南京信息工程大学 编译原理 实验（实习）报告

实验（实习）名称 词法分析程序的设计与实现 日期 2016.04.23 得分 指导教师 闫雷鸣
系 计软院 专业 软件工程 年级 2013 班次 1 姓名 张少华 学号 20131344022

一、实验目的

1. 学会针对 DFA 转换图实现相应的高级语言源程序。
2. 深刻领会状态转换图的含义，逐步理解有限自动机。
3. 掌握手工生成词法分析器的方法，了解词法分析器的内部工作原理。

二、实验内容

计算机语言的编译程序的词法分析部分实现。从左到右扫描每行该语言源程序的符号，拼成单词，换成统一的内部表示（token）送给语法分析程序。为了简化程序的编写，有具体的要求如下：

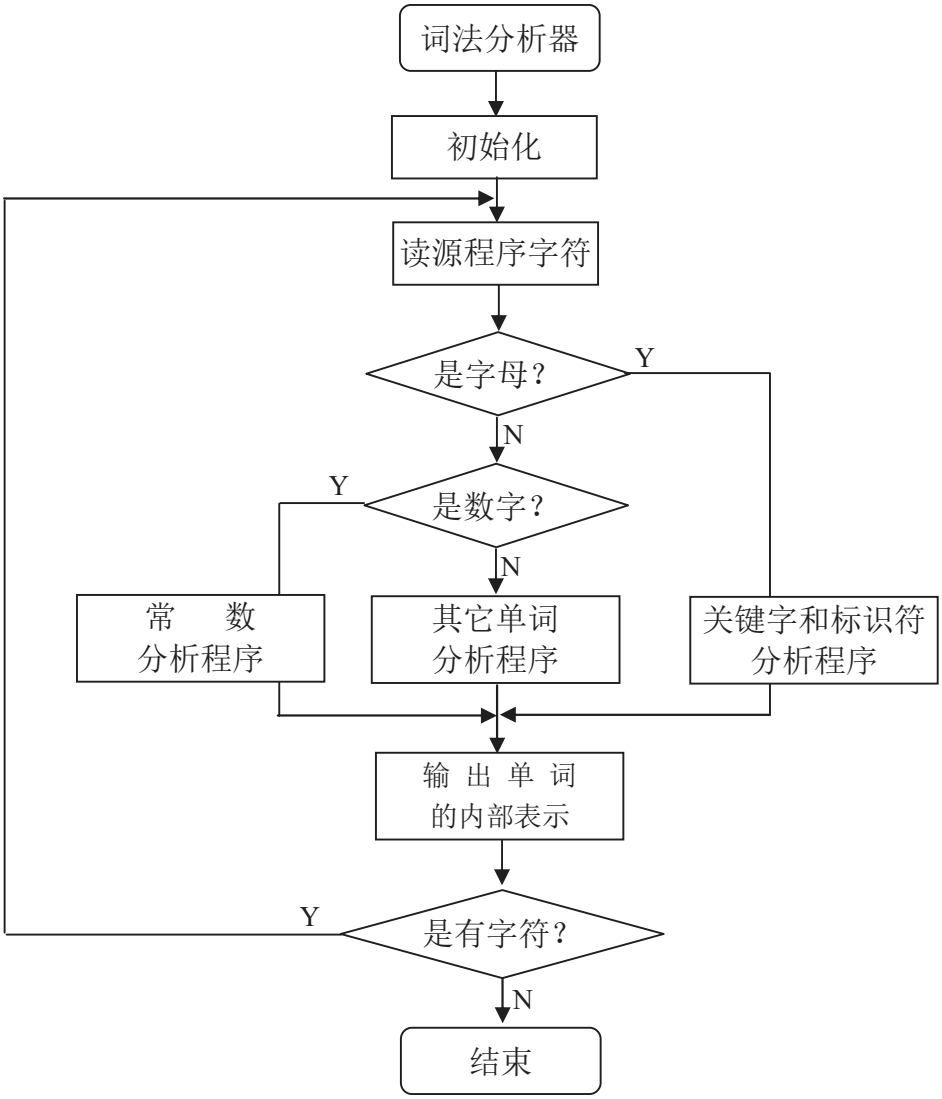
1) 整数或浮点数。
2) 空白符仅仅是空格、回车符\n、制表符\t。
3) 代码是自由格式。
4) 注释应放在花括号或者/* */或者 // 之内，并且不允许嵌套

