# **《编译技术》实验报告**

实验名称：语法分析程序

姓名：牟洪锋

学号：U201417220

班级：软件工程1404班

指导教师：胡福林

2016年12月17日

1. 实验题目

对下述单词表与语法定义的语言设计编制一个语法分析器。单词符号及种别表、语法及语法分析器功能、基本要求如下：

（1）单词符号及种别表

|  |  |  |
| --- | --- | --- |
| 单词符号 | 种别编码 | 单词值 |
| main | 1 |  |
| int | 2 |  |
| float | 3 |  |
| double | 4 |  |
| char | 5 |  |
| if | 6 |  |
| else | 7 |  |
| do | 8 |  |
| while | 9 |  |
| l(l|d)\* | 10 | 内部字符串 |
| ( +|-|ε ) dd\*(.dd\* | ε)( e ( +|-|ε ) dd\*|ε) | 20 | 二进制数值表示 |
| = | 21 |  |
| + | 22 |  |
| - | 23 |  |
| \* | 24 |  |
| / | 25 |  |
| ( | 26 |  |
| ) | 27 |  |
| { | 28 |  |
| } | 29 |  |
| , | 30 |  |
| ; | 31 |  |
| > | 32 |  |
| >= | 33 |  |
| < | 34 |  |
| <= | 35 |  |
| == | 36 |  |
| != | 37 |  |
| # | 0 |  |

（2）语法结构定义

<表达式> ::= <项>{ +<项>|-<项>}

<项> ::= <因子>{\*<因子>|/<因子>}

<因子> ::=ID|num|(<表达式>)

num::= ( +|-|ε ) 数字数字\*(.数字数字\* | ε)( e ( +|-|ε ) 数字数字\*|ε)

ID::=字母(字母|数字)\*

字母::=a|b|c…|z|A|B|C…|Z

数字::=0|1|2…|9

1. 实验目的

通过该实验,熟练应用编译原理关于语法分析的基本理论和方法；学会用C/C++高级程序设计语言设计一个语法分析器；加深对编译原理理论的分析理解，提高实际操作和解决具体问题的能力。

1. 实验要求

词法分析程序的基础上完成语法分析的功能：

处理用户给出的符合文法的语句并判断其语法是否正确。

例如：用户输入 a+23.7e+15 程序对其进行判断

输出：success

用户输入 (123.456+-456.789e-120)\*m2+(a++456)\*c123

输出：success

用户输入 (123.456+-456.789e-120)\*m2\*-c123

输出：error

1. 实验步骤

基本设计思路

1. 首先将基本字作为特殊符号处理，识别基本字，查表，给出其种别码。基本字的初始值定为：

**char\*rwtab[9]={"main","int","float","double","char","if","else","do","while"}; （字符指针数组）**

1. 识别无符号整数是将数字串转换为无符号整数。我们在getchar()的时候是把数字当做字符从外部输出读取的。将数字串789#转换为整数：(7\*10+8)\*10+9=789送到sum中
2. 程序由两个函数组成：**main()**和**scanner()**。每次读取一个符号并进行判断，主程序控制输出。
3. 本程序采用递归下降法进行语法判断：对文法的每个非终结符编制一个递归过程（函数），按规则右部符号串的顺序编写：若为终结符，则读下一个单词符号；非终结符，调用相应的递归过程（函数）。

正规式

将正规文法转换成如下表达式

E -> T{ +T|-T}

T -> F{\*F|/F}

F -> ID|num|(E)

状态转换图

流程框图

E函数

出错处理

+,-?

调用T函数

调用T函数

调用scanner函数

否

是

T函数

出错处理

\*,/?

调用F函数

调用F函数

调用scanner函数

否

是

F函数

是

是否标识符

否

否

否

出错处理

是

调用scanner函数

是

是

调用scanner函数

是否)

调用E函数

调用scanner函数

是否(

否

是否数字

算法设计

语法分析程序所用的较为重要的全局变量和需调用的函数如下：

1. ch字符变量，存放当前读进的源程序**字符**。
2. token[8]字符数组，存放构成单词符号的**字符串**。
3. prog[80]字符数组，存放**所有**用户输入的字符。
4. syn整数，存放当前单词的种别码。
5. sum双精度浮点型变量，存放无符号整数，或者浮点数。
6. isDecimal整数，是否为浮点数。isDecimal为1，则为浮点数。
7. decimal双精度浮点型变量(double)，浮点数的小数部分。
8. isExp整数，是否为指数形式表示的浮点数（即是否存在符号E或者e）。isExp为1， 则为指数形式。
9. index整数，指数形式的幂。
10. isNegative整数，是否为负数幂。isNegative为1，则为负数幂，如123E-2。
11. scanner()扫描子程序。
12. getchar()从控制台读取一个字符数据。
13. double pow(double x,double y)，计算x的y次幂。
14. int strcmp(char \*str1,char #str2)，字符串比较。
15. int isSignal; 是否带正负号(0不带，1负号，2正号）
16. int repeat; 是否连续出现+,-。
17. void E(); 递归下降分析程序中，为非终结符E编写的函数
18. void T(); 递归下降分析程序中，为非终结符T编写的函数
19. void F(); 递归下降分析程序中，为非终结符F编写的函数

