

编译原理实验二 实验报告

李全昊

201220180

一. 实现的功能及实验内容

实现的功能：完成了所有的必做和选做内容

在这个实验中，我主要实现了一个基于十字链表和 open hashing 散列表的符号表，并且对应的写了基于 lab1 的语法树的语义分析程序

主体代码分为两个文件：semantic.c 和 symbol_table.c

在 semantic.c 中主要用递归调用来实现对于各个语法单元的语义分析的处理

Symbol_table.c 中主要是对于符号表的实现，包括插入 symbol，找到 symbol 等功能（这一块我当时写的时候经常出现对于空指针的引用的 bug！应该多多注意）

这个符号表还包括了一个 domain 的栈，每个元素对应于不同深度的 domain，注意这里不用担心为什么每个深度只用设置一个 domain，因为比如说一个函数的作用域退出之后，会调用 exit_domain 函数，然后这个作用域就会被 pop 出作用域栈！下次同一个深度的作用域到来的时候就不会重复了

二. 实验中遇到的 bug 和一些思考

这次试验真的痛苦，遇到了好多好多 bug，debug 了好久。。。

1. 不能把结构体当作函数的作用域来看！否则定义完一个结构体就直接 exit_domain 就会把结构体踢出符号表，所以要为结构体单独设置一个表！

每次插入一个结构体变量的时候，我会把这个结构体插入到整体的符号表中，然后把结构体内定义的变量插入到另外一个结构体表中

2. 在碰到 int i,j;这种定义的时候，不能把 i 和 j 的 type 指向同一个元素，否则删除的时候可能两个都删除了（不过后来我直接不 free node type 了，所以这也不是问题了）

3. 注意 deflist 和 deflist_struct 中调用的 find_symbol 应该只找同一层作用域的（外层作用域即使有，也可以定义）！意思是定义变量的时候应该只找同一层作用域中是否有重复名字的变量！外层作用域即使有也没关系，我 find_symbol 的时候只需要找深度最深的那个 symbol 就可以了

三. 编译方法

使用 make 编译，make test 测试相应 test 文件夹的对应文件，或者 ./parser test.cmm 测试

四. 总结

这次试验真的工作量太大了。。。debug 了一周多总算搞完了。（再次感叹一声）