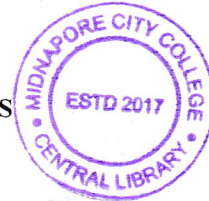


PG
MCA Semester- II Examination, 2025
MASTER OF COMPUTER APPLICATIONS
PAPER: MCA 204
(COMPILER DESIGN)



Full Marks: 100

Time: 3 Hours

The figures in the right-hand margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable.

Group-AAnswer any **FIVE** of the following questions:

2×5=10

1. Differentiate tokens, patterns and lexeme.
2. Define decorated parse tree.
3. What is Bootstrapping?
4. What do you mean by left recursion?
5. Define T-diagram.
6. Find the number of tokens in the following code:
printf("&i=%x i=%d", &i, i);
7. What are basic blocks of code optimization?
8. What are the functions of error handler?

Group-BAnswer any **FOUR** of the following questions:

15×4=60

9. (a) Calculate FIRST and FOLLOW for the following grammar?

S → a B D h

B → c C

C → b C | ε

D → EF

E → g | ε

F → f | ε

- (b) Eliminate left recursion in the following Grammar:

S → Aa | b

A → Ac | Sd | f

- (c) Consider the following Grammar:

A → abB | aB | cdg | cdeB | cdfB

Remove left factoring.

5+5+5

(P.T.O.)

(2)

10. (a) Build an LL (1) parsing table for the following grammar.

Program $\rightarrow$ begin d semi X end

$X \rightarrow d \text{ semi } X \mid SY$

$Y \rightarrow \text{semi } SY \mid \epsilon$

- (b) Consider the following grammar:

$E \rightarrow EBE \mid id$

$B \rightarrow + \mid * \mid =$

Convert the following grammar into operator grammar. Construct the

Operator parsing table. Show how the operator parser parses the input string

"id + id * id".

5+(2+5+3)

11. (a) Consider the following grammar:

$S \rightarrow AaAb \mid BbBa$

$A \rightarrow \epsilon$

$B \rightarrow \epsilon$

Compute augmented grammar and construct DFA.

- (b) Check whether it is LR (0) or SLR (1) justify. Generate the LR (0) and SLR (1) parsing table for the grammar given in 11(a). (1+4) + (5+5)

12. (a) Write quadruple, triples and indirect triples for the following expression-

$a = b * -(c-d) + b * -(c-d)$

- (b) Draw the Syntax tree and DAG for the following expression-

$-(a+b) * (c+d) + (a+b+c)$ 7+8

13. (a) If the following grammar is not LL (1), not LR (0) and not SLR (1).

Check whether it is CLR (1) or LALR (1). Generate the LR (1) parsing table for the following grammar.

$S \rightarrow Aa \mid bAc \mid dcbda$

$A \rightarrow d$

- (b) Compare SLR, CLR and LALR parsers.

- (c) Convert the following infix notation into postfix notation and generates the three-address code.

$-a + b / c \uparrow d \uparrow e * f / g$

7+4+(2+2)



(P.T.O.)

(3)

14. (a) What is SDT? Differentiate between S-attributed definition and L-attributed definition.

- (b) Consider the SDT as given below

$E \rightarrow E * T$ {E.val = E.val * T.val;}

$E \rightarrow T$ {E.val = T.val;}

$T \rightarrow F - T$ {T.val = F.val - T.val;}

$T \rightarrow F$ {T.val = F.val;}

$F \rightarrow 2$ {F.val = 2;}

$F \rightarrow 4$ {F.val = 4;}

Using the above STD construct parse tree and evaluate the string "4-2-4*2".

- (c) Define Dead-code elimination with example. (2+4) + 6+3

15. (a) What is common sub-expression and how to eliminate it? Explain with example.

- (b) Define flow graph. Give one example.

- (d) Explain with example the various techniques in Peephole optimization.

(3+1) + (2+1) + 8

3x5

16. Write short note (any three):

- (a) Cross Compiler

- (b) Code optimization

- (c) Dependency graph

- (d) Symbol table.

- (e) Phases of a compiler

(Internal Assessment: 30 Marks)

<<<<<<>>>>>

