CSL302: Compiler Design

Lab-8 (Intermediate Code Generation)

Due Date: 24-October-2025

Instructions:

1. Prepare a zip file and upload your solutions on canvas
2. Include a readme file

Task-1: Execute the demo files

Download the demo lex and yacc files discussed in the class from canvas (corresponding to TAC tutorial). Try to compile the lex files and yacc files. Execute them and check if you are getting the expected output after executing them.

You can use the following steps for compiling and running

1. lex <filename>.l
2. yacc -d <filename.y>
3. gcc lex.yy.c y.tab.c
4. ./a.out

Task-2: Three Address Code Generation

1. Extend the arithmetic expression grammar to generate the three address codes for the below boolean expressions.
   a. E relop E
   b. E or E
   c. E and E
   d. not E
   e. true
   f. false

   Here relop can be <, >, <=, >=, ==, !=

2. Extend the above grammar to implement semantics for the variable declaration. Once a variable is declared, its value should be entered in the symbol table. Two data types int and float are supported.

   Example of variable declaration:
   int a, b, c; should be valid
   float e, f, g; should be valid.

   At the time of variable declaration, you are required to check if the variable is already defined; if yes, then you should throw an error. Otherwise, its name and type should be entered in the symbol table.

   Note: You can use the demo code discussed for the symbol table lecture (Lab-6).