A Mealy machine is a type of finite state automaton (FSA) that produces output on transitions between states.

Unlike a Moore machine, which produces output for each state-input symbol pair, a Mealy machine only produces output when it changes states.

This means that Mealy machines are more efficient than Moore machines in terms of output generation, as they do not produce unnecessary output for states that do not change.

In the Mealy machine, the output symbol at a given time is a function of the present input as well as the present state of the machine.

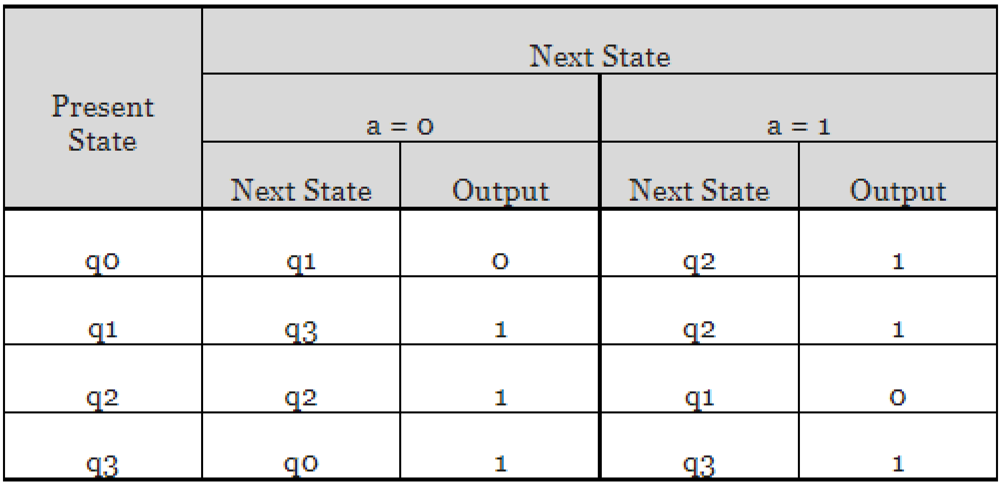
Therefore the transition graph of the Mealy machine cannot be further simplified.

> *Mealy machine is defined as a 6 tuple (Q, Σ, Δ, δ, λ, q0)*
>
> *where,*
>
> - *Q: A finite set of states.*
> - *Σ: A finite set of input symbols.*
> - *δ: A transition function: δ: Q x Σ → Q.*
> - *λ: An output function: λ: δ(Q x Σ) → O.*
> - *q_0: The start state.*
> - *F: A set of accepting states.*
>
> *Reference: Introduction to the Theory of Computation” by Michael*

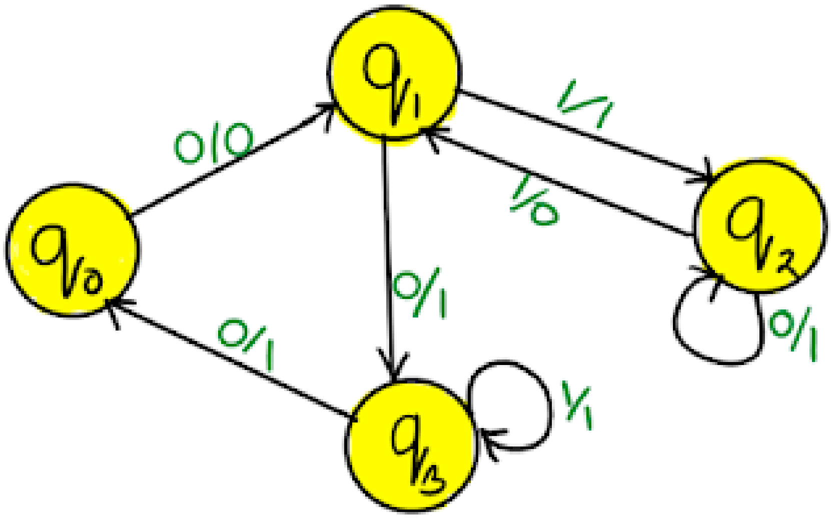
> *Sipser.*

For example,

Mealy machine Transition table



Present State	Next State			
	a = 0		a = 1	
	Next State	Output	Next State	Output
q0	q1	0	q2	1
q1	q3	1	q2	1
q2	q2	1	q1	0
q3	q0	1	q3	1



Mealy machine

Transition Diagram

Related Posts:

1. Definition of Deterministic Finite Automata
2. Notations for DFA
3. How do a DFA Process Strings?
4. DFA solved examples
5. Definition Non Deterministic Finite Automata
6. Moore machine
7. Regular Expression Examples
8. Regular expression
9. Arden’s Law
10. NFA with ∈-Moves
11. NFA with ∈ to DFA Indirect Method
12. Define Mealy and Moore Machine
13. What is Trap state ?
14. Equivalent of DFA and NFA
15. Properties of transition functions
16. Mealy to Moore Machine
17. Moore to Mealy machine
18. Diiference between Mealy and Moore machine
19. Pushdown Automata
20. Remove ∈ transitions from NFA
21. TOC 1
22. Diiference between Mealy and Moore machine
23. RGPV TOC What do you understand by DFA how to represent it

24. What is Regular Expression
25. What is Regular Set in TOC
26. RGPV short note on automata
27. RGPV TOC properties of transition functions
28. RGPV TOC What is Trap state
29. DFA which accept 00 and 11 at the end of a string
30. CFL are not closed under intersection
31. NFA to DFA | RGPV TOC
32. Moore to Mealy | RGPV TOC PYQ
33. DFA accept even 0 and even 1 |RGPV TOC PYQ
34. Short note on automata | RGPV TOC PYQ
35. DFA ending with 00 start with 0 no epsilon | RGPV TOC PYQ
36. DFA ending with 101 | RGPV TOC PYQ
37. Construct DFA for a power n, n>=0 || RGPV TOC
38. Construct FA divisible by 3 | RGPV TOC PYQ
39. Construct DFA equivalent to NFA | RGPV TOC PYQ
40. RGPV Define Mealy and Moore Machine
41. RGPV TOC Short note on equivalent of DFA and NFA
42. RGPV notes Write short note on NDFA
43. Minimization of DFA
44. Construct NFA without ∈
45. CNF from S–>aAD;A->aB/bAB;B->b,D->d.
46. NDFA accepting two consecutive a’s or two consecutive b’s.
47. Regular expresion to CFG
48. Regular expression to Regular grammar
49. Grammar is ambiguous. S → aSbS|bSaS|∈
50. leftmost and rightmost derivations

51. Construct Moore machine for Mealy machine
52. RGPV TOC PYQs
53. Introduction to Automata Theory
54. Design a NFA that accepts the language over the alphabet, Σ= {0, 1,2} where the decimal equivalent of the language is divisible by 3.