23/6/23 - FN.

Reg. No. : □□□□□□□□□□□□

Question Paper Code : 50434

B.E./B.Tech. DEGREE EXAMINATIONS, APRIL/MAY 2023.

Sixth Semester

Computer Science and Engineering

CS 8602 — COMPILER DESIGN

(Common to Computer Science and Business Systems)

(Regulations 2017)

Time : Three hours Maximum : 100 marks

Answer ALL questions.

PART A — (10 × 2 = 20 marks)

1. State the role of lexical analyzer. Identify the lexemes and their corresponding tokens in the following statement: printf (“Total = %d\n”, score);

2. What is the difference between compiler and interpreter?

3. For what type of grammar, recursive descent parser cannot be constructed? Show the steps involved in recursive descent parsing with backtracking for the string cad with the given grammar: S -> cAd A -> ab | a?

4. Construct a parse tree and syntax tree for 4–6/3*5+7.

5. What are syntax directed translation schemes?

6. Determine the types and relative addresses for the identifiers in the following sequence of declarations:

   float x;

   record { float x; float y; } p;

   record { int tag; float x; float y; } q;

7. What is static allocation strategy? State its limitations.

8. State how a task is divided between calling and called program for stack updating?

9. What is peephole optimization?

10. What is a flow graph? State its role in compilation process.

PART B — (5 ×13 = 65 marks)

11. (a) List out the functions of a Lexical Analyzer? State the reasons for the Separation of analyses of programs into Lexical, Syntax, and Semantic Analyses. (13)

Or

(b) Discuss the phases of a compiler indicating the inputs and outputs of each phase in translating the statement "amount = principle + rate * 36.0". (13)

12. (a) Explain the usage of YACC parser generator in construction of a Parser with one example. (13)

Or

(b) Define an LL(1) grammar. How do you check whether the grammar is LL (1) or not? Generate LL(1) parsing table for the Grammar

$S \rightarrow iEtS \mid iEtSeS \mid a$

$E \rightarrow b$

Is Grammar in LL(1) or not? (13)

13. (a) Define syntax tree. What is s-attributed definition? Explain construction of syntax tree for the expression a-4+c using SDD. (13)

Or

(b) With a neat diagram explain the format of the Symbol Table. Discuss the tree structures representation of scope information. (13)

14. (a) Discuss how induction variables can be detected and eliminated from the given intermediate code

```
B2: i:= i+1
    t1: = 4*j
    t2: = a[t1]
    if t2 < 10 goto B2
```
(13)

Or

(b) What is an activation record? Explain stack allocation of activation records using example. (13)

15. (a) Explain different code optimization techniques available in local and global optimizations? (13)

Or

(b) Construct the DAG for the following basic block :

(i) t1:=4*i

(ii) t2:=a[t1]

(iii) t3:=4*i

(iv) t4:=b[t3]

(v) t5:=t2*t4

(vi) t6:=prod+t5

(vii) prod:=t6

(viii) t7:=i+1

(ix) i:=t7

(x) if i<=20 goto (i) (13)

PART C — (1 × 15 = 15 marks)

16. (a) Consider the following basic block of 3-address instructions:

a:=b+c x:=a+b b:=a–d c:=b+c d:=a–d y:=a–d

Write the next-use information for each line in the basic block. (15)

Or

(b) Draw transition diagrams corresponding to production rules for arithmetic expressions consisting of operators + and ^ for predictive parser. Explain how parsing takes place for the same using transition diagrams. (15)