H 1193

B.E./B.Tech. DEGREE EXAMINATION, MAY/JUNE 2006.

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. Draw a transition diagram to represent relational operators.

2. List four software tools that analyses the source program.

3. Compare NFA and DFA based on time and space complexity.

4. What are the goals of error handler in a parser?

5. What is phrase level error recovery?

6. What are the disadvantages of operator precedence parsing?

7. What are the methods of representing a syntax tree?

8. Give the syntax directed definition for if–else statement.

9. What are basic blocks?

10. What are the properties of optimizing compilers?

PART B — (5 × 16 = 80 marks)

11. (i) What is a compiler? Explain the various phases of compiler in detail. (10)

(ii) Give a detailed note on the compiler–construction tools. (6)

12. (a) (i) Construct a minimum state DFA for the regular expression

$(a\ b) * abc$. (16)

(ii) Explain the role of lexical analyzer. (6)

Or

(b) (i) Explain in detail about the error recovery strategies in parsing. (8)

(ii) Consider the grammar

$E \rightarrow E + E / E * E / (E) / id$. (8)

Show the sequence of moves made by the shift–reduce parser on the input $id_1 + id_2 * id_3$ and determine whether the given string is accepted by the parser or not.

13. (a) (i) For the operators given below, calculate the operator–precedence relations and operator–precedence function

id, +, *, \$. (8)

(ii) Explain the LR parsing algorithm in detail. (8)

Or

(b) Construct a canonical parsing table for the grammar given below.

$S \rightarrow CC$

$C \rightarrow cC \mid d$. (16)

14. (a) (i) Explain the various methods of implementing symbol table. (8)

(ii) Describe the various methods of implementing three–address statements. (8)

Or

(b) (i) Explain procedure calls with a neat example. (8)

(ii) Describe the method of generating syntax–directed definition for control statements. (8)

5. (a) Explain the various issues involved in the design of a code generator. (16)

Or

(b) (i) What is bootstrapping? Explain the approaches to compiler development. (10)

(ii) Discuss about the following :

Copy propagation, Dead–code elimination, and Code motion. (6)
