

实验二

171860572 侯策 171860572@smail.nju.edu.cn

171860588 史文泰 171860588@smail.nju.edu.cn

实验内容

数据结构设计

- 类型结构体设计

- 涉及的类型有：基本类型 BASIC，数组的类型 ARRAY，函数类型 FUNCTION，结构体类型 STRUCTURETYPE，结构体变量类型 STRUCTURE。
- 结构体类型 STRUCTURETYPE 和结构体变量类型 STRUCTURE：
 - 结构体类型：定义一个结构体，该结构体类型对应的 kind 域为 STRUCTURETYPE。
 - 结构体变量类型：用一个结构体类型来定义一个变量，该变量的类型称为结构体变量类型，对应的 kind 域为 STRUCTURE。
- 上述两种表项中均包括结构体名称 name 以及各域类型和名称 fieldList，使用 struct { char* name; FieldList fieldList; } structure; 来存储其 Type。
- 函数类型 Function_：
 - 由选做 2.1，添加文法：ExtDef: Specifier FunDec SEMI，用于表示函数声明。
 - 由选做 2.1，额外设置一个域 isDefined; 来标记该函数是仅声明还是已定义。

```
struct Function_{
    Type returnType;           // 返回值类型
    FieldList paramType;       // 参数类型
    int paramNum;              // 参数数量
    int isDefined;             // 是否被定义过
};
```

- 符号表类型设计

- 符号表采用散列表，开地址闭散列，桶的个数为1024，采用文档中给出的 hash_pjw() 函数进行散列。
- 符号表内存的符号类型有：变量（包括普通变量，数组变量和结构体变量），函数，结构体类型。

- 此处举例解释：结构体类型与结构体变量类型在符号表里的具体存储形式。

```
struct A{
    int x;
    float y;
} vector;
```

name	type		Lineno
	type.kind	type.u	
A	STRUCTURETYPE		
vector	STRUCTURE		

两者的 `type.u` 和 `lineno` 属性相同。

语义分析实现要点

- 语义分析的整体流程即在**实验一构造的语法分析树（假定没有语法错误/词法错误）**上进行遍历，在语法树上计算 SDD ，并根据语法错误的种类在特定的终结符或非终结符结点上进行语法错误分析。对于每种语法树结点，定义相应的处理函数。
- 实验中的**特殊处理**如下：

- 采用**统一的符号表** `symbolTable` 存储所有的符号（变量，函数类型，`struct` 类型）。
- `struct` 本身类型和 `struct` 所定义的变量均存入符号表，但是以不同的 `Type` 加以区分。

定义 `struct` 类型符号的目的是：

- 当用 `struct` 类型 `A` 来定义变量 `a` 时，可以查找符号表判断 `A` 类型是否已经被定义过，并获取 `A` 类型的 `Type` 作为变量 `a` 的 `Type`。
 - 可以判断 `struct` 类型 `A` 是否与前面定义过的结构体或变量名字重复。
- 由于结构体在定义时的特殊性，需要在定义时判断其域名是否与之前的域名重复，且将 `fieldList` 域填写完整，因此将结构体定义的相关部分与非结构体定义的部分拆分，在函数名前加上 `structure` 作为区分。即：在 `StructureSpecifier` 的后续推导中出现的函数命名为：`structureDefList`，`structureDef`，`structureDecList`，`structureDec`，`structureVarDec`（叶节点）。在上述函数中将 `struct` 类型符号 `A` 的 `Type` 域作为继承属性传入，并在叶结点处填充其 `fieldList` 域，而不在 `struct` 的推导过程中的变量定义无需做上述处理，故名称正常为：`DefList`，`Def`，`DecList`，`Dec`，`VarDec`。
- 对于函数参数，实验中的选择为：
 - 无论为函数定义还是函数声明，函数类型符号都需要存入符号表
 - 只有函数定义中的参数才会作为普通变量存入符号表，并正常进行错误判断。

只存入函数定义中的参数的原因是：

- 同一函数的函数声明可能存在多次，且其参数名称可能重复（或与函数定义的参数名称重复），这并不应报错。只有函数定义中的参数名称才被认为是全局变量，无法重复。
- 函数类型符号的处理与 `struct` 类型符号相似，需要将其 `Type` 作为继承属性自参数分析 `VarList` 开始，传递至参数定义处并填充其 `fieldList` 域，可调用 `structure` 对应函数处理。
- 名等价处理**：对于匿名结构体，给予其独有名字：利用 `ID` 不能以数字开头的特性，将整数利用 `sprintf()` 函数转换为字符串，给予匿名结构体，使之能够正常进行名等价处理。

编译方法

- 实验严格按照提供的 `makefile` 文件进行编译与执行。命令如下：

```
make clean  
make parser  
make test
```

若需要添加新的测试文件,则在 `Test` 文件夹中添加文件,在 `makefile` 中修改对应的指令即可。

实验感悟

- `Specifier()` 函数返回类型 `type` , 并将其传递给后续表达式处理函数, 直至到达叶子结点, 进行类型判断与后续检查。这里的 `type` 即为理论学习中的继承属性, 向下层层传递。