Department of CSE/IT
Assignment-III
CS6660-Compiler Design

Part-A

1. Define a context free grammar.
2. What is meant by left recursion? Eliminate left recursion for the grammar. E→E+T/T, T→T*F/F F→(E)/id
3. Write the algorithm for FIRST and FOLOW
4. What is LL(1) grammar?
5. Write the algorithm for the construction of a predictive parsing table?
6. Define LR (0) item.
7. Left factor the following grammar
   S → iEtS | iEtSeS |a
   E → b.
8. What is the syntax for YACC source specification program?

Part-B

1) a) Consider the grammar given below:
   E → E+T
   E → T
   T → T*F
   T → F
   F → (E)
   F → id
   Construct an LR parsing side for the above grammar.
   Give the moves of LR parser on id*id+id
   b) Check whether the following grammar is a LL(1) grammar
   S → iEtS | iEtSeS | a
   E → b Also define the FIRST and FOLLOW procedures.

2) a) Consider the following grammar(
   E→E+T | T
   T→TF | F
   F→F* | a | b construct the SLR parsing table for this grammar.
   Also parse the input a*b+a .
   b) Construct LALR parsing table for the grammar.
   E→ E+T | T
   T→ T*F | F
   F→ (E) | id
3) a) Construct non recursion predictive parsing table for the following grammar.
   E→E or E /E and E/ not E / (E) / 0 /1.
   b) Write the algorithm to eliminate left-recursion and left-factoring and apply both to the following grammar.
   E-->E+T|E-T|T
   T-->a|b|(E)
4) a) Construct predictive parsing table and parse the string NOT(true OR false)
   bexpr→bexpr OR bterm | bterm
   bterm→bterm AND bfactor | bfactor
   bfactor→NOT bfactor | (bexpr) | true | false
   b) Write short notes on Yacc and also explain the design of a Syntax Analyzer for a Sample Language