**编译原理实验报告**

实验名称：\_\_\_\_语法分析器\_\_\_\_\_\_\_\_\_\_\_\_\_\_

指导教师：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

专业班级：\_\_ \_\_计131\_ \_\_\_\_\_\_\_

姓 名：\_\_\_ \_王振\_ \_\_\_\_\_\_\_\_\_

学 号：\_\_\_\_\_\_\_\_ 201358505136\_\_\_\_\_\_\_\_

电子邮件：\_\_\_\_\_\_4013465w@sina.com\_\_\_\_\_

实验地点：\_\_\_\_\_计算机中心\_\_\_\_\_\_\_\_\_\_\_

实验成绩：\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

日期： 2015 年 11 月 日

**一、实验题目**

对下述文法和单词表定义的语言设计编制一个语法分析器。

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 |  |

1. 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

**二、实验目的**

1、通过该实验,熟练应用编译原理的基本理论和方法

2、学会用C/C++高级程序设计语言设计一个语法分析器的技术

3、加深对编译原理的分析理论的理解，培养动手实践能力

**三、实验要求**

词法分析程序需具备语法分析的功能：

处理用户提交的符合上述文法的源代码序列，进行语法分析，并给出语法是否正确的结论。

例如：控制台输入d+-11.7e-17#，经过语法分析后输出如下序列：

success!

例如：控制台输入(123.456+-456.789e-120)\*m2+(a++456)\*c123#，经过语法分析后输出如下序列：

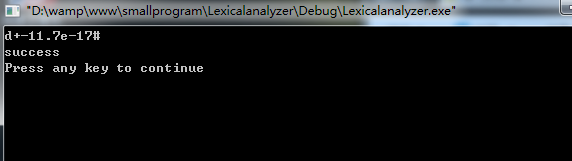
success!

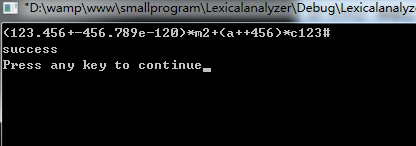
例如：控制台输入(123.456+-456.789e-120)\*m2+(a++456)\*-c123#，经过语法分析后输出如下序列：

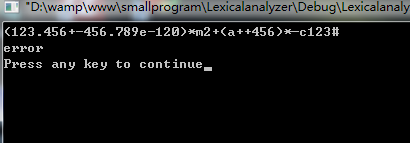
error!

**四、实验步骤**

（包括基本设计思路、流程框图、算法设计、函数相关说明、输入与输出以及程序运行结果）







**五、实验心得体会**

**六、源程序清单（代码）**

#include<iostream>

#include<string>

using namespace std;

string data[100];

int c[100],indexd=0,ans[38][38],indexc=0,j=0,isError=0;

int i=0;int index\_ck=0,syn;

string demo[38];

bool is\_letter(char s){//判断是否为字母

if(s>='a'&&s<='z'||s>='A'&&s<='Z'){

return 1;

}else{

return 0;

}

}

bool is\_num(char s){//判断手否为数字

if(s>='0'&&s<='9'){

return 1;

}else{

return 0;

}

}

bool is\_bit2(){//是否为二进制数

bool result=0;int m=0;

if(data[indexd][j]=='+'||data[indexd][j]=='-'){

m++;

}

while(is\_num(data[indexd][j+m])){

m++;result=1;

}

if(data[indexd][j+m]=='.'&&result){

m++;

while(is\_num(data[indexd][j+m])){

m++;

}

}

if(data[indexd][j+m]=='e'){

m++;

if(data[indexd][j]=='+'||data[indexd][j]=='-'){

m++;

}

while(is\_num(data[indexd][j+m])){

m++;

}

}

if(result){

j+=m-1;

c[indexc]=20;

indexc++;

}

return result;

}

bool two\_str(int s){//计算>=等字符

bool result=0;

if(demo[s][0]==data[indexd][j]&&j+2<data[indexd].size()&&demo[s][1]==data[indexd][j+1]){

result=1;

}

if(result){

j+=2;

c[indexc]=s;

indexc++;

}

return result;

}

bool other\_str(){//判断其他变量名

bool result=0;

while(j<data[indexd].size()&&(is\_num(data[indexd][j])||is\_letter(data[indexd][j]))){

j++;

result=1;

}

if(result){

c[indexc]=10;

indexc++;

j--;

}

return result;

}

bool cmpstr(int s){//判断是否为第s个关键词

bool result=1;

for(int m=0;m<demo[s].size()&&j+m<data[indexd].size();m++){

if(demo[s][m]!=data[indexd][j+m]){

result=0;

break;

}

}

if(result &&!is\_letter(data[indexd][j+m+1])&&!is\_num(data[indexd][j+m+1])){

j+=m;

c[indexc]=s;

indexc++;

}else{

result=0;

}

return result;

