A Moore machine is a type of finite state automaton (FSA) that is characterized by its output function. Unlike a Mealy machine, which produces output only when transitioning between states, a Moore machine produces output for each state and input symbol combination.

This means that a Moore machine can be used to generate output strings as well as recognize input strings.

In this type of machine, the future state of the machine is decided by the current state and current input symbol of the machine.

The output symbol at a given time depends only on the present state of the machine.

Moore machine is defined as a 6 tuple $(Q, \Sigma, \Delta, \delta, \lambda, q0)$.

where,

- Q: A finite set of states.
- Σ: A finite set of input symbols.
- δ : A transition function: δ : $Q \times \Sigma \rightarrow Q$.
- λ : An output function: λ : $Q \times \Sigma \to O$.
- qo: The start state.
- F: A set of accepting states.

Example Moore machine:





Moore Machine
Transition Diagram

In Moore machine, each state has a fixed output, i.e. in above diagram,

- When machine is in state q1, the output is 0.
- When machine is in state q2, the output is 0.
- When machine is in state q3, the output is 1.

Referencees:

• 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. Mealy 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, $n \ge 0$ || 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