

实验二 语义分析

一、程序功能及其实现

1. 功能概述

本程序基于实验一的词法分析和语法分析程序，实现了 C - - 语言的语义分析。基于实验指导书的七个假设，本程序实现了对全部 17 种错误的检测。

2. 实现细节

(1) **类型表示**：为实现类型检查和针对不同类型的报错，程序定义了结构体 `Type` 用来存放类型信息。`Type` 结构体包含两个字段：`kind` 表示类型的种属，包括 `BASIC`、`ARRAY`、`STRUCT` 和 `FUNCTION` 四种；`u` 是一个联合体，根据不同的类型种属，存放不同的类型信息。

(2) **符号表**：程序使用哈希表来存放符号。在语义分析前，程序会开辟一个大数组作为符号表，数组的元素为指向 `FieldList` 结构体的指针，`FieldList` 结构体存放了变量的类型和名字，对于结构体类型，`FieldList` 还存放了指向下一个域的指针。

(3) **语义分析**：C - - 源程序可以看作是由若干个 `ExtDef` 构成的。在 `main` 中，程序调用 `traverseTree`，按照语法树的先序遍历顺序进行语义分析。对于每个 `ExtDef` 结点，程序会调用 `analyseExtDef` 进行分析，`analyseExtDef` 再根据结点的类型调用不同的分析函数。

(4) **错误处理**：在语义分析过程中，当发现错误时，程序会构造错误消息并调用自定义的 `error` 函数报错。错误信息的格式为

Error type [错误类型]: at Line [行号]: [说明文字].
其中说明文字尽量与实验指导书保持一致。

二、程序编译方式

1. 实验环境

- 操作系统: Ubuntu 20.04.6 LTS
- 编译器: gcc version 9.4.0
- Flex: 2.6.4
- Bison: 3.5.1

2. 文件结构

```
lab2-code/ (根目录)
├── build.sh (编译脚本)
├── lexical.l (词法分析器源代码)
├── main.c (主程序源代码)
├── nodeType.h (语法树结点类型定义)
├── semantic.c (语义分析器源代码)
├── semantic.h (语义分析器头文件)
├── syntax.y (语法分析器源代码)
├── tree.c (语法树操作函数)
├── tree.h (语法树头文件)
├── utils.c (工具函数)
└── utils.h (工具函数头文件)
```

3. 编译方式

在根目录下执行 `./build.sh` 即可编译程序。编译完成后, 会生成一个名为 `main` 的可执行文件。可以通过 `./main path/to/test.cmm` 运行程序, 其中 `test.cmm` 是待分析的 C++ 源代码文件。

三、程序创新与亮点

1. 分析 StructSpecifier 时区分定义和声明

首先区分如下的几个概念: 结构体声明、结构体定义和结构体变量的定义/声明。在

C - - 的语法规范中，允许对结构体先声明、再定义，即允许如下的语句：

```
struct A;  
struct A {  
    int a;  
};
```

如果在分析结构体时不区分定义和声明，由于结构体名 A 重复出现，程序会在第二个结构体定义处报错误类型 16。因此，本程序在 `analyseStructSpecifier` 函数中通过外层传参的形式对定义和声明进行了区分。当语句为结构体声明时，只需检查符号表中是否存在该结构体名即可；当语句为结构体定义时，如果符号表中没有该结构体名，则正常插入；否则，检查对应记录的 `u` 字段是否为空，若为空则用当前的结构体定义填充，若不为空则发生了重复定义，需报错。对于结构体变量的定义/声明，程序只允许使用已定义的结构体名，若使用不存在或仅声明的结构体名，则报错。

2. 函数相关检查

为支持与函数相关的检查，本程序在指导书给出的 `Type` 结构体的基础上增加了种属 `FUNCTION`，并在 `u` 字段中添加了 `Function` 结构体，用于存放函数的返回类型和参数列表。方便起见，本程序将函数的参数列表设为一个定长数组，最多支持 8 个参数。现在想来也可以类比结构体的做法，使用 `FieldList` 链表来存放参数列表。

3. 多维数组的处理

多维数组的难点主要在类型获取上。本程序在 `analyseVarDec` 函数中，采用先获取类型，再获取变量名的方法。程序通过调用 `analyseVarDecType` 函数获取变量的类型，它是一个递归函数，以 `baseType` 为基础，层层向上封装类型，直到最外层的调用返回。