

# 哈尔滨工业大学

## 编译系统 2022 春

### 实验二

学院:	计算学部
姓名:	冯开来
学号:	1190201215
指导教师:	陈鄞

#### 一、实验目的

- 巩固对词法分析与语法分析的基本功能和原理的认识
- 能够应用自动机的知识进行词法与语法分析
- 理解并处理词法分析与语法分析中的异常和错误

#### 二、实验内容

##### (一) 实验环境

Ubuntu 12.04, kernel version 3.2.0-29; GCC version 4.6.3  
GNU Flex version 2.5.35; GUN Bison version 2.5

##### (二) 实验过程

###### 1. 实现功能

本次实验实现了必做的错误检测（未标黄项目）。以及部分标黄的选做内容。

###### 2. 数据结构

本次实验主要用到的数据结构包括变量类型 Type，还有字符表中的一个域 FieldList。这一部分的书写参照实验指导书即可。

###### 3. 语义分析

语义分析是在前两次的实验基础上完成。大致流程就是在上一次成功建立语法树之后，再遍历一遍树，遍历树的过程中检查语义错误。我以几类错误检测为例，简单讲一下语义分析过程。

**3.1 变量出现重复定义或变量定义过的的结构体名字重复。**如语法单元 VarDec, 给出的语法规则为  $\text{VarDec} \rightarrow \text{ID} \mid \text{VarDec LB INT RB}$ 。这是一个声明语句，可以申明一个普通的变量名或者一个数组。当语法树遍历到这个语法单元时，我们构造函数 `void VarDec(Tree root, Type type)`，形参是此时的根节点和综合属性类型。然后通过儿子和兄弟的遍历这棵树找到他的下一个叶节点。并且申请一个域用来存放下一个叶节点（也可能是子树）的信息。

```
void VarDec(Tree root, Type type)
{
    // VarDec -> ID
    //           | VarDec LB INT RB
    // printf("in VarDec\n");

    Tree id = root;
    // 不断迭代找到变量名
    while (id->child)
        id = id->child;

    FieldList tmp = createField();
    strcpy(tmp->name, id->s);
}
```

如果是 ID，那么我们就将 ID 的信息放入这个域 tmp。如果产生式是 VarDec LB INT RB，那么说明此时是一个数组类型，同理，我们创建一个数组变量信息放入这个域 tmp。

最后，我们搜索我们之前创建的符号表是否包含 tmp 这个域，如果有则说明之前已经定义该变量名，程序输出 Redefined variable ‘tmp->name’。如果符号表中没有该信息，我们则将该变量加入符号表。

```
if (!check(tmp))
{
    printf("Error type 3 at Line %d: Redefined variable \"%s\".\n",
        root->line, tmp->name);
    freeField(tmp);
}
else
    addTable(tmp);
```

**3.2 赋值号两边的类型表达式不匹配或者赋值号左边只有一个右值的表达式**这类错误。对应于语法分析的  $\text{EXP} \rightarrow \text{Exp ASSIGNOP EXP}$  这一部分。我们同样声明一个域，当匹配到 root->child->brother->unit 为 ASSIGNOP 时，我们进行错误检测，首先判断左边是否是 INT 或是 FLOAT，如果是，则形如 1=... 这样的形式，这显然时左边是一个右值表达式的错误，输出**错误类型 5**。如果左边 id 的 basic 和右边 id 的 basic 不一样说明两边表达式类型不匹配，说明不可进行赋值操作，输出**错误类型 6**。

**3.3 操作数类型不匹配或操作数类型与操作符不匹配。**在进行移入-规约操作时，即对应  $\text{Exp} \rightarrow \text{Exp PLUS Exp} \mid \text{Exp MINUS Exp}$  时，程序对运算表达式两侧的表达式类型进行检验。若两侧表达式类型不相等则输出**错误类型 7**，类似上面的赋值号两边不等。

**3.4 数组访问操作符[.]中出现非整数。**每遇到数组访问语句的时候，对应于  $\text{Exp} \rightarrow \text{Exp LB Exp RB}$ ，程序对字节节点第三个属性值 root->child->brother->brother，若不为 int，说明出现非整数，输出**错误类型 12**。

**3.5 对非数组型变量使用数组访问操作符。**同样对子节点的第三个属性值进行判断，若它的 kind 不为数组，则说明该节点对应的是非数组变量，输出**错误类型 10**。

#### 4. 编译过程

本次实验采用 Makefile 进行编译。具体内容如下图所示：

```
objs = lex.yy.c scanner syntax.yy.c \
       syntax.tab.c syntax.tab.h parser \
       syntax.output

cc: main.c lex.yy.c syntax.tab.c semantic.c
    gcc main.c syntax.tab.c semantic.c -lfl -ly -o cc

scanner: main.c lex.yy.c
    gcc main.c lex.yy.c -lfl -o scanner

lex.yy.c: lexical.l
    flex lexical.l

syntax.tab.c: syntax.y
    bison -d -v syntax.y

syntax.tab.h: syntax.y
    bison -d syntax.y

.PHONY: clean
clean:
    rm -rf $(objs)
```

### (三) 实验结果

必做内容的几个结果：

```
carlofkl@ubuntu:~$ ./parser 1.cmm
Error type 1 at Line 4: Undefined variable "j".
carlofkl@ubuntu:~$ ./parser 3.cmm
Error type 3 at Line 4: Redefined variable "i".
carlofkl@ubuntu:~$ ./parser 5.cmm
Error type 5 at Line 4: Type mismatched for assignment.
carlofkl@ubuntu:~$ ./parser 6.cmm
Error type 6 at Line 4: The left-hand side of an assignment must be a variable.
carlofkl@ubuntu:~$ ./parser 7.cmm
Error type 7 at Line 4: Type mismatched for operands.
carlofkl@ubuntu:~$ ./parser 10.cmm
Error type 10 at Line 4: "i" is not an array.
carlofkl@ubuntu:~$ ./parser 12.cmm
Error type 12 at Line 4: "1.5" is not an integer.
```

选做内容的几个结果：

```
carlofkl@ubuntu:~$ ./parser 2.cmm
Error type 2 at Line 4: Undefined function "inc".
carlofkl@ubuntu:~$ ./parser 4.cmm
Error type 4 at Line 6: Redefined function "func".
carlofkl@ubuntu:~$ ./parser 8.cmm
Error type 8 at Line 4: Type mismatched for return.
carlofkl@ubuntu:~$ ./parser 9.cmm
Error type 9 at Line 8: too many arguments to function "func", except 1 args.
carlofkl@ubuntu:~$ ./parser 11.cmm
Error type 11 at Line 4: "i" is not a function.
carlofkl@ubuntu:~$ ./parser 13.cmm
Error type 13 at Line 9: Illegal use of ".".
carlofkl@ubuntu:~$ ./parser 14.cmm
Error type 14 at Line 9: Non-existent file Exp.
carlofkl@ubuntu:~$ ./parser 16.cmm
Error type 16 at Line 6: Duplicated name "Position".
carlofkl@ubuntu:~$ ./parser 17.cmm
Error type 17 at Line 3: Undefined structure "Position".
```

## 三、实验总结

1. 学会了如何实现简单的语义分析任务。