函数相关说明

void E()

{

T();

while((syn==22)||(syn==23)) //’+’,’-‘

{

scanner();

T();

}

}

void T()

{

F();

while((syn==24)||(syn==25)) //’\*’,’/’

{

scanner();

F();

}

}

void F()

{

if((syn==20)||(syn==10)) //字符串，数字

scanner();

else if(syn==26) //’(‘

{

scanner();

E();

if(syn==27) //’)’

{

scanner();

}

else isError=1;

}

else isError=1;

}

输入与输出

语法分析程序的输入输出情况

输入：判断文法的一段字符串（以#为结束符）

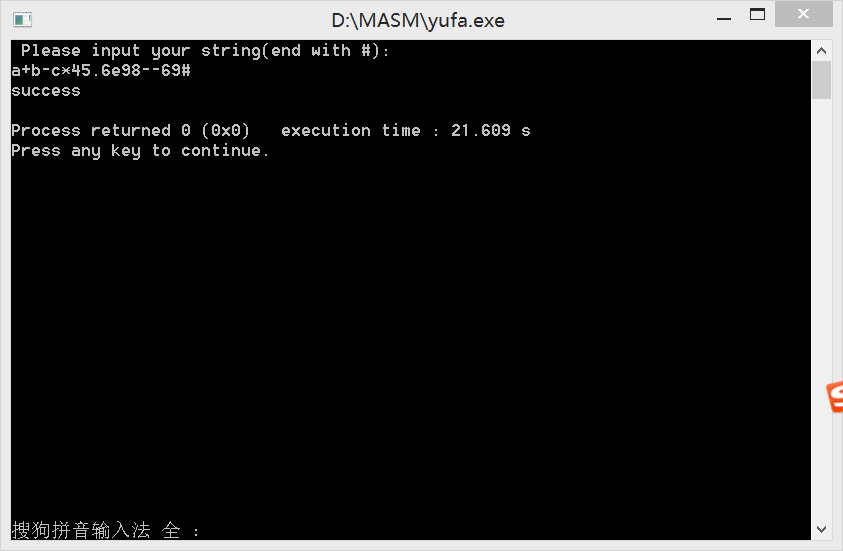
