Qn. Set Code-1

Semester: 5th
Programme: B.Tech
Branch: IT

AUTUMN END SEMESTER EXAMINATION-2022

5th Semester B.Tech

COMPILER DESIGN

CS 3008

(For 2020 (L.E), 2019 & Previous Admitted Batches)

Time: 3 Hours Full Marks: 50

Answer any SIX questions.
Question paper consists of four SECTIONS i.e. A, B, C and D.
Section A is compulsory.
Attempt minimum one question each from Sections B, C, D.
The figures in the margin indicate full marks.
Candidates are required to give their answers in their own words as far as practicable and all parts of a question should be answered at one place only.

SECTION-A

1. Answer the following questions. [1 × 10]

(a) What is compiler? What is an Interpreter? Explain and differentiate between them.

(b) Describe all the viable prefixes for the following grammar:

S→ bSb | b.

(c) Explain address descriptor and register descriptor.

(d) Define flow graph and basic blocks.

(e) Write short notes on buffer pair.

(f) Write the rule to eliminate left recursion in a grammar. Prepare and Eliminate the left recursion for the grammar.

S → Aa | b

A →Ac | Sd |ε

(g) Count the number of tokens in the following C snippet.

```
int main()
{  printf("Roll %d", ++&&***a); /*abc*/
return 0;
}
```

(h) Draw DAG for the expression
a+a+ a+a+ a+a+ a+a

(i) What is SSA (Static Single Assignment), explain with an example.

(j) What are the various methods of implementing three address statements?

SECTION-B

2. (a) Find the FIRST and FOLLOW of the following: [4]

$S \rightarrow aBDh / bBc$
$B \rightarrow eC$
$C \rightarrow bC / \epsilon$
$D \rightarrow EF$
$E \rightarrow g / \epsilon$
$F \rightarrow f / \epsilon$
$S \rightarrow ACB / cbB / Ba$
$A \rightarrow da / BC$
$B \rightarrow g / \epsilon$
$C \rightarrow h / \epsilon$

(b) Construct the LL(1) parsing table for the following grammar: [4]

$S \rightarrow aAC | Bb$
$A \rightarrow eD$
$B \rightarrow f | g$
$C \rightarrow h | i$
$D \rightarrow bE | \epsilon$
$E \rightarrow eD | dD$

3. (a) Describe the various phases of the compiler and describe in detail about symbol table. [4]

(b) Consider the following statement : Float i , j; i = i * 70 + j + 2 Develop the output at all phases of the compiler for this code. [4]

SECTION-C

4. (a) Construct the Canonical LR parsing table for the grammar [4]

   S→ aAb/ bB
   A→ Aa/ ϵ
   B→Bb/ ϵ

   (b) Construct the LALR parsing table for the following grammar. [4]

   S→ aAd | bBd | aBe | bAe
   A→ c
   B→ c

5. (a) Explain Synthesized and Inherited attributes. Evaluate the value of following SDT, given the integer values as 3,4 and 5 respectively. Also provide the topological sorting. [4]

   E -> E+T { E.val = E.val + T.val }
   E -> T { E.val = T.val }
   T -> T*F { T.val = T.val * F.val }
   T -> F { T.val = F.val }
   F -> INTLIT { F.val = INT.lexval }

   (b) Evaluate the expressions for the SDD annotated parse tree for the follow expressions. [4]

   3 * 5 + 4n
   3 * 5

   Explain a type checker which can handle expressions, statements and functions.

6. (a) Explain various methods of code optimizations. [4]

   (b) Translate the statement x=(a+b)*-c/d into: [4]

   i. Quadruples.
   ii. Triples.
   iii. Indirect Triples.

SECTION-D

7. (a) C program. [4]

```
i = 0,j=0, s = 0;
for(i=0;i<=10;i++)
{
    for(j=0;j<=10;j++)
    {
     a[i][j]+=b[i][j]
       s = s+ a[i] [j];
    }
}
```

Write down the three address code for the following code segment.

(b) For the above program, define DAG and construct the DAG for the following three address codes. [4]

8. (a) Explain in detail about optimization of basic blocks. [4]

(b) Construct the DAG for the following Basic block & explain it. [4]

```
1. t1: = 4 * i
2. t2:= a [t1]
3. t3: = 4 * i
4. t4:= b [t3]
5. t5:=t2*t4
6. t6:=Prod+t5
7. Prod:=t6
8. t7:=i+1
9. i:= t7
10. if i<= 20 goto (1).
```

*****