

RGPV 2010

Q. Write short note on automata ?

Ans. An automaton is an abstract model of a digital computer.

Some types of automata :

1. Finite-state machine.
2. Pushdown automata.
3. Turing machine

A finite automaton (FA) is a simple idealized machine used to recognize patterns within input taken from some character set (or alphabet) C. The job of an FA is to accept or reject an input depending on whether the pattern defined by the FA occurs in the input.

Finite automata are used in text processing, compilers, and hardware design. Context-free grammar (CFGs) are used in programming languages and artificial intelligence.

Finite automata are very useful for communication protocols and for matching strings against regular expressions.

Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages.

A pushdown automaton is a way to implement a context-free grammar in a similar way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.

A pushdown automaton reads a given input string from left to right. In each step, it chooses a transition by indexing a table by input symbol, current state, and the symbol at the top of the stack.

A PDA is more powerful than FA. Any language which can be acceptable by FA can also be acceptable by PDA. PDA also accepts a class of language which even cannot be accepted by FA. Thus PDA is much more superior to FA.

A Turing machine is a system of rules, states and transitions rather than a real machine.

The example a Turing machine handles a string of 0s and 1s, with 0 represented by the blank symbol. Its task is to double any series of 1s encountered on the tape by writing a 0 between them. For example, when the head reads "1111", it will write a 0, then "1111". The output will be "111101111".

Related posts:

1. RGPV TOC What do you understand by DFA how to represent it
2. RGPV short note on automata
3. RGPV TOC properties of transition functions
4. RGPV TOC What is Trap state
5. CFL are not closed under intersection
6. NFA to DFA | RGPV TOC
7. Moore to Mealy | RGPV TOC PYQ
8. DFA accept even 0 and even 1 | RGPV TOC PYQ
9. DFA ending with 00 start with 0 no epsilon | RGPV TOC PYQ
10. DFA ending with 101 | RGPV TOC PYQ
11. Construct DFA for a power n , $n \geq 0$ || RGPV TOC
12. Construct FA divisible by 3 | RGPV TOC PYQ
13. Construct DFA equivalent to NFA | RGPV TOC PYQ
14. RGPV Define Mealy and Moore Machine
15. RGPV TOC Short note on equivalent of DFA and NFA
16. RGPV notes Write short note on NDFA
17. CNF from $S \rightarrow aAD; A \rightarrow aB/bAB; B \rightarrow b, D \rightarrow d$.
18. NDFA accepting two consecutive a's or two consecutive b's.
19. Regular expression to CFG

20. Regular expression to Regular grammar
21. Grammar is ambiguous. $S \rightarrow aSbS|bSaS|\epsilon$
22. leftmost and rightmost derivations
23. Construct Moore machine for Mealy machine
24. Definition of Deterministic Finite Automata
25. Notations for DFA
26. How do a DFA Process Strings?
27. DFA solved examples
28. Definition Non Deterministic Finite Automata
29. Moore machine
30. Mealy Machine
31. Regular Expression Examples
32. Regular expression
33. Arden's Law
34. NFA with ϵ -Moves
35. NFA with ϵ to DFA Indirect Method
36. Define Mealy and Moore Machine
37. What is Trap state ?
38. Equivalent of DFA and NFA
39. Properties of transition functions
40. Mealy to Moore Machine
41. Moore to Mealy machine
42. Difference between Mealy and Moore machine
43. Pushdown Automata
44. Remove ϵ transitions from NFA
45. TOC 1
46. Difference between Mealy and Moore machine

- 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 ϵ
- 54. RGPV TOC PYQs
- 55. Introduction to Automata Theory