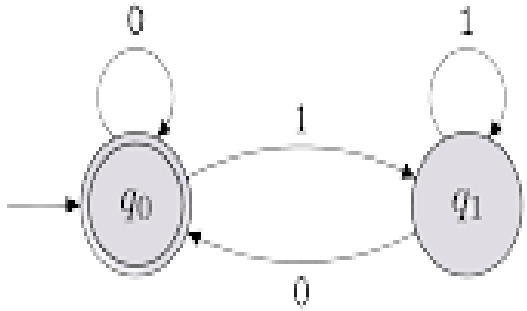
2 WAY DETERMINISTIC FINITE AUTOMATA



Made by: Yamini Bamal.

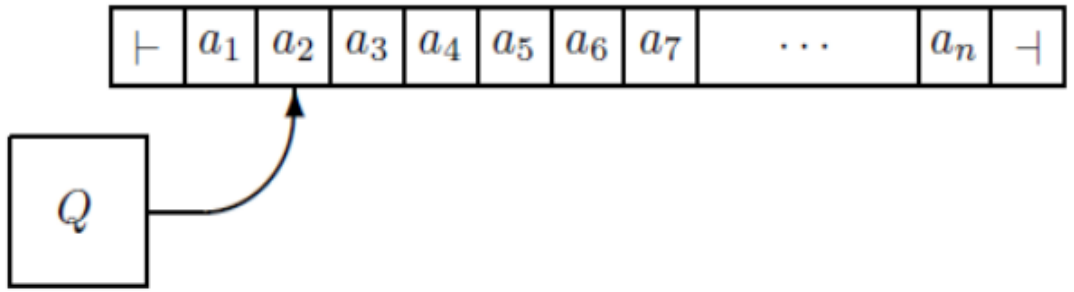
[0905cs201201]

overview

- Introduction.
- Formal construction.
- Example.
- 2DFA vs DFA.

introduction

- Generalized version of DFA.
- Process the input in either direction.
  - ➢Have read only head which can move in both direction over the input string.
  - ➢Revisit the characters again and again.
- Like a Turing Machine but.
  - ➢Have read only head.
  - ➢Have finite memory like DFA



- 2DFA has finite set of states Q like DFA.
- Input string
  - Input string is stored on finite tape.
  - One character per cell.
  - Input string is stored in between two extra symbol called left end marker(`) and right end marker(a).
- At any time instance the machine is in state p and scan some symbol ai ∈ Σ or an end markers {|-, -|}, based on p and current symbol it will move its head one cell in direction d ∈ {L, R} and enter in new state q .
- Machine head never go outside the end markers.

- Accept and reject states.
  - 2DFA needs only single accept and single reject state.
  - It will accept the input string by entering in a special accept state t.
  - It will reject the input string by entering in a special reject state r.
  - Accept and reject states are like sink state.
- The machine action on a present state and head symbol is depend on transition function δ.
- Transition function take present state and head symbol as input argument and return next state and direction of movement of head.

Formal definition of 2DFA

- 2DFA is represented by octuple.

$$M = \{Q, \Sigma, \delta, s, t, r, |\text{-}, \text{-}|\}$$

- where :

➢ Q is a finite set of states.

➢ Σ is a finite set of input symbol.

➢ $\delta : Q \times (\Sigma \cup \{`, a\}) = Q \times \{L, R\}$ is a transition function.

➢ $s \in Q$ is a start state.

➢ $t \in Q$ is a accept state.

➢ $r \in Q$ is a reject state.

➢ |- is left end marker.

➢ -| is right end marker

Some properties of transition function

- Input is end marker.
  - $\delta(p, |-) = (q, R)$
  - $\delta(p, -|) = (q, L)$
- Accept and reject states are t , r respectively and current input symbol is $a \in \Sigma \cup \{|-\}$.
  - $\delta(t, a) = (t, R)$ and $\delta(t, -|) = (t, L)$
  - $\delta(r, a) = (r, R)$ and $\delta(r, -|) = (r, L)$
- In general
  - $\delta(p, a) = (q, d)$ where $p, q \in Q$ and $d \in \{L, R\}$

Example (Constructing a normal DFA)

