B.Tech. (Sem. VII) (Main) Examination, Nov-Dec - 2011
Computer Science
7CS3 Compiler Construction

Time : 3 Hours

Total Marks : 80
Min. Passing Marks : 24

Attempt any five questions, selecting one question from each unit. All questions carry equal marks. (Schematic diagrams must be shown wherever necessary. Any data you feel missing suitably be assumed and stated clearly. Units of quantities used/calculated must be stated clearly.)

UNIT - I

1. (a) Explain the different phases of compiler design with the help of suitable diagram. **(8)**

   (b) Explain the following terms

      (i) Translators, compilers and interpreters

      (ii) Bootstrapping **(2+2)**

   (c) Illustrate the translation of the following statement on all phase of compiler A: = B * C + D/E. **(4)**

OR

1. (a) Write short note on "Input Buffering" **(2)**

   (b) Construct minimum state DFA for the following regular expressions.

   b(a/b)*a

   (a/b)* - a(a/b) (c/b)

   ((a/b) / (b/c))* **(6)**

   (c) What is role of lexical analyzer **(2)**

   (d) Define the term NFA and DFA with an example. What are the rules to get a NFA for a regular expressions? And also describe the applications and limitations of finite automata. **(6)**

UNIT - II

2. **(a)** Consider the following grammar G:

$E \rightarrow TE'$

$E' \rightarrow + TE'/\in$

$T \rightarrow FT'$

$T' \rightarrow * FT'/\in$

$F \rightarrow (E)/id$

Where $\in$ denotes the empty string of symbols

(i) Complete FIRST and FOLLOW for each non terminal of the grammar G.

(ii) Construct a predictive parsing table for grammar G. **(3+3)**

(b) What do you mean by context free grammar? Give distinction between regular and context free grammar and limitations of context free grammar. **(4)**

(c) Show whether the following grammar is LL(1) or not

$E \rightarrow TE/ + TE/\in$

$T \rightarrow FT/ * FT/ \in$

$F \rightarrow (E)/id$

And explain the model of a predictive parser. **(3+3)**

OR

2. Explain top down and bottom up parsing technique detail. **(16)**

UNIT - III

3. Write a program to translate an infix expression into postfix form. Also write syntax directed definition for the same. **(16)**

OR

3. **(a)** Define syntax directed definition. Explain the various forms of syntax directed definition. **(10)**

(b) Write specifications of a simple type checker with example. **(6)**

UNIT - IV

4. **(a)** Write short notes on

(i) Activation Record

(ii) Parameter Passing (6×2)

(b) Differentiate between stack and heap allocation. (4)

OR

4. (a) What are various runtime storage management techniques? Explain with programming examples. (10)

(b) Write short note on symbol table (6)

UNIT - V

5. (a) Construct the DAG and generate the code for given block

d; = b + c

e: = a * b

b: = b – c

a: = e * d (8)

(b) Write short note on code optimization and DAG. (8)

OR

5. Construct the tree for following expression and apply labeling algorithm for optimal ordering $x*(y + z) - z/(u - v)$. (16)