CComp

Generated by Doxygen 1.8.3.1

Wed Apr 10 2013 10:38:04

Contents

Chapter 1

README

#CS 460/660 - Compilers

University of Nevada, Reno - Spring 2013

Authors: Alex Fiannaca & Sandeep Mathew

Date: 03/25/2013

Submitted Materials: `website`

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 3

Class Index

3.1 Class List

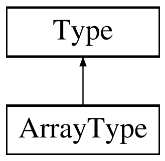
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

Class Documentation

4.1 ArrayType Class Reference

Inheritance diagram for ArrayType:



Public Member Functions

- **ArrayType** (Type ∗baseType, string name, int dims)
- int **SetCapacity** (int cap)
- int **GetCapacity** (int dim)
- Type ∗ **GetBase** ()
- void **SetBase** (Type ∗base)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

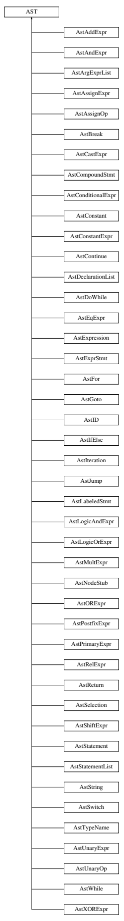
- Type ∗ **baseType**
- int **dimensions**
- vector< int > **capacities**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.2 AST Class Reference

Inheritance diagram for AST:



Public Member Functions

- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()
- virtual void **Visit** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

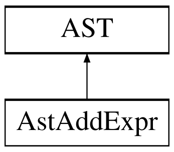
- int **uid**
- string **label**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.3 AstAddExpr Class Reference

Inheritance diagram for AstAddExpr:



Public Types

- enum **Operator** { **NONE**, **PLUS**, **MINUS** }

Public Member Functions

- **AstAddExpr** (AstMultExpr ∗m)
- **AstAddExpr** (AstAddExpr ∗a, Operator o, AstMultExpr ∗m)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstAddExpr::Operator **op**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

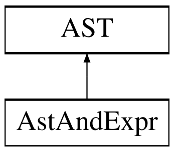
- AstMultExpr ∗ **mult**
- AstAddExpr ∗ **add**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.4 AstAndExpr Class Reference

Inheritance diagram for AstAndExpr:



Public Member Functions

- **AstAndExpr** (AstEqExpr ∗e)
- **AstAndExpr** (AstAndExpr ∗a, AstEqExpr ∗e)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

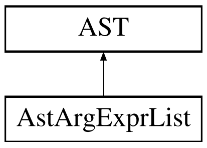
- AstEqExpr ∗ **eq**
- AstAndExpr ∗ **a**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.5 AstArgExprList Class Reference

Inheritance diagram for AstArgExprList:



Public Member Functions

- **AstArgExprList** (AstArgExprList ∗list, AstAssignExpr ∗expr)
- **AstArgExprList** (AstAssignExpr ∗expr)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

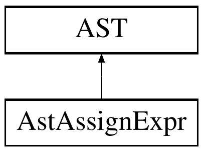
- AstArgExprList ∗ **list**
- AstAssignExpr ∗ **expr**
- bool **isLastItem**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.6 AstAssignExpr Class Reference

Inheritance diagram for AstAssignExpr:



Public Member Functions

- **AstAssignExpr** (AstConditionalExpr ∗c)
- **AstAssignExpr** (AstUnaryExpr ∗u, AstAssignOp ∗a, AstAssignExpr ∗e)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

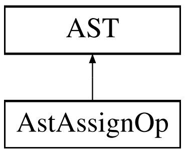
- AstConditionalExpr ∗ **cond**
- AstUnaryExpr ∗ **uni**
- AstAssignOp ∗ **op**
- AstAssignExpr ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.7 AstAssignOp Class Reference

Inheritance diagram for AstAssignOp:



Public Types

- enum **Operator** {
EQ, **MUL_ASSIGN**, **DIV_ASSIGN**, **MOD_ASSIGN**,
ADD_ASSIGN, **SUB_ASSIGN**, **LEFT_ASSIGN**, **RIGHT_ASSIGN**,
AND_ASSIGN, **XOR_ASSIGN**, **OR_ASSIGN** }

Public Member Functions

- **AstAssignOp** (Operator o)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

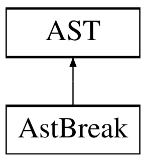
- Operator **op**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.8 AstBreak Class Reference

Inheritance diagram for AstBreak:



Public Member Functions

- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

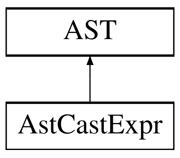
- int **uid**
- string **label**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.9 AstCastExpr Class Reference

Inheritance diagram for AstCastExpr:



Public Member Functions

- **AstCastExpr** (AstUnaryExpr ∗u)
- **AstCastExpr** (AstTypeName ∗t, AstCastExpr ∗c)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

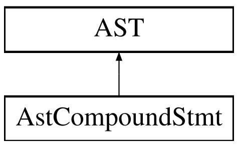
- AstUnaryExpr ∗ **uniexpr**
- AstCastExpr ∗ **cast**
- AstTypeName ∗ **tname**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.10 AstCompoundStmt Class Reference

Inheritance diagram for AstCompoundStmt:



Public Member Functions

- **AstCompoundStmt** (AstDeclarationList ∗d, AstStatementList ∗s)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

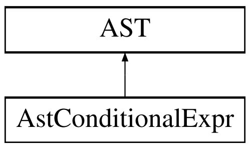
- AstStatementList ∗ **stmtList**
- AstDeclarationList ∗ **declList**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.11 AstConditionalExpr Class Reference

Inheritance diagram for AstConditionalExpr:



Public Member Functions

- **AstConditionalExpr** (AstLogicOrExpr ∗o)
- **AstConditionalExpr** (AstLogicOrExpr ∗o, AstExpression ∗e, AstConditionalExpr ∗ce)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

