

**2019**

**MCA**

**4th Semester Examination**

**COMPILER DESIGN**

**PAPER – MCA-403**

**Full Marks : 100**

**Time : 3 Hours**

The figures in the right-hand margin indicate full marks.

Candidates are required to give their answers in their

Own words as far as practicable.

Illustrate the answers wherever necessary.

Answer any **FIVE** of the following questions : (14x5=70)

1. a) What is Compiler ? What are the difference between compiler and interpreter? 2+2

b) Briefly describe all phases of a compiler including symbol table and error handler . 10

2. a) Compute NFA of  $(a/b)^*abb$  . 4

b) Convert the NFA to its equivalent DFA . 4

c) Write the algorithm for minimizing the number of states of an DFA . 6

3. a) Eliminate the left recursion and left factoring from the following grammar .

$E \rightarrow E + T / T$  4+4

$T \rightarrow id / id [ ] / id [ X ]$

$X \rightarrow E, E / E$

b) Find FIRST and FOLLOW for each non – terminal : 6

$S \rightarrow ABCDE$

$A \rightarrow a / \epsilon$

$B \rightarrow b / \epsilon$

$C \rightarrow c / \epsilon$

$D \rightarrow d / \epsilon$

$E \rightarrow e / \epsilon$

4. Design an LL (1) parsing table for the following grammar . 14

$S \rightarrow a A c d / B c A e$

$A \rightarrow b / \epsilon$

$B \rightarrow cf / d$

$C \rightarrow fe$

And parse the input "fefcbe"

5. Construct the LR ( 0 ) Parsing table for the following grammar 14

$S \rightarrow L = R$

$S \rightarrow R$

$L \rightarrow *R$

$L \rightarrow id$

$R \rightarrow L$

6. Construct CLR parsing table from 14

$S \rightarrow Aa$

$S \rightarrow bAc$

$S \rightarrow dc$

$S \rightarrow bda$

$A \rightarrow d$

7. Construct LALR Parsing table from 14

$S \rightarrow aAd / bBd / aBc / bAc$

$A \rightarrow c$

$B \rightarrow e$

8. a) What is Three address code ? describe different types of three address code . 2+3

Represent three address code as

i) Quadruples (ii) Triples (iii) Indirect triples 3+3+3

9. Consider the following program segment

```

For ( i = 2 ; i<=n ; i++)
    a [i] = TRUE
Count = 0 ;
S = sqrt (n) ;
for ( i = 2 ; i<= n ; i++)
    If (a [i] == TRUE )
        { Count ++ ;
for ( j = 2 * i ; j <=n ; j = j+2 )
        a [j] = FALSE ;
    }

```

- |   |   |
|---|---|
| a) Translate the program into three address code        | 6 |
| b) Identity all Basic Blocks on your three address code | 4 |
| c) Build the flow graph for three address code          | 4 |

**(Internal Assessment = 30 )**