编译原理Lab2实验报告

任务号：11
组长：董启轩 181860016
组员：田浩 181860088
联系方式：1273440689@qq.com

一、编译及测试环境

虚拟机：Ubuntu16.04.5
版本：gcc 5.4.0
flex 2.6.0
bison 3.0.4
编译方式：使用提供的makefile编译
分工：田浩：根据语法树进行语义分析，产生语义错误
董启轩：符号表和相关功能接口的实现

二、实验部分

（本次实现完成选做2.1，实现函数声明与定义）

(一)符号表的实现

本次实验的符号表在指导提供的数据结构的基础上作出一定修改。

Type_结构中，将枚举类型拿到了外面重新定义为KIND枚举类型，区分int和float，添加FUNCTION；联合体u中的structure除了保存域列表，还要保存该符号所属结构体名称；u添加function，function中保存函数声明所在行号，返回值类型，参数列表。FieldList_中再额外保存定义所在行号。

定义结构体TableNode用于创建符号表，保存符号名，所属Type，用于表示是否定义(只用于函数)的ifdef和一个指向下一节点的指针。其实主要目的是区分FieldList(参数列表或域列表)和符号表。

符号表采用哈希表，开散列的形式，使用提供的hash函数。同时为了便于区分和语义分析，将结构体名放在一个StructTable中，函数名放在一个FunctionTable中。将函数单独分开是因为指导中没有提及变量等不能与函数名重名，所以决定将函数单独分开(但形参依旧需要放进符号表)；将结构体名单独分开是因为一个结构体是一种自定义的类型(即类似于int 和 float)，所以为其单独建表，但依旧不能与变量等重名。

同时为了完成向语义分析器传递声明未定义的函数报错信息，定义一个报错用结构体UndefinedFunction，用于存储报错信息。

基于名等价机制，为所有匿名结构体起名，这个名字需要避开ID的表示范围，所以使用单纯的数字串为其取名，由一个int数namelessStructNumber转换为数字字符串。

提供的函数大致有如下分类：

（1）创建基类，数组类，结构体类，函数类的Type的函数create_xxx_Type(…)

（2）涉及FieldList链表的节点创建，插入，删除，查重等一系列操作的函数

（3）插入节点insert_Node(Type type_in,char* name)，在这里将完成向3个表插入节点的操作以及匿名结构体命名；查询对应名称的Type，以及各类查询Type中存储的数据的函数

（4）判断数据状态的函数，如if_define(char* name)函数用于查看函数的ifdef值，判断其是否定义；以及判断匹配性的函数，如检查type是否等价的函数same(Type A,Type B)

（5）报错函数，如生成报错字符串的generateStr(FieldList f)和generateFuncStr(char* name)和debug用函数，为打印符号表和FieldList信息的printTable( )和printFieldList(FieldList f);

语义分析器将调用前三类函数创建对应的节点并插入符号表，查询节点信息；使用第四类函数获取返回值决定是否报错、报错类型和执行顺序；使用第五类函数获取其自身获取不到的错误信息，其中debug函数主要使用在debug阶段。

(二)语义分析与报错

先为语法树的节点类型添加继承属性与综合属性，其类型定义如下：

```
union SDDVal{
    struct{
        char *name;
        FieldList f;
    } func;
    Type t;
    char *str;
    FieldList f;
};
```

然后从根节点开始进行语法制导，对每个产生式，有以handle_开头，后面跟产生式名的函数用以处理产生式，其中也会调用handle_开头的函数用以处理产生式中的语法单元。

因为结构定义中的DefList和其他地方的DefList里面逻辑有所不同，为了加以区分，使用一个全局变量instruct，初始值为0。在进入一个结构定义的语句时(即在StructSpecifier节点的处理中)，有以下操作：

```
int old_instruct = instruct;
instruct = 1;
处理结构定义中的语句;
instruct = old_instruct;
```

以下是一些重要的语法单元的处理方式：

Vardec:节点有继承得来的类型属性，判断产生式的体为ID还是Vardec LB INT RB。若为后者，则用继承来的类型属性创建数组类型继承给体中的Vardec并调用handle_VarDec，将处理过后的VarDec的综合属性作为该节点的综合属性返回；若为前者，则用类型与ID名创建一个FieldList节点并用写入综合属性返回。最终，Vardec的综合属性为由这个变量的ID与类型生成的FieldList。

Dec:有继承得来的类型属性，将类型属性继承给产生式体中的Vardec然后进行处理。之后根据instruct的值判断是否在结构定义当中，不是的话则用Vardec综合属性里的FieldList中的信息将符号插入符号表。并且，Dec将会将Vardec的综合属性作为自己的综合属性返回。

DefList,Def,DecList:有继承得来的类型属性，将其向下继承。子节点的综合属性都为FieldList，将其拼接起来作为自己的综合属性。

StructSpecifier：判断产生式的体为定义一个新的结构体还是使用之前定义好的结构体，后者查表得到结构体类型，前者处理DefList节点，得到里面所有变量信息构成的FieldList，将里面的所有变量插入符号表，然后用这些信息创建一个新的结构体类型插入表中。将类型作为综合属性返回。

ExtDef:分辨产生式的体为变量定义，单独一个类型符号，函数声明，还是函数定义。前两个调用对应的处理函数即可，对于函数，将处理得到的函数插入表中，如果为函数定义，则再将其设为已定义。

Exp:传回表达式的类型作为综合属性。

完成错误处理，主要在原本的语法制导当中添加各种检查，将插表等函数和其相关的检查动作封装成函数，在合适位置传入合适的参数进行调用即可。发生错误时，一部分语句的处理方式是直接返回，不能完全被执行。