输出：若文法正确则输出 success

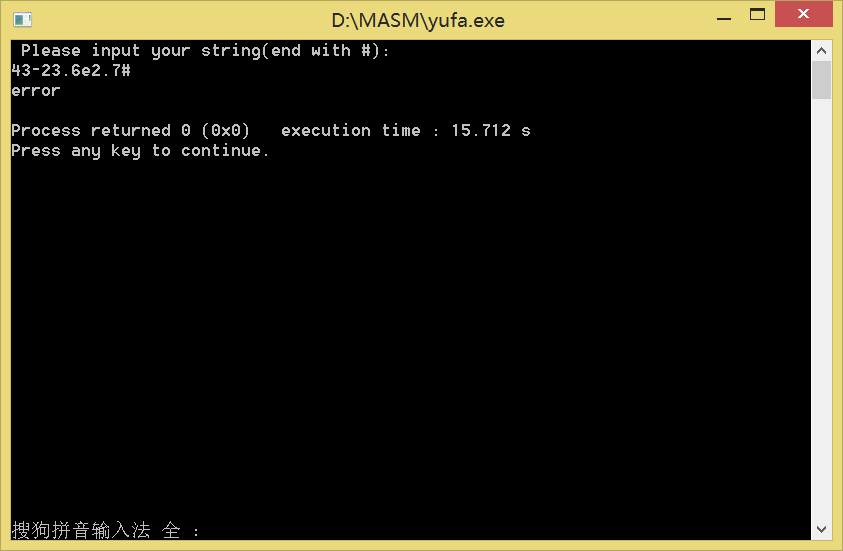
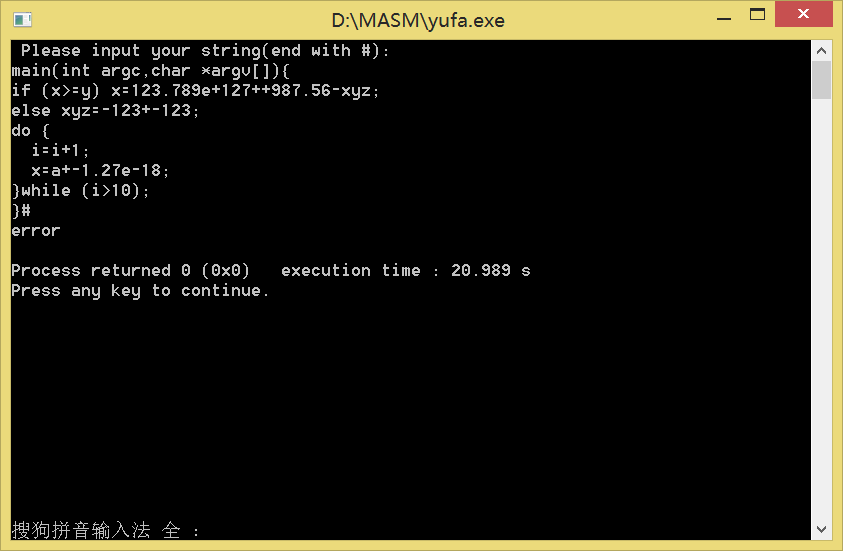
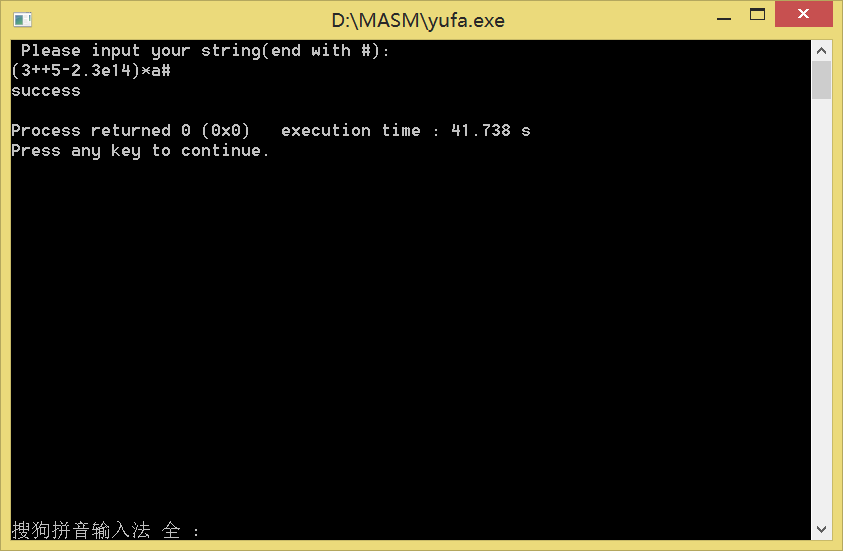
若文法错误输出error

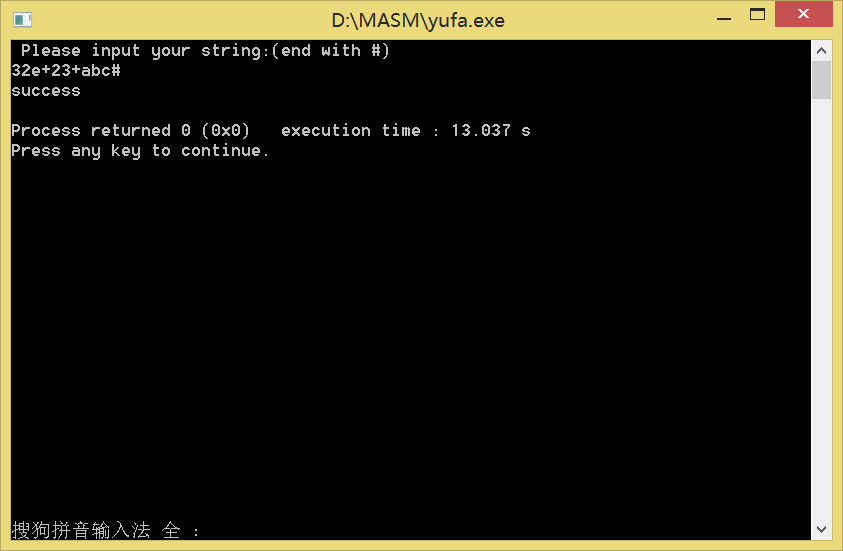
例如：c\*a+b\*12.e45#

输出 success

程序运行结果







1. 实验方案设计实现

代码由c语言编写，语法分析由递归下降法实现

1. 程序亮点描述

借鉴了词法分析程序的经验并进行了扩充，语法分析运用递归下降法解决问题，代码比词法分析程序简洁，变量名更清晰。

1. 实验程序使用说明

用户输入待识别字符串（并以“#”结尾，表示字符串输入结束），回车后程序自动输出语法分析结果。

1. 实验心得体会