Mealy machine has 6 tuples: (Q, q0, Σ , O, δ , λ ')

- 1. Q : Finite set of states
 - In diagram below Q = {A,B,C,D}
- 2. q0 : Initial state/ Starting state
 - In diagram below A is initial state
- 3. Σ : Input alphabet
 - In diagram below input alphabets are {0,1}
- 4. O : Output alphabet
 - In diagram below output alphabets are {0,1}
- 5. δ is transition function which maps $Q \times \Sigma \rightarrow Q$
- 6. ' λ ' is the output function which maps $Q \times \Sigma \rightarrow O$



Moore machine has 6 tuples: (Q, q0, Σ , O, δ , λ')

- 1. Q : Finite set of states
 - In diagram below $Q = \{A,B,C,D\}$

- 2. q0 : Initial state/ Starting state
 - In diagram below A is initial state
- 3. Σ : Input alphabet
 - In diagram below input alphabets are {0,1}
- 4. O : Output alphabet
 - In diagram below output alphabets are {0,1}
- 5. δ is transition function which maps $Q \times \Sigma \to Q$
- 6. ' λ' is the output function which maps $Q \rightarrow O$



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.

Reference:

• Introduction to the Theory of Computation" by Michael Sipser.

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. Mealy Machine
- 8. Regular Expression Examples
- 9. Regular expression
- 10. Arden's Law
- 11. NFA with \in -Moves
- 12. NFA with \in to DFA Indirect Method

- 13. Define Mealy and Moore Machine
- 14. What is Trap state ?
- 15. Equivalent of DFA and NFA
- 16. Properties of transition functions
- 17. Mealy to Moore Machine
- 18. Moore to Mealy machine
- 19. Pushdown Automata
- 20. Remove \in 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 \in
- 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 \rightarrow aSbS|bSaS| \in
- 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, $\Sigma = \{0, 1, 2\}$ where the decimal equivalent of the language is divisible by 3.