

(6)

16. (a) Discuss how instruction selection and common subexpression elimination help in target code generation. 6
- (b) Mention the roles of each units of activation record. 5
- (c) Give an example of semantic error and explain why the error is not detected in syntax analysis phase. 4

[Internal Assessment—30]

★ ★ ★

Total Pages—06

PG/2nd Sem/MCA-204/24

2 0 2 4

M.Sc. 2nd Semester Examination

MCA

PAPER : MCA-204

(Compiler Design)

Full Marks : 100

Time : 3 hours

The figures in the right-hand margin indicate marks.

Candidates are required to give their answers in their own words as far as practicable.

Illustrate the answers wherever necessary.

Answer from **both** the Groups as directed.

GROUP—A

Answer *any five* questions : 2×5=10

1. What do you mean by ambiguous grammar?
Give one example.
2. State the function of lexical analysis phases of a compiler.

(2)

3. Differentiate between NFA and DFA.
4. "No left recursive grammar can be LL(1)." Explain.
5. What do you mean by code optimization?
6. What are basic blocks? Give one example.
7. "Every SLR(1) grammar is unambiguous but there are many unambiguous grammar that are not SLR." Give one example.
8. Define annotated parse tree.

GROUP—B

Answer *any four* questions : 15×4=60

9. (a) Let G be a context free grammar for which the production rules are given below :
$$E \rightarrow E E \mid E \mid E \mid E \mid E \mid E \mid a \mid b \mid B B$$

Derive the string $a b a b$ using the above grammar (using Left Most Derivation and Right Most Derivation). 5

/984

(Continued)

(5)

13. (a) What are basic blocks? Write the algorithm for partitioning into blocks. 2+2=4
(b) What is flow graph? Give one example. 2+2=4
(c) What do you mean by dependency graph? What is cross compiler? 2+2=4
(d) Define dead-code elimination with an example. 3
14. (a) Explain the function of symbol table. 5
(b) Write a short note on 'bootstrapping'. 5
(c) Draw the syntax tree and DAG for the following expression : 5
(a b) (c d) (a b) b
15. (a) Construct DFA of the following : 5
(0 1) (1 00)(0 1)
(b) Construct CLR parsing table of the following grammar :
$$\begin{aligned} S &\rightarrow xAy \mid xBy \mid xAz \\ A &\rightarrow qS \mid q \\ B &\rightarrow q \end{aligned}$$

Check whether the sentence 'xqy' can be derived from the grammar. 10

/984

(Turn Over)

(4)

11. (a) Consider the following grammar :

S AB|eDa
A ab|c
B dC
C eC|
D fD

Compute sets of SLR closure items. 5

(b) Construct SLR parsing table for the grammar given in 11(a). 5

(c) Show how the SLR parser parses the string 'abde'. 5

12. (a) What is intermediate code and write two benefits of intermediate code generation. 2+2=4

(b) Generate three address code : 5

While (A<C and B>D) do
 If A=1, then C=C+1;
 Else
 While A<=D do
 A=A+3;

(c) Write quadruples, triples and indirect triples for the expression : 2+2+2=6

(a b) (c d) (a b c d)

/984

(Continued)

(3)

(b) Eliminate left recursion from the following grammar : 5

R R&R |R@R|R |(R) |a|b|c

(c) Remove left factoring from the following grammar : 5

E E T |T
T id|id[] |id[X]
X E, E|E

10. (a) Calculate FIRST and FOLLOW for the following grammar : 5

S Aba|bCA
A cBCD |
B CdA ad
C eC
D bSf|a

(b) Construct predictive parsing table for the given grammar in 10(a). Show whether the given grammar is LL(1) or not. 5

(c) Show how the predictive parser parses the string 'becada'. 5

/984

(Turn Over)