要求实现编译器的以下功能：

1) 按规则拼单词,并转换成二元式形式
2) 删除注释行
3) 删除空白符 (空格、回车符、制表符)，即中间表示形式中不含空白符。
4) 列表打印源程序，按照源程序的行打印，在每行的前面加上行号，并且打印出每行包含的记号的二元形式
5) 能发现并定位词法错误

五、实验步骤

词法分析过程：



字符表：

符号	编号	符号	编号	符号	编号
main	1	if	2	then	3
while	4	do	5	static	6
int	7	double	8	struct	9
break	10	else	11	long	12
switch	13	case	14	typedef	15
char	16	return	17	const	18
float	19	short	20	continue	21
for	22	void	23	sizeof	24
Include	25	define	26	+	51
-	52	*	53	/	54
=	55	;	56	(	57
)	58	#	59	{	60
}	61	<	62	>	63
ID	100	NUM	101	<=	103

<<	104	>=	105	==	106
++	107	+=	108	--	109
-=	110				

词法分析代码实现：

```
#include<iostream>
#include<stdio.h>
#include<string.h>
#include<stdlib.h>
using namespace std;
const int totKeyWords = 50;//定义关键字的数量
const int totSymbols = 50;//定义定界符的数量
#define ID 100//定义标识符
#define NUM 101//定义常数
//存放处理后的字符串
char tempstr[255] = {};
//空格标志
bool temp = false;
//临时数组
char word[255] = {};
//keyword 关键字
string keyword[totKeyWords] = { "main", "if", "then", "while", "do", "static", "defualt",
"do", "int", "double", "struct", "break", "else", "long", "swtich", "case", "typedf",
"char", "return", "const", "float", "short", "continue", "for", "void", "sizeof", "include",
"define"};
//部分运算符，定界符等
char symbol[totSymbols] = { '+', '-', '*', '/', '=', ';', '(', ')', '#','{','}','<','>' };
//判断是否为字母
bool IsLetter(char ch) {
    if ((ch >= 'a'&&ch <= 'z') || (ch >= 'A'&&ch <= 'Z'))
        return true;
    return false;
}
//判断是否为数字
bool IsDigit(char ch) {
    if (ch >= '0'&&ch <= '9')
        return true;
    return false;
}
//判断是否为定界符等
int IsSymbol(char ch) {
    for (int i = 0; i < totSymbols; i++) {
        if (ch == symbol[i])
            return i;
    }
    return -1;
}
//判断是否为关键字
int IsKeyword(string str) {
    for (int i = 0; i < totKeyWords; i++) {
        if (str == keyword[i]) {
            return i;
```

```
            }
        }
        //在关键字表中没有找到
        return -1;
}
//空格处理
void HandleSpace(char a[]) {
        int j = 0;
        memset(word, 0, 255);//需要清空，不然可能残留上次的字符串
        temp = false;
        for (int i = 0; i < strlen(a); i++) {
                if (a[i] != ' ' && a[i] != '\t' ) {
                        word[j++] = a[i];
                        temp = false;
                } else {
                        if (!temp&&a[i] != '\t') {
                                word[j++] = a[i];
                                temp = true;
                        }
                }
        }
}
//处理"//"注释
void prePro() {
        int j = 0;
        memset(tempstr, 0, 255);
        for (int i = 0; i < strlen(word); i++) {
                if (word[i] == '/'&&word[i + 1] == '/') {
                        while (i < strlen(word)) {
                                i++;
                        }
                } else {
                        tempstr[j++] = word[i];
                }
        }
}
int main() {
        char instr[255] = {}; //接收输入字符串
        bool isComment = false; //多行注释标志,false 为未处于注释区域
        string token;//存放字符串
        char *str = NULL;//存放每行的字符串
        freopen("test.cpp", "r", stdin);
        // freopen("result.txt", "w", stdout);//输出到文件
        int linenumer = 1;
        while (fgets(instr,100,stdin) != NULL) {
                cout << "line:" << linenumer++;
                HandleSpace(instr);
                prePro();
                str = tempstr;
                while (str != NULL) {
                        //一行一行开始词法分析
                        for (int i = 0; i < strlen(str); i++) {
                                        //注释处理开始
                                if (*(str + i) == '/') {
                                        if (*(str + i + 1) == '*') {
                                                isComment = true;
```

```
        }
    }
    //注释处理: */,注释区域结束
    if (*(str + i) == '*'&&isComment) {
        if (*(str + i + 1) == '/') {
            isComment = false;
            i+=2;
        }
    }
    //标识符，关键词
    if (IsLetter(*(str + i)) && (!isComment)) {
        while (IsLetter(*(str + i)) || IsDigit(*(str + i)) || *(str + i) == '_') {
            token += *(str + i);
            i++;
        }
        if (IsKeyword(token) != -1) {
            printf("(%d,%s) ", IsKeyword(token), token.c_str());
        } else {
            // if(judge(token))//判断是否为合法变量
                printf("(%d,%s) ", ID, token.c_str());
        }
        token = "";
    }
    if (IsDigit(*(str + i)) && (!isComment)) {
        bool mistake = false;
        while (IsDigit(*(str + i)) || IsLetter(*(str + i))) {
            if(IsLetter(*(str + i))){//找出词法分析的错误定位
                mistake = true;
            }
            token += *(str + i);
            i++;
        }
        if(!mistake)
            printf("(%d,%s) ",NUM, token.c_str());
        else
            printf("(错误 ： %s 不能作为变量名)", token.c_str());
        token = "";
    }
    if (*(str + i) == '<' && (!isComment)) {
        //<,<=,!=
        if (*(str + i) == '=') {
            printf("(103,<=) ");//<=
            i++;
        }
        if (*(str + i) == '<') {//<<
            printf("(104,<<) ");
            i++;
        } else {
            printf("(%d,<) ", IsSymbol('<'));
        }
    } else if (*(str + i) == '>' && (!isComment)) {
        //>,>=
        if (*(str + i + 1) == '=') {
            printf("(105,>=) ");
        } else {
            printf("(%d,>) ", IsSymbol('<'));
        }
```

```
                } else if (*(str + i) == '=' && (!isComment)) {
                    //== =
                    if (*(str + i + 1) == '=') {
                        printf("(106,==) ");
                        i ++;
                    } else {
                        printf("(%d,=) ", IsSymbol('='));
                    }
                } else if (*(str + i) == '+' && (!isComment)) {
                    //== =
                    if (*(str + i + 1) == '+') {
                        printf("(107,++) ");
                        i ++;
                    }
                    if(*(str + i + 1) == '='){
                        printf("(108,+=) ");
                        i ++;
                    }
                    else {
                        printf("(%d,+) ", IsSymbol('+'));
                    }
                }else if (*(str + i) == '-' && (!isComment)) {
                    //== =
                    if (*(str + i + 1) == '-') {
                        printf("(109,--) ");
                        i ++;
                    }
                    if(*(str + i + 1) == '='){
                        printf("(110,-=) ");
                        i ++;
                    }
                    else {
                        printf("(%d,+) ", IsSymbol('+'));
                    }
                }else if (IsSymbol(*(str + i)) != -1 && (!isComment)) {
                    //余下定界符等
                    printf("(%d,%c) ", IsSymbol(*(str + i)), *(str + i));
                }
            }
            str = NULL;
        }
        //控制输出换行
      printf("\n");
    }
    return 0;
}
```