}

void compare(){//路由

bool result;

switch(data[indexd][j]){

case 'm':

result=cmpstr(1);

if(result){

break;

}else if(other\_str()){

break;

}

case 'i':

result=cmpstr(2);

if(result){

break;

}

result=cmpstr(6);

if(result){

break;

}if(other\_str()){

break;

}

case 'f':

result=cmpstr(3);

if(result){

break;

}if(other\_str()){

break;

}

case 'd':

result=cmpstr(4);

if(result){

break;

}

result=cmpstr(8);

if(result){

break;

}if(other\_str()){

break;

}

case 'c':

result=cmpstr(5);

if(result){

break;

}

if(other\_str()){

break;

}

case 'e':

result=cmpstr(7);

if(result){

break;

}if(other\_str()){

break;

}

case 'w':

result=cmpstr(9);

if(result){

break;

}if(other\_str()){

break;

}

case '='://==

result = two\_str(36);

if(result){

break;

}

c[indexc]=21;

indexc++;

break;

case '+'://单独处理

if(indexc>0&&(c[indexc-1]==22||c[indexc-1]==23||c[indexc-1]==24||c[indexc-1]==225)){

if(is\_bit2()){

break;

}

}else{

//result=two\_str();

c[indexc]=22;

indexc++;

break;

}

case '-'://单独处理

if(indexc>0&&(c[indexc-1]==22||c[indexc-1]==23||c[indexc-1]==24||c[indexc-1]==225)){

if(is\_bit2()){

break;

}

}else{

c[indexc]=23;

indexc++;

break;

}

case '\*':

c[indexc]=24;

indexc++;

break;

case '/':

c[indexc]=25;

indexc++;

break;

case '(':

c[indexc]=26;

indexc++;

break;

case ')':

c[indexc]=27;

indexc++;

break;

case '{':

c[indexc]=28;

indexc++;

break;

case '}':

c[indexc]=29;

indexc++;

break;

case ',':

c[indexc]=30;

indexc++;

break;

case ';':

c[indexc]=31;

indexc++;

break;

case '>'://///>=

result = two\_str(33);

if(result){

break;

}

c[indexc]=32;

indexc++;

break;

case '<':

result = two\_str(35);//<=

if(result){

break;

}

c[indexc]=34;

indexc++;

break;

case '!'://///!=

result = two\_str(37);

if(result){

break;

}

case '\n':

break;

default:

if(is\_bit2()){

break;

}

if(other\_str()){

break;

}else{

c[indexc]=-34;

indexc++;

}

}

}

/\*\*

语法分析

\*\*/

void E(); void T(); void F();

void E(){

T();

while((syn==22)||(syn==23))

{

index\_ck++;

syn=c[index\_ck];

T();

}

}

void T(){

F();

while((syn==24)||(syn==25))

{

index\_ck++;

syn=c[index\_ck];

F();

}

}

void F(){

if((syn==20)||(syn==10)){

index\_ck++;

syn=c[index\_ck];}

else if(syn==26)

{

index\_ck++;

syn=c[index\_ck];

E();

if(syn==27)

{

index\_ck++;

syn=c[index\_ck];

}

else

isError=1;

}

else

isError=1;

}

int main(){

demo[0]="#",

demo[1]="main",

demo[2]="int",

demo[3]="float",

demo[4]="double",

demo[5]="char",

demo[6]="if",

demo[7]="else",

demo[8]="do",

demo[9]="while",

demo[10]="l(l|d)\*",

demo[20]="( +|-|ε ) dd\*(.dd\* | ε)( e ( +|-|ε ) dd\*|ε)",

demo[21]="=",

demo[22]="+",

demo[23]="- ",

demo[24]="\*",

demo[25]="/",

demo[26]="(",

demo[27]=")",

demo[28]="{",

demo[29]="}",

demo[30]=",",

demo[31]=";",

demo[32]=">",

demo[33]=">=",

demo[34]="<",

demo[35]="<=",

demo[36]="==",

demo[37]="!=";

i=0;

while(cin>>data[i]&&data[i][data[i].size()-1]!='#'){

// cout<<Data[i][Data[i].length()-1]<<endl;

i++;

}

int n=i;

for(indexd=0;indexd<=n;indexd++){

for (j=0;j<data[indexd].size();j++){

//cout<<Data[i][j]<<endl;

// cout<<data[indexd][j]<<" "<<c[indexc-1]<<endl;

compare();

}

}

for(i=0;i<38;i++){

for(int q=0;q<38;q++){

ans[i][q]=0;

}

}

/\* for(i=0;i<=indexc;i++){

cout<<c[i]<<endl;

}\*/

syn=c[index\_ck];

if((syn==20)||(syn==10)||(syn==26))

{

E();

}

else{

isError=1;

}

if(isError==0)

printf("success\n");

else

printf("error\n");

return 0;

}