

16114(J)

June-16

B. Tech 6th Semester Examination

Compiler Design (NS)

CS-322/IT-325

Time : 3 Hours

Max. Marks : 100

The candidates shall limit their answers precisely within the answer-book (40 pages) issued to them and no supplementary/continuation sheet will be issued.

**Note :** Attempt five questions in all, selecting one question from each of the sections A, B, C & D of the question paper and all subpart of question no. 9 (section-E) which is compulsory.

SECTION - A

1. (a) What is a translator? Discuss the role of various phases of the compiler in the translation of source program to object code. (10)
- (b) Describe the architecture of compiler. What are the different components and types of compiler? (10)

OR

2. (a) Prove the theorem that if L is accepted by deterministic finite automata, then L is denoted by a regular expression. (10)
- (b) Symbol table is necessary for compiler construction, Justify your statement with example. Also discuss the advantages and disadvantages for single and multipass compiler. (10)

[P.T.O.]

2

16114

SECTION - B

3. Translate the executable statement of the following C program

```
main()
{
    int i;
    int a [10];
    i=1;
    while (i<=10)
    {
        A[i] = 0;
        i = i+1;
    }
}
```

into

- (i) a syntax tree
- (ii) postfix notation
- (iii) three address code. (20)

OR

4. After Computing FIRST and FOLLOW functions, construct predictive parsing table for the following grammar:

```
E → TE'
E' → TE' / ε
T → FT'
T' → *FT / ε
F → (E) / id
```

(20)

## SECTION - C

5. Define Directed Acyclic Graph. By taking a suitable example show how DAG representation could be helpful in code optimization. (20)

OR

6. Write the semantic actions to generate the three-address code for function call and return statements of any language you are familiar with. (20)

## SECTION - D

7. (a) What is DAG? What are its advantages in context of optimization? How does it help in elimination of common subexpression? (10)
- (b) Specify the necessary and sufficient conditions for performing:
- (i) Constant propagation
  - (ii) Dead code elimination (10)

OR

8. Construct DAG for the following code sequence:

```

A[I] := B
*P := C
D := A[J]
E := *P
*P := A[I]

```

Assume that:

- (i) P can point anywhere.
- (ii) P point to only B or D.

(20)

[P.T.O.]

## SECTION - E

9. Answer short answer type questions:

- (a) What is lexeme? Define a regular set.
- (b) Define LR(0) ITEMS
- (c) How semantic rules are defined?
- (d) What are the benefits of using machine-independent intermediate form?
- (e) Mention the application of DAG.
- (f) Define back patching.
- (g) What are the characteristics of peephole optimization?
- (h) Give the block diagram of organization of code optimizer?
- (i) Do we really need a two-pass assembler? Why?
- (j) What are the two standard storage allocation strategies? (2×10=20)