SAVEETHA SCHOOL OF ENGINEERING

SAVEETHA INSTITUTE OF MEDICAL AND TECHNICAL SCIENCES

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LIST OF EXPERIMENTS

COURSE CODE : CSA13

COURSE NAME : THEORY OF COMPUTATION

Software Requirement:

1. Simulation Tool : http://www.cburch.com/proj/autosim/download.html

Once JAVA is installed this above JRE file(tool) will work

2. Turbo C++

C Programming   UNIT-1

1. Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with ‘a’ and end with ‘a’
2. Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with 0 and end with 1
3. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

   $S \rightarrow 0A1 \quad A \rightarrow 0A \mid 1A \mid \varepsilon$
4. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

   $S \rightarrow 0S0 \mid 1S1 \mid 0 \mid 1 \mid \varepsilon$
5. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

   $S \rightarrow 0S0 \mid A \quad A \rightarrow 1A \mid \varepsilon$
6. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

   $S \rightarrow 0S1 \mid \varepsilon$
7. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

   $S \rightarrow A101A, \quad A \rightarrow 0A \mid 1A \mid \varepsilon$
8. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

   $S \rightarrow aS/\varepsilon$

9. Write a C program to simulate a Finite Automata (NFA) for the given language representing strings that start with b and end with a
10. Write a C program to simulate a Finite Automata (NFA) for the given language representing strings that start with 1 and end with 1
11. Write a C program to find ε -closure for all the states in a Non-Deterministic Finite Automata (NFA) with ε -moves.
12. Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with b and end with b
13. Write a C program to simulate a Deterministic Finite Automata (DFA) for the given language representing strings that start with a and end with b

14. Write a C program to simulate a Non-Deterministic Finite Automaton (NFA) for a given language. The language consists of strings that start with the character 'b' and end with the character 'a'.
15. Write a C program to find ε -closure for all the states in a Non-Deterministic Finite Automata (NFA) with ε -moves.
16. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

    $S \rightarrow A00A, \quad A \rightarrow 0A \mid 1A \mid \varepsilon$
17. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

    S->aAb

    A->aA/bA/ε
18. Write a C program to simulate a Finite Automata (NFA) for the given language representing strings that start with 00 and end with 11
19. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

    S->AaAaA

    A->aA/bA/ε
20. Write a C program to check whether a given string belongs to the language defined by a Context Free Grammar (CFG)

    A->aA/bA/ε

<u>Finite Automata UNIT-1I</u>

21. Design Deterministic Finite Automata using simulator to accept odd number of a's
22. Design Deterministic Finite Automata using simulator to accept the string the end with ab over set {a,b)

    W= aaabab
23. Design Deterministic Finite Automata using simulator to accept the string having 'ab' as substring over the set {a,b}
24. Draw a Deterministic Finite Automata for the language accepting strings ending with 'abba' over input alphabets ∑ = {a, b}
25. Design Deterministic Finite Automata using simulator to accept even number of a's.
26. Draw a Deterministic Finite Automata that accepts a language L over input alphabets ∑ = {0, 1} such that L is the set of all strings starting with 'aa'.

27. Design Deterministic Finite Automata using simulator to accept strings in which a's always appear tripled over input {a,b}
28. Design Deterministic Finite Automata using simulator to accept strings in which b's always appear tripled over input {a,b}
29. Design Non Deterministic Finite Automata using simulator to accept the string the start with a and end with b over set {a,b} and check W= abaab is accepted or not.
30. Design Non Deterministic Finite Automata using simulator to accept the string that start and end with different symbols over the input {a,b}.
31. Design DFA using simulator to accept the string the end with abc over set {a,b,c) W= abbaababc
32. Design Deterministic Finite Automata using simulator to accept string which start and end with different symbols .
33. Design DFA using simulator to accept the string the end with cb over set {a,b,c) W= abbaabacb
34. Design Deterministic Finite Automata using simulator to accept string which consist of even number of a's or even number b's
35. Design DFA which checks whether the given unary number is divisible by 3.
36. Design DFA for accepting all the strings of L={$a^m b^n$ / m>=a and n>=b}
37. Design NFA which accept a language consisting the strings of any no.of a's followed by any no.of b's followed by any no.of c's
38. Draw a Deterministic Finite Automata for the language accepting strings staring with 'aa' and ending with 'bb'over input alphabets ∑ = {a, b}.
39. Design Deterministic Finite Automata using simulator to accept the input string "a" ,"ac",and "bac".
40. Design a Deterministic Finite Automaton (DFA) that accepts the union of languages L1 and L2, where L1 accepts the string "0" and L2 accepts the string "1", we need to create a DFA that accepts strings "0" or "1".
41. Design DFA for accepting all the strings of L={a* / number of a >=0 }

