[Apr-24]

GITAM (Deemed to be University)
[CSEN3031]
GST/GSS/GSB/GSHS. Degree Examination

VI Semester

COMPILER DESIGN

(Effective from the admitted batch 2021-22)

Time: 2 Hours **Max. Marks: 30**

Instructions: All parts of the unit must be answered in one place only.

Section-A

1. **Answer all Questions:** **(5×1=5)**
   a) Differentiate between Token, Lexeme, Pattern with suitable example.
   b) Define FOLLOW() and list out the rules for it.
   c) Define closure(I) in LR parsers.
   d) What are the applications of Syntax Direct Translation?
   e) Define flow graph.

Section-B

Answer the following: **(5×5=25)**

UNIT-I

2. Describe the role of Lexical Analyzer in developing a compiler.

OR

3. Explain the input buffering strategy used in the lexical analysis with a neat diagram.

UNIT-II

4. Construct predictive parsing table for the given grammar and check whether it is LL(1) or not.

   S→xABC
   A→a | bbD
   B→a | ε
   C→b | ε
   D→c | ε

OR

5. Define Left Recursion. Find different ways left recursion exist in the grammar. Eliminate left recursion in the following grammar. E→E+E | E*E | id

UNIT-III

6. Prove that the following grammar is not SLR grammar.
S→AaAb | BbBa
A→ε
B→ε

OR

7. Construct LR(1) parser for the following grammar and check whether the string cdd accepted by the parser or not.
{S→AA A→cA | d }

UNIT-IV

8. Design S-attributed SDD for generating syntax tree of a given arithmetic expression using the grammar:
G {E→E+E | E*E | (E) |id } and obtain syntax tree for the given arithmetic expression " (a + b) * c".

OR

9. Write three address statements for the following expression and represent them in to quadruple, triple and indirect triples.
x=-(a+b)*(c+d)-(a-b+c)

UNIT-V

10. Eliminate the common sub expression from the following basic block using DAG.
a=b+c
b=a-d
c=b+c
d=a-d
where 'b' is not used in further blocks.

OR

11. What is peephole optimization? Explain various techniques in it.