U19CS076

Kuthikha
Balamurug...

Tutorial 5

1- Write difference between Moore Machine and Mealy machine.

Basis of Comparison	Mealy Machine	Moore Machine
Definition.	Defined as a machine in theory of computation whos outputs depends on current state and current inputs.	Defined as a machine in theory of computation whose output depend on only its current state.
Tuple	It has 6 tuples $(Q, q_0, \Sigma, O, \delta, \lambda')$	It has also 6 tuples $(Q, q_0, \Sigma, O, \delta, \lambda)$
state	Less states than Moore machine	More states than Mealy.
Output	Output depends on present state, present input	Output depends only on present state
	Output is placed on transition	Output is placed on states.
Reaction to input	Read faster to inputs. They react in same clock cycle.	More logic required to decode output resulting in more circuit delays.

U19CS076

~~They react~~		They react one clock later
Value of output function	Function of transition and changes when input logic on present state is done.	Function of current state, changes at clock edges whenever state change occur.
State requirement	Fewer states for synthesis.	More states for synthes.
Hardware Req.	Less hardware to design	More hardware to design
Counter	Not a Mealy machine	IS a Moore mach.
Design	Not always easy	Easy.

(2). Define Moore machine and Mealy machine

There are 2 types of finite automata state machines that generate output.

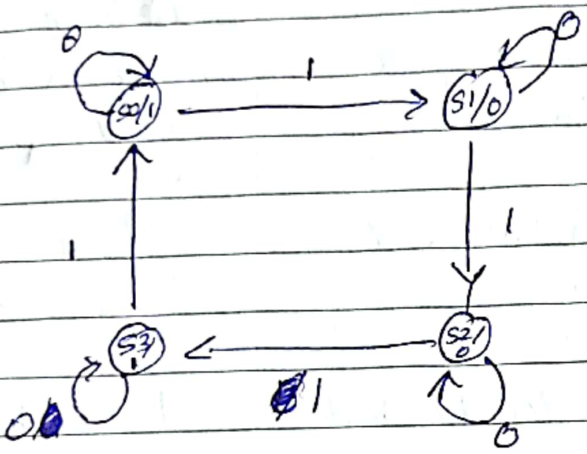
Moore Machine

It is a finite state machine whose outputs depend on only present state

U19CS076

It can be described by 6 tuple
$(Q, \Sigma, O, \delta, q_0, X)$

- Q is finite set of states
- Σ is finite set of symbols - input alphabet
- O is " " " " - output alphabet
- δ is input transition function $\delta : Q \times \Sigma \to Q$
- X is output " " $X : Q \to O$
- q_0 is initial state



Mealy machine

It is a finite state machine whose output depends on present state as well as present input

It has 6 tuple $(Q, \Sigma, O, \delta, X, q_0)$

Q is finite set of states
Σ is finite set of symbols - input alphabet
O " " " " " - output "
δ is input transition function