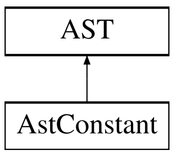
- AstLogicOrExpr ∗ **o**
- AstExpression ∗ **e**
- AstConditionalExpr ∗ **ce**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.12 AstConstant Class Reference

Inheritance diagram for AstConstant:



Public Member Functions

- **AstConstant** (int val)
- **AstConstant** (string val)
- **AstConstant** (double val)
- **AstConstant** (int val, string name)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Types

- enum **ConstType** { **INT**, **CHAR**, **FLOAT**, **ENUM** }

Private Attributes

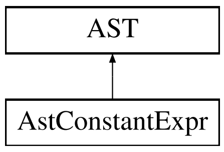
- ConstType **type**
- int **ival**
- string **str**
- double **dval**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.13 AstConstantExpr Class Reference

Inheritance diagram for AstConstantExpr:



Public Member Functions

- **AstConstantExpr** (AstConditionalExpr ∗e)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

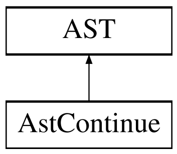
- AstConditionalExpr ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.14 AstContinue Class Reference

Inheritance diagram for AstContinue:



Public Member Functions

- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

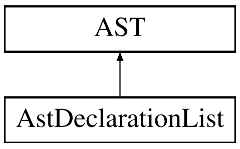
- int **uid**
- string **label**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.15 AstDeclarationList Class Reference

Inheritance diagram for AstDeclarationList:



Public Member Functions

- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

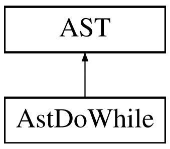
- int **uid**
- string **label**

The documentation for this class was generated from the following file:

- Ast.h

4.16 AstDoWhile Class Reference

Inheritance diagram for AstDoWhile:



Public Member Functions

- **AstDoWhile** (AstStatement ∗s, AstExpression ∗t)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

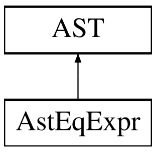
- AstExpression ∗ **test**
- AstStatement ∗ **statement**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.17 AstEqExpr Class Reference

Inheritance diagram for AstEqExpr:



Public Types

- enum **Operator** { **NONE**, **EQ_OP**, **NE_OP** }

Public Member Functions

- **AstEqExpr** (AstRelExpr ∗r)
- **AstEqExpr** (AstEqExpr ∗e, Operator o, AstRelExpr ∗r)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstEqExpr::Operator **op**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

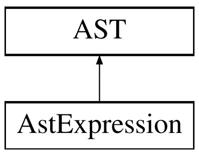
- AstRelExpr ∗ **rel**
- AstEqExpr ∗ **eq**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.18 AstExpression Class Reference

Inheritance diagram for AstExpression:



Public Member Functions

- **AstExpression** (AstAssignExpr ∗a)
- **AstExpression** (AstExpression ∗e, AstAssignExpr ∗a)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

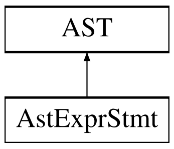
- AstAssignExpr ∗ **ass**
- AstExpression ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.19 AstExprStmt Class Reference

Inheritance diagram for AstExprStmt:



Public Member Functions

- **AstExprStmt** (AstExpression ∗e)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

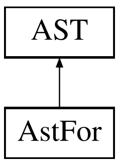
- AstExpression ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.20 AstFor Class Reference

Inheritance diagram for AstFor:



Public Member Functions

- **AstFor** (AstExpression ∗init, AstExpression ∗test, AstExpression ∗increment, AstStatement ∗statement)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

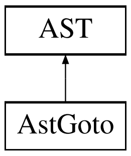
- AstExpression ∗ **init**
- AstExpression ∗ **test**
- AstExpression ∗ **increment**
- AstStatement ∗ **statement**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.21 AstGoto Class Reference

Inheritance diagram for AstGoto:



Public Member Functions

- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

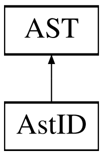
- int **uid**
- string **label**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.22 AstID Class Reference

Inheritance diagram for AstID:



Public Member Functions

- **AstID** (string s)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

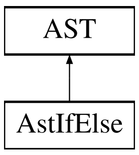
- string **str**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.23 AstIfElse Class Reference

Inheritance diagram for AstIfElse:



Public Member Functions

- **AstIfElse** (AstExpression ∗test, AstStatement ∗statement, AstStatement ∗elseStatement)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

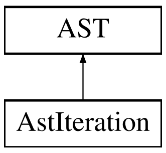
- AstExpression ∗ **test**
- AstStatement ∗ **statement**
- AstStatement ∗ **elseStatement**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.24 AstIteration Class Reference

Inheritance diagram for AstIteration:



Public Types

- enum **Type** { **DOWHILE**, **WHILE**, **FOR** }

Public Member Functions

- **AstIteration** (AstDoWhile ∗d)
- **AstIteration** (AstWhile ∗w)
- **AstIteration** (AstFor ∗f)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstIteration::Type **t**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

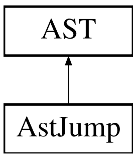
- AstDoWhile ∗ **dwl**
- AstWhile ∗ **wl**
- AstFor ∗ **fr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.25 AstJump Class Reference

Inheritance diagram for AstJump:



Public Types

- enum **Type** {
  GOTO, **CONTINUE**, **BREAK**, **EMPTY_RETURN**,
  RETURN }

Public Member Functions

- **AstJump** (AstGoto ∗g, AstID ∗i)
- **AstJump** (AstContinue ∗c)
- **AstJump** (AstBreak ∗b)
- **AstJump** (AstReturn ∗r)
- **AstJump** (AstReturn ∗r, AstExpression ∗e)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstJump::Type **t**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

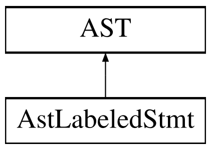
- AstGoto ∗ **go**
- AstID ∗ **id**
- AstContinue ∗ **cont**
- AstBreak ∗ **br**
- AstReturn ∗ **ret**
- AstExpression ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.26 AstLabeledStmt Class Reference

