Sixth Semester B.E. Degree Examination, Jan./Feb. 2021
Compiler Design

Time: 3 hrs.   Max. Marks:100

Note: *Answer any FIVE full questions, selecting atleast TWO questions from each part.*

PART – A

1 a. List the phases of a Compiler. Describe the a analysis part and synthesis part of the compiler. **(05 Marks)**
   b. Write a note on Compiler Construction tools. **(05 Marks)**
   c. Draw the transition diagram for relation operators in C : $<$, $>$, $<=$, $>=$, $==$, $!=$. **(05 Marks)**
   d. Explain the concept of Input buffering scheme. **(05 Marks)**

2 a. Enlist the error recovery strategies in parser and briefly explain any three recovery strategies in parser. **(06 Marks)**
   b. Construct FIRST and FOLLOW set for the following grammar :
   $D \rightarrow T$ id ; $D \mid \in$
   $T \rightarrow BC \mid$ Struct id {D}
   $B \rightarrow$ int $\mid$ float $\mid$ char
   $C \rightarrow$ [num] $C \mid \in$. **(06 Marks)**
   c. Write down the algorithm for construction of predictive parsing table and also construction the parsing table for the given grammar.
   $E \rightarrow T$ E' E' $\rightarrow +$ T E' $\mid \in$ $T \rightarrow$ id|(E). **(08 Marks)**

3 a. Construct LR(0) automaton using CLOSURE and GOTO functions for the grammar given below. Check whether the grammar is in SLR. Justify your answer.
   $S \rightarrow L = R \mid R$
   $S \rightarrow * R \mid$ id
   $R \rightarrow L$. **(08 Marks)**
   b. Figure out different types of conflicts occur during shift reduce parsing. Discuss the situations in which these conflicts occur. **(04 Marks)**
   c. Write down the algorithms for constructing SLR parsing table and LR parsing for the given input. **(08 Marks)**

4 a. What are the limitations of SLR parser? How do you overcome these limitations? Write down the method to calculate look ahead token for canonical items. **(06 Marks)**
   b. Construct the canonical LR(1) items and the GOTO graph as well as canonical LR(1) parsing table for the following grammar $S \rightarrow$ (S) S $\mid \in$. **(10 Marks)**
   c. Build LALR automaton or parsing table for the grammar given in Q4(b). **(04 Marks)**

PART – B

5 a. Explain the concept of Syntax directed definition and translation. Define synthesized and inherited attributes. Mention the types of attributes used in bottom up and top down parsers. **(08 Marks)**

   b. Write down the Syntax directed definition for simple calculator. Construct annotated parse tree and the Syntax tree for the input string $5 * 6 + 2 * 7$. **(06 Marks)**

   c. Give semantic rules for declaration of data types and Syntax directed translation for the same using the given grammar.

$T \rightarrow B\ C \quad B \rightarrow int\ |\ float$

$C \rightarrow [num]\ C\ |\ \in.$ **(06 Marks)**

6   a. Demonstrate the concept of three address code, quadruples. Translate the arithmetic expression $f = a - (b + c) * d$ into i) Quadruples ii) Triples iii) Indirect triples. **(08 Marks)**

   b. Describe the Syntax directed translation for switch statement. **(08 Marks)**

   c. Justify the role of control statements in programming language. Write down the Syntax directed definition for flow of control statements. **(04 Marks)**

7   a. Describe the structure of activation record with neat diagram. **(05 Marks)**

   b. List out the functions and properties of memory manager, a subsystem of heap management. **(05 Marks)**

   c. Mention the steps involved in calling a function and returning from a function with the diagram. **(05 Marks)**

   d. Using the below given code for finding n^{th} Fibonacci number, build activation tree for finding 5th Fibonacci number.

```
int fib (int n)
    {if (n < 2) return 1 ;
     else return (fib (n-1) + fib (n-2));}.
```
(05 Marks)

8   a. For the following program

```
For I = 1 to 10 do
For J = 1 to 10 do
      A[I, J] = 0
   For I = 1 to 10 do
      A = [I, I] = 1.
```
(10 Marks)

   b. Explain the concept of dead code elimination and finding local common sub expressions with examples. **(10 Marks)**

*****