Construct the DFA to accept the language

L = {x ∈ Σ ∗ |#a(x) are multiple of 3, #b(x) are multiple of 2}

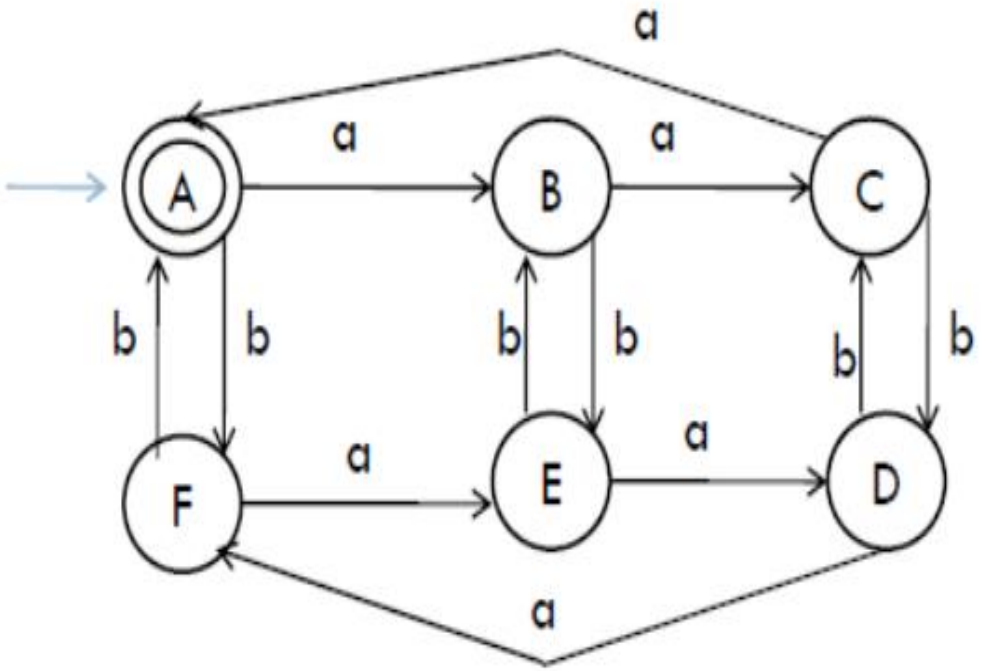
Construct a normal DFA

- DFA M1 accepting L1 = {x ∈ Σ ∗ |#a(x) are multiple of 3}

  M1 = {Q1, Σ, δ1,s1, F1} 2 DFA
- DFA M2 accepting L2 = {x ∈ Σ ∗ |#b(x) are multiple of 2}

  M2 = {Q2, Σ, δ2,s2, F2}
- DFA M accepting L such that M = M1 × M2

  M = {Q, Σ, δ,s, F}



Example(Constructing a 2DFA)

Construct a 2DFA accepting the set

L = {x ∈ Σ ∗ |#a(x) are multiple of 3, #b(x) are multiple of 2}

- Machine start scanning from the left end marker.
- Scan input string from left to right consider only a and ignore b.
  if #a(x) are not multiple of 3 then rejects x and enters in state r.
- if #a(x) are multiple of 3 then start scanning from right consider only b and ignore a.
  if #b(x) are not multiple of 2 then enters in t otherwise enters in state r.

Example (Formal construction of 2DFA)

M = {Q, Σ, δ , s, t, r, |-, -|}

where Σ = {a, b}, Q = {q0, q1, q2, p0, p1,t,r} and the transition function δ is given by following table.

states	$\vdash$	a	b	$\dashv$
q_0	(q_0, R)	(q_1, R)	(q_0, R)	(p_0, L)
q_1	-	(q_2, R)	(q_1, R)	(r, L)
q_2	-	(q_0, R)	(q_2, R)	(r, L)
p_0	(t, R)	(p_0, L)	(p_1, L)	-
p_1	(r, R)	(p_1, L)	(p_0, L)	-
t	(t, R)	(t, R)	(t, R)	(t, L)
r	(r, R)	(r, R)	(r, R)	(r, L)

2DFA vs DFA

DFA

Input is processed once from left to right. After an input has been read, the DFA decides whether the input is accepted or rejected.

2DFA

- Can read the input back and forth with no limit on how many times an input symbol can be read.
- As in the case of DFA ,the 2DFA decides whether a given input is accepted or rejected.


Thank you