

编译原理Lab1实验报告

201840281 史成璐 201840281@smail.nju.edu.cn

实验一的任务是编写一个程序对使用C++语言书写的源代码进行词法分析和语法分析，并打印分析结果。实验要求使用词法分析工具GNU Flex和语法分析工具GNU Bison，并使用C语言来完成。

功能

程序可以正确输出实验一介绍中的所有必做与选做样例的结果。

1.查出源代码中可能包含的下述几类错误

(1) 词法错误（错误类型A）：即出现C++词法中未定义的字符以及任何不符合C++词法单元定义的字符。

该功能的实现是使用flex完成，在lexical.l中定义了一系列正则表达式和相应的响应函数。其中一部分正则表达式在Appendix文件中有提供，自行完成了INT、FLOAT、ID的正则表达式定义。

其中，INT型整数并非直接实现，在阅读实验要求后，我是用无符号二进制、八进制、十进制和十六进制整数来表示。二进制整数以0b|0B开头，由01串组成，八进制整数由0开头，由[0-7]组成，十六进制以0x|0X开头，由[0-9a-fA-F]组成。

FLOAT正则表达式可以识别小数点形式以及指数形式的浮点数。

ID正则表达式可以识别以字母或下划线开头的字符串，表示标识符。

为了识别错误的二进制、八进制、十进制和十六进制整数以及浮点小数、标识符，还定义了识别以上类型错误的正则表达式，如错误的八进制数09，十六进制数0x1G，指数形式的浮点数1.05e等。

```

BIN 0[bB][01]+
WRONG_BIN 0[bB][01]*[2-9a-zA-Z]+[0-9]*|[0-9][bB]
OCT 0[0-7]+
WRONG_OCT 0[0-7]*[8-9]+[0-9]*
DEC 0[1-9][0-9]*
WRONG_DEC [0-9]+[AaC-Wc-wY-Zy-z]+[0-9A-Da-dF-Zf-z]*
HEX 0[xX][0-9a-fA-F]+
WRONG_HEX 0[xX][0-9a-fA-F]*[g-zA-Z]+[0-9a-fA-F]*|0[xX]

FLOAT ([0-9]+\.[0-9]*\.[0-9]+)([eE][+-]?[0-9]+)?
WRONG_FLOAT ([0-9]+\.[0-9]*\.[0-9]+)[eE][+-]?([0-9]+\.[0-9]*\.[0-9]+)|([0-9]+\.[0-9]*\.[0-9]+)[eE][+-]?

ID [A-Za-z_][0-9A-Za-z_z]*

```

还能够识别“//”和“/**/”形式的注释。若输入文件中包含符合定义注释，程序能够滤除这样的注释；若输入文件中包含不符合定义的注释，包括“/**/”嵌套，程序可以给出错误的提示信息。

(2) 语法错误（错误类型B）

使用Bison构建语法树，`syntax.y`中包含了具体的语法和相应的语义动作。为了能够构建和表示语法树，终结符或非终结符属性值的类型为应为多叉树类型，语义动作中要将产生式右边的每一个符号所对应的树结点作为产生式左边的非终结符所对应的树结点的子结点逐个进行插入。

在`syntaxTree.h`中定义了树节点结构，包含节点名、节点类型、子节点数量、子节点数组、错误行号，以及一个联合体类型，根据符号的不同来访问联合体中不同的域以实现多种类型的效果。声明了节点初始化与节点插入函数。在节点结构中含有一个子节点指针数组，构造语法树时，首先初始化该节点，再根据语法规则，将其子节点依次放入数组，最后将子节点添加到该节点上。

书写了一系列包含error的产生式，用以检查输入文件中存在的一些语法错误。尽量消除了二义性与冲突，因此还有一些可能的错误恢复没有完成。

2. 输出运行结果

通过标准输出打印程序的运行结果。对于那些包含词法或者语法错误的输入文件，输出相关的词法或语法有误的信息，包括错误类型、出错的行号以及说明文字。对于没有错误的文件，在`syntaxTree.h`中声明了辅助输出语法树的函数`tree_display`，本质上是一个先序遍历语法树的尾递归函数。程序有一个全局变量用于统计错误的数量，在`main`函数中，如果文件没有错误，则调用函数输出语法树。否则会在词法、语法分析过程中输出错误信息。

我重写yyerror函数后发现输出时仍然会执行默认函数，输出"syntax error"后才会执行我写的输出语句，无法解决这一问题，只好将yyerror函数改为不进行任何输出，然后新定义了函数print_error来输出要求的格式。

编译

可以直接使用Makefile，执行make命令即可生成parser程序。

示例

```
Program (1)
  ExtDefList (1)
    ExtDef (1)
      Specifier (1)
        TYPE: int
      FunDec (1)
        ID: main
        LP
        RP
      CompSt (2)
        LC
        DefList (3)
          Def (3)
            Specifier (3)
              TYPE: int
            Declist (3)
              Dec (3)
                VarDec (3)
                  ID: i
                  ASSIGNOP
                  Exp (3)
                    INT: 83
            SEMI
          DefList (4)
            Def (4)
              Specifier (4)
                TYPE: int
              Declist (4)
                Dec (4)
                  VarDec (4)
                    ID: j
                    ASSIGNOP
                    Exp (4)
                      INT: 63
            SEMI
          RC
  Error type B at Line 4: Missing "]"
  Error type B at Line 5: Missing ";"
```