Overview

Name: Mantvydas Luksas

Student ID: R00150390

This is an overview of the clang generated Abstract Syntax Tree for the while loop of the t2.c file. The Abstract Syntax tree file known as t2.ast was generated for the c function fred().

Starting with line 5 of the t2.c file, the while statement is modelled through the instance of the class WhileStmt which is in the AST file in line 25.

```
|-WhileStmt 0x1c54138 <line:5:1, line:9:1>
```

This represents the while statement which is followed by the expression (i < 10), followed by the opening curly brackets for a series of statements of the while loop. In t2.ast the expression (i < 10) is first represented by the conditional operator < which is modelled as an instance of the "BinaryOperator" class.

```
|-BinaryOperator 0x1c08a80 <line:5:8, col:10> 'int' '<'
```

This node represents a built-in binary operation expression such as "x + y" or "x <= y". The node has children on either side of the operator "<" in the code at line 5 in t2.c, with the left child node being the variable "i" and the right child node being the integer value of 10. These children are represented by the "-" symbol below the parent node which is "BinaryOperator".

```
-WhileStmt 0x1c54138 <line:5:1, line:9:1>
 |-<<<NULL>>>
 |-BinaryOperator 0x1c08a80 <line:5:8, col:10> 'int' '<'
 | |-ImplicitCastExpr 0x1c08a68 <col:8> 'int' <LValueToRValue>
 | | `-DeclRefExpr 0x1c08a20 <col:8> 'int' lvalue Var 0x1c08820 'i' 'int'
 | `-IntegerLiteral 0x1c08a48 <col:10> 'int' 10
```

The left child node is modelled by the instance of the "DeclRefExpr" class which represents a reference to a declared variable, function, enum, etc. This will generate an error if the variable "i" was not previously declared in the code. The right child node represents the value 10 which is modelled as an instance of the "IntegerLiteral" class to represent any integer value. The "BinaryOperator" node is a child of the "WhileStmt" node as displayed in the picture above. The "BinaryOperator" together with its children are displayed from lines 27 to 30 in the t2.ast file.

As mentioned previously, the while loop begins with the opening of the curly braces for a series of statements within the while loop. This is modelled through an instance of the "CompoundStmt" class. The node represents a group of statements like { statement statement}.

The children of the "CompoundStmt" node are the if statement in line 6 of t2.c file and the expression "i=i+1;" in line 8 of t2.c file. The if statement is modelled by the instance of the class "IfStmt" which represents an if/then/else. This node is present in line 32 of the ast file and its children are two expressions represented by two nodes of BinaryOperator class from line 34 to 38. The first child represents the (i&1) statement in line 6 of the t2.c code. The "BinaryOperator" node is the "&" operator with its left child being the node "DeclRefExpr" for variable i and the right child being the IntegerLiteral for the value of 1. This value of 1 is clearly shown in the ast in line 37 where it states "int 1".

```
-IfStmt 0x1c54008 <line:6:4, line:7:13>
 |-<<<NULL>>>
 |-BinaryOperator 0x1c53ea0 <line:6:8, col:10> 'int' '&'
 | |-ImplicitCastExpr 0x1c08af0 <col:8> 'int' <LValueToRValue>
 | | `-DeclRefExpr 0x1c08aa8 <col:8> 'int' lvalue Var 0x1c08820 'i' 'int'
 | `-IntegerLiteral 0x1c08ad0 <col:10> 'int' 1
```

The second child being another “BinaryOperator” node for the expression part of the if statement in line 7 of the t2.c file.

The code “j=j+i+10;” expression is represented primarily by this second child “BinaryOperator” node which exists on line 38 of t2.ast file. This node represents the “=” sign in the expression. To the left of this is the variable “j” which is modelled through the node “DeclRefExpr” as the first child of the “BinaryOperator” node at line 39 of the t2.ast file. To the right of the “=” sign is the second child which is another BinaryOperator node at line 40 of the t2.ast file to represent the second “+” operator at column 13.

This is not the first “+” operator, but the last. On either side of it are its children. The child on the left is the other “+” operator at column 11 and the child on the right is the integer value of 10. This is clearly visible as this is modelled in the ast file as “BinaryOperator” node with the children being the other “BinaryOperator” node at line 41 and the IntegerLiteral node at line 46.

```
-BinaryOperator 0x1c53fe0 <line:7:7, col:13> 'int' '='
 |-DeclRefExpr 0x1c53ec8 <col:7> 'int' lvalue Var 0x1c08890 'j' 'int'
 `-BinaryOperator 0x1c53fb8 <col:9, col:13> 'int' '+'
   |-BinaryOperator 0x1c53f70 <col:9, col:11> 'int' '+'
   | |-ImplicitCastExpr 0x1c53f40 <col:9> 'int' <LValueToRValue>
   | | `-DeclRefExpr 0x1c53ef0 <col:9> 'int' lvalue Var 0x1c08890 'j' 'int'
   | `-ImplicitCastExpr 0x1c53f58 <col:11> 'int' <LValueToRValue>
   |   `-DeclRefExpr 0x1c53f18 <col:11> 'int' lvalue Var 0x1c08820 'i' 'int'
   `-IntegerLiteral 0x1c53f98 <col:13> 'int' 10
```

The “BinaryOperator” node located at line 41 has two children nodes at line 43 and 45 of the t2.ast file. The nodes are “DeclRefExpr” to represent the “j” variable reference and the “i” variable reference.

The last expression present in the function is outside of the if statement. The expression “i=i+1;” at line 8 of the t2.c file is modelled by another instance of the “BinaryOperator” class. The node at line 48 of the t2.ast file is the second child of the “CompoundStmt” node representing the “=” symbol. To the left of “=” symbol is the variable “i” which is represented as the first child of the node, modelled as an instance of the “DeclRefExpr” class referencing the variable “i”. This is visible at line 49 of the t2.ast file. The second child to the right of the “=” symbol is another instance of the BinaryOperator class representing the “+” operator in the expression. This node is present in line 50 of the t2.ast file. Its first child is a reference to the variable “i” through the node “DeclRefExpr” and the second child represents the integer value 1 through the node “IntegerLiteral”.

```
-BinaryOperator 0x1c540e8 <line:8:4, col:8> 'int' '='
 |-DeclRefExpr 0x1c54038 <col:4> 'int' lvalue Var 0x1c08820 'i' 'int'
 `-BinaryOperator 0x1c540c0 <col:6, col:8> 'int' '+'
   |-ImplicitCastExpr 0x1c540a8 <col:6> 'int' <LValueToRValue>
   | `-DeclRefExpr 0x1c54060 <col:6> 'int' lvalue Var 0x1c08820 'i' 'int'
   `-IntegerLiteral 0x1c54088 <col:8> 'int' 1
```

The NULL children of the nodes represent future expressions that can be added to the statements. The first child of the WhileStmt node is the BinaryOperator node at line 27 and the second child is the CompoundStmt node at line 31 of the t2.ast file.