Turing Machine **UNIT-1II**

42. Design Turing Machine using simulator over the set {a,b} to accept the input string $a^n b^n$
43. Design Turing Machine using simulator over the set {a,b} to accept the input string $a^n c^n$

44. Design Turing Machine using simulator over the set {a,b} to accept the input string $a^n b^{2n}$
45. Design Turing Machine using simulator to accept the input string for odd length of Palindrome over the set {a,b}.
46. Design Turing Machine using simulator over the set {a,b} to accept the input string $a^n b^{3n}$
47. Design Turing Machine using simulator to accept the input string ww over input alphabets $\sum = \{a, b\}$
48. Design Turing Machine using simulator to perform addition of ‘aa’ and ‘aaa’
49. Design Turing Machine using simulator to perform subtraction of aaa-aa
50. Design Turing Machine using simulator to accept the input string for even length of Palindrome
51. Design a Turing Machine using a simulator to accept the input string "wcw" over the alphabet {a, b}, where 'C' is the check-off symbol and w = "ab".
52. Design Turing Machine using simulator to perform string comparison where w={aba aba}
53. Design Turing Machine using simulator to perform subtraction of bb-bb
54. Design Turing Machine using simulator to accept all palindrome strings of all length over the set {a,b}.
55. Design Turing Machine using simulator to accept the input string w=abba
56. Design Turing Machine using simulator to accept the input string w=baab
57. Design Turing Machine using simulator to accept the input string w=ababa
58. Design Turing Machine using simulator over the set {c,b} to accept the input string $C^n b^n$
59. Design Turing Machine using simulator to accept the input string w=caac
60. Design Turing Machine using simulator over the set {c,b} to accept the input string $C^n a^n$

PDA UNIT-IV

61. Design Push Down Automata using simulator to accept the input string aabb
62. Design Push Down Automata using simulator to accept the input string a^nb^{2n}
63. Design Push Down Automata using simulator to accept the input string a^nb^n over input alphabets $\sum = \{a, b\}$.
64. Design Push Down Automata to represent the language L ={W/W belongs to (a+b)* and na(w)>nb(w) where na(w)=Number of a's in w, nb(w)=Number of b's in w.
65. Design Push Down Automata using simulator to accept the input string aaabbb
66. Design Push Down Automata using simulator to accept the input string a^mb^m
67. Design Push Down Automata to represent the language L ={W/W belongs to (a+b)* and na(w)=nb(w) where na(w)=Number of a's in w, nb(w)=Number of b's in w.
68. Design Push Down Automata using simulator to accept the input string aabbbcc
69. Design Push Down Automata using simulator to accept the input string aaccbb
70. Design Push Down Automata using simulator to accept the input string bbbaaccc
71. Design Push Down Automata using simulator to accept the input string bbbccaaa
72. Design Push Down Automata using simulator to accept the input string ccbbaa
73. Design Push Down Automata using simulator to accept the input string ccaabb
74. Design Push Down Automata to represent the language L ={W/W belongs to (b+c)* and nb(w)=nc(w) where nb(w)=Number of b's in w, nc(w)=Number of c's in w.
75. Design Push Down Automata to represent the language L ={W/W belongs to (a+c)* and na(w)=nc(w) where na(w)=Number of a's in w, nc(w)=Number of c's in w.
76. Design Push Down Automata using simulator to accept the input string a^nc^n over input alphabets $\sum = \{a, c\}$.
77. Design Push Down Automata using simulator to accept the input string

$c^n a^n$ over input alphabets $\sum = \{c, a\}$.

78. Design Push Down Automata using simulator to accept the input string $a^m c^m$
79. Design Push Down Automata using simulator to accept the input string $b^m a^m$
80. Design Push Down Automata using simulator to accept the input string $c^m b^m$