Static program analysis in Clang

"Don't do your computer's job!"

```
TranslationUnitDecl 0x4825700 <<invalid sloc>> <invalid sloc>
|-TypedefDecl 0x4825c90 <<invalid sloc>> <invalid sloc> implicit __int128_t '__int128'
| `-BuiltinType 0x4825970 '__int128'
|-TypedefDecl 0x4825d00 <<invalid sloc>> <invalid sloc> implicit __uint128_t 'unsigned __int128'
| `-BuiltinType 0x4825990 'unsigned __int128'
|-TypedefDecl 0x4826048 <<invalid sloc>> <invalid sloc> implicit __NSConstantString 'struct __NSConstantString_tag'
| `-RecordType 0x4825df0 'struct __NSConstantString_tag'
|   `-CXXRecord 0x4825d58 '__NSConstantString_tag'
|-TypedefDecl 0x48260e0 <<invalid sloc>> <invalid sloc> implicit __builtin_ms_va_list 'char *'
| `-PointerType 0x48260a0 'char *'
|   `-BuiltinType 0x4825790 'char'
|-TypedefDecl 0x485b590 <<invalid sloc>> <invalid sloc> implicit __builtin_va_list 'struct __va_list_tag [1]'
| `-ConstantArrayType 0x48263c0 'struct __va_list_tag [1]' 1
|   `-RecordType 0x48261d0 'struct __va_list_tag'
|     `-CXXRecord 0x4826138 '__va_list_tag'
`-FunctionDecl 0x485b6c8 </home/prazek/troll.cc:2:1, line:17:1> line:2:7 Q_rsqrt 'float (float)'
  |-ParmVarDecl 0x485b600 <col:16, col:22> col:22 used number 'float'
  `-CompoundStmt 0x485c260 <line:3:1, line:17:1>
    |-DeclStmt 0x485b820 <line:4:2, col:8>
    | `-VarDecl 0x485b7c0 <col:2, col:7> col:7 used i 'long'
    |-DeclStmt 0x485b940 <line:5:2, col:13>
    | |-VarDecl 0x485b850 <col:2, col:8> col:8 used x2 'float'
    | `-VarDecl 0x485b8c8 <col:2, col:12> col:12 used y 'float'
    |-DeclStmt 0x485b9f0 <line:6:2, col:31>
    | `-VarDecl 0x485b970 <col:2, col:27> col:14 used threehalfs 'const float' cinit
    |   `-FloatingLiteral 0x485b9d0 <col:27> 'float' 1.500000e+00
    |-BinaryOperator 0x485bab8 <line:8:2, col:16> 'float' lvalue '='
    | |-DeclRefExpr 0x485ba08 <col:2> 'float' lvalue Var 0x485b850 'x2' 'float'
    | `-BinaryOperator 0x485ba90 <col:7, col:16> 'float' '*'
    |   |-ImplicitCastExpr 0x485ba78 <col:7> 'float' <LValueToRValue>
    |   | `-DeclRefExpr 0x485ba30 <col:7> 'float' lvalue ParmVar 0x485b600 'number' 'float'
    |   `-FloatingLiteral 0x485ba58 <col:16> 'float' 5.000000e-01
    |-BinaryOperator 0x485bb48 <line:9:2, col:7> 'float' lvalue '='
    | |-DeclRefExpr 0x485bae0 <col:2> 'float' lvalue Var 0x485b8c8 'y' 'float'
    | `-ImplicitCastExpr 0x485bb30 <col:7> 'float' <LValueToRValue>
    |   `-DeclRefExpr 0x485bb08 <col:7> 'float' lvalue ParmVar 0x485b600 'number' 'float'
    |-BinaryOperator 0x485bcb0 <line:10:2, col:21> 'long' lvalue '='
    | |-DeclRefExpr 0x485bb70 <col:2> 'long' lvalue Var 0x485b7c0 'i' 'long'
    | `-ImplicitCastExpr 0x485bc98 <col:7, col:21> 'long' <LValueToRValue>
    |   `-UnaryOperator 0x485bc78 <col:7, col:21> 'long' lvalue prefix '*'
    |     `-CStyleCastExpr 0x485bc50 <col:9, col:21> 'long *' <BitCast>
    |       `-UnaryOperator 0x485bbe8 <col:20, col:21> 'float *' prefix '&'
