PVP 14

Code: CS4T1

II B.Tech - II Semester–Regular/Supplementary Examinations
October - 2020

COMPILER DESIGN

(COMPUTER SCIENCE AND ENGINEERING)

Duration: 3 hours Max. Marks: 70

PART – A

Answer ***all*** the questions. All questions carry equal marks

11x 2 = 22 M

1.

a) How do you differentiate Compiler and Interpreter?
b) Can you explain role of preprocessor?
c) How do you find the FIRST and FOLLOW for the following grammar?
   i. S→aAB | bA | ϵ
   ii. A→aAb | ϵ
   iii. B→bB | ϵ
d) How do you eliminate left recursion for the following grammar?
   i. E→E+T | T
   ii. T→T*F | F
e) Can you draw the model of an LR parser?
f) How do you explain Triple with suitable example?
g) How do you define Peephole optimization?
h) How the Garbage collection works through reference counting?

i) Can you explain LEX file structure?
j) How do you explain the Error recovery in LR parser?
k) Can you explain loop optimization?

PART – B

Answer any ***THREE*** questions. All questions carry equal marks.
3 x 16 = 48 M

2. a) Can you write LEX regular expressions? 4 M

b) What is the motivation behind the maintenance of Symbol Table? 6 M

c) Can you list out and explain the various phases of compiler with neat diagram? 6 M

3. a) Can you explain the Parsing techniques in brief? 6 M

b) How would you show the following grammar is Ambiguous

$S \rightarrow aSbS$
$S \rightarrow bSaS$
$S \rightarrow \epsilon$ 6 M

c) What are the problems with Top-Down Parsing? Explain with suitable examples. 4 M

4. a) What in Handle Pruning? Consider the grammar

E→E*E | id

Consider the string **id*id*id** using RMD explain handle. 6 M

b) Consider the following grammar

E→E+T | T

T→T*F | F

F→(E) | id

Construct SLR parsing Table. 10 M

5. a) How would you compare Static, Stack, and Heap allocations? 6 M

b) How do you 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<=D do
        A=A+B
```

10 M

6. a) Can you explain DAG representation of Block. Consider the following code and construct the DAG

```
sum=0;
for( i=0; i<=10; i++)
sum=sum+a[i];
```

10 M

b) Can you explain Machine dependent code optimization?

6 M