# A 1156

B.E./B.Tech. DEGREE EXAMINATION, MAY/JUNE 2007.

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. Define Compiler.
2. What is Regular Expression?
3. What are Handles?
4. List out the actions involved in shift-reduce parsing.
5. Eliminate Left Recursion for the given grammar:

   E –> E + T | T

   T –> T * F | F

   F –> (E) | id.
6. What is three address code(TAC) and generate TAC for the statement :

   a = 2 * b + 3 * c
7. What is SDT?
8. Define Directed Acyclic Graph (DAG).
9. Optimize following the code by eliminating common sub expression :

   $y = A + B^*x + C^*(x^{**}2) + D^*(x^{**}3)$
10. What is peep-hole optimization?

PART B — (5 × 16 = 80 marks)

11. (a) Explain the various phases of compiler in detail with neat sketch. (16)

Or

(b) (i) Explain with neat sketch implementation of Lexical analyzer. (10)

(ii) Describe in detail about any three compiler construction tools. (6)

12. (a) (i) Explain the working model of top down parsing and bottom up parsing? (10)

(ii) Describe the error detection and recovery process involved in the lexical phase. (6)

Or

(b) Write the algorithm for predictive parser and parse the Input expression:

x – 2 * y using the below given grammar G : (16)

E –> E+T | E-T| T

T –> T*F | T/F | F

F –> num | id

13. (a) (i) Perform LR parsing and derive the input a (a , a ( a , a)) using the below given grammar: (10)

S' –> S

S –> a (L)

S –> a

L –> S, L

L –> S.

(ii) Perform shift reduce parsing for the input 2 * (1+3) using the grammar: (6)

E ⟶ E + T | T

T ⟶ T * F | F

F ⟶ (E) | id

Or

(b) (i) Write an algorithm for generating LR item sets and constructing SLR Parsing table. (12)

(ii) Write about LALR Parser. (4)

14. (a) Brief Intermediate code generation for Basic block, Control flow and Boolean expressions. (16)

Or

(b) (i) Explain the datastructure used for implementing symbol table. (10)

(ii) Write about Quadruple and Triple with its structure. (6)

15. (a)

(b)

15. (a) (i) Explain various code optimization techniques in detail. (10)

(ii) Generate target code for the given program segment: (6)

```
main ( )
{
    int i, j;
    i = 4;
    j = i + 5;
}
```

Or

(b) (i) Explain the various issues involved in the design of Code generation. (10)

(ii) Describe in detail about Run-time storage Management. (6)

---