Code No: 155EG **R18**

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY HYDERABAD
B. Tech III Year I Semester Examinations, January/February - 2023
COMPILER DESIGN
(Computer Science and Engineering – Cyber Security)

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)	Define linker and loader and explain briefly.	[3]
c)	Define ambiguous grammar.	[2]
d)	Compare SLR, CLR and LALR parsers.	[3]
e)	What is type a syntax tree?	[2]
f)	How to find evaluation order for SDD's?	[3]
g)	What are the limitations of static allocation?	[2]
h)	Write the fields and uses of symbol table.	[3]
i)	What is common sub-expression elimination? Explain.	[2]
j)	What are induction variables? What is induction variable elimination?	[3]

PART – B

(50 Marks)

2.a) Identify various Error Recovery strategies in Lexical analysis.
b) Write down the steps in constructing DFA for the regular expression (a/b)*aab(a/b)*. [5+5]

OR

3.a) How to specify the Tokens? Differentiate Token, Lexeme and Pattern with suitable examples.
b) Examine the analysis-synthesis model of a compiler. [5+5]

4.a) Consider the following grammar.
bexpr → bexpr or bterm | bterm
bterm →bterm and bfactor | bfactor
bfactor → not bfactor | (bexpr) | true | false
i) Construct a parse tree for the sentence not (true or false)
ii) Is this grammar ambiguous? Why?
b) Distinguish Top-Down Parsing and Bottom-Up Parsing. [6+4]

OR

5. Construct CLR parsing table for the following Grammar
S →L=R
S→R
L→*R
L→id
R→L (Write all necessary procedures). [10]

6. Examine syntax directed translation of Boolean expressions with and without back patching. [10]

OR

7.a) Classify the three forms of intermediate code representations? Explain them.

b) Convert the following arithmetic expression into Syntax Tree and Three Address Code b*4(a+b+c). [5+5]

8.a) Write down the code generation algorithm and explain briefly.

b) Explain Peephole optimization with an illustrative example [5+5]

OR

9.a) Construct DAG for the following basic block:

d:=b+c
e:=a+b
b:=b*c
a:=e-d

b) Explain different methods for register allocation and assignment with example. [4+6]

10. How to construct the basic block and compute DAG for the code fragment? Explain with the following code fragment.

```
Procedure fun(x,y,z)
        begin
        y=z+1;
         z=z+x;
        end fun
        begin main()
        a=2;
        b=3; use
        fun(A+B,A,B)
        print(A)
        end main;
```

[10]

OR

11. Explain the following:

a) Basic elements of Data-Flow Analysis

b) Register Allocation. [5+5]

---ooOoo---