RGPV PYQs

Q. What is a regular expression?

Solution. The language accepted by finite automata can be easily described by simple expressions called regular expressions.

Let,
Σ denotes the input set.

1. Φ is a regular expression that denotes the empty set.
2. ε is a regular expression and denotes the set {ε}, and it is a null string.
3. For each 'a' in Σ, 'a' is a regular expression and denotes the set (a).
4. If R1 and R2 are regular expressions denoting the Languages L1 and L2, respectively, then
    - R1+R2 is equivalent to LI ∪ L2, i.e., union.
    - R1R2 is equivalent to L1 ∩ L2, i.e. concatenation
    - R* is equivalent to L1*, i.e., closure.

The R* is known as kleen closure or closure, which indicates the occurrence of R an infinite number of times.

Some other examples of regular expressions are:

1. R = a, *i.e., all combinations of a.*
2. *R = a+, i.e., all combinations of a without a null string.*
3. *R = (a+b)*, i.e., strings contain any number of a and b

Related Posts:

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

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