**SET – 2128**

1. Construct a DFA for the following regular expression: a \* (a | b) a a

2. Consider the following grammar:

tuple --> ( list )

list --> list a

list --> a

Start symbol is ‘tuple’. (a) It is a left-recursive grammar. Remove left-recursion and compute FIRST and FOLLOW sets of the modified grammar. (b) Write at least two strings generated from this grammar. Show their rightmost derivation using the original grammar. At each step show the handle of the derivation. (c) Construct LR(0) item set. Construct the SLR parsing table. 8+6+(2+6+8)=30

**SET-503:-**

1. What is the importance of regular grammar in compiler design?

2. (a) Write Regular expression for all strings of digits that represent odd numbers.

(b) Construct an NFA for the above regular expression.

3. Explain the difference between CFG and CSG.

4. For the following grammar, compute the FOLLOW set for each non terminal.

Construct an SLR parsing table (terminals of the grammar are for, to, id, := , num and print. S is the start symbol)

S-> for id := E to E S

S-> print E

S-> id

S-> num

(3+10+3+10 =26 )

**SET-211: -**

1. What is the purpose of semantic analysis? What is syntax-directed definition (SDD) ? What are synthesized and inherited attributes?

2. Write an SDD for declaration of list of variables in a ‘C’ like language. Consider the types ‘integer’, ‘real’ and ‘char’. Using the SDD, give an annotated parse tree for a declaration statement of two variables of type ‘real’. Also draw a dependency graph.

3. Optimize the following code and discuss each optimization technique that you have applied stating their advantages:

for (i=0 ; i<n ; i++)

{

for(j=0; j<n; ++)

{

If(i%2)

{

x +=(4\*j + 5\*i);

y+=(7 + 4\*j );

}  
 }

}

4. With an example explain what are ‘liveness’ and ‘next use’ of variables. (6+7+7+6=26)