    |         `-DeclRefExpr 0x485bb98 <col:21> 'float' lvalue Var 0x485b8c8 'y' 'float'
    |-BinaryOperator 0x485be08 <line:11:2, col:29> 'long' lvalue '='
    | |-DeclRefExpr 0x485bcd8 <col:2> 'long' lvalue Var 0x485b7c0 'i' 'long'
    | `-BinaryOperator 0x485bde0 <col:7, col:29> 'long' '-'
    |   |-ImplicitCastExpr 0x485bdc8 <col:7> 'long' <IntegralCast>
    |   | `-IntegerLiteral 0x485bd00 <col:7> 'int' 1597463007
    |   `-ParenExpr 0x485bda8 <col:20, col:29> 'long'
    |     `-BinaryOperator 0x485bd80 <col:22, col:27> 'long' '>>'
    |       |-ImplicitCastExpr 0x485bd68 <col:22> 'long' <LValueToRValue>
    |       | `-DeclRefExpr 0x485bd20 <col:22> 'long' lvalue Var 0x485b7c0 'i' 'long'
    |       `-IntegerLiteral 0x485bd48 <col:27> 'int' 1
    |-BinaryOperator 0x485bf18 <line:12:2, col:22> 'float' lvalue '='
    | |-DeclRefExpr 0x485be30 <col:2> 'float' lvalue Var 0x485b8c8 'y' 'float'
    | `-ImplicitCastExpr 0x485bf00 <col:7, col:22> 'float' <LValueToRValue>
    |   `-UnaryOperator 0x485bee0 <col:7, col:22> 'float' lvalue prefix '*'
    |     `-CStyleCastExpr 0x485beb8 <col:9, col:22> 'float *' <BitCast>
    |       `-UnaryOperator 0x485be80 <col:21, col:22> 'long *' prefix '&'
    |         `-DeclRefExpr 0x485be58 <col:22> 'long' lvalue Var 0x485b7c0 'i' 'long'
    |-BinaryOperator 0x485c188 <line:13:2, col:41> 'float' lvalue '='
    | |-DeclRefExpr 0x485bf40 <col:2> 'float' lvalue Var 0x485b8c8 'y' 'float'
    | `-BinaryOperator 0x485c160 <col:7, col:41> 'float' '*'
    |   |-ImplicitCastExpr 0x485c148 <col:7> 'float' <LValueToRValue>
    |   | `-DeclRefExpr 0x485bf68 <col:7> 'float' lvalue Var 0x485b8c8 'y' 'float'
    |   `-ParenExpr 0x485c128 <col:11, col:41> 'float'
    |     `-BinaryOperator 0x485c100 <col:13, col:39> 'float' '-'
    |       |-ImplicitCastExpr 0x485c0e8 <col:13> 'float' <LValueToRValue>
    |       | `-DeclRefExpr 0x485bf90 <col:13> 'const float' lvalue Var 0x485b970 'threehalfs' 'const float'
    |       `-ParenExpr 0x485c0c8 <col:26, col:39> 'float'
    |         `-BinaryOperator 0x485c0a0 <col:28, col:37> 'float' '*'
    |           |-BinaryOperator 0x485c038 <col:28, col:33> 'float' '*'
    |           | |-ImplicitCastExpr 0x485c008 <col:28> 'float' <LValueToRValue>
    |           | | `-DeclRefExpr 0x485bfb8 <col:28> 'float' lvalue Var 0x485b850 'x2' 'float'
    |           | `-ImplicitCastExpr 0x485c020 <col:33> 'float' <LValueToRValue>
    |           |   `-DeclRefExpr 0x485bfe0 <col:33> 'float' lvalue Var 0x485b8c8 'y' 'float'
    |           `-ImplicitCastExpr 0x485c088 <col:37> 'float' <LValueToRValue>
    |             `-DeclRefExpr 0x485c060 <col:37> 'float' lvalue Var 0x485b8c8 'y' 'float'
    `-ReturnStmt 0x485c248 <line:16:2, col:9>
      `-ImplicitCastExpr 0x485c230 <col:9> 'float' <LValueToRValue>
        `-DeclRefExpr 0x485c208 <col:9> 'float' lvalue Var 0x485b8c8 'y' 'float'
```

Thesis supervisor:
Konrad Durnoga

Team:
Stanisław Barzowski
Piotr Padlewski
Marek Sokołowski
Jakub Staroń

INNOVATION IN IT