Inheritance diagram for AstLabeledStmt:



Public Types

- enum **Type** { **NO_CASE**, **CASE**, **DEFAULT** }

Public Member Functions

- **AstLabeledStmt** (AstID ∗i, AstStatement ∗s)
- **AstLabeledStmt** (AstConstantExpr ∗c, AstStatement ∗s)
- **AstLabeledStmt** (AstStatement ∗s)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstLabeledStmt::Type **t**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

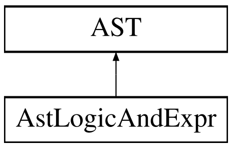
- AstID ∗ **id**
- AstStatement ∗ **stmt**
- AstConstantExpr ∗ **constExpr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.27 AstLogicAndExpr Class Reference

Inheritance diagram for AstLogicAndExpr:



Public Member Functions

- **AstLogicAndExpr** (AstORExpr ∗o)
- **AstLogicAndExpr** (AstLogicAndExpr ∗a, AstORExpr ∗o)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

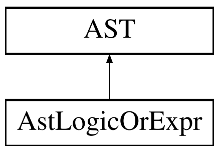
- AstORExpr ∗ **o**
- AstLogicAndExpr ∗ **a**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.28 AstLogicOrExpr Class Reference

Inheritance diagram for AstLogicOrExpr:



Public Member Functions

- **AstLogicOrExpr** (AstLogicAndExpr ∗a)
- **AstLogicOrExpr** (AstLogicOrExpr ∗o, AstLogicAndExpr ∗a)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

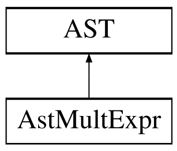
- AstLogicAndExpr ∗ **a**
- AstLogicOrExpr ∗ **o**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.29 AstMultExpr Class Reference

Inheritance diagram for AstMultExpr:



Public Types

- enum **Operator** { **NONE**, **STAR**, **DIV**, **MOD** }

Public Member Functions

- **AstMultExpr** (AstCastExpr ∗c)
- **AstMultExpr** (AstMultExpr ∗m, Operator o, AstCastExpr ∗c)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstMultExpr::Operator **op**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

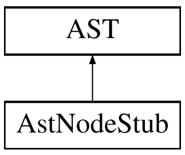
- AstCastExpr ∗ **cast**
- AstMultExpr ∗ **mult**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.30 AstNodeStub Class Reference

Inheritance diagram for AstNodeStub:



Public Member Functions

- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

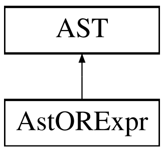
- int **uid**
- string **label**

The documentation for this class was generated from the following file:

- Ast.h

4.31 AstORExpr Class Reference

Inheritance diagram for AstORExpr:



Public Member Functions

- **AstORExpr** (AstXORExpr ∗x)
- **AstORExpr** (AstORExpr ∗o, AstXORExpr ∗x)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

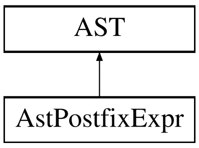
- AstXORExpr ∗ **x**
- AstORExpr ∗ **o**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.32 AstPostfixExpr Class Reference

Inheritance diagram for AstPostfixExpr:



Public Types

- enum **Operator** {
  NONE, **DOT_OP**, **PTR_OP**, **INC_OP**,
  DEC_OP }
- enum **Type** {
  PRIMARY, **BRACKETS**, **EMPTY_PARENS**, **PARENS**,
  DOT, **PTR**, **INC**, **DEC** }

Public Member Functions

- **AstPostfixExpr** (AstPrimaryExpr ∗p)
- **AstPostfixExpr** (AstPostfixExpr ∗p, AstExpression ∗e)
- **AstPostfixExpr** (AstPostfixExpr ∗p)
- **AstPostfixExpr** (AstPostfixExpr ∗p, AstArgExprList ∗a)
- **AstPostfixExpr** (AstPostfixExpr ∗p, Operator o, AstID ∗i)
- **AstPostfixExpr** (AstPostfixExpr ∗p, Operator o)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

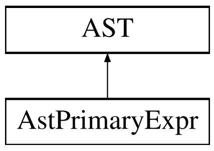
- AstPrimaryExpr ∗ **priexpr**
- AstPostfixExpr ∗ **ptfExpr**
- AstExpression ∗ **brakExpr**
- AstArgExprList ∗ **argExprList**
- AstID ∗ **id**
- Operator **op**
- Type **t**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.33 AstPrimaryExpr Class Reference

Inheritance diagram for AstPrimaryExpr:



Public Member Functions

- **AstPrimaryExpr** (AstID ∗id)
- **AstPrimaryExpr** (AstConstant ∗c)
- **AstPrimaryExpr** (AstString ∗s)
- **AstPrimaryExpr** (AstExpression ∗e)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Types

- enum **ExprType** { **ID**, **CONST**, **STRING**, **EXPR** }

Private Attributes

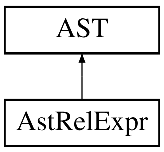
- ExprType **type**
- AstID ∗ **id**
- AstConstant ∗ **constant**
- AstString ∗ **str**
- AstExpression ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.34 AstRelExpr Class Reference

Inheritance diagram for AstRelExpr:



Public Types

- enum **Operator** {
  NONE, **LT_OP**, **GT_OP**, **LE_OP**,
  GE_OP }

Public Member Functions

- **AstRelExpr** (AstShiftExpr ∗s)
- **AstRelExpr** (AstRelExpr ∗r, Operator o, AstShiftExpr ∗s)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstRelExpr::Operator **op**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

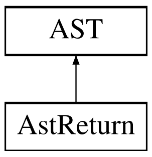
- AstShiftExpr ∗ **shift**
- AstRelExpr ∗ **rel**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.35 AstReturn Class Reference

