53 (CS 601) CPDG

2018

COMPILER DESIGN

Paper : CS 601

Full Marks : 100

Time : Three hours

The figures in the margin indicate full marks for the questions.

Answer Question No. ***1*** *and* ***any three*** *from the rest.*

1. Consider the following grammar

$$A \rightarrow A + B \mid B,\ B \rightarrow P,\ P \rightarrow id$$

(a) Compute FIRST and FOLLOW.

(b) Construct the LR (1) items for the above grammar.

(c) Create canonical LR (1) parsing table using those LR (1) items.

(d) Create LALR parsing table.

(e) Check whether $id+id+id$ will be accepted by your LALR parser.

10+10+10+5+5

2. Consider the regular expression

$$ab(ab)^*(a|b)^*$$

(a) Design the corresponding NFA using the McNaughton-Yamada-Thompson algorithm.

(b) Convert the NFA to an equivalent DFA.

(c) Minimize the number of states.

5+10+5

3. (a) With an example, discuss why left recursion is removed in top down parsing but not in bottom up parsing.

(b) Why there is a requirement to convert an augmented grammar in bottom up parsing?

(c) Discuss about the semantic errors with examples. 10+5+5

4. Consider the following function :

```
void mult (int n)
{
    int i,j,k ;
    for (i=0 ; i<n ; i++)
    {   for (j=0 ; j<n ; j++)
            a[i] [j]=0 ;
    }
    for (i=0 ; i<n ; i++)
    {
        for (j=0 ; j<n ; j++)
    {
        for (k=0 ; k<n ; k++)
            a[i] [j]=a[i] [j]+b[i] [k] * c[k] [j]
    }
}
```

(a) Generate three address code for the above fragment.

(b) Create flow graph for it.

(c) Remove local and global common sub-expression. 5+5+10

5. *(a)* Consider the following instruction of a program.

$$X = Y + Z/IO$$

Show the outputs of the different phases of compiler.

(b) Write short notes on :

(i) left factoring

(ii) dead code elimination.

10+(5+5)