| **Semester: V** | | | | | | |
| --- | --- | --- | --- | --- | --- | --- |
| **FINITE AUTOMATA AND FORMAL LANGUAGES**  **(Theory)** | | | | | | |
| **Course Code** | **:** | 18CS52 |  | **CIE Marks** | **:** | 100 |
| **Credits: L:T:P** | : | 3:0:0 |  | **SEE Marks** | **:** | 100 |
| **Total Hours** | : | 39L |  | **SEE Duration** | **:** | 3 Hrs |

1. Find a grammar that generates L={ an bn+1 : n>=0}
2. Find a grammar that generates the language L = {w: na (w) = nb (w)}
3. Consider the grammar G= ({A,S},{a,b},S,P1 )

with P1 consisting of the production S→ aAb│λ, A→aAb│. Find the language generated by this grammar.

1. Find grammars for Σ = {a,b} that generate the sets of (a) all strings with exactly one a, (b) all strings with atleast one a.
2. Find grammars for the following languages on Σ = {0}. (a) L= {w:|w|mod3 =0},(b) L = { w: |w|mod3>0}.
3. Convert the following NFA to a DFA

δ 0 1

→p {p,q} {p}

q {r} {r}

r {s} Ф

\*s {s} {s}

1. Convert to a DFA the following NFA

δ 0 1

→p {q,s} {q}

\*q {r} {q,r}

r {s} {p}

\*s Ф {p}.

1. Consider the following ε-NFA.(a) Compute the ε-closure of each state. (b) Convert the automaton to a DFA.

δ ε a b c

→p Ф {p} {q} {r}

q {p} {q} {r} Ф

\*r {q} {r} Ф {p}

1. Do same as (8) for

δ ε a b c

→p {q,r} Ф {q} {r}

q Ф {p} {r} {p,q}

\*r Ф Ф Ф Ф

1. Let L be the language accepted by a NFA MN. Show that there exists a DFA MD such that L(MN) = L(MD).
2. Define a regular expression(RE). Write RE for the language consisting of the set of strings over alphabet {a,b,c} containing at least one a & at least one b.
3. Write REs for the following languages over{0,1} (a) the set of all strings of 0’s & 1’s such that every pair of adjacent 0’s appears before any pair of adjacent 1’s. (b) All strings ending in 01.
4. Find the language generated by the following REs (a) r = (a+b)\*(a+bb) (b) r = (aa)\*(bb)\*b
5. Show that if r is a RE then there exists a finite automata which accepts L(r).
6. Define regular language. If L is a regular language then prove that there exists a RE r such that L = L(r).
7. Transition table for a DFA is give below

δ 0 1

→p q p

q r p

\*r r q

(a) give all the REs Rij(0). (b) Give all the REs Rij(1). Simplify the expressions as much as possible. (c) Construct the transition diagram for the DFA & give a RE for its language by eliminating state q.

28. Convert the following DFA to a RE, using the state elimination technique

δ 0 1

→\*p s p

q p s

r r q

s q r

29. Convert the following REs to NFA’s with ε-transitions. (a) 01\* , (b) 00(0+1)\*.

30. Write a note on applications of REs.

31. State and prove Pumping Lemma(PL) for regular languages.

32. Prove that following languages are not regular using PL. (a) {0n1n : n≥1}

(b) {0n10n : n≥1}.

33. Using PL show that the following languages are not regular. (a) {0n : n is a perfect square} (b) {0n1m : m≥n}.

34. If L & M are regular languages then show that LUM is also regular language.

35. If L is a regular language over alphabet Σ then prove that Σ\* - L is also a regular language.

36. If L & M are regular languages then prove that L∩M is also regular.

37. Show that L-M is regular language if L & M are regular.

38. If L is a regular language prove that LR is also regular.

39. If L is a regular language over alphabet Σ & h is a homomorphism on Σ then prove that h(L) is also regular.

40. Let h be the homomorphism from the alphabet {0,1,2} to the alphabet {a,b} defined by h(0) = a, h(1) = ab, & h(2) = ba. (a) What is h(0120), h(21120)? (b) If L is the language L(01\*2), what is h(L)? (c) If L is the language L(a(ba)\*  what is h-1(L)?

41. Define equivalence of states. What are distinguishable & indistinguishable states.

42. Minimize the states of the following DFA using table-filling algorithm.

δ 0 1

→A B A

B A C

C D B

\*D D A

E D F

F G E

G F G

H G D

43. Do same as (42) for the following DFA

δ 0 1

→A B E

B C F

\*C D H

D E H

E F I

\*F G B

G H B

H I C

\*I A E

44. Define CFG, leftmost derivation, rightmost derivation, language of a grammar, sentential form, derivation tree, parse tree.

45. Design CFG for the following languages (a) {0n1n : n≥1 } (b) {aibjck : i≠j or j≠k}.

46. Consider the grammar G with productions S→A│B, A→0A│ε, B→0B│1B│ε. Give leftmost & rightmost derivations of the following strings (a) 00101 (b) 1001.