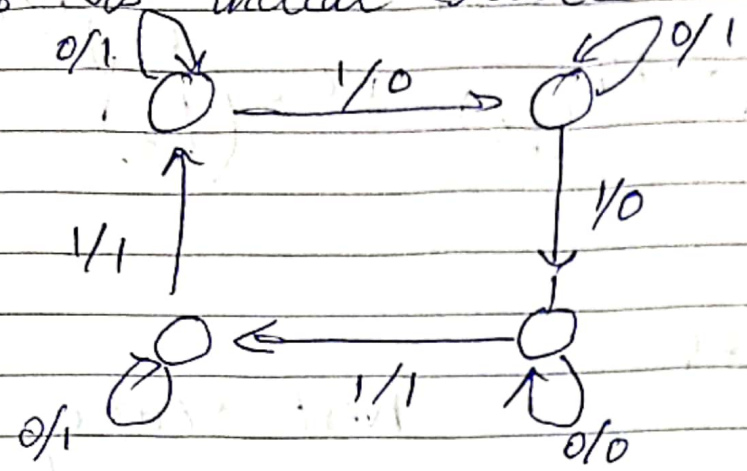
$$\delta : Q \times \Sigma \to Q$$

X is output transition function

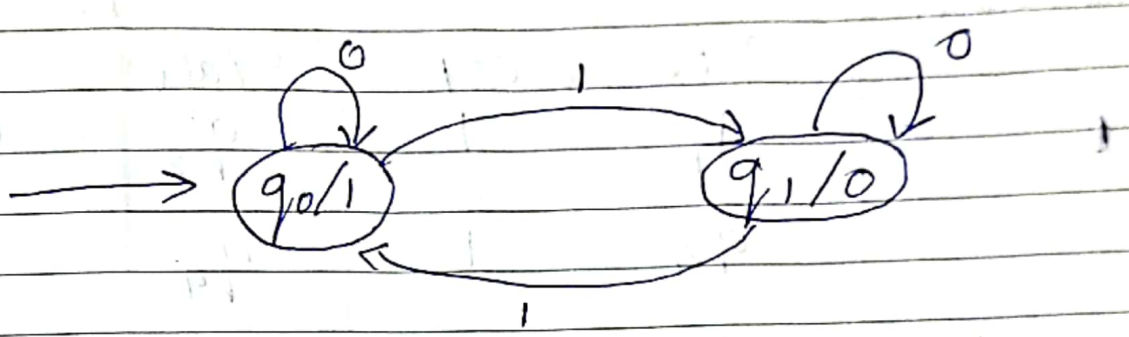
$$X : Q \times \Sigma \to O$$

U19CS076

q_0 is initial state



3.

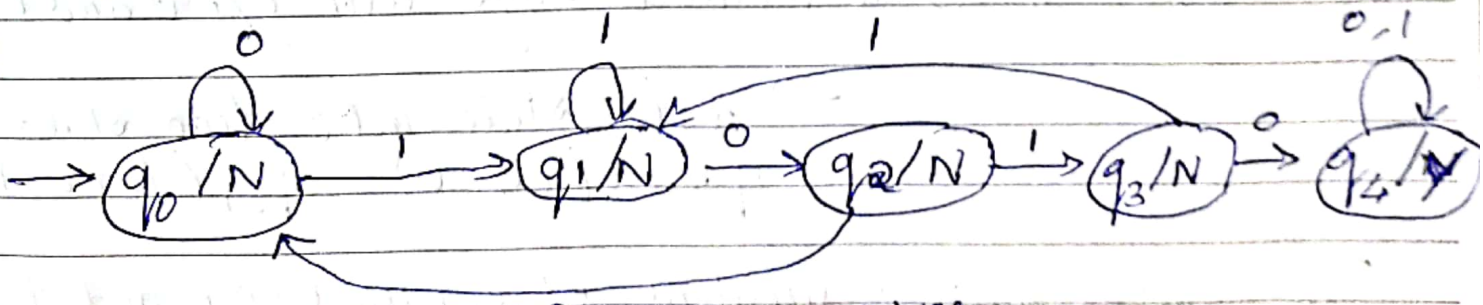


q_0 accepts even number of 1s
q_1 accepts odd number of 1s.

When at state q_0, output is 1 otherwise all other states have 0

We move from q_1 to q_0 when input is 1 since there is no restriction on number of zeroes, self loop of 0 applied both states.

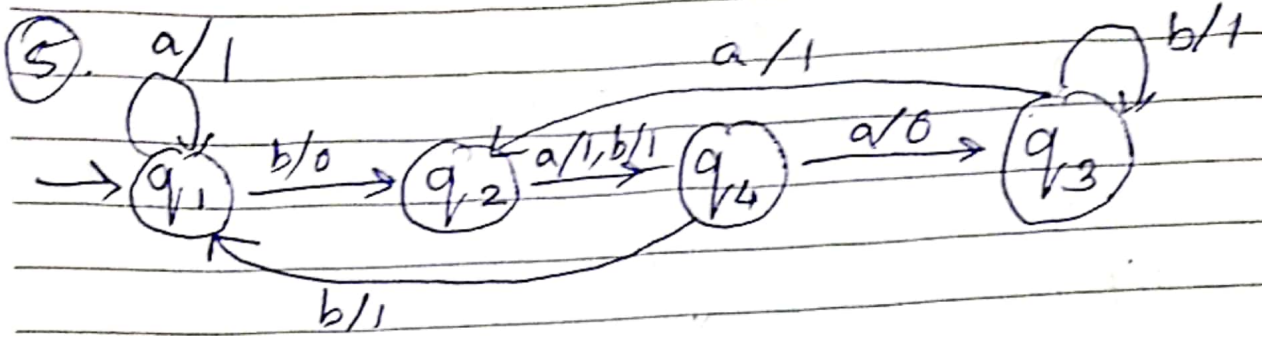
4



Here only final state q_4 gives accepts Y since 1010 is formed. All other states produces N or does self loop.

