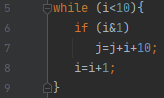
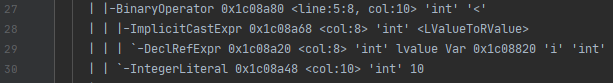


t2.ast t2.c





In t2.ast, line 25 represents a while statement. This corresponds with line 5 of t2.c where the while loop is initialised. A loop executes a block of code continuously, stopping either when the condition has been satisfied, or at a break. In t2.ast, the <line:5:1, line:9:1> corresponds with the block of code inside the while loop in t2.c.

Lines 27-30 of t2.ast correspond to line 5 of t2.c. A binary operator refers to a binary expression, in this case using relational operators. The test condition of the while loop is such that the loop is true when the value of integer i is less than that of the value of integer 10. While this condition is true, the loop will continue to iterate. Line 28 implicitly casts the expression as an int. In the C programming language, this will always product RValues. RValue is used to represent the result of the evaluated expression. LValue references may be an LLVM pointer, or a pointer plus a bitrange. Line 29 declares a reference to the variable i of type integer. Line 30 declares an integer literal of 10, as this value is explicitly represented in the source code.



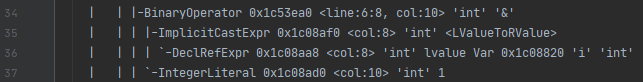


Line 31 refers to a group of statements within the curly brackets. This CompoundStmt is the closing bracket of the while loop.



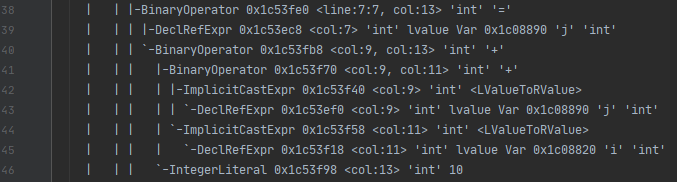


Line 32 is where the if statement begins.



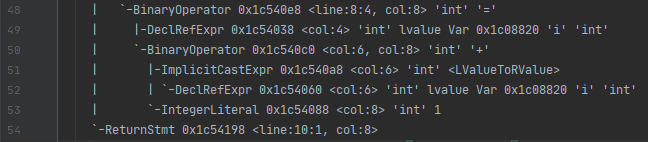


Line 34 again shows BinaryOperator, in this case it is for the bitwise operator AND. This relates to line 6’s (i&1). AND stipulates that both sides must be true for the whole to be true. In C, AND copies a bit to the result if it true for both. Lines 35-37 again implicitly casts the expression as an integer, references the declared variable int i, and declares a literal integer of 1, as it appears in the source code.





Lines 38-46 relate to just one line of the while loop, line 7. Here, we perform a binary operation, reference a declared variable int j, perform a binary operator adding the operands i+10, followed by another adding the operands j+i. Next is an implicit type conversion of integer, with a reference to declared variable int j. As follows, we have an implicit type conversion of integer, with a reference to declared variable int i. 10 is a literal integer as it appears in the source code. Essentially, we are setting the new value of j to be the sum of (j+i+10).





Lines 48-54 relates to line 8 of the while loop. Here, we perform a binary operation, with a reference to the declared variable int i. We perform a binary operation, adding the operands i+j. We have an implicit type conversion of integer, another reference to the declared variable int i, with a declaration of a literal integer 1. In effect, we are setting the new value of i to be the sum of (i+1). Thus concludes the while loop statement block.