Inheritance diagram for AstReturn:



Public Member Functions

- **AstReturn** (AstExpression ∗r)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

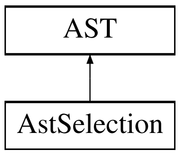
- AstExpression ∗ **expr**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.36 AstSelection Class Reference

Inheritance diagram for AstSelection:



Public Types

- enum **Type** { **SWITCH**, **IFELSE** }

Public Member Functions

- **AstSelection** (AstSwitch ∗s)
- **AstSelection** (AstIfElse ∗ie)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstSelection::Type **t**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

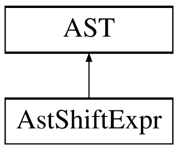
- AstSwitch ∗ **swtch**
- AstIfElse ∗ **ifelse**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.37 AstShiftExpr Class Reference

Inheritance diagram for AstShiftExpr:



Public Types

- enum **Operator** { **NONE**, **LEFT_OP**, **RIGHT_OP** }

Public Member Functions

- **AstShiftExpr** (AstAddExpr ∗a)
- **AstShiftExpr** (AstShiftExpr ∗s, Operator o, AstAddExpr ∗a)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstShiftExpr::Operator **op**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

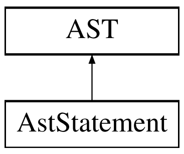
- AstAddExpr ∗ **add**
- AstShiftExpr ∗ **shift**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.38 AstStatement Class Reference

Inheritance diagram for AstStatement:



Public Types

- enum **Type** {
  LABELED, **COMPOUND**, **EXPR**, **SELECT**,
  ITER, **JUMP** }

Public Member Functions

- **AstStatement** (AstLabeledStmt ∗l)
- **AstStatement** (AstCompoundStmt ∗c)
- **AstStatement** (AstExprStmt ∗e)
- **AstStatement** (AstSelection ∗s)
- **AstStatement** (AstIteration ∗i)
- **AstStatement** (AstJump ∗j)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstStatement::Type **t**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

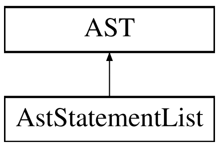
- AstLabeledStmt ∗ **lbl**
- AstCompoundStmt ∗ **cmp**
- AstExprStmt ∗ **expr**
- AstSelection ∗ **slct**
- AstIteration ∗ **iter**
- AstJump ∗ **jump**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.39 AstStatementList Class Reference

Inheritance diagram for AstStatementList:



Public Member Functions

- **AstStatementList** (AstStatement ∗s)
- **AstStatementList** (AstStatementList ∗l, AstStatement ∗s)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

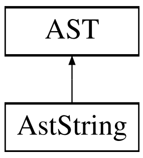
- AstStatement ∗ **stmt**
- AstStatementList ∗ **list**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.40 AstString Class Reference

Inheritance diagram for AstString:



Public Member Functions

- **AstString** (string str)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

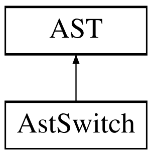
- string **val**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.41 AstSwitch Class Reference

Inheritance diagram for AstSwitch:



Public Member Functions

- **AstSwitch** (AstExpression ∗e, AstStatement ∗s)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

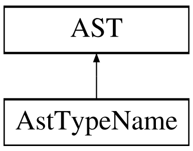
- AstExpression ∗ **expr**
- AstStatement ∗ **stmt**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.42 AstTypeName Class Reference

Inheritance diagram for AstTypeName:



Public Member Functions

- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

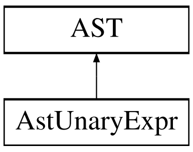
- int **uid**
- string **label**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.43 AstUnaryExpr Class Reference

Inheritance diagram for AstUnaryExpr:



Public Types

- enum **Type** {
  POSTFIX, **INC**, **DEC**, **CAST**,
  SIZEOF, **SIZEOF_TYPE** }

Public Member Functions

- **AstUnaryExpr** (AstPostfixExpr ∗e)
- **AstUnaryExpr** (AstUnaryExpr ∗e, bool inc)
- **AstUnaryExpr** (AstUnaryOp ∗o, AstCastExpr ∗c)
- **AstUnaryExpr** (AstUnaryExpr ∗e)
- **AstUnaryExpr** (AstTypeName ∗t)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Public Attributes

- enum AstUnaryExpr::Type **t**

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

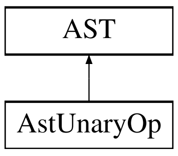
- AstPostfixExpr ∗ **expr**
- bool **isINC**
- bool **isDEC**
- AstUnaryOp ∗ **op**
- AstCastExpr ∗ **cast**
- AstUnaryExpr ∗ **uniexpr**
- AstTypeName ∗ **tname**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.44 AstUnaryOp Class Reference

Inheritance diagram for AstUnaryOp:



Public Types

- enum **Operator** {
  BIN_AND, **STAR**, **PLUS**, **MINUS**,
  TILDE, **BANG** }

Public Member Functions

- **AstUnaryOp** (Operator o)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

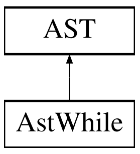
- Operator **op**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.45 AstWhile Class Reference

Inheritance diagram for AstWhile:



Public Member Functions

- **AstWhile** (AstExpression ∗test, AstStatement ∗statement)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

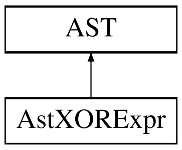
- AstExpression ∗ **test**
- AstStatement ∗ **statement**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.46 AstXORExpr Class Reference

Inheritance diagram for AstXORExpr:



Public Member Functions

- **AstXORExpr** (AstAndExpr ∗a)
- **AstXORExpr** (AstXORExpr ∗x, AstAndExpr ∗a)
- void **Visit** ()
- void **setLabel** (string l)
- int **getUID** ()
- string **getLabel** ()

