Roll No. .... Total Pages : 4

220302

December, 2019

MCA- III SEMESTER

Principle of System Programming and Compiler Design

(MCA-17-203)

Time : 3 Hours] [Max. Marks : 75

Instructions :

1. *It is compulsory to answer all the questions (1.5 marks each) of Part-A in short.*
2. *Answer any four questions from Part-B in detail.*
3. *Different sub-parts of a question are to be attempted adjacent to each other.*

PART - A

1. (a) Generate language for the grammar:

$\mathbf{A \rightarrow B \mid C;\ B \rightarrow a\ B\ b \mid \in\ l;\ C \rightarrow a\ C\ b \mid \in}$

(1.5)

(b) How many minimum number of terminals are required to derive grammar for **(01/1)*(01)*?** (1.5)

(c) Remove the ambiguity from the following grammar:

E → E+E/E–E/E%E/E*E/E↑E

Where ↑ has top most priority, *, and % have second highest, + and – occupy lowest priority. All operators except ↑ are left associative. (1.5)

(d) Write the lexemes, tokens and patterns occurring in the following statement:

int j;

scanf ("j=%d, &j=%x", j,&j); (1.5)

(e) What is difference between parsing and derivation? (1.5)

(f) Write the algorithm to remove left recursion from a grammar. (1.5)

(g) Define peephole optimization. (1.5)

(h) Write difference between a phase and pass. (1.5)

(i) Write quadruple, triple and indirect triple for expression:

A=(B*C/D)+C/D. (1.5)

(j) What is significance of abstract syntax tree? (1.5)

PART - B

2. (a) For the following grammar

   D→T L ;

   L→L , id | id

   T→int | float

   Remove left recursion (if required) and Find first and follow for each non-terminal for Resultant grammar. (10)

   (b) Construct LL(1) parsing table for above grammar. (5)

3. (a) Discuss the necessity of code optimization in compilation process. (5)

   (b) Construct a Syntax-Directed Translation scheme that translates arithmetic expressions from infix into postfix notation. Show the application of your scheme to the string "3*4+5*2". (10)

4. Explain two pass assembler in detail. (15)

5. Write short note on following :

   (a) Code generation. (5)

   (b) Algorithm to convert minimizing states of DFA. (5)

   (c) Canonical collection of LR(0) item set. (5)

6. (a) Explain machine dependent and machine independent code optimization. Write the three address code for following program fragment:

If x then if a+b then c+d else c–d else c*d. (5)

(b) What is the significance of number of pass of compiler? Briefly describe how do various system programs facilitate the execution of program. (10)

7. Consider the grammar:

S → *L=R/R

L → **R/id

R → L

Construct CLR parsing table for the above grammar. (15)