**实验报告:词法分析**

**一、实验目的**

通过设计、开发一个高级语言的词法分析程序，加深对课堂教 学内容（包括正规文法、正规表达式、有限自动机、NFA到DFA的 转换、DFA的最小化）的理解，提高词法分析方法的实践能力。

**二、实验要求**

(1) 深入理解、掌握有限自动机及其应用；

(2) 掌握根据语言的词法规则构造识别其单词的有限自动机的方 法；

(3) 掌握NFA到DFA的等价变换方法、DFA最小化的方法；

(4) 掌握设计、编码、调试词法分析程序的技术与方法，具体实现 S语言的词法分析程序。

**三、实验原理**

词法分析是编译过程的第一个阶段。它的任务是对输入的字符 串形式的源程序按顺序进行扫描，根据源程序的词法规则识别具 有独立意义的单词，并输出单词的序列。有限自动机是描述程序设计语言单词结构的工具，而状态转换 图是有限自动机的比较直观的描述方法。根据程序设计语言的词 法规则构造描述该语言单词结构的有限自动机，获取识别各类单 词的形式模型，进而通过编程模拟该形式模型的运行，可实现词法分析程序。

**四、实验步骤**

(1)根据S语言的词法规则，总结S语言的单词种类与各类单词的结 构特征;

(2)设计描述S语言各类单词结构的状态转换图(即有限自动机 FA);

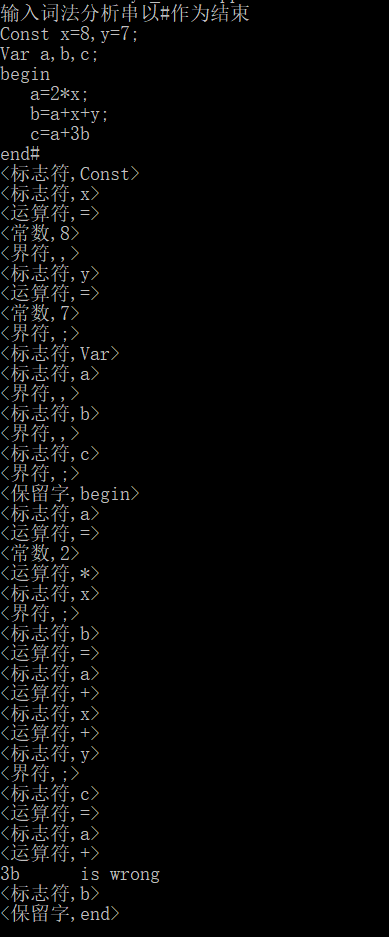
(3)对描述各类单词结构的状态转换图进行合并(将各状态转换图 的初始状态合并为一个唯一的初态，然后对状态重新编号)， 构成一个能识别S语言所有单词的状态转换图(NFA)。

(4)对能识别所有单词的NFA进行确定化操作，将其转换成等价的 DFA;

(5)对DFA进行最小化操作;

(6)编写程序，模拟最小化DFA的运行，实现S语言的词法分析程序;

**五、结果展示**

  
可以看出可以识别出错误的标识符 3b

**六、实验程序**

**//**sy2\_test.c

#include<stdio.h>

#include<string.h>

#include<stdbool.h>

void Print();

void Scanner();

int syn;//存放单词的类型

char \*wordtype[5] = {"保留字","运算符","标志符","常数","界符"};

int p;

char ch;

int sum;//用来保存数字的值

char program[1000],token[15];

char \*rwtab[13]={"begin","if","then","procedure","odd","call","else","while","do","const","var","end","do"};

int m;

bool isDigital(char ch)

{

if(ch<='9'&&ch>='0')

return true;+

else

return false;

}

bool isAlpha(char ch)

{

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

return true;

else

return false;

}

void Scanner()

{

int num=0;

int n;

for(m=0;m<15;m++)

{

token[m]='\0';

}

m=0;

ch=program[p++];

while(ch==' '||ch=='\n'||ch=='\t')

{

ch=program[p++];

}

if(isAlpha(ch)){

do{

token[m++]=ch;

ch=program[p++];

}while(isAlpha(ch)||isDigital(ch));

p--;

syn=2;

token[m++]='\0';

for(n=0;n<13;n++)

{

if(strcmp(token,rwtab[n])==0)

{

syn=0;

break;

}

}

return;

}

else if(isDigital(ch))

{

sum=0;

while(isDigital(ch))

{

sum=sum\*10+ch-'0';

ch=program[p++];

}

p--;

syn=3;

if(isAlpha(ch))

syn=-1;

return;

}

else

{

token[0]=ch;

switch(ch)

{

case '<':ch=program[p++];

if(ch=='>')

{

syn=1;

token[1]=ch;

}

else if(ch=='=')

{

syn=1;

token[1]=ch;

}

else{

syn=1;

p--;

}

break;

case '>':ch=program[p++];

if(ch=='=')

{

syn=1;

token[1]=ch;

}

else{

syn=1;

p--;

}

break;

case '=':ch=program[p++];

if(ch=='=')

{

syn=1;

token[1]=ch;

}

else{

syn=1;

p--;

}

break;

case '+':syn=1;break;

case '-':syn=1;break;

case '\*':syn=1;break;

case '/':syn=1;break;

case ';':syn=4;break;

case '(':syn=4;break;

case ')':syn=4;break;

case ',':syn=4;break;

case '#':syn=-2;break;

case ':':ch=program[p++];

if(ch=='=')

{

syn=1;

token[1]=ch;

}

else{

syn=4;

p--;

};break;

default:syn=-1;break;

}

return;

}

}

void Print()

{

if(syn==3)

printf("<%s,%d>\n",wordtype[syn],sum);

else if(syn==-2) {}

else

printf("<%s,%s>\n",wordtype[syn],token);

}

int main()

{

printf("输入词法分析串以#作为结束\n");

do

{

ch=getchar();

program[p++]=ch;

}while(ch!='#');

p=0;

int i=0;

//printf("%s",wordtype[0]);

//printf("%s\n",program);

do{

Scanner();

switch(syn)

{

case -1:for(i=p-1;i<=p;i++) printf("%c",program[i]);printf(" "); printf("is wrong\n",token);break;

default:Print();break;

}

}while(syn!=-2);

if(syn==0)printf("success");

}