B 269

B.E/B.Tech. DEGREE EXAMINATION, NOVEMBER/DECEMBER 2005.

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 the various compiler construction tools?

2. What is the need for separating the analysis phase into lexical analysis and parsing?

3. Draw an NFA for (a* | b*)

4. What is an ambiguous grammar? Give an example.

5. What are kernel and non-kernel items?

6. How will YACC resolve the parsing action conflicts?

7. What are the various methods of implementing three-address statements?

8. Suggest a suitable approach for computing hash function.

9. What are the steps involved in partitioning a sequence of three address statements into basic blocks?

10. What is a cross-compiler? Give an example.

PART B — (5 × 16 = 80 marks)

11. (i) With a neat block diagram, explain the various phases of a compiler in detail. Assuming an expression, give the output of each phase. (10)

(ii) Write a note on token specification. (6)

12. (a) (i) Explain input buffering in detail. (6)

(ii) Construct a DFA for the given regular expression using Subset construction method. Also minimize the number of states. (10)

(a|b)*abb

Or

(b) (i) Give an algorithm for finding the FIRST and FOLLOW positions for a given non-terminal. (4)

(ii) Consider the grammar. (12)

$E \rightarrow TE'$

$E' \rightarrow + TE' \mid \varepsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT' \mid \varepsilon$

$F \rightarrow (E) \mid id.$

Construct a predictive parsing table for the grammar given above. Verify whether the input string id + id*id is accepted by the grammar or not.

13. (a) (i) What is a shift-reduce parser? Explain in detail the conflicts that may occur during shift-reduce parsing. (6)

(ii) For the grammar given below, calculate the operator precedence relation and the precedence functions. (10)

$E \rightarrow E + E \mid E - E \mid E * E \mid E / E \mid E \uparrow E \mid (E) \mid - E \mid id$

Or

(b) (i) Consider the grammar given below. (12)

$E \rightarrow E + T$

$E \rightarrow T$

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow (E)$

$F \rightarrow id.$

Construct an LR Parsing table for the above grammar. Give the moves of LR parser on id * id +id.

(ii) Briefly explain error recovery in LR Parsing. (4)

14. (a) (i) Explain procedure calls with a suitable example. (6)

(ii) Explain the various data structures used for symbol table construction. Compare them. (10)

Or

(b) How would you generate the intermediate code for the flow of control statements? Explain with examples. (16)

(i) Explain the simple code generator with a suitable example. (8)

(ii) Explain the various approaches to compiler development. (8)

Or

(b) (i) Explain peephole optimization with a suitable examples. (8)

(ii) Discuss in detail about the run time management. (8)
