Name : .....

Roll No. : .....

Invigilator's Signature : .....

CS/M.Tech(SE, CSE)/SEM-2/PGSE-203, PGCSE-204A/2013

2013

PRINCIPLES OF LANGUAGE TRANSLATION

Time Allotted : 3 Hours *Full Marks* : 70

The figures in the margin indicate full marks.

Candidates are required to give their answers in their own words as far as practicable.

Answer any *five* questions. 5 × 14 = 70

1. a) What are meant by 'lexeme' and 'token' ?

   b) Give regular expressions for the following languages on $\sum = \{ a, b \}$:

      i) All strings containing at least one '*b*'

      ii) All strings containing at least two '*a*'.

   c) Convert the following regular expression to NFA :

      (a|b)a*b*

   d) Convert the NFA to DFA and minimize the DFA.

   2 + 2 + 2 + 8

2. a) What is meant by 'terminal' and 'non-terminal' in a context free grammar ?

b) Eliminate Left Recursion in the following grammar :

S → a|↑|(T)|T

T → T,S|a|↑|(T)

c) Test ambiguity in the following grammar :

S → +SS|–SS|a

d) Consider the following grammar :

S → L = R|R

L → *R|id

R → L

i) Construct a predictive parse table for the above mentioned grammar.

ii) Show the actions of the parser for the input string "id = *id" 2 + 2 + 2 + 8

3. a) Show the shift-reduce conflict in SLR parsing table for the following grammar :

S → iCtS

S → iCtSeS

S → a

C → e

b) Test whether the CLR parser can resolve the conflict.

6 + 8

4. a) Construct the operator precedence parsing table for the following grammar :

   exp → exp + exp|exp*exp|(exp)|id|a

   b) Consider the sentence id*id + id for parsing and show the parsing action. 9 + 5

5. a) What is the type expression for "int [2][3]" ?

   b) Write SDD for the following grammar and show the dependency graph for "int [2][3]" :

   T → BC

   B → int

   B → float

   C → [num]C_1

   C → E

   c) Write SDD for the following grammar and show the dependency graph for "a – b + 5"

   E → T$E^{|}$

   $E^{|}$ → + T$E^{|}$

   $E^{|}$ → – T$E_1^{|}$

   $E^{|}$ → – E

   T → (E)

   T → id

   T → num 2 + 6 + 6

6. a) Write three address code for the following source code

```
while ( x < y ) {
    z = 5;
    for ( a = 0; a < = 10; a++ ) {
        if ( c < d ) k = 1;
        else k = 2;
    }
}
```

b) Generate code for the following expression using labelled tree DAG.

(a – b) * (a – c – b) + (a – b – c) 7 + 7

7. Write short notes on any *two* of the following : 7 + 7

a) Symbol table

b) LALR parsing

c) Register allocation problem

d) Predictive parsing.