

编译原理 实验三 实验报告

181250117 秦锐鑫

设计与实现

实现功能

1. 在假设1-7的情况下，完成错误类型1-17的检测
2. 完成要求2.3

实现方式

自定义 Function_ 结构体，成员包括函数名 char* name、返回值类型 Type returnType 以及参数列表 FieldList parameter；自定义 Structure_ 结构体成员包括结构体名 char* name、域成员 FieldList domain

类型结构体 Type_，在实验手册的 Type_ 结构体的基础上 新增成员 LR_value 用于记录该类型属于左值还是右值；此外 enum kind 新增 FUNCTION，union 新增 Function指针，Structure指针，FieldList结构体同实验手册。

定义结构 Sysmtable_item 作为符号表项

```
struct Sysmtable_item{
    enum{VARIABLE, FUNCTION, STRUCTURE,CONST}kind;
    std::string name;
    Type type;
    int row;
};
```

借助 cpp 中的 map 构造符号表，std::map< std::string, struct Sysmtable_item> Sysmtable;

在获得抽象语法树的基础上，从根基点开始，根据节点语法单元的不同进行语义分析，一个例子如下图

```
void AnalasysForExtDef(tree_node* ptr){
    // 全局变量 结构体 函数
    // ExtDef : Specifier ExtDecList SEMI | Specifier SEMI| Specifier FunDec CompSt
    ...
    tree_node* Specifier_ = ptr->child_node[0];
    Type specifier_type = AnalasysForSpecifier(Specifier_);
    ...
    if(ptr->child_num==3){
        ... // ExtDecList
        AnalasysForExtDecList(ptr->child_node[1], specifier_type);
        ... // or FunDec
        Function function = AnalasysForFunDec(ptr->
        >child_node[1],specifier_type);
        ...
        AnalasysForCompSt(ptr->child_node[2], specifier_type);
    }
}
}
```

分析过程中

- 综合属性作为 AnalasysFor...() 函数返回值，从子节点传递到父节点，或在兄弟节点间传递；（如以上例子的 AnalasysForFunDec 的函数返回值 specifier_type 作为参数传递）
- 继承属性作为 AnalasysFor...() 函数参数，从父节点传递到子节点，或在兄弟节点间传递。（如以上例子的 AnalasysForCompSt 将 AnalasysForSpecifier 的函数返回值 specifier_type 作为参数从父节点传递到子节点以便子节点去判断函数体内返回值是否匹配）

AnalasysForExp 函数最为复杂，包含最多情况，用 if-else 一个个判断。

在遇到检查函数调用是否正确，赋值运算符左右标识符类型是否相同，运算符左右标识符类型是否相同，需要通过 TypeEq(Type type1, Type type2) 函数进行判断

- 首先两者为 nullptr 需要特殊判断；
- 其次 type1.kind 要与 type2.kind 相同：
 - 若 kind 都为 BASIC，则 u.basic 要相同；
 - 若 kind 都为 ARRAY，则 u.array.elem 应该相同，递归调用 TypeEq(Type type1, Type type2)；
 - 若 kind 都为 STRUCTURE，则 u.structure->domain 应该相同，调用 FieldListEq(FieldList fieldlist1, FieldList fieldlist2) 函数进行判断。

有趣的现象 or 印象深刻的 bug

AnalasysForDefList 过程中，可能出现 int a,b,c; float f; 的情况，一开始分析的时候 AnalasysForDef 返回的是首变量，如 int a,b,c; 返回的 FieldList res 是 a，而在 AnalasysForDefList 中，后续直接将 res->tail = AnalasysForDefList(ptr->child_node[1], src) 会导致 b, c 被 f 覆盖，后来改为先迭代取到 FieldList temp 为 c 再让 temp->tail = AnalasysForDefList(ptr->child_node[1], src)

```
FieldList AnalasysForDefList(tree_node* ptr, int src){
//DefList : Def DefList | Eps1
    if(ptr==nullptr)
        return nullptr;
    tree_node* Def_ = ptr->child_node[0];
    tree_node* DefList_ = ptr->child_node[1];
    FieldList res = AnalasysForDef(Def_, src);
    FieldList temp = res;
    while (temp->tail!=nullptr)
    {
        temp = temp->tail;
    }
    temp->tail = AnalasysForDefList(DefList_, src);
    return res;
}
```

字符数组比较字符串内容不能使用 ==

TypeEq(Type type1,Type type2)空指针异常