47. Given the following CFG E→E+T│T, T→T\*F│F, F→(E)│a│b│c draw parse tree for the following sentences (a) (a+b)\*c (b) (a)+b\*c.

1. Define ambiguous grammar & inherent ambiguity in a grammar with an example.
2. Show that the following grammar ambiguous S→AB│aaB, A→a│Aa, B→b. Construct an unambiguous grammar equivalent to above grammar.
3. Show that the following grammar is ambiguous. Also obtain unambiguous grammar for the following grammar E→E+E│E\*E│E/E│E-E│(E)│a│b.
4. Let G be the grammar S→aA│a│SS, A→SbA│ba. For the string aabaa find (a) leftmost derivation (b) rightmost derivation (c) derivation tree.
5. Define CNF & GNF, Useless productions, Unit productions, λ-productions.
6. Eliminate all useless productions from the grammar S→aS│AB, A→bA, B→AA.
7. Eliminate all useless productions from the grammar S→AB│CA, A→a, B→BC│AB, C→aB│b.
8. Eliminate useless productions from S→a│aA│B│C, A→aB│λ, B→Aa, C→cCD, D→ddd.
9. Eliminate all λ-productions from S→AaB│aaB, A→λ, B→bbA│λ.
10. Find a CFG without λ-productions equivalent to the grammar defined by S→ABaC, A→BC, B→b│λ, C→D│λ, D→d.
11. Eliminate all unit productions from the grammar S→Aa│B, B→A│bb, A→a│bc│B.
12. Eliminate all unit productions from the grammar E→T│E+T, T→F│T\*F, F→I│(E), I→a│b│Ia│Ib│I0│I1.
13. Put the grammar with productions given below into CNF S→ASB│ε, A→aAS│a, B→SbS│A│bb.
14. Convert the following grammar into CNF S→0A0│1B1│BB, A→C, B→S│A, C→S│ε.
15. Transform the grammar with productions S→abAB, A→bAB│ε, B→Baa│A│ε into CNF.
16. Convert the following grammar into GNF S→AA│0, A→SS│1.
17. Convert the grammar S→ab│aS│aaS into GNF.
18. Convert the grammar S→ABb│a, A→aaA│B, B→bAb into GNF.
19. State & prove Pumping Lemma(PL) for Context Free Languages(CFL).
20. Using CFL PL show that { aibjck │ i<j<k } is not CFL.
21. Using CFL PL show that {0i1j │ j = i2 } is not CFL.
22. Show that CFL’s are closed under union & concatenation.
23. Prove that CFL’s are closed under closure & homomorphism.
24. Show that CFL’s are not closed under intersection.
25. Define PDA. Design a PDA to accept the language L = { wwR │ w is a string consisting of 0’s and 1’s}. Draw transition diagram.
26. Define NPDA. Design a PDA to accept the language L = { aibjck : i+j = k, i≥0, j≥0}. Draw the transition diagram.
27. Design an NPDA for L = {anb2n : n≥0}.
28. Obtain a PDA to accept the language L = { anbn │ n≥0} by a final state. Give the graphical representation for PDA obtained. Show the moves made by the PDA for the string aaabbb.
29. For any CFL L, show that there exists an NPDA M such that L = L(M).
30. Construct a PDA that accepts the language generated by grammar with productions S→aSbb│a,
31. Construct an NPDA that accepts the language generated by the grammar S→aA, A→aABC│bB│a, B→b, C→c.
32. If L = L(M) for some NPDA M then show that L is a CFL.
33. Define Turing Machine(TM). Design a TM that accepts the strings of the form 0n1n. Write transition diagram.
34. Design a TM that accepts the language L = { anbncn : n≥1}. Give graphical representation of the TM.
35. Do the same as Q(81) for L = {wwR : w is any string of 0’s & 1’s}.
36. Define counter machines. Describe a counter machine that accepts the language {0n1m : 1≤m≤n}.
37. Prove that every language accepted by a multitape TM is recursively enumerable.
38. Write short notes on (a) Multitape TM, (b) Nondeterministic TM.
39. Explain the general structure of multitape & multidimensional TMs, show that those are equivalent to standard TM.
40. Explain the programming techniques for TMs.
41. Explain the techniques for simulating a TM by computer.
42. Explain the method for simulating computer by a TM.
43. Write short notes on (a) TMs with semi-infinite tapes (b) Multistack machines.
44. What is meant by halting problem of TM? Explain the blank tape halting problem.
45. Define recursively enumerable language, recursive language & universal languages.
46. Show that if L is a recursive language so its complement.
47. Define Post’s correspondence Problem(PCP) & the modified PCP
48. Write notes on (a) PCP (b) Halting problem
49. Write notes on (a) Application of Finite Automata (b) Applications of CFG’s
50. Write notes on (a) Homomorphism (b) Multitape TMs.
51. Solve the PCP given below

List A List B

i wi xi

1 10 101

2 011 11

3 101 011

99. Solve the PCP given below

List A List B

i wi xi

1 11 111

2 100 001

3 111 11

100. Solve the PCP given below

List A List B

i wi xi

1 1 111

2 10111 10

3 10 0