Static Public Attributes

- static Visualizer **vis**

Protected Attributes

- int **uid**
- string **label**

Private Attributes

- AstAndExpr ∗ **a**
- AstXORExpr ∗ **x**

The documentation for this class was generated from the following files:

- Ast.h
- Ast.cpp

4.47 AVLTree< DataItem > Class Template Reference

Classes

- struct Node

Public Member Functions

- void **Insert** (DataItem item)
- void **Insert** (DataItem item, Node ∗&node, int &change)
- DataItem ∗ **Fetch** (DataItem itemToFind)
- Node ∗ **Find** (DataItem itemToFind)
- bool **Contains** (DataItem itemToFind)
- void **Dump** ()

Private Member Functions

- int **SingleRotate** (Node ∗&rootNode, int direction)
- int **DoubleRotate** (Node ∗&rootNode, int direction)
- int **Balance** (Node ∗&rootNode)
- void **Dump** (Node ∗node)

Private Attributes

- Node ∗ **root**

The documentation for this class was generated from the following file:

- AvlTree.h

4.48 CCompiler Class Reference

Public Member Functions

- void **scan_begin** (bool debug_scanning)
- void **scan_end** ()
- int **parse** (const std::string &fname)
- void **setOutfile** (std::string fname)
- yy::CParser::token::yytokentype **checkType** (char ∗key, const yy::location &loc, SymbolInfo ∗sym)
- void **allocateSymbol** ()
- void **globalScope** ()

- void **enterScope** ()
- void **leaveScope** ()
- void **set_insert_mode** (bool iMode)
- bool **get_insert_mode** ()
- void **error** (const yy::location &loc, const std::string &msg)
- void **error** (const std::string &msg)
- void **warning** (const yy::location &loc, const std::string &msg)
- void **warning** (const std::string &msg)
- void **printTok** (std::string)
- void **printTok** (std::string, char ∗)
- void **printRed** (std::string)
- void **turnDebugOn** (bool)
- void **printDebug** (std::string)
- void **save_line** (int i, string s)

Public Attributes

- int **result**
- bool **trace_scanning**
- std::string **fname**
- bool **trace_parsing**
- SymbolInfo ∗ **currentSymbol**
- Type ∗ **structMemberType**
- SymTab **SymbolTable**
- bool **anonymousEnum**
- int **structUnionMode**
- list< string > **enumConsts**
- list< SymbolInfo > **structUnionTypes**
- EnumType ∗ **enumType**
- SymbolInfo ∗ **enumSym**
- int **structVarCount**
- bool **trace_symtab**
- char **linebuf** [500]
- fstream **ydbFile**
- map< int, string > **input_text**

Static Public Attributes

- static TAC_Generator **tacGen**

Private Attributes

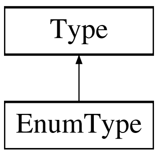
- bool **debug_on**
- bool **insert_mode**
- bool **outfile_set**
- fstream **tFile**
- fstream **rFile**
- fstream **outfile**

The documentation for this class was generated from the following files:

- CCompiler.h
- CCompiler.cpp

4.49 EnumType Class Reference

Inheritance diagram for EnumType:



Public Member Functions

- **EnumType** (string n, int startVal)
- int **GetConstVal** (string s)
- void **AddEnumConst** (string s)
- void **AddEnumConst** (string s, int val)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

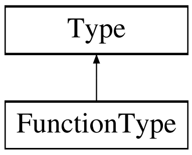
- map$<$ string, int $>$ **enumConsts**
- int **currentVal**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.50 FunctionType Class Reference

Inheritance diagram for FunctionType:



Public Member Functions

- **FunctionType** (string n)
- void **AddParam** (Type $*$t)
- void **SetReturnType** (Type $*$t)
- int **GetParamCount** ()
- Type $*$ **GetReturnType** ()
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

- vector< Type * > **params**
- Type * **returnType**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.51 InputLine Struct Reference

Public Member Functions

- **InputLine** (int l, string s)

Public Attributes

- int **line**
- string **text**

The documentation for this struct was generated from the following file:

- CCompiler.h

4.52 AVLTree< DataItem >::Node Struct Reference

Public Attributes

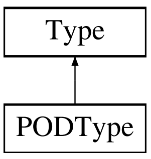
- DataItem **data**
- Node * **children** [CHILD_SIZE]
- int **balanceFactor**

The documentation for this struct was generated from the following file:

- AvlTree.h

4.53 PODType Class Reference

Inheritance diagram for PODType:



Public Member Functions

- **PODType** (string n, int s)
- bool **isSigned** ()
- void **SetSigned** (bool isSigned)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

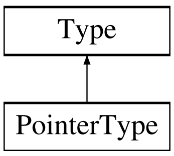
- bool **is_signed**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.54 PointerType Class Reference

Inheritance diagram for PointerType:



Public Member Functions

- **PointerType** (Type ∗base, string n, int d)
- **PointerType** (Type ∗base, bool baseIsPtr, string n)
- Type ∗ **GetBase** ()
- void **SetBaseType** (Type ∗base)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

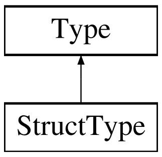
- Type ∗ **baseType**
- int **ptrDepth**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.55 StructType Class Reference

Inheritance diagram for StructType:



Public Member Functions

- **StructType** (string n)
- void **AddMember** (string s, Type ∗t)
- bool **MemberExists** (string s)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

- vector< string > **memberNames**
- vector< Type ∗ > **memberTypes**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.56 SymbolInfo Struct Reference

Public Member Functions

- **SymbolInfo** (const SymbolInfo &sym)
- int **operator<** (SymbolInfo inf)
- int **operator==** (SymbolInfo inf)
- int **operator>** (SymbolInfo inf)
- string **GetKey** ()

Public Attributes

- string **symbol_name**
- Type ∗ **symbolType**
- int **type_qualifier**
- string **function_name**
- bool **isEnumConst**
- bool **struct_union_name**
- bool **isStrunctOrUnionItem**
- int **typeTableIndex**
- int **storage_class**
- int **flags**

