

RGPV 2010

Q. Formally define the following (with example)-

1. Mealy machine

1. Moore machine

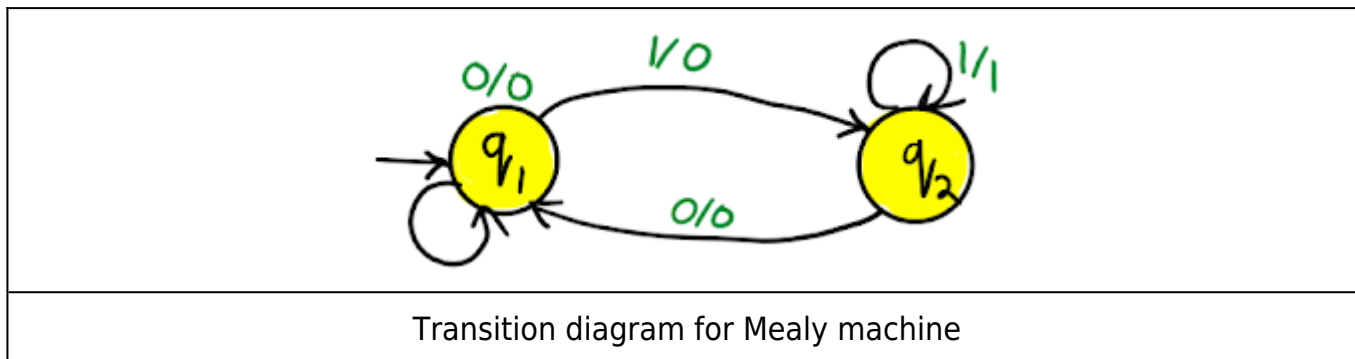
1. Mealy machine: Mealy machine is a six tuple machine.

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

1. Q is finite set of states.
2.  $\Sigma$  is the input alphabet.
3.  $\Delta$  is the output alphabet.
4.  $\delta$  is transition function which maps  $Q \times \Sigma \rightarrow Q$ .
5. ' $\lambda$ ' is the output function which maps  $Q \times \Sigma \rightarrow \Delta$ .
6.  $q_0$  is the initial state.

Transition table for Mealy machine

Present state	Next state			
	a = 0		a = 1	
	State	Output	State	Output
->q1	q1	0	q2	0
q2	q1	0	q2	1



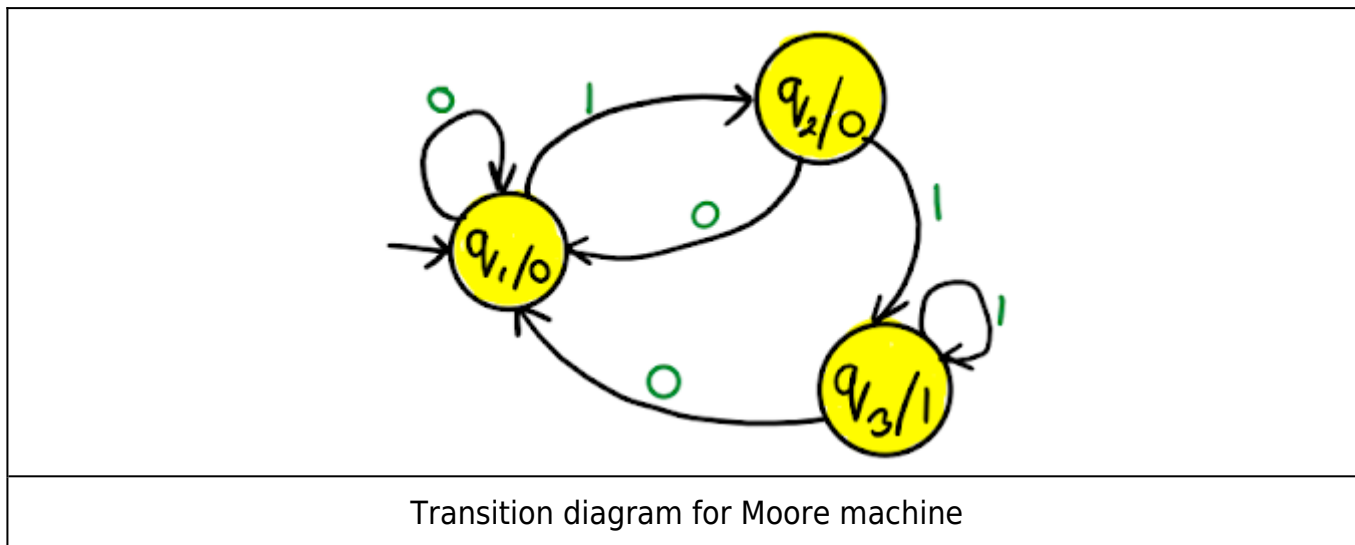
1. **Moore machine:** Moore machine is a six tuple machine.

