Code No: 155FG **R18**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, July/August - 2023
FINITE AUTOMATA AND COMPILER DESIGN
(Computer Science and Engineering - IOT)

Time: 3 Hours **Max. Marks: 75**

Note: i) Question paper consists of Part A, Part B.
ii) Part A is compulsory, which carries 25 marks. In Part A, Answer all questions.
iii) In Part B, Answer any one question from each unit. Each question carries 10 marks and may have a, b as sub questions.

PART – A

(25 Marks)

1.a)	Define Regular Expression.	[2]
b)	What do you mean by Left Most Derivation?	[3]
c)	Define Syntax directed definition.	[2]
d)	List out the rules for comparing FOLLOW SET.	[3]
e)	What is type checking?	[2]
f)	Define type O Grammar.	[3]
g)	Define Flow Graph.	[2]
h)	What is dynamic storage allocation?	[3]
i)	Define Code generation.	[2]
j)	List out the issues in the design of a code generation.	[3]

PART - B

(50 Marks)

2.a) Construct Finite Automata for the regular Expression 01(0+10)*00.
b) Construct a derivation tree for the string abcd from the grammar
S→aAB,A→bC,B→d,C→cd [5+5]

OR

3.a) Define Ambiguous Grammar. Check whether the grammar
S→aAB,
A→bC/cd,
C→cd,
B→c/d
Is Ambiguous or not?
b) List out the applications of CFL. [5+5]

4. Construct SLR Parsing table for the grammar S→L=R/R, L→*R/id, R→L. (Write all necessary procedures). [10]

OR

5. Contrast SLR Parser for the following grammar:
E→ E+T/T, T→T*F/F, F →(E)/id
(Write all necessary procedures). [10]

6. Explain in detail about Chomsky hierarchy of Languages. [10]

OR

7.a) Explain in brief about overloading of operators with examples.
b) Discuss about equivalence of type expressions. [5+5]

8.a) List out and explain about the criteria are required for Code improving Transformations.
b) Define Basic block. Explain in brief about optimization of basic blocks. [5+5]

OR

9.a) Explain in brief about function preserving transformations on basic blocks.
b) List out and explain about the approaches for implementing dynamic scope. [5+5]

10.a) Explain reducible and non-reducible flow graphs with examples.
b) Explain in detail about the garbage collection via Reference Counting. [5+5]

OR

11. Explain in detail about Generic Code Generation algorithm. [10]

---ooOoo---