The important thing about DFA is to know that it identifies the acceptance of strings.

> *A DFA processes a string by starting in the start state and reading the input symbols one at a time. For each input symbol, the DFA transitions to the next state according to the transition function, and it accepts the string if it ends in an accepting state.*

An algorithm that describes how a DFA processes a string:

1. Start in the start state.
2. Read the next input symbol.
3. Transition to the next state according to the transition function.
4. If the current state is an accepting state, then accept the string.
5. If the current state is not an accepting state, and there are no more input symbols to read, then reject the string.
6. Repeat steps 2-5 until the end of the string is reached.

For example,

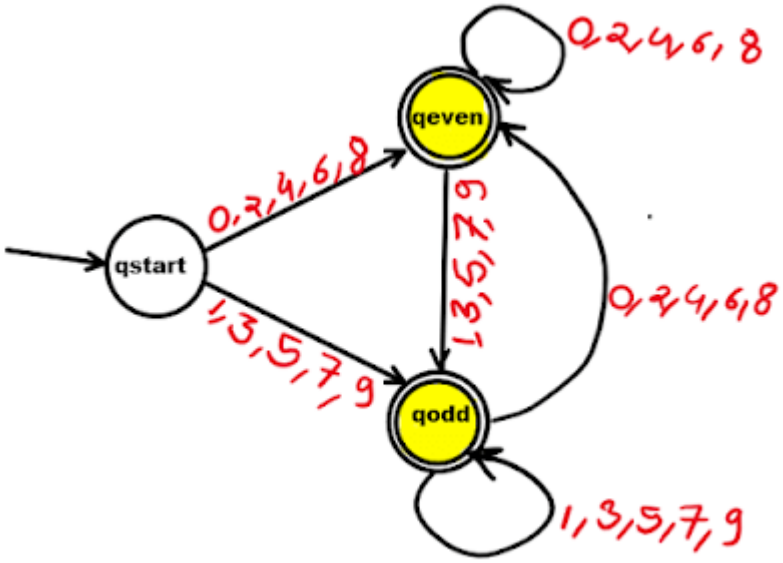
Consider the DFA that identifies whether the given decimal is even or odd.

Here we consider 3 states, one start state qstart, one even state qeven and one odd state

qodd.

If the machine stops at qeven the given number is even.

If it stops at qodd the given number is odd.



DFA that identifies ODD and EVEN numbers

State	Input Symbol	
	0, 2, 4, 6, 8	1, 3, 5, 7, 9
qstart	qeven	qodd
qeven	qeven	qodd
qodd	qeven	qodd

References:

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

Related Posts:

1. Definition of Deterministic Finite Automata
2. Notations for DFA
3. DFA solved examples
4. Definition Non Deterministic Finite Automata
5. Moore machine
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>=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.