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

1. Q is finite set of states.
2.  $\Sigma$  is the input alphabet.
3.  $\Delta$  is the output alphabet.
4.  $\delta$  is transition function which maps  $Q \times \Sigma \rightarrow Q$ .
5. ' $\lambda$ ' is the output function which maps  $Q \rightarrow \Delta$ .
6.  $q_0$  is the initial state.

Transition table for Moore machine

Present state	Next state		Output
	a = 0	a = 1	
->q1	q1	q2	0
q2	q1	q3	0
q3	q1	q3	1



Transition diagram for Moore machine

### Mealy machine vs Moore machine

Mealy machine	Moore machine
Output depends on present state as well as present input.	Output depends on the present state.
If input changes, output also changes	If input changes, output does not changes.
Compare to Moore less number of states are required. Because states do not depends on output.	Compare to Mealy more number of states are required. Because states depends on number of output.
Difficult to develop. Difficulty due to input affects output.	Easy to develop.
Output is placed on transition arrow.	Output is placed with state.

Related posts:

1. RGPV TOC Short note on equivalent of DFA and NFA
2. RGPV notes Write short note on NFA
3. Construct Moore machine for Mealy machine
4. Difference between Mealy and Moore machine
5. RGPV TOC What do you understand by DFA how to represent it
6. RGPV short note on automata
7. RGPV TOC properties of transition functions
8. RGPV TOC What is Trap state
9. CFL are not closed under intersection
10. NFA to DFA | RGPV TOC
11. Moore to Mealy | RGPV TOC PYQ
12. DFA accept even 0 and even 1 | RGPV TOC PYQ
13. Short note on automata | RGPV TOC PYQ
14. DFA ending with 00 start with 0 no epsilon | RGPV TOC PYQ
15. DFA ending with 101 | RGPV TOC PYQ
16. Construct DFA for a power  $n$ ,  $n \geq 0$  || RGPV TOC
17. Construct FA divisible by 3 | RGPV TOC PYQ
18. Construct DFA equivalent to NFA | RGPV TOC PYQ
19. CNF from  $S \rightarrow aAD; A \rightarrow aB/bAB; B \rightarrow b, D \rightarrow d$ .
20. NFA accepting two consecutive a's or two consecutive b's.
21. Regular expression to CFG
22. Regular expression to Regular grammar
23. Grammar is ambiguous.  $S \rightarrow aSbS|bSaS|\epsilon$
24. leftmost and rightmost derivations
25. Definition of Deterministic Finite Automata
26. Notations for DFA
27. How do a DFA Process Strings?

28. DFA solved examples
29. Definition Non Deterministic Finite Automata
30. Moore machine
31. Mealy Machine
32. Regular Expression Examples
33. Regular expression
34. Arden's Law
35. NFA with  $\epsilon$ -Moves
36. NFA with  $\epsilon$  to DFA Indirect Method
37. Define Mealy and Moore Machine
38. What is Trap state ?
39. Equivalent of DFA and NFA
40. Properties of transition functions
41. Mealy to Moore Machine
42. Moore to Mealy machine
43. Difference between Mealy and Moore machine
44. Pushdown Automata
45. Remove  $\epsilon$  transitions from NFA
46. TOC 1
47. What is Regular Expression
48. What is Regular Set in TOC
49. DFA which accept 00 and 11 at the end of a string
50. DFA end with 1 contain 00 | RGPV TOC draw
51. RGPV TOC design finite automata problems
52. Minimization of DFA
53. Construct NFA without  $\epsilon$
54. RGPV TOC PYQs

55. Introduction to Automata Theory