Lab2 实验报告

161220117 唐诗美

一、功能

实现了讲义上以那七个假设为前提，能分析出 17 种语义错误；

实现了结构等价（选做 2.3）；

类型等价比较还实现了数组不同维数的比较；

二、实现方法

首先在 gtree.h 中加入了 Type_结构，可以用于记录一个符号的类型（整型浮点、数组、结构体）信息，并且对语法树的结构也做了一些调整，在其中加入了一个 union，记录一个符号是普通类型还是一个函数；还加入了一个 n_type，用于记录是结构体还是函数还是一般的变量，另一个加入的变量是 rule，用于记录是对于那条语法规则的分析，用于后面的分析。还添加了两个用于记录数组维数和符号左右值的变量。

symbol_table.h 主要是对于符号的处理，这里把函数和一般的变量符号分成两张表记录，其中每个表都是用链表实现的。其中 Symbol 结构是记录了一般符号的种类（一般变量还是结构），符号的名字和类型，SymbolF 是记录了函数名，函数的返回值参数数量和参数。通过 add_symbol(F)和 find_symbol(F)在符号表中进行添加、查找。

```
struct Symbol_ {
    IDkind idkind;
    char name[55];
    Type type;

    Symbol next;
};
struct SymbolF_ {
    char name[55];
    Type retType;
    int argc;
    Type* argv;

    SymbolF next;
};
```

主要的语义分析是在 semantic.c 这个文件中，对于每个规则都有一个处理的函数，从最开始的规则逐步递归地分析它的每个孩子，分析完后再把一些属性赋给父节点。写的过程中主要的难点有结构体的处理，它的实现是通过创建一个 FieldList 类型名为 tmp_table 的数组，利用一个全局变量 top 代表栈顶，被初始化为-1，当解析到符号是结构后，把符号压栈，接着分析它其中的变量声明。解析变量时，如果 top 不为-1，表示现在解析的符号是在结构体中的，于是把当前分析的变量存到 tmp_table[top]最后一个 fieldlist 的 tail 上，从而把整个结构体连起来。最后分析完了再把这个结构体变量放进符号表中，连带着它的 fieldlist 中的内容。

对于函数的分析，在节点中有一个 Type* argv 的属性，就像一个指针数组，argv 可以存储一个函数的所有参数内容。

还有就是类型判断问题，对于类型判断我把数组的判断和其他类型的判断分开处理，对于是数组的类型，首先根据这个节点的维数分析到这个维数的数组的类型是什么后再把他们进行比较；具体实现如下：

```
1.  int isEqual_array(Type a, Type b, int a_dim, int b_dim) {
2.      Type a_cmp = a;
3.      Type b_cmp = b;
4.      for(int i = 0; i<a_dim; i++) {
5.          a_cmp = a_cmp->u.array.elem;
6.      }    //找出把 a 所有维数拆到底的元素类型
7.      for(int i = 0; i<b_dim; i++) {
8.          b_cmp = b_cmp->u.array.elem;
9.      }    //找出把 b 所有维数拆到底的元素类型
10.     if(!isEqual(a_cmp, b_cmp))    //再用判断类型等价函数进行判断
11.         return 0;
12.     return 1;
13. }
```

对于结构类型的结构体等价，通过把它的 FieldList 遍历一遍，对于每个类型再用判断类型是否相等的函数进行比较达到。

三、数据结构

本次实验利用了链表结构去实现符号表，在很多地方，比如 structure 中的 Fieldlist 也是利用到了链表结构。

四、编译运行方法

本次实验是在 ubuntu16.04 中进行的。

通过在 code 文件夹中输入“make”指令，得到 parser，运行 parser 进行解析即可。test_Project_2 中的测试是讲义中的 test，my_test 是自己写的测试。

五、实验总结

通过本次实验，对于如何翻译一棵语法树有了更深的印象。这次实验一开始不太明白的地方主要是对于不同符号需要怎么存储，特别是结构体。实验写好后，在调试的过程中最主要发生的错误提示就是 segmentation fault，这是因为对于每次指针的使用没有分配内存造成的，而且在访问指针结构体中的域的时候，没有事先判断这个指针是否存在。还有一些错误主要是因为没看清变量输入错误造成。