

B.TECH. SEM -VII (COMPUTER) 2014 COURSE (CBCS) :
SUMMER - 2018
SUBJECT: COMPILER DESIGN

Day: **Thursday**
Date: **24/05/2018**

S-2018-2484

Time: **02.30 PM TO 05.30 PM**
Max Marks. 60

N.B. :

- 1) All questions are **COMPULSORY**.
- 2) Figures to the right indicate **FULL** marks.
- 3) Assume suitable data, if necessary.
- 4) Draw neat and labeled diagram **WHEREVER** necessary.

Q.1 Explain in detail the role of Lexical analyzer with the help of input buffering scheme. **(10)**

OR

Convert the regular expression $(a | b)^* abb$ into NFA with ϵ and then into a DFA using subset construction algorithm. **(10)**

Q.2 Consider the following grammar **(10)**

$S \rightarrow (L) | a$

$S \rightarrow L, S | S$

parse the input string $(a, (a, a))$ using shift-reduce parser.

OR

a) Explain with suitable example Ambiguity and Left Recursion. **(05)**

b) Write the YACC specification of a simple desk calculator. **(05)**

Q.3 Illustrate with suitable example the different forms of intermediate languages. **(10)**

Also generate the three – address code for the following program fragment :

while ($A < C$ and $B > D$) *do*

if $A = 1$ *then* $C = C + 1$

else

while $A \leq D$ *do*

$A = A + 3$

OR

Describe various representations of three address codes. Translate the expression given below into quadruples and triples. **(10)**

$x = -(a + b) * (c + d) + (a + b + c)$

Q.4 Explain the need of code optimization with example. Also illustrate loop optimization with suitable example. **(10)**

OR

Construct the DAG for the following basic block. **(10)**

$O = L * N$

$P = M + L$

$L = L * N$

$M = P - O$

Explain the various applications of DAG.

P.T.O.

- Q.5** a) Explain the various issues of code generator. (04)
b) Construct a DAG for the following three-address code statement : (06)
- $$T_1 = A + B$$
- $$T_2 = C + D$$
- $$T_3 = E - T_2$$
- $$T_4 = T_1 - T_3$$

OR

Consider the following expression (10)

W : (a-b) + (a-c) + (a-c)

- a) Translate it into three-address code.
b) Apply the algorithm code generation on it.

- Q.6** Explain with suitable example different levels of optimization and discuss preprocessor unit in detail. (10)

OR

Explain with neat block diagram of gcc architecture. Discuss different command line options for gcc compiler. (10)

* * * * *