
编译原理实验2：语义分析

实验报告

吴澄杰

151220122

wuchengjie@smail.nju.edu.cn

2018年5月27日

一、实验进度

此次试验，在词法分析和语法分析的基础上，我实现了C语言代码的语义分析的程序，可以对词法语法正确的代码执行语义分析和类型检查，并能够报告错误。

我完成了实验的必做要求：程序可以在所有变量不重名、结构体类型名等价等基础上进行结构体定义、类型声明、类型检查等功能。若C语言代码没有语义错误则不输出信息，反之打印错误信息。

我通过了所有必做测试样例。

二、编译与实验

在工程目录下，输入 make 命令，即可完成编译，可生成程序 parser。用C语言源代码文件名作为参数运行程序 parser 就可以得到在该文件上的词法分析和语法分析结果。

工程目录下的 example2 文件夹中有实验2的17个必做样例的源代码和正确结果。若要检测程序在样例上运行结果的正误，只需将样例文件名作为参数给到 parser 运行即可。比如，`./parser ./example2/example1.txt` 就可以得到程序在第一个测试样例上的运行结果。

三、具体功能描述

1. 语义分析概述

整个语义分析模块由 type.c、symbol.c、semantic.c 及其对应头文件来实现。其中 type 定义了综合表示基本数据类型、结构体、数组及其嵌套结构的数据类型，并且实现了对 Specifier 非终结符的语义分析。

symbol 定义了符号和符号表。在我的实现中，变量、结构体、函数的名称存储在一张符号表中，以名称作为它们的唯一标识符。符号表以链表的方式进行存储。并且实现了符号表中的插入、查找等操作。

semantic 是语义分析的主要过程。其基本结构是，对每一个非终结符定义相应函数进行语义分析，对每一条产生式规则定义合适的语义分析过程。在涉及到符号、数据类型等信息的传递的时候，则通过函数参数传递和返回值来进行属性值的传递。

2. type 模块

数据类型的定义如下：

```
enum typeKind {INTtype, FLOATtype, ARRAY, STRUCTURE};
struct Type {
    enum typeKind kind;
    union {
        struct {
            struct Type *eleType;
            int size;
        } array;
        struct FieldList *structure;
    };
};
struct FieldList {
    char name[FILED_NAME_MAX_LEN];
    struct Type *fieldType;
    struct FieldList *next;
};
```

数据类型要么是基础数据类型，要么是结构体或者数组。数组需要记录数组的长度和数组元素类型，结构体则需要记录每一个域的名称和类型。

3. symbol 模块

符号的定义如下：

```
enum symbolKind {VARIABLE, FUNCTION, STRUCT};
struct Symbol {
    enum symbolKind kind;                // 符号类型
    char name[SYMBOL_NAME_MAX_LEN];      // 符号标识符
    struct Type *structure;               // 表示数据类型。在FUNCTION中表示返回值类型
    struct Symbol *params;                // 维护函数定义的参数表中的指针
    struct Symbol *next;                  // 维护符号表中的指针
};
```

每个符号需要记录符号名、数据类型及指向符号表中下一个符号的指针。其中next是一个比较特殊的指针。它被用于函数声明时维护参数列表这一链表。

4. semantic 模块

这一模块是语义分析实现的主体。它包含了很多函数，对每一个非终结符的每一个产生式都定义了规则进行语义处理。比如，处理定义的结构体、变量、函数，并将相应信息记录在符号表中；对表达式进行类型检查；检查函数参数和返回值是否匹配等等。在此过程中，若发现语义错误，则报告相应的错误类型和编号。此过程是一个利用综合属性和继承属性进行语义分析的过程。继承属性可以作为函数的参数一层层向下传递，综合属性则可以通过函数的返回值向上传递。

四、实验总结

此次实验，我基于上一步实验所实现的语法树，利用综合属性和继承属性实现了语义分析、类型检查等内容。总体思路为，在定义时记录好变量、结构体、函数的各项属性信

息，然后在使用每一个产生式的时候检查相应类型是否有误。实验复杂的地方是对每一条产生式都要具体地定义语义分析方案，并且在整个语义分析过程中，要通过函数参数和返回值的传递来实现综合属性和继承属性的传递。