Assignment No 4

Aim: Generating intermediate Code for assignment statement using LEX and YACC.

Objective:

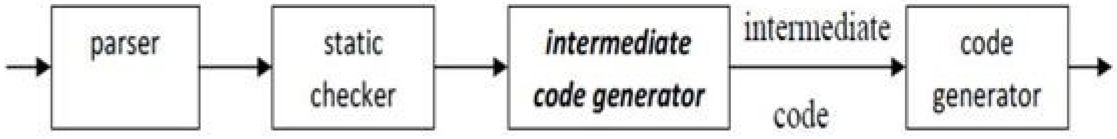
1. To understand fourth phase of compiler: Intermediate code generation.
2. To learn and use compiler writing tools.
3. To learn how to write three address code for given assignment statement.

Software Requirement:

1. Linux Operating System
2. Lex compiler
3. Yacc compiler

Mathematical Model:
Consider a set S consisting of all the elements related to a program.The mathematical model is given as below,
S=s,e,X,Y,Fme,DD,NDD,Mem shared
Where, s = Initial State
e = End State
X = Input data.Here it is assignment statement.
Y = Output.Here output is intermediate code for assignment statement.
Fme = Algorithm/Function used in program.for eg.addquad(),display()
DD = Deterministic Data
NDD = Non deterministic Data

Position of intermediate code generator



THEORY :

Introduction:

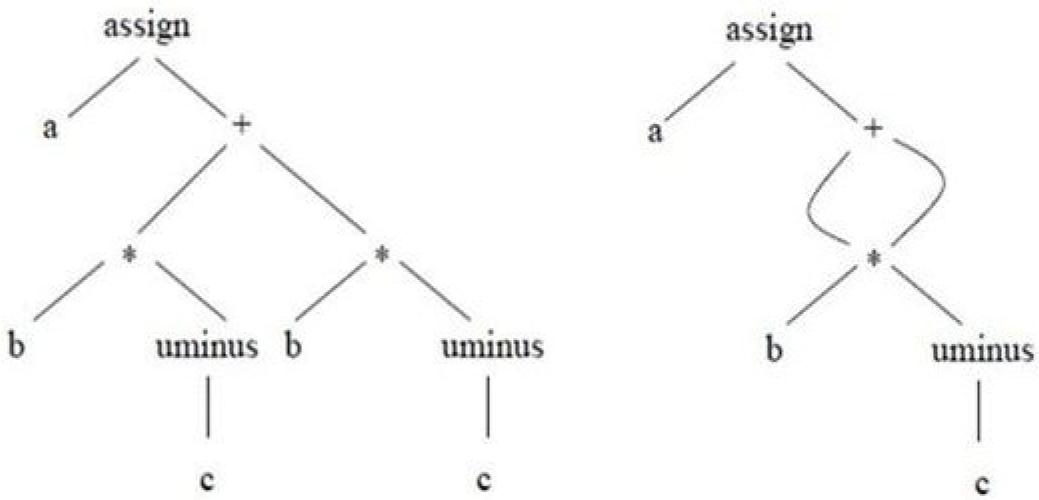
In the analysis-synthesis model of a compiler, the front end analyzes a source program and creates an intermediate representation, from which the back end generates target code. Ideally, details of the source language are confined to the front end, and details of the target machine to the back end. The front end translates a source program into an intermediate representation from which the back end generates target code.With a suitably defined intermediate representation, a compiler for language i and machine j can then be built by combining the front end for language i with the back end for machine j. This approach to creating suite of compilers can save a considerable amount of effort: m x n compilers can be built by writing just m front ends and n back ends.

Benefits of using a machine-independent intermediate form are:

1. Retargeting is facilitated. That is, a compiler for a different machinecan be created by attaching a back end for the new machine to an existing front end.
2. A machine-independent code optimizer can be applied to the intermediate representation.

Intermediate Languages:
Three ways of intermediate language representation:

1. Syntax Tree:
   A syntax tree depicts the natural hierarchical structure of a source program. A dag (Directed Acyclic Graph) gives the same information but in a more compact way because common subexpressions.



(a) Syntax tree (b) DAG

2. Postfix notation
   Postfix notation is a linearized representation of a syntax tree; it is a list of the nodes of the tree in which a node appears immediately after its children. The postfix notation for the syntax tree given above is, a b c uminus * b c uminus ∗+ assign

3. Three Address Code

Three-address code is a sequence of statements of the general form,

$$x := yopz$$

where x, y and z are names, constants, or compiler-generated temporaries; op stands for any operator, such as a fixed- or floating-point arithmetic operator, or a logical operator on Boolean valued data. Thus a source language expression like x+ y*z might be translated into a sequence,

$$t1 := y * z$$

$$t2 := x + t1$$

where t1 and t2 are compiler-generated temporary names. The reason for the term three-address code is that each statement usually contains three addresses, two for the operands and one for the result.

