N 1253

B.E./B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2004.

Sixth Semester

Computer Science and Engineering

CS 337 — PRINCIPLES OF COMPILER DESIGN

Time : Three hours   Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1.   What are compiler construction tools?

2.   What is a preprocessor?

3.   What is a recognizer?

4.   What are the issues of the lexical analyzer?

5.   What do you mean by Viable prefixes?

6.   Define Handle.

7.   Draw the DAG for a:= b * – C + b * – C.

8.   Translate a or b and not c into three address code.

9.   How do you calculate the cost of an instruction?

10. What is a Basic block?

PART B — (5 × 16 = 80 marks)

11. (i)   What are the phases of the compiler? Explain each phase in detail. Write
           down the output of each phase for the expression a := b + c * 50.   (10)

     (ii) Describe how various phases could be combined as a pass in a compiler.
                                                                                  (6)

12. (a) Explain the process of constructing an NFA from a regular expression. Find NFA for the expression (a / b) * abb. Also convert the obtained NFA into DFA. (16)

Or

(b) (i) How would you construct a DFA directly from a regular expression? Explain with a suitable example. (10)

(ii) What are preliminary steps that are to be carried out during parsing? Explain with suitable examples. (6)

13. (a) Check whether the following grammar is a LL(1) grammar

S -> iEtS |iEtSeS| a

E -> b

Also define the FIRST and FOLLOW procedures. (16)

Or

(b) (i) Explain the error recovery in predictive parsing. (8)

(ii) Write a parser generator program for desk calculator. (8)

14. (a) Explain the various data structures used for implementing the symbol table. (16)

Or

(b) (i) What are the various ways of calling the procedures? Explain in detail. (8)

(ii) What is a Three address code? Mention its types. How would you implement the three address statements? Explain with suitable examples. (8)

15. (a) (i) What are the issues in design of a code generator? Explain in detail. (8)

(ii) Discuss about the run time storage management of a code generator. (8)

Or

(b) Explain the principle sources of code optimization in detail. (16)