Friends

- ostream & **operator**<< (ostream &outStream, const SymbolInfo &inf)

The documentation for this struct was generated from the following file:

- SymTab.h

4.57 SymTab Class Reference

Public Member Functions

- **SymTab** (CCompiler ∗ref)
- void **EnterScope** ()
- void **LeaveScope** ()
- void **insert_symbol** (SymbolInfo symbolInfo)
- void **insert_symbol** (SymbolInfo symbolInfo, int level)
- bool **find_symbol** (SymbolInfo symbolInfo, int &level)
- void **dump_table** ()
- void **dump_table** (int level)

Private Member Functions

- void **error** (string msg)
- void **warning** (string msg)

Private Attributes

- int **currentLevel**
- vector< AVLTree< SymbolInfo > > **symTable**
- CCompiler ∗ **driver**

The documentation for this class was generated from the following files:

- SymTab.h
- SymTab.cpp

4.58 TAC_Generator Class Reference

A class for generating three address code.

```
#include <TAC_Generator.h>
```

Public Types

- enum ThreeOpInstructions {
  ADD, SUB, MULT, DIV,
  EQ, GT, LT, GE,
  LE, NE, BREQ, BRGT,
  BRLT, BRGE, BRLE, BRNE,
  PROCENTRY, BOUND }

Enumeration of 3 operand instructions.
- enum TwoOpInstructions {
NEG, NOT, ASSIGN, ADDR,
GLOBAL, STRING }

  Enum of 2 operand instructions.
- enum OneOpInstructions {
LABEL, BR, ARGS, REFOUT,
VALOUT, CALL, COMMENT }

  Enum of 1 operand instructions.
- enum NoOpInstructions { HALT, ENDPROC, RETURN }

  Enum of instructions without operands.

Public Member Functions

- TAC_Generator (const string &filename)

  The paramaterized constructor.
- TAC_Generator ()

  The default constructor.
- ∼TAC_Generator ()

  The destructor.
- void toTAC (ThreeOpInstructions t, void ∗op1, void ∗op2, void ∗op3, string c="")

  Generate a 3AC string.
- void toTAC (TwoOpInstructions t, void ∗op1, void ∗op2, string c="")

  Generate a 3AC string.
- void toTAC (OneOpInstructions t, void ∗op1, string c="")

  Generate a 3AC string.
- void toTAC (NoOpInstructions t, string c="")

  Generate a 3AC string.
- void SetCommentStart (string commentStart)

  Sets the symbol which should appear at the end of all comments.
- void SetCommentEnd (string commentEnd)

  Sets the symbol which should appear at the beginning of all comments.
- void WriteComment (string comment)

  Writes a comment string to the 3AC output.
- void Blank ()

  Puts a blank line in the 3AC output.
- void SetFile (const string &filename)

  Sets the name of the file in which the output 3AC should be saved.
- void SetColumnWidth (int w)

  Sets the fixed column width for outputting 3AC statements.
- void SetFormatFlags (ios_base::fmtflags ff)

  Sets the ios_base format flags used when generating formatted 3AC strings.
- void SetBlankBeforeComments (bool flag)

  Sets the blankBeforeComments flag.

Static Public Member Functions

- static string GetLabelName ()

  Generates a unique label string.
- static string GetIVarName ()

  Generates a unique string for integer temps.
- static string GetFVarName ()

  Generates a unique string for floating-point temps.

Private Member Functions

- void Emit (string CodeToEmit)

  This function saves the string passed in to a STL list for later output.

Private Attributes

- list< string > buffer

  A buffer for the generated 3AC.
- ofstream fout

  Output stream.
- string commentStart

  String to be placed at the beginning of every comment.
- string commentEnd

  String to be placed at the end of every comment.
- bool blankBeforeComment

  Flag for placing blank lines before comments.
- int width

  Fixed column width of the output 3AC.
- ios_base::fmtflags flags

  Format flags.

Static Private Attributes

- static int lCount = 0

  Current label counter for generating unique labels.
- static int iCount = 0

  Current integer counter for generating unique integer labels.
- static int fCount = 0

  Current float counter for generating unique float labels.

4.58.1 Detailed Description

A class for generating three address code.

The TAC_Generator class is responsible for generating well-formatted three address code (3AC or TAC). The generator stores all generated 3AC in a STL list of strings during runtime, and outputs the 3AC to a file when the destructor is called. This allows for the 3AC to be manipulated prior to output (i.e. putting all function decls at the top of the 3AC).

4.58.2 Member Enumeration Documentation

4.58.2.1 enum TAC_Generator::NoOpInstructions

Enum of instructions without operands.

These enum values serve as flags to the toTAC functions in order to indicate which 3AC statement should be generated, and what the types of the void $*$ parameters to the toTAC functions are.

Enumerator

HALT Immediately halt execution.

ENDPROC Mark the end of a procedure.

RETURN Return control to the caller.

4.58.2.2 enum TAC_Generator::OneOpInstructions

Enum of 1 operand instructions.

These enum values serve as flags to the toTAC functions in order to indicate which 3AC statement should be generated, and what the types of the void $*$ parameters to the toTAC functions are.

Enumerator

LABEL Generate a label.

BR Branch to a label.

ARGS Specify the number of arguments being sent to the next call.

REFOUT Pass op1 by reference.

VALOUT Pass op1 by value.

CALL Call the procedure named op1.

COMMENT Output op1 as a comment.

4.58.2.3 enum TAC_Generator::ThreeOpInstructions

Enumeration of 3 operand instructions.

These enum values serve as flags to the toTAC functions in order to indicate which 3AC statement should be generated, and what the types of the void $*$ parameters to the toTAC functions are.

Enumerator

ADD Add the value of two temps.

SUB Subtract the value of two temps.

MULT Multiply the value of two temps.

DIV Divide the value of two temps.

EQ Set op3 to 1 is op1 == op2, or 0 otherwise.