U19CS076

(5)



Transition table

Present State	Next State a		Next State b	
	State	o/p	State	o/p
q_1	q_1	1	q_2	0
q_2	q_4	1	q_4	1
q_3	q_2	1	q_3	1
q_4	q_3	0	q_1	1

For state q_1, it has single incident edge with o/p 0.

→ No need to split data in Moore machine

For state q_2, 2 incident edge with o/p 0 and 1.

→ Split state into two states q_{20} (for o/p 0), q_{21} (for o/p 1)

For state q_3, 2 incident edge with o/p 0 and 1.

→ Split state into two states q_{30} (for o/p 0), q_{31} (for o/p 1)

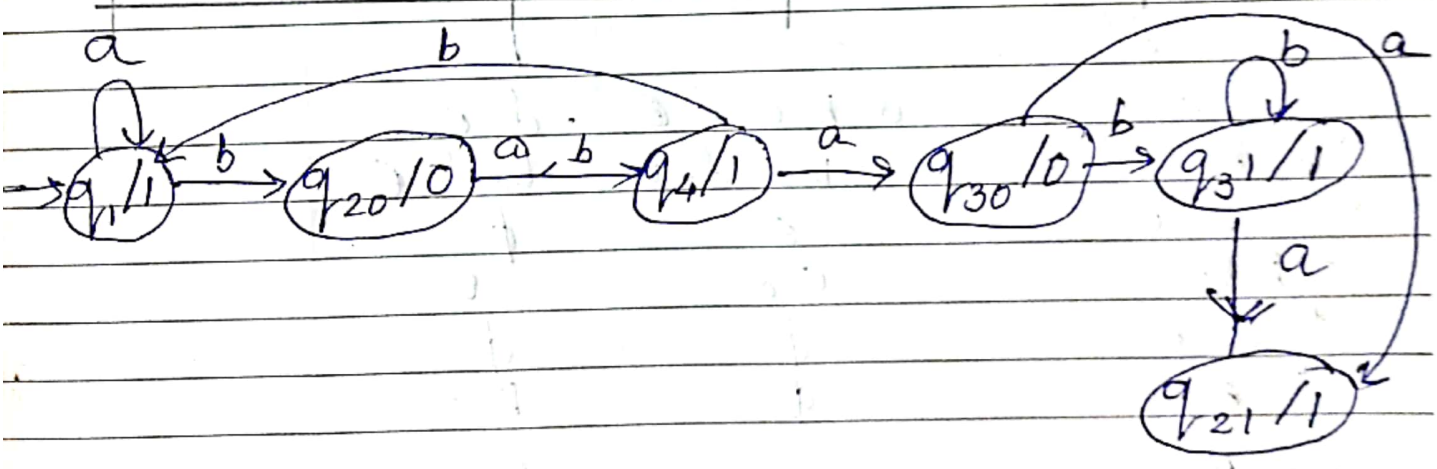
U19CS076

For state q_4, only one incident edge with O/P = 0.

$\Rightarrow$ No need to split

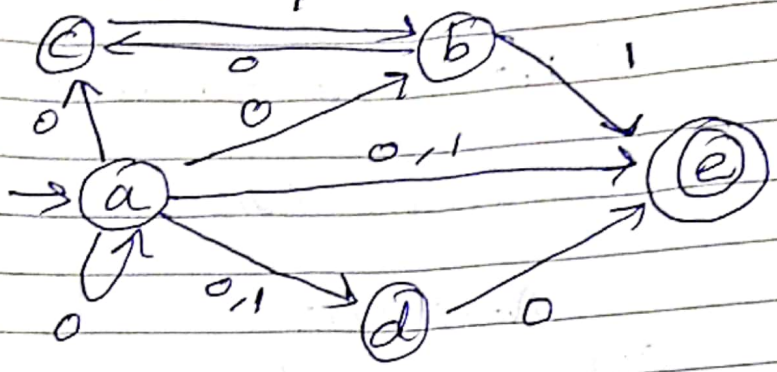
$\Rightarrow$ Moore machine Transition table

Present state	Next State		Output
	a=0	a=1	
q_1	q_1	q_2	1
q_{20}	q_4	q_4	0
q_{21}	ϕ	ϕ	1
q_{30}	q_{21}	q_{31}	0
q_{31}	q_{21}	q_{31}	1
q_4	q_3	q_4	1