Types of Three-Address Statements :

1. Assignment Statements: X:= Y op Z
2. Unary Assignment Statements: X:= op Z
3. Copy Statements: X:= Y
4. Unconditional Jump: goto L,with L a label of a statement.
5. Conditional Juml: if X relop Y goto L
6. Procedure Call: param x, and call p,n for caling a procedure,p,with nparameters.return Y is the returned value of the procedure:
   param x1 param x2
   ...
   param xn
   call p,n

7. Indexed Assignments: X:=Y[i] or x[i]:=Y

8. Pointer Assignments: X:=&Y,X:= ∗Y,or *X:=Y;where &Y stands for the address of Y, and *Y the value of Y.

Translation scheme for Assignment Statement :
Consider the following grammar for assignment statement.
S − >id=E
E − > E1 + E2
E − > E1 * E2
E − > -E1
E − > (E1)
E − > id

Advantages of three-address code :

1. The unraveling of complicated arithmetic expressions and of nestedflow-of-control statements makes three-address code desirable for target code generation and optimization.
2. The use of names for the intermediate values computed by a programallows three address code to be easily rearranged unlike postfix notation.

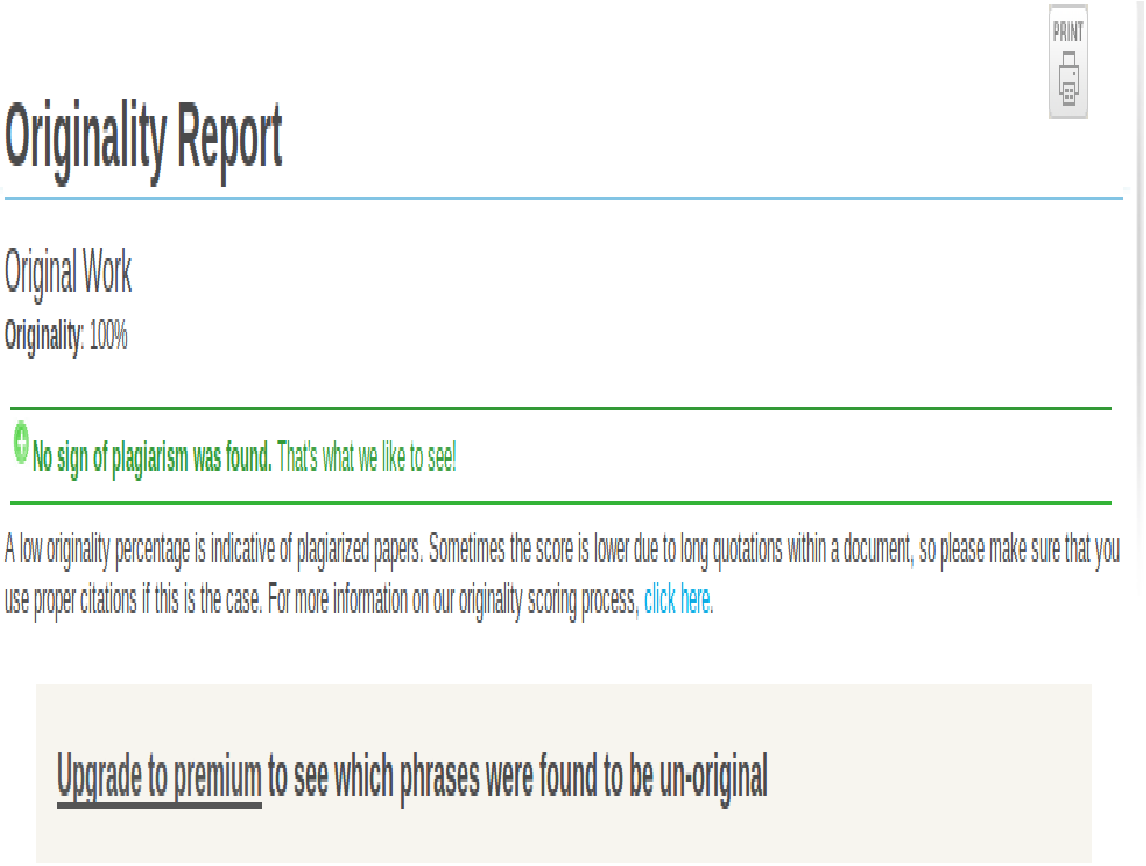
Command

```
: $ lex <program name>.l
$ yacc -d <program name>.y
$ gcc lex.yy.c y.tab.c -ll -ly
$ ./a.out<input.txt
```

CONCLUSION :
Thus, we have implemented LEX and YACC program to generate an intermediate code for assignment statement.

Roll No	*Name of Student*	*Date of performance*	*Date of Checking*	*Signature of Sta*
*BECOC*357	Sunny Shah	01 / 09 / 2017	20 / 09 / 2017	

1 PLAGARISM REPORT :



Originality Report

Original Work

Originality: 100%

No sign of plagiarism was found. That's what we like to see!

A low originality percentage is indicative of plagiarized papers. Sometimes the score is lower due to long quotations within a document, so please make sure that you use proper citations if this is the case. For more information on our originality scoring process, click here.

Upgrade to premium to see which phrases were found to be un-original

Figure 1: Plagarism Checker www.smallseotools.com/plagarism-checker