GT Set op3 to 1 is op1 $>$ op2, or 0 otherwise.

LT Set op3 to 1 is op1 $<$ op2, or 0 otherwise.

GE Set op3 to 1 is op1 $>$= op2, or 0 otherwise.

LE Set op3 to 1 is op1 $<$= op2, or 0 otherwise.

NE Set op3 to 1 is op1 != op2, or 0 otherwise.

BREQ If(op1 == op2) goto op3.

BRGT If(op1 $>$ op2) goto op3.

BRLT If(op1 $<$ op2) goto op3.

BRGE If(op1 $>$= op2) goto op3.

BRLE If(op1 $<$= op2) goto op3.

BRNE If(op1 != op2) goto op3.

PROCENTRY Marks the beginning of a procedure.

BOUND Checks the bounds of an array access.

4.58.2.4 enum TAC_Generator::TwoOpInstructions

Enum of 2 operand instructions.

These enum values serve as flags to the toTAC functions in order to indicate which 3AC statement should be generated, and what the types of the void $*$ parameters to the toTAC functions are.

Enumerator

NEG	op2 = -(op1)
NOT	Set op2 to 1 if op1 == 0, or 0 otherwise.
ASSIGN	Assign the value of op1 to op2.
ADDR	Assign the address of op1 to op2.
GLOBAL	Declare op1 as a global of size op2.
STRING	Associate string op1 with label op2.

4.58.3 Constructor & Destructor Documentation

4.58.3.1 TAC_Generator::TAC_Generator (const string & *filename*)

The paramaterized constructor.

This constructor opens the 3AC file with the given filename.

Parameters

filename	The name of the file in which to output 3AC

4.58.3.2 TAC_Generator::TAC_Generator ( )

The default constructor.

This constructor does not open an output file. If this constructor is used, then the function SetFile must be called.

See Also

SetFile()

4.58.3.3 TAC_Generator::∼TAC_Generator ( )

The destructor.

This destructor is responsible for outputting the 3AC from the list of strings to the output file and then closing the output file.

4.58.4 Member Function Documentation

4.58.4.1 void TAC_Generator::Emit (string *CodeToEmit*) `[private]`

This function saves the string passed in to a STL list for later output.

NOTE: The 3AC Generator "emits" code to a list first, and then after all code has been emitted, it is pushed to a file. This is done so as to allow for changes to be made to the 3AC before it is finalized (ie: moving all function decls to the top of the code)

4.58.4.2 void TAC_Generator::SetBlankBeforeComments (bool *flag*)

Sets the blankBeforeComments flag.

If true, a blank line will be output in the final 3AC before each comment.

Parameters

flag	True if there should be an empty line before each comment

4.58.4.3 void TAC_Generator::SetColumnWidth (int *w*)

Sets the fixed column width for outputting 3AC statements.

Parameters

w	Integer indicating the width of the columns to print the 3AC in.

4.58.4.4 void TAC_Generator::SetCommentEnd (string *commentEnd*)

Sets the symbol which should appear at the beginning of all comments.

Parameters

commentEnd	String to be placed at the end of every comment

4.58.4.5 void TAC_Generator::SetCommentStart (string *commentStart*)

Sets the symbol which should appear at the end of all comments.

Parameters

commentStart	String to be place at the beginning of every comment

4.58.4.6 void TAC_Generator::SetFile (const string & *filename*)

Sets the name of the file in which the output 3AC should be saved.

Parameters

filename	The name of the file in which to output 3AC

4.58.4.7 void TAC_Generator::SetFormatFlags (ios_base::fmtflags *ff*)

Sets the ios_base format flags used when generating formatted 3AC strings.

Parameters

ff	Format flags (i.e. left, right, etc.)

4.58.4.8 void TAC_Generator::toTAC (ThreeOpInstructions *t,* void ∗ *op1,* void ∗ *op2,* void ∗ *op3,* string *c* = "")

Generate a 3AC string.

The toTAC overloads take in a flag to indicate the type of three address code statement and a series of parameters required by the particular statement, in order to generate a formatted 3AC statement. NOTE: this function determines the type of the operands based on the flag passed as the first parameter. If the incorrect flag is passed, then the program will cast the address to the wrong type, so be careful.

Parameters

t	Flag indicating the type of 3AC statement to generate
op1	A pointer to the first operand (cast as a void∗)
op2	A pointer to the second operand (cast as a void∗)
op3	A pointer to the third operand (cast as a void∗)
c	An optional comment to prepend to the 3AC statement (useful for outputting the original input code as comments to the 3AC file)

4.58.4.9 void TAC_Generator::toTAC (TwoOpInstructions *t*, void ∗ *op1*, void ∗ *op2*, string *c* = " ")

Generate a 3AC string.

The toTAC overloads take in a flag to indicate the type of three address code statement and a series of parameters required by the particular statement, in order to generate a formatted 3AC statement. NOTE: this function determines the type of the operands based on the flag passed as the first parameter. If the incorrect flag is passed, then the program will cast the address to the wrong type, so be careful.

Parameters

t	Flag indicating the type of 3AC statement to generate
op1	A pointer to the first operand (cast as a void∗)
op2	A pointer to the second operand (cast as a void∗)
c	An optional comment to prepend to the 3AC statement (useful for outputting the original input code as comments to the 3AC file)

4.58.4.10 void TAC_Generator::toTAC (OneOpInstructions *t*, void ∗ *op1*, string *c* = " ")

Generate a 3AC string.

The toTAC overloads take in a flag to indicate the type of three address code statement and a series of parameters required by the particular statement, in order to generate a formatted 3AC statement. NOTE: this function determines the type of the operand based on the flag passed as the first parameter. If the incorrect flag is passed, then the program will cast the address to the wrong type, so be careful.

Parameters

t	Flag indicating the type of 3AC statement to generate
op1	A pointer to the first operand (cast as a void∗)
c	An optional comment to prepend to the 3AC statement (useful for outputting the original input code as comments to the 3AC file)

4.58.4.11 void TAC_Generator::toTAC (NoOpInstructions *t*, string *c* = " ")

Generate a 3AC string.

