Lab2 实验报告

201850187, 周琰轲，zyk@nju.edu.cn

2023 年 4 月 14 日

1 功能

完成了实验手册要求的所有功能，并支持所有 19 个文法错误的报错。通过了实验手册上的所有测试样例。

2 设计特色

2.1 符号表

采用十字交叉哈希表存储符号，利用栈指针 *stack_top* 指定当前在哪个作用域下；相对应的，每一个符号表表项是一个结构体 *sym*，该结构体的两个指针分别指向下一个哈希表符号和下一个在同一作用域下的符号。同时，该结构体有一项无符号整型成员 *StackDepth* 用于记录该符号在哪个作用域下。我们约定处于 *StackDepth* 高的作用域下可以访问 *StackDepth* 低的作用域的符号，反之则不行。

同时为了应对额外要求 2.1，即在出现同名符号的情况下优先选择内部作用域的符号，在向符号表插入符号时优先将符号插在头部，保证后被定义的符号优先被选取。

2.2 作用域处理

在初次实现中我并没有实现深拷贝等手段，而是直接采用指针赋值，这样以来使得哪些内存在哪些地方被申请、应该在哪些地方被释放尤其混乱，因此我没有实现内存的释放。

由于不释放内存，为了满足额外要求 2.1，在进入一个 LC RC 时，仅仅做栈指针的变换是不够的，为了保证栈 pop 时当前作用域下各符号失效，我设计了 StackPop(int teardown) 函数，如果要销毁当前作用域下的符号，该函数会将当前作用域下所有符号的 StackDepth 成员设为 (unsigned int) (-1)，这样一来，这些符号的 StackDepth 变成理论上最大值，根据我们之前的假设，这些符号不可能被任何低于 (unsigned int) (-1) 的作用域访问到，等价于实现了销毁。

2.3 类型结构体

按照讲义设计了 BASIC，ARRAY，STRUCTURE，FUNCTION 四种类型，对后三种设计了三种相应的结构体。除了按照讲义的建议设计以外在 STRUCTURE 内部设置变

量 IsAnony 判断该结构体是否是匿名的；在 FUNCTION 内部设置变量 isDef 判断该函数是定义还是声明。

针对错误 18，19，我之前的设计是使用一个全局数组，记录下那些被声明的函数，在最后对这个数组进行遍历，如果在符号表中能找到同名函数且 isDef 属性为 1，说明不存在错误 18，反之报错。但这样存在一个问题，例如下面的例子：

```
int func(int a);
float func(int a);
int func(float a) {
    return 1;
}
```

正确报错应该是 2，3 行报错 19，但我的上述实现使得最后还会报一句 line 1 function undefined，这是因为第三行检测到冲突后就没有把这个 func 加入符号表了。因此我额外设计了一个判定数组，每当出现文法 Specifier FunDec CompSt，直接把 FunDec 对应的 ID 所对应的判定数组的值设为真。在最后的对错误 18 的判断中，如果判定数组为真，说明要么该函数已经定义过了，要么定义过但已经报错 19，不需要再报错。

3 致谢

感谢邓文泰和张旭清两位同学，相互之间的交流和讨论解决了不少实验中碰到的困惑和难点。