27/4/20 ①

Lecture - 6

Finite state Transducers

- FSM are used as language recognizers. i.e, given a language, we find out if it is accepted or rejected.
- For every transition ~~an~~ or state an output is attached.

∴ An input sequence can be converted into a sequence of one or more outputs.

⇓

process of conversion of input data from one form to another is called a transducer.

Finite State Transducer (FST)

A FSM where an output is associated with either a transition or a state.

FST types

- Mealy Machines
- Moore Machines.

Mealy Machines

- output values are determined by both the current state and current input.

It is a 6-tuple.

$$M = (Q, \Sigma, \delta, q_0, \lambda, F)$$

where,

Q = set of states

Σ = input alphabet

δ ← transition function ($Q \times \Sigma$)

q_0 ← start state

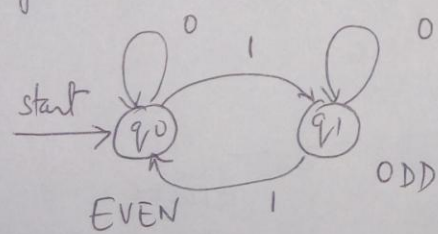
λ ← output function ($Q \times \Sigma \rightarrow \Gamma$)

~~F ← set of final states.~~

Γ ← set of output alphabets

Ex. Construct a Mealy machine which accepts strings of 0's and 1's, and output EVEN when the input string has even number of 1's and output ODD when the input string has odd number of 1's.

Sol.



For q_0

$\delta(q_0, 0) = q_0$ $\quad\lambda(q_0, 0) = E$

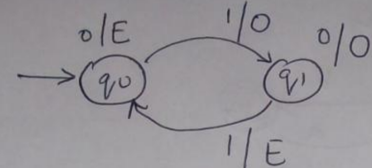
$\delta(q_1, 1) = q_0$ $\quad\lambda(q_1, 1) = E$

For q_1

$\delta(q_1, 0) = q_1$ $\lambda(q_1, 0) = 0$

$\delta(q_0, 1) = q_1$ $\lambda(q_0, 1) = 0$

$\therefore$



$M = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$

$Q = (q_0, q_1)$

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

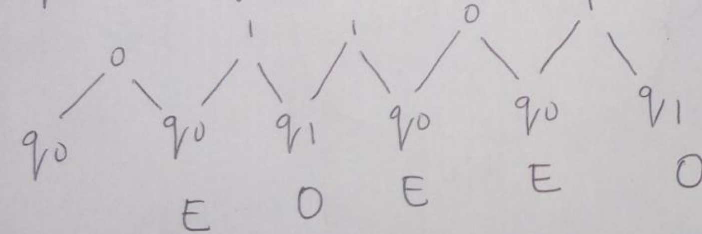
$\Gamma = \{E, 0\}$

δ	0	1
q_0	q_0	q_1
q_1	q_1	q_0

λ	0	1
$\rightarrow q_0$	E	O
q_1	O	E

q_0 is the start state.

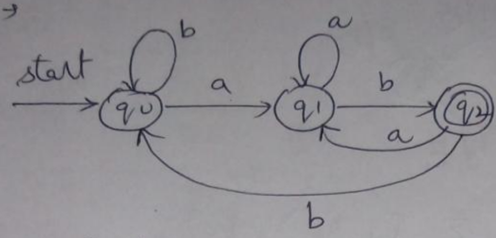
sequence of moves for a string 01101



Ex

Construct a Mealy Machine ~~and~~ which accepts strings of a's and b's and count the number of times the pattern ab is present in the string.

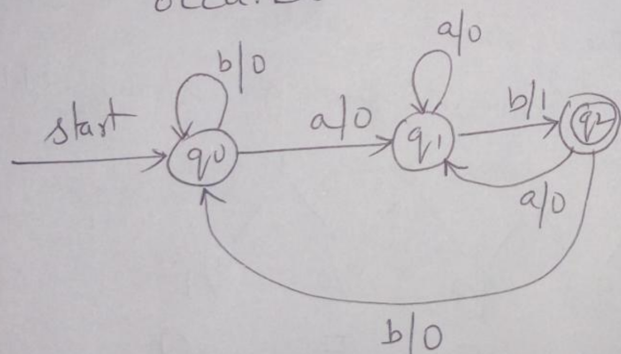
Sol :→



FSM.

Change into a Mealy machine.

- as we encounter a pattern ab., we move into final state ⇒ count = 1
- for other transitions, output count = 0
- count the number of 1's to get final number of times string ab occured.



$$M = (Q, \Sigma, \Gamma, \delta, q_0, \lambda)$$

$$Q = \{q_0, q_1, q_2\}$$

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

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

$q_0 \leftarrow$ start state

δ	a	b
$\rightarrow q_0$	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0

λ	a	b
q_0	0	0
q_1	0	1
q_2	0	0

check with string abab

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{a} q_1 \xrightarrow{b} q_2$

outputs: 0 1 0 1

number of 1's = 2

$\Downarrow$

implies ab appeared twice

Moore Machines

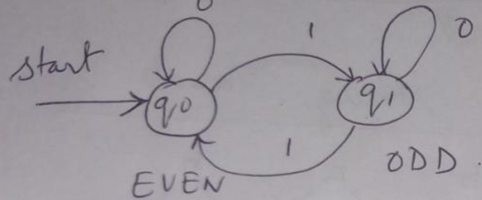
Def :→ A FSM where an output is associated with each state.

$M = (Q, \Sigma, \Gamma, \delta, \lambda, q_0)$

- Q ← finite set of state
- Σ ← input alphabet
- Γ ← set of output alphabets
- δ ← transition function from ($Q \times \Sigma$ to Q)
- λ ← output function
- q_0 ← initial state

Ex. Construct a Moore Machine which accept strings of 0's and 1's as the input, ~~accepting~~ and produces output EVEN when the input string has even number of 1's and ODD when input string has ODD number of 1's.

Sol. Let us first draw the FSM



In state q_0, machine accepts strings of 0's and 1's with even number of 1's. So output this as EVEN.

In state q_1, machine accepts strings of of 0's and 1's with ODD number of 1's, so output this as ODD.