

# 编译原理 Lab2

## 语义分析

实现思路：按照 SDT 的格式对树上的每一个节点进行分析遍历（见下图 1），然后记录符号，其中实验中的所有符号（包括函数）使用**同一张表**，在实验讲义的提供的参考的记录结构体重量中加入了“bool isFunc”来记录当前的节点是否是函数，对于普通的变量采用讲义上提供的数据结构记录相关信息，而对于函数则记录其函数名及一个链表链接各个参数的名称，因为并不需要实现局部作用域，则可以直接记录名称以查全局符号表（一开始没想清楚还记录了各个符号的类型，用了二级指针，写的还累，还多此一举 Orz），所有函数和变量实现均放置在“symbol\_table.c”中，对应声明放置在同名头文件“symbol\_table.h”中；

```
/* SDT FUNCTION */
void ppr_Program(node* root);

void ppr_ExtDefList(node* edlrt);

void ppr_ExtDef(node* edrt);

void ppr_ExtDecList(node*edlrt,vtype stype);

vtype ppr_Specifier(node* srt, int belong, void* st);

vtype ppr_StructSpecifier(node* ssrt, int belong, void* st);

vtype* ppr_VarDec(node* vdrt,vtype stype, int belong, void * gnrc, bool is_dec);

st_node ppr_FunDec(node* fdrt, vtype rtype, bool is_dec);

void ppr_VarList(node* vlrt, void* st, bool is_dec);

void ppr_ParamDec(node* pdrt, void* st, bool is_dec);

void ppr_CompSt(node* csrt, vtype rtype);

void ppr_StmtList(node* slrt, vtype rtype);

void ppr_Stmt(node* srt, vtype rtype);
```

图 1 SDT 函数定义 其中“ppr”代表“prepare”

这次的实验可说的地方好像不是很多，所有函数在做的事情就是两件“存类型”和“查类型”，并分为“生产者”存类型和“消费者”查类型，而具体的 symbol\_table 节点的定义如下：

```
struct st_node_ { //这个是真正的符号表节点，里面既包含了 该符号的名字 也
    int isFunc;
    char* n;
    union {
        vtype vi; //normal variable info
        func_info fi;
    };
    int funcDecLineNum;
    st_node next;
};
```

两种不同的  
数据类型

而本次实验我需要完成选做 2.1（函数的声明的检查），考虑到函数的声明对于中间代码生成并不会实际上造成影响，则本次实验并没有采取讲义推荐的实现：在记录函数节点时同时记录该函数是声明还是定义，觉得这样会污染符号表而且显得比较笨重（平白对符号表节点加了一个域），于是另开了一个链表队列，专门记录一个函数的声明，而判断当前是一个函数的定义还是声明只需要修改 stntax.y 从产生式部分判别即可，在进行完正文的所有扫描后，再去遍历这个队列，然后根据是否存在对应定义以及与定义是否等价来判断额外的错误

18 和错误 19;

除此之外值得一提的是对于 **VarDec** 由于一个变量可能定义于另一个变量中（例如函数的参数及结构体的域），则需要将当前这个 **VarDec** 挂到上层变量的链表中，所以记录了当前变量的定义“**belong**（属于）”谁，由于分析是自顶向下的，则只需要从最高层变量传递下来这个信息(继承属性)以及对应的指针来指向当前底层变量，于是在 **VarDec** 的参数中：

```
vtype* ppr_VarDec(node* vdr, vtype stype, int belong, void * gn timer, bool is_dec);
```

我们加入了“**int belong**”以及“**void \* gn timer**”，**belong** 可以有三种：**BSELF**（自己就是顶层变量）（此时 **gn timer == NULL**）、**BFUNC**（此时 **gn timer** 为传下来函数参数链表的指针）、**BSTCT**（此时 **gn timer** 为结构体传下来的域指针），由于他们类型不同，故需要用通用指针，再根据 **belong** 按照需要进行转换；

好像确实没什么可说的，剩下的调 **BUG** 的艰辛路程都是脑子抽的风就不说了。。

171860580 甘宇航  
gyh8884@gmail.com