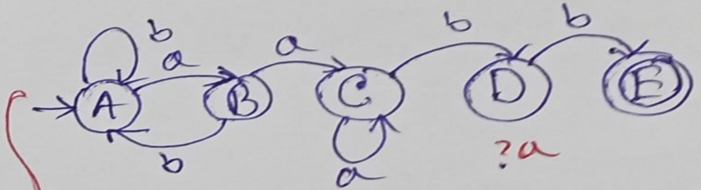
STEPS:

1


2


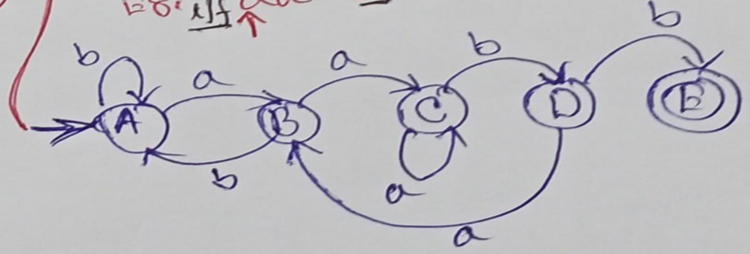
3


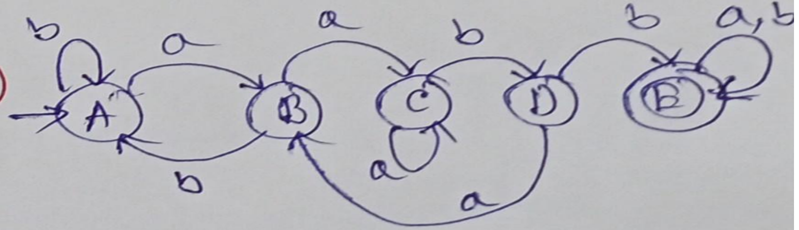
aaabb

4


?a

Eg: iff aaba — wrong



5


(8)

DFA (Example 3)

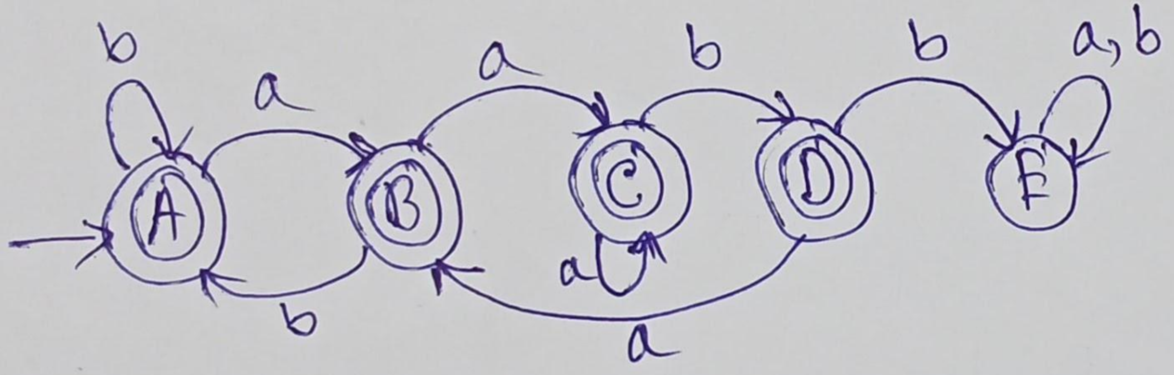
⇒ Construct a DFA that accepts any strings over {a, b} that does not contain the string aabb in it.

⇒ Flip the States

- make the final state into non-final state.

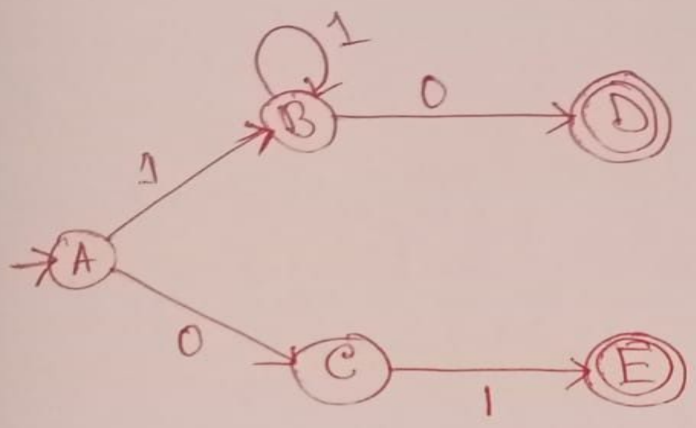
and

- make the non-final states into final state.



DFA (Example-4)

⇒ How to figure out what a DFA recognizes?



10 ✓

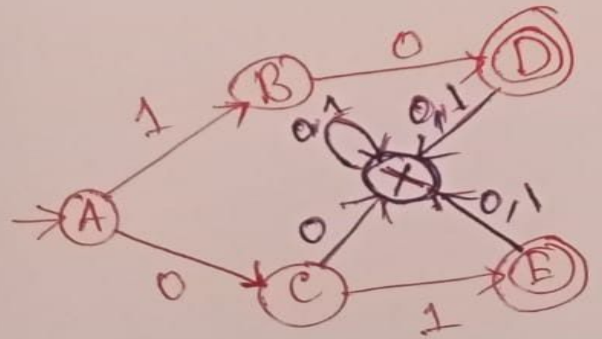
$\underset{\uparrow}{1}\underbrace{1111}_{B}\underset{\uparrow}{0}$ ✓
A D

01 ✓

Atleast one binary digit '1'

L = { Accepts the string 01 or a string of atleast one '1' followed by a '0' }

Eg. 001, 010, 011, 1101, 1100



→ OK, Complete.

Thank You