U19CS076

⑥. Convert NFA to DFA



NFA

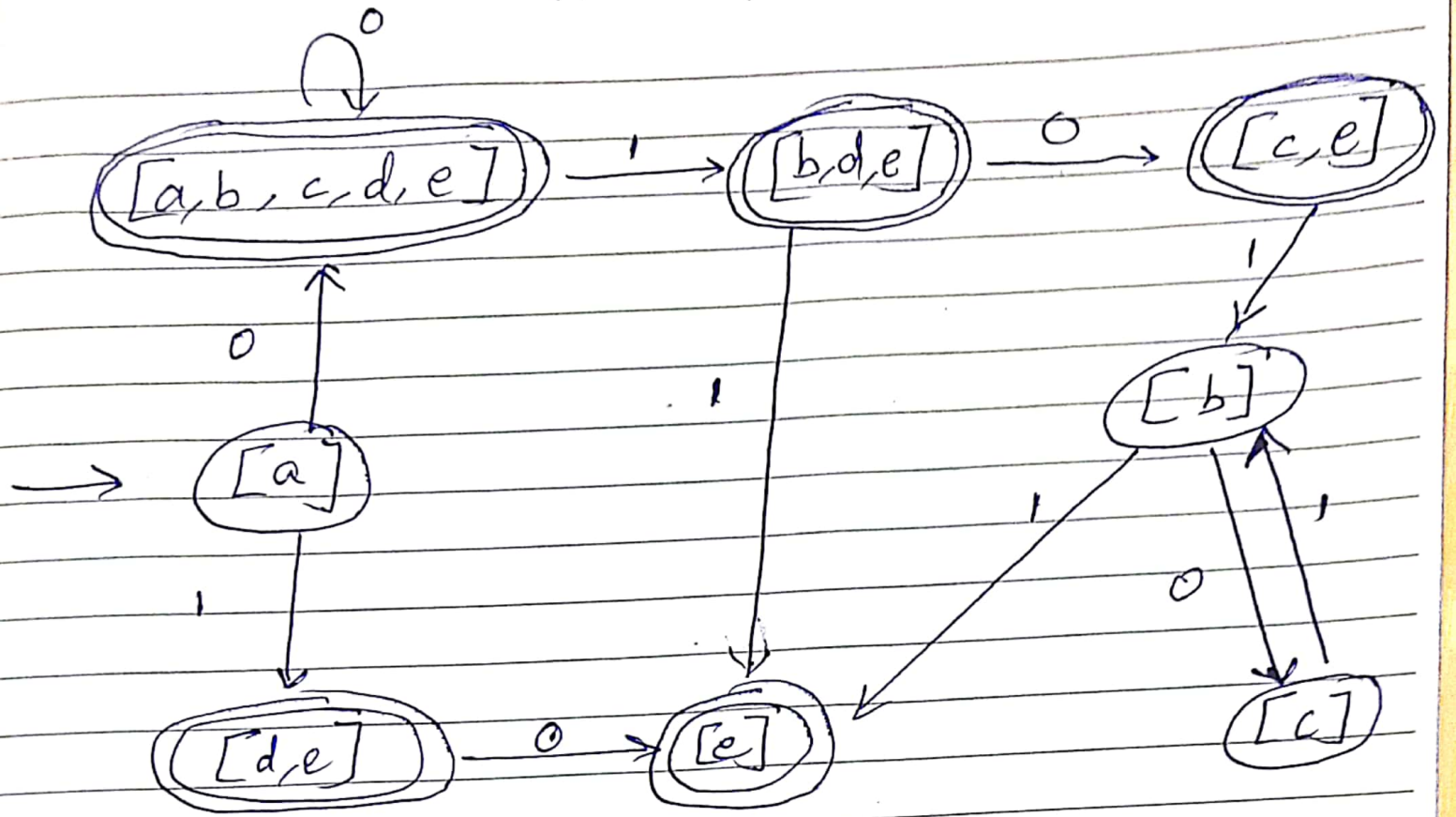
	δ	0	1
	→ a	{a, b, c, d, e}	{d, e}
	b	{c}	{e}
	c	ϕ	{b}
	d	{e}	ϕ
[Final State]	*e	ϕ	ϕ

NFA → DFA

	δ'	0	1
→	a	a, b, c, d, e	d, e
*	a, b, c, d, e	a, b, c, d, e	b, d, e
*	d, e	e	ϕ
*	b, d, e	c, e	e
*	e	ϕ	ϕ
*	c, e	ϕ	b
	b	c	e
	c	ϕ	b

All * are final states

U19CS076



Dfa chart.