(Following Paper ID and Roll No. to be filled in your Answer Books)

Paper ID : 110603 Roll No. | | | | | | | | | | |

B.TECH.

Theory Examination (Semester-VI) 2015-16

COMPILER DESIGN

Time : 3 Hours *Max. Marks : 100*

Note: Attempt questions from all Sections as per directions.

Section-A

Attempt all parts of this section. Answer in brief. **(2×10=20)**

Q1. (a) What is cross compiler?

(b) Whal do you mean by a regular expression?

(c) State the problems associated with the top down parsing.

(d) Differentiate quadruples and triples.

Discuss the challenges in compiler design.

Section-B

2. **Attempt any five questions from this section.** **(10×5=50)**

(a) Construct an SLR (1) parsing table for the following grammar

S→A)

A → A, P) (P, P

P → {num, num}

(b) Give the algorithm for computing precedence function. Consider the following operator precedence matrix draw precedence graph and compute the precedence function:-

(c) Define backpatching and semantic rules for Boolean expression. Derive the three address code for the following expression

P < Q or R < S and T < U

(d) Generate three address code for the following code

```
switch a+b
{
case 1: x =x+1
case 2: y = y+2
case 3: z=z+3
default :c=c-I
}
```

$S \rightarrow Aa \mid bAc \mid Be \mid bBa$

$A \rightarrow d$

$B \rightarrow d$

is LR (1) but not LALR (1).

(g) What are lexical phase errors, syntactic phase errors and semantic phase errors? Explain with suitable example.

(h) Describe symbol table and its entries. Also discuss various data structure used for symbol table.

Section-C

Attempt *any two questions* from this section. (15×2=30)

3. How DAG is different from Syntax Tree? Construct the DAG for the following basic blocks.

4. Consider the following sequence of three address codes:

1. Prod: =0
2. I: =1
3. T1:=4*I
4. T2:=addr (A)-4
5. T3:=T2 [T1]
6. T4:=addr (B)-4
7. T5:=T4 [T1]
8. T6:=T3*T5
9. Prod: =Prod+T6

10: I=I+1

11: If I<=20 goto (3)

Perform Loop Optimization.

5. Write short notes:

(i) Global Data Flow Analysis

(ii) Loop Unrolling

(iii) Loop Jamming