# 实验 2：词法分析实验

### 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（见附录A）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；

(7)撰写实验报告。

### 实现代码

#include<stdlib.h>

#include<iostream>

#include<fstream>

#include <stdio.h>

#include <string.h>

#define MAX\_mid 100

using namespace std;

//######################词法分析部分###############################

char word[1000];

char scan[100];

int line=1;

int t=0,i=0,sum=0;

char c;

bool ci\_flag = true;

void scanner()

{

for(i=0;i<100;i++)

scan[i]=NULL;

i=0;

c=word[t++];

while(c==' ')

{

c=word[t];

t++;

}

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

{

while(c>='a' && c<='z' || c>='A' && c<='Z' || c>='0' && c<='9')

{

scan[i++]=c;

c=word[t++];

}

scan[i++]='\0';

if(!strcmp(scan,"Const") || !strcmp(scan,"Var") || !strcmp(scan,"if") || !strcmp(scan,"then") || !strcmp(scan,"else") || !strcmp(scan,"while") || !strcmp(scan,"do") || !strcmp(scan,"begin") || !strcmp(scan,"end"))

printf("%d: Reserved words : %s\n",line,scan);

else

printf("%d: Identifier : %s\n",line,scan);

t--;

i=0;

}

else if(c>='0' && c<='9')

{

while(c>='0' && c<='9')

{

scan[i++]=c;

c=word[t++];

}

char temp = c;

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

{

t--;

c = word[t];

printf("Identifier error: %d \n",line );

ci\_flag = false;

while((c>='a' && c<='z')|| (c >= 'A' && c <= 'Z') || (c >= '0' && c<= '9' ))

{

c = word[t++];

}

//exit(0);

}

else

printf("%d: Number : %s\n",line,scan);

t--;

i=0;

}

else if(c=='\n')

line++;

else

{

switch(c)

{

case '>':

i=0;

scan[i++]=c;

c=word[t++];

t--;

if(c=='=')

{

scan[i++]=c;

scan[i]='\0';

i=0;

printf("%d: %s\n",line,scan);

break;

}

else

{

i=0;

printf("%d: %s\n",line,scan);

break;

}

case '<':

i=0;

scan[i++]=c;

c=word[t++];

t--;

if(c=='=')

{

scan[i++]=c;

scan[i]='\0';

i=0;

printf("%d: %s\n",line,scan);

break;

}

else if(c=='>')

{

scan[i++]=c;

scan[i]='\0';

i=0;

printf("%d: %s\n",line,scan);

break;

}

else

{

i=0;

printf("%d: %s\n",line,scan);

break;

}

case '=':

i=0;

scan[i++]=c;

c=word[t++];

t--;

if(c=='=')

{

scan[i++]=c;

scan[i]='\0';

i=0;

printf("%d: %s\n",line,scan);

break;

}

else

{

i=0;

printf("%d: %s\n",line,scan);

break;

}

case ',':scan[0]=c;printf("%d: %s\n",line,scan);break;

case ';':scan[0]=c;printf("%d: %s\n",line,scan);break;

case '(':scan[0]=c;printf("%d: %s\n",line,scan);break;

case ')':scan[0]=c;printf("%d: %s\n",line,scan);break;

case '+':scan[0]=c;printf("%d: %s\n",line,scan);break;

case '-':scan[0]=c;printf("%d: %s\n",line,scan);break;

case '\*':scan[0]=c;printf("%d: %s\n",line,scan);break;

case '/':scan[0]=c;printf("%d: %s\n",line,scan);break;

default:printf("Error!");ci\_flag = false;

}

}

}

int main(){

char in[MAX\_mid];

FILE \*fin;

cout<<"请输入程序文件名：";

cin>>in;

cout<<endl;

if ((fin=fopen(in,"r"))==NULL)

{

cout<<endl<<"打开词法分析输入文件出错!"<<endl;

}

char CiFaChar;

int ci\_i = 0;

while((CiFaChar = fgetc(fin))!=EOF)

word[ci\_i++]=CiFaChar;

while(t!=ci\_i)// t一开始为0 直到加到和ci\_i一样长（读到的词一样长）

scanner();

if(ci\_flag == false)

{

cout <<"词法分析有错，请更正后，在进行下一步分析"<<endl;

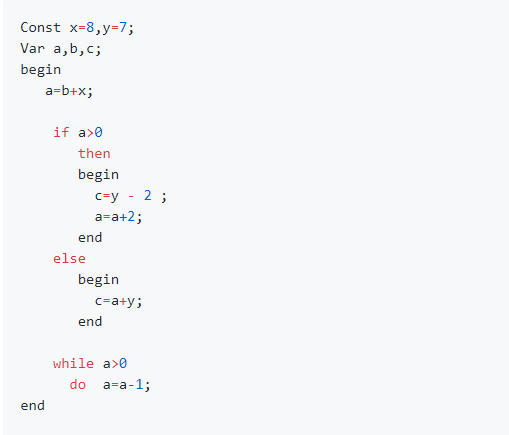
exit(0);

}

return 0;

}

### 测试代码



### 结果展示

