Example 1: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contains only a's or only b's of length zero or more.

Solution: $r = a^* + b^*$

Example 2: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w is of length one or more and contains only a's or only b's. r = a+ + b+

Solution: $r = a^+ + b^+$

Example 3: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contains zero or more a's followed by zero or more b's

Solution: $r = a^*b^*$

Example 4: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length even

Solution: r = [(a + b) (a + b)]*

Example 5: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length odd

Solution: r = (a + b) [(a + b) (a + b)]*

Example 6: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length three

Solution: r = (a + b) (a + b) (a + b)

Example 7: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length atmost three

Solution: r = (a + b + ∈) (a + b + ∈) (a + b + ∈)

Example 8: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w of length odd containing only b's

Solution: r = (bb)* b

Example 9: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w starting with a always

Solution: r = a(a + b)*

Example 10: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w starting and ending with b and having only a's in between.

Solution: r = b a* b

Example 11: Let Σ = {a, b}. Write regular expression to define language consisting of strings

w such that, w starting and ending with same double letter

Solution: r = {(aa (a + b)* aa) | (bb (a + b)* bb)

Example 12: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with starting and ending with different letters

Solution: r = (a(a+b)* b) | (b (a + b)* a)

Example 13: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at least two occurrence of a

Solution: r = (a + b)* a (a + b)* a (a + b)*

Example 14: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with exactly two occurrence of a

Solution: r = b* a b* a b*

Example 15: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at most two occurrence of a

Solution: r = b* (a + ∈) b* (a + ∈) b*

Example 16: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with begin or end with aa or bb

Solution: r = ((aa + bb) (a + b)*) + ((a + b) * (aa + bb))

Example 17: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with begin and end with aa or bb

Solution: r = ((aa + bb) (a + b)* (aa + bb)) + aa + bb

Example 18: Let Σ = {a, b}. Write regular expression to define language consisting of strings

w such that, w with total length multiple of 3 always

Solution: r = [(a + b) (a + b) (a + b)]*

Example 19: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w containing total a’s as multiple of 3 always

Solution: r = [b* a b* a b* a b*]*

Example 20: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with exactly two or three b’s

Solution: r = a* b a* b a* (b + ∈) a*

Example 21: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with number of a’s even

Solution: r = b* + (b* a b* a b*)*

Example 22: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w in which b is always tripled

Solution: r = (a + bbb)*

Example 23: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at least one occurrence of substring aa or bb

Solution: r = (a + b)* (aa + bb) (a + b)*

Example 24: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w with at the most one occurrence of sub-string bb

Solution: r = (a + ba)* (bb + ∈) (a + ab)*

Example 25: Let Σ = {a, b}. Write regular expression to define language consisting of strings

w such that, w without sub-string ab

Solution: $r = b^* a^*$

Example 26: Let $\Sigma = \{a, b\}$. Write regular expression to define language consisting of strings w such that, w without sub-string aba

Solution: $r = (a + \in) (b + aa+)^* (a + \in)$

Example 27: Let $\Sigma = \{a, b\}$. Write regular expression to define language consisting of strings w such that, w in which 3rd character from right end is always a

Solution: $r = (a + b)^* a (a + b) (a + b)$

Example 28: Let $\Sigma = \{a, b\}$. Write regular expression to define language consisting of strings w such that, w always start with ‘a’ and the strings in which each ‘b’ is preceded by ‘a’.

Solution: $(a + ab)^*$

Example 29: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contains atleast one ‘a’.

Solution: (a + b)* a (a + b)*

Example 30: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contain atleast two ‘a’s or any number of ‘b’s.

Solution: (a* a b* a b*) + b*

Example 31: Let Σ = {a, b}. Write regular expression to define language consisting of strings w such that, w contain atleast one ‘a’ followed by any number of ‘b’s followed by atleast one ‘c’.

Solution: $a^+ b^* c^+$

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
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.