**PRACTICAL # 14**

**OBJECT:**

Implementation of while loop statement CFG.

**THEORY:**

Loops are used to repeat a block of code. There are different types of loops in a programming language. In the previous lab, we designed CFG for while loop statement as shown below.

*<Wh\_st> -> while (<Cond>) <Lp\_body>*

*<Cond> -> <ID\_Const> <Rel\_Exp>*

*<ID\_Const> -> ID | <Const>*

*<Rel\_Exp> -> Rel\_Op <ID\_Const> | ε*

*<Lp\_body> -> ; | <S\_st> | {<M\_st>}*

*<S\_st> -> <Wh\_st> | <DoWh\_st> | <If\_st> | <For\_st> | <Dec\_st> | return<Const> | break; | continue;*

*<M\_st> -> <S\_st> <M\_st> | ε*

**Program:**

The program below implements the simplified while loop CFG.

*using System;*

*class SyntaxAnalyser {*

*// <while\_st> -> while ( <cond> ) <while\_body>*

*// <cond> -> ID RELOP ID*

*// <while\_body> -> ; | <s\_st> | { <m\_st> }*

*// <s\_st> -> <while\_st> | <assign\_st> | <if\_st>*

*// <m\_st> -> <s\_st> <m\_st> | E*

*public static bool while\_St (List<Token> tokens, ref int i){*

*if (tokens[i].TokenClass == "while") {*

*i++;*

*if (tokens[i].TokenClass == "(")*

*{*

*i++;*

*if ( cond(tokens, ref i) == false) {*

*Console.WriteLine("condition syntax error at line: {0}", tokens[i].LineNum);*

*return false;*

*}*

*i++;*

*if (tokens[i].TokenClass == ")")*

*{*

*i++;*

*if (while\_body(tokens, ref i) == true) {*

*Console.WriteLine("while loop parsed successfully");*

*}*

*}*

*else {*

*Console.WriteLine(" ) missing at line number {0}", tokens[i].LineNum);*

*return false;*

*}*

*}*

*else {*

*Console.WriteLine("syntax error at line: {0}", tokens[i].LineNum);*

*return false;*

*}*

*}*

*return true;*

*}*

*public static bool cond(List<Token> tokens, ref int i) {*

*if (tokens[i].TokenClass == "ID")*

*{*

*i++;*

*if (tokens[i].TokenClass == "relop") {*

*i++;*

*if (tokens[i].TokenClass == "ID")*

*{*

*return true;*

*}*

*else {*

*Console.WriteLine("ID missing after relational operator");*

*}*

*}*

*}*

*return false;*

*}*

*public static bool while\_body(List<Token> tokens, ref int i)*

*{*

*if (tokens[i].TokenClass == ";")*

*{*

*//Console.WriteLine("while/if loop parsed successfully");*

*return true;*

*}*

*else if (tokens[i].TokenClass == "{") {*

*i++;*

*m\_st(tokens, ref i);*

*if (tokens[i].TokenClass == "}")*

*{*

*//Console.WriteLine("while loop parsed successfully");*

*return true;*

*}*

*else {*

*Console.WriteLine(" Syntax error in while loop ");*

*return false;*

*}*

*}*

*else*

*{*

*s\_st(tokens, ref i);*

*}*

*return true;*

*}*

*public static bool s\_st(List<Token> tokens, ref int i)*

*{*

*if (while\_St(tokens, ref i) == false*

*&& if\_st(tokens, ref i) == false)*

*{*

*return false;*

*}*

*return true;*

*}*

*public static void m\_st(List<Token> tokens, ref int i)*

*{*

*if (tokens[i].TokenClass == "while")*

*{*

*if (s\_st(tokens, ref i))*

*{*

*i++;*

*m\_st(tokens, ref i);*

*}*

*}*

*else*

*{*

*return;*

*}*

*}*

*}*

Execute the program to observe the output.

**ACTIVITIES**

**Activity 1**

Implement the while loop statement CFG of your own language.

**Activity 2**

Implement the for loop statement CFG of your own language.

**REVIEW QUESTIONS**

1. What purpose do loop statement serves?
2. What is an infinite loop?
3. How is a loop statement terminated?