

Lab3实验报告

1.实现的功能

设计符号表，以此为基础进行语义分析和符号检查，函数定义部分实现了变量声明（定义部分），赋值部分（stmt）有一部分还没有实现。

2.实现方法与设计

编写的代码主要在CmmSemanticVisitor()类中，一开始我将函数的返回值统一改为自己设计的ReturnStruct类，子节点向父节点传值，比如一个int a，在父节点分别收集类型和变量名称。但是这样的问题在于，最终的ReturnStruct类内容过于繁杂，不便于代码维护，bug频出，而且多维数组（超过一维）的访问也难以实现。

所以推翻了第一次写的代码，重新设计，不适用返回值传值，而是使用栈，栈的好处在于值的传递更加简单，而且操作只有出入栈，不会有像ReturnStruct类内容繁杂的情况。配合文法嵌入语义动作（如下的代码）使用，更加易于管理。但是出入栈的顺序需要十分注意，仍然很容易搞错，所以这个结构还是有很多缺陷的。

```
decList[boolean inStruct]: dec[$inStruct] (COMMA dec[$inStruct])*;
```

不同数据类型的表示：

不同的类型全部继承自一个父类

```
public class Type {  
    public Kind kind;  
    public String name;  
    public boolean notLeftValue;  
}
```

notLeftValue是处理左值的时候加入用于标注的，其实应该放在别的地方更加合理，这种设计有一些冗余。

3.印象较深的bug

采用第一次的设计时，没有很好的使用antlr已经创建好的访问树，导致多维数组难以访问，类似 $i=j+1*2$ 以及 $\text{return sub}(1,2)$ （即不是只返回一个数）的式子访问也会出问题，这些式子的处理后来都利用visitChildren()进行，层次比较清晰。