编译原理（2）实验报告

张语婷-181860140-普通班-1051570722@qq.com

一、 实现功能 基础+选做 1.1

1. 加入函数声明的文法： ExtDef->Specifier FunDec SEMI
2. 符号表及函数表：

```
struct TABLE* table[16384];//储存结构体或变量
struct FUNCTION* function[16384];//储存函数定义或声明
```

```
struct TABLE{
    int is_def_struct;
    FieldList field;
    struct TABLE* next_for_openhash;
    int linenumber;
};
```

```
struct FUNCTION{
    char* name;//函数名
    FieldList field;//参数列表
    Type return_type;//返回值
    int declaration;//声明个数
    int definition;//定义个数
    int linenumber;//用于错误打印
    struct FUNCTION* next_for_openhash;
};
```

3. 每一个语法单元设置相应函数，根据产生式进行调用
   1) `Program`
      调用 ExtDef( )
      在所有递归调用完成返回后检查 error 18，是否存在函数只声明无定义
   2) `Specifier`
      由 TYPE 类型生成，则直接设置 type=int/float
      由 StructSpecifier 类型生成，则调用 StructSpecifier( )函数
   3) `StructSpecifier`
      - STRUCT OptTag LC DefList RC：读取 OptTag 中的 ID 信息，生成对应的结构体名称，判断该命名是否已被使用(error 16)，调用 DefList(judge=0)函数生成域，将该 struct 加入符号表中
      - STRUCT Tag：读取 Tag 中的 ID 信息，在符号表中查找，找到则返回该结构体对应的 Type，未找到则对应 error 17
   4) `DefList\ DecList\ VarList`
      根据产生式递归调用，并将相继产生的 FieldList 通过指针 tail 连接
   5) `ExtDecList\ StmtList`
      根据产生式递归调用

6) Dec
- VarDec：调用 FieldList=VarDec( )
- VarDec ASSIGNOP Exp：根据传递的 judge 值，结构体定义(judge=0)对应 error 15(定义时对域进行初始化)，否则调用 exp_type=Exp( )，若 exp_type 与 type 不同，则对应 error 5(赋值号两边表达式类型不匹配)

7) VarDec
- VarDec LB INT RB：设置当前 vardec_type 为 ARRAY，并向上传递调用 VarDec(child,vardec_type,judge)，即递归生成的下一个 FieldList 的 type 类型为 vardec_type

```
struct Node* int_num=fir_bro->brother;
Type vardec_type=(Type)malloc(sizeof(struct Type_));
vardec_type->kind=ARRAY;
vardec_type->u.array.size=int_num->int_number;
vardec_type->u.array.elem=type;
FieldList find_upper=VarDec(child,vardec_type,judge);
return find_upper;
```

- ID：创建一个 FieldList，读取 ID 的值
  1-函数声明的参数：直接返回 type，不加入符号表
  2-函数定义的参数或变量定义：符号表中查重(error 3)，不重复则加入符号表
  3-结构体的定义：符号表中查重(error 15)，不重复则加入符号表

8) FunDec
创建一个 FUNCTION，设置对应的 type\name 等值
- 函数声明：判断是否第一次声明(加入 function 表)，声明冲突(error 19)，或同一函数多次声明(declaration++)
- 函数定义：判断是否第一次定义(加入 function 表)，与声明冲突(error 19)或多次定义(error 4)

9) Stmt
- Exp SEMI：调用 Exp ( )
- CompSt：调用 CompSt ( )
- RETURN Exp SEMI：调用 exp_type=Exp ( )，比较 exp_type 与 type 的值，若不同则对应 error 8(return 语句的返回类型与函数定义的返回类型不匹配)
- IF \ WHILE 语句：调用 Exp ( )，若 Exp 不为 int 类型，则 error 7(操作数类型不匹配)，若匹配则调用 Stmt( )

10) Exp
- INT \ FLOAT：直接生成对应 type
- ID：查找符号表是否存在，不存在则对应 error 1(变量在使用时未经定义)，存在则直接返回该 FieldList 的 Type
- MINUS Exp \ LP Exp RP：调用 Exp( )
- NOT Exp \ AND \ OR：调用 Exp( )，若不是 int 类型，则对应 error 7
- ASSIGNOP：判断左端 Exp 是否满足左值的三个产生式，不满足则对

应 error 6(赋值号左边出现一个只有右值的表达式)，若左右 Exp 类型不同，则对应 error 5(赋值号两边的表达式类型不匹配)

- RELOP \ PLUS \ MINUS \ STAR \ DIV：若左右 Exp 类型不同，则对应 error 7(操作数类型不匹配)
- Exp DOT ID：判断左端 Exp 是否为 STRUCTURE 类型，不满足则对应 error 13(对非结构体型变量使用“.”操作符)，判断右端 ID 是否为定义过的域，不满足则对应 error 14(访问结构体中未定义过的域)
- Exp LB Exp RB：判断左端 Exp 是否为 ARRAY 类型，不满足则对应 error 10(对非数组型变量使用“[…]”操作符)，判断右端 Exp 是否为 int 类型，不满足则对应 error 12(数组访问操作符“[…]”中出现非整数)
- ID LP Args RP \ ID LP RP：判断左端 ID 是否在 table 表，在则对应 error 11(对普通变量使用“(…)”或“()”操作符)，判断左端 ID 是否在 function 表，不在则对应 error 2(函数在调用时未经定义)，调用 Args( ) 判断函数参数是否符合规范，不满足则对应 error 9(函数调用时实参与形参的数目或类型不匹配)

11) Args

调用 FieldList->type= Exp ( )

并将相连的 Args 产生的 FieldList 通过指针 tail 连接

二、 编译语句

make

./parser name.cmm （name.cmm 为 cmm 文件名字）