The toTAC overloads take in a flag to indicate the type of three address code statement and a series of parameters required by the particular statement, in order to generate a formatted 3AC statement.

Parameters

t	Flag indicating the type of 3AC statement to generate
c	An optional comment to prepend to the 3AC statement (useful for outputting the original input code as comments to the 3AC file)

4.58.4.12 void TAC_Generator::WriteComment (string *comment*)

Writes a comment string to the 3AC output.

Parameters

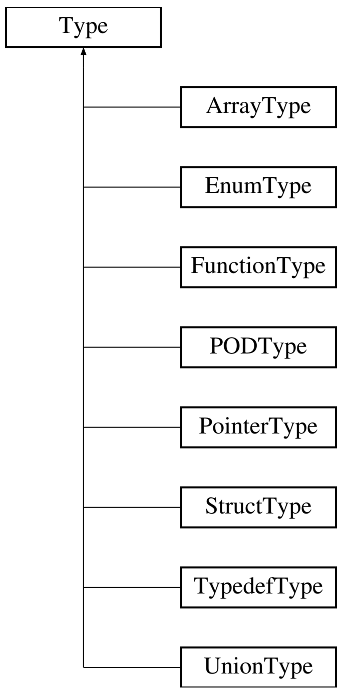
comment	String to output as a comment

The documentation for this class was generated from the following files:

- TAC_Generator.h
- TAC_Generator.cpp

4.59 Type Class Reference

Inheritance diagram for Type:



Public Member Functions

- **Type** (string n, int s)
- **Type** (Type &t)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

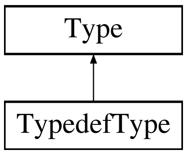
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.60 TypedefType Class Reference

Inheritance diagram for TypedefType:



Public Member Functions

- **TypedefType** (Type ∗actual, string tdname)
- Type ∗ **GetActual** ()
- string **GetTypedefName** ()
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

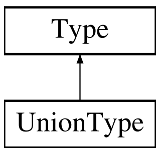
- Type ∗ **actualType**
- string **typedefName**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.61 UnionType Class Reference

Inheritance diagram for UnionType:



Public Member Functions

- **UnionType** (string n)
- void **AddMember** (string s, Type ∗t)
- bool **MemberExists** (string s)
- string **GetName** ()
- int **GetSize** ()
- void **SetName** (string n)

Protected Attributes

- vector< string > **memberNames**
- vector< Type * > **memberTypes**
- string **name**
- int **size**

The documentation for this class was generated from the following files:

- Type.h
- Type.cpp

4.62 Visualizer Class Reference

A class for visualizing the generation of the AST.

```
#include <Visualizer.h>
```

Public Member Functions

- Visualizer ()

  Default constructor.
- Visualizer (string fname)

  Parameterized constructor.
- ~Visualizer ()

  Destructor.
- void begin ()

  Creates the opening part of the GraphViz file.
- void end ()

  Creates the closing part of the GraphViz file.
- void addNode (int uid, string label)

  Adds a node to the graph with a unique id and a label.
- void addNode (int parentid, int childid, string parent_label)

  Adds a node to the graph with a unique id and a label, then creates an edge from the new node to a child node.
- void addDummyNode (int parentid, string label)

  Adds a node to the graph which is only for visualizing extra info rather than visualizing an actual node in the AST.
- void addEdge (int parent, int child)

  Adds an edge from a parent node to a child node.

Static Public Member Functions

- static int GetNextUID ()

  Gets a unique id to be used in adding nodes to the AST.

Private Attributes

- fstream file

  The filename of the output GraphViz file.
- string gname

  The name of the graph in the GraphViz file.

Static Private Attributes

- static int nextUID = 0

  The next unique id value.

4.62.1 Detailed Description

A class for visualizing the generation of the AST.

The Visualizer class provides a method for generating a GraphViz output file which can be converted to a graphic representing the AST.

4.62.2 Constructor & Destructor Documentation

4.62.2.1 Visualizer::Visualizer ( )

Default constructor.

This version of the constructor defaults the output filename to "vis.dot". It attempts to open the file, and exits with EXIT_FAILURE if the file cannot be opened.

4.62.2.2 Visualizer::Visualizer (string *fname*)

Parameterized constructor.

This version of the constructor takes the output filename as a parameter. It attempts to open the file, and exits with EXIT_FAILURE if the file cannot be opened.

Parameters

fname	The output GraphVis file filename

4.62.2.3 Visualizer::∼Visualizer ( )

Destructor.

Completes the GraphViz file and closes the filestream.

4.62.3 Member Function Documentation

4.62.3.1 void Visualizer::addDummyNode (int *parentid*, string *label*)

Adds a node to the graph which is only for visualizing extra info rather than visualizing an actual node in the AST.

Parameters

uid	The unique id of the parent node off of which the dummy node should be attached
label	The label to go inside the dummy node

4.62.3.2 void Visualizer::addEdge (int *parent*, int *child*)

Adds an edge from a parent node to a child node.

Parameters

parent	The unique id of the parent
child	The unique id of the child

4.62.3.3 void Visualizer::addNode (int *uid*, string *label*)

Adds a node to the graph with a unique id and a label.

Parameters

uid	A unique id
label	The label to go inside the node in the graph

4.62.3.4 void Visualizer::addNode (int *parentid*, int *childid*, string *parent_label*)

Adds a node to the graph with a unique id and a label, then creates an edge from the new node to a child node.

Parameters

parentid	A unique id for the new node
childid	The unique id of the child node to create an edge to
parent_label	The label to go inside the node in the graph

4.62.3.5 static int Visualizer::GetNextUID ( ) `[inline]`, `[static]`

Gets a unique id to be used in adding nodes to the AST.

Returns

A unique integer valued id

The documentation for this class was generated from the following files:

- Visualizer.h
- Visualizer.cpp

Index