测试代码：

```
zsh@zsh-pc:~/Projects/Compiler$ cat test.cpp
#include<iostream>
int main(){
  //comment
  /* asdfa dsaf as*/if(int a == 5){
  int 2h = 10;
}
  while (i == 1) {
    int s = 100;
    int sa = i++;
    int a = 100;
    for(int i = 1 ; i <= 100 ; i--){
      a++;
    }
    int 32a = 8;
  }
}
```

词法分析结果：

```
zsh@zsh-pc:~/Projects/Compiler$ g++ Compiler.cpp && ./a.out
line:1(8,#) (26,include) (104,<<) (100,ostream) (11,>)
line:2(8,int) (0,main) (6,() (7,)) (9,{)
line:3
line:4(1,if) (6,() (8,int) (100,a) (106,==) (101,5) (7,)) (9,{)
line:5(8,int) (错误 ：2h 不能作为变量名)(4,=) (101,10) (5,;)
line:6(10,})
line:7(3,while) (6,() (100,i) (106,==) (101,1) (7,)) (9,{)
line:8(8,int) (100,s) (4,=) (101,100) (5,;)
line:9(8,int) (100,sa) (4,=) (100,i) (107,++) (0,+) (5,;)
line:10(8,int) (100,a) (4,=) (101,100) (5,;)
line:11(23,for) (6,() (8,int) (100,i) (4,=) (101,1) (5,;) (100,i) (104,<<) (101,100) (5,;) (100,i) (109,--) (0,+) (7,)) (9,{)
line:12(100,a) (107,++) (0,+) (5,;)
line:13(10,})
line:14(8,int) (错误 ：32a 不能作为变量名)(4,=) (101,8) (5,;)
line:15(10,})
line:16(10,})
```