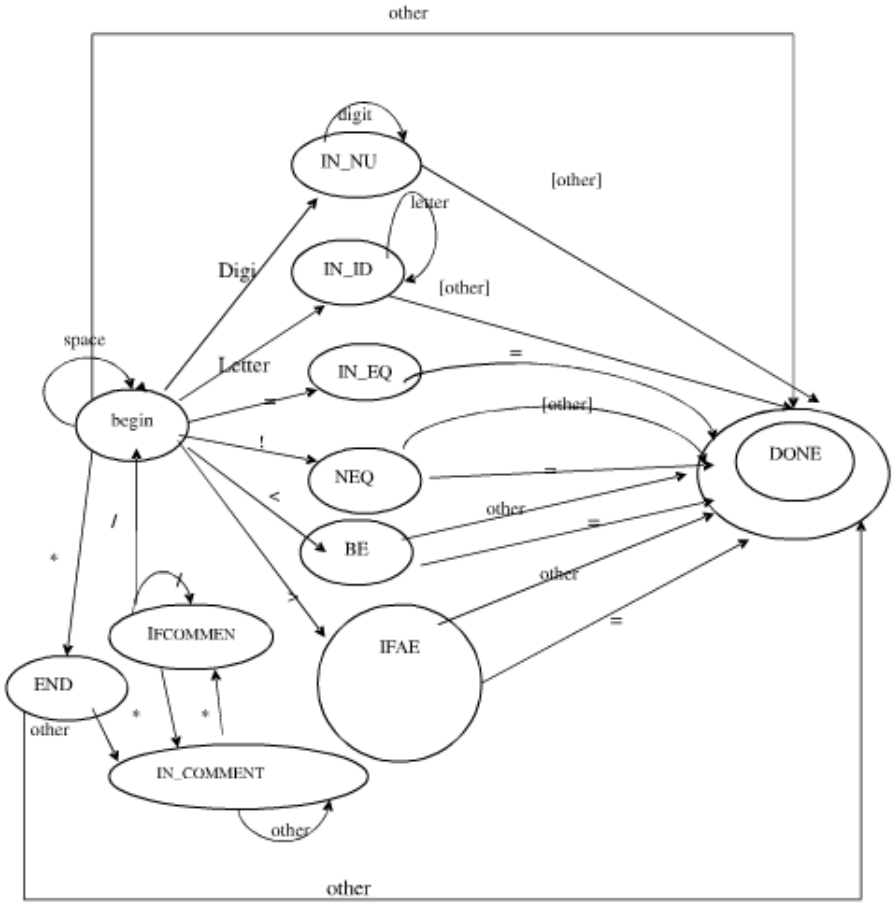
四 川 大 学 计 算 机 学 院、软 件 学 院

实 验 报 告

学号：2014141462082 姓名：胡 研 专业： 计算机 班级： 4 第 3 周

课程名称	编译原理课程设计	实验课时	2
实验项目	C-Minus 词法分析	实验时间	2017.4.17
实验目的	熟悉 C-Minus 语言的词法 手工构造 C-Minus 的词法分析器 构造 DFA,设计数据类型、数据结构		
参考资料	《编译原理及实践》 第四版 《程序设计语言编译原理》 第一版		
C-Minus 语言词法规则特点	C-Minus 的词法规则 （1）关键字： if else int return void while （2）专用符号： + - * / < <= > >= == != = ; , ( ) [ ] { } /* */ （3）其他标记为 ID 和 NUM ，通过下列正则表达式定义： ID = letter letter* NUM = digit digit* Letter = a|..|z|A|..|Z Digit = 0|..|9 （4）空格由空白、换行符、制表符组成。 （5）注释由 /*...*/ 围起来。		

C-Minus DFA



DFA 方案

初始状态为 begin，并过滤掉空格，读取到/进入判断并读取下一个字符，检验其是否为*，是则进入注释 INCOMMENT，同理读到*/则结束注释，回到开始状态 Begin；结束符直接结束；读到数字和字母进入 INNUM 和 INID 一直循环，直到读到非数字或者字母，结束循环，进入 Done；符号则判断符号类型，注意<=这样的符号需要读取下一个判断其内容，并且需要调用 ungetChar 方法回退一个字符。

数据类型、数据结构设计	

```
enum TokensType                          //枚举变量，表示Token的类型
{
    ENDFILE, ERROR,
    IF, ELSE, RETURN, INT, VOID, WHILE, ASSIGN,
    ID, NUM,
    PLUS, MINUS, TIMES, OVER, LT, LTP, RT, RTP, EQUAL, NEQUAL, EQ, SEMI,
COMMA, LFKH, RFLH, LPAREN, RPAREN, LHKH, RHKH, LZS, RLS};


extern FILE* source;                    //外部文件指针，读取的目标文件
extern FILE* listing;                   //外部文件指针，输出文件

extern int lineno;                      //行号
extern int TraceScan;                   //路径扫描
extern int EchoSource;                  //回显


typedef enum                            //枚举变量，程序状态
{
    START, INASSIGN, INCOMMENT, INNUM, INID, DONE,
}StateType;

char tokenString[MAXTOKENLEN + 1];     //存储读到的Token
#define BUFLEN 256                      //一行最大长度
#define EOF_FILE -1                     //自定义文件结束符

static char lineBuf[BUFLEN];           //存储一行
static int linepos = 0;                //行中的位置
static int bufsize = 0;                //缓冲区大小
static int EOF_flag = false;       //  int 也可以使用true!

static struct                          //定义结构，表示关键字
{
    char* str;
    TokensType tok;
}reservedWords[MAXRESERVED]
= { { "if", IF }, { "else", ELSE }, { "int", INT }, { "return", RETURN },
{ "void", VOID }, { "while", WHILE } };
```

关键代码的分析	

```
static char getNextChar(void)       //获取下一个非空字符，lineBuf满了就装到一个新行里
{
    if (!(linepos < bufsize))
    {
        lineno++;
        if (fgets(lineBuf, BUFLEN - 1, source))  //fgets(_Out_writes_z_(_MaxCount) char * _Buf, _In_ int _MaxCount, _Inout_ FILE * _File)
            //                  输出文件，最大个数，读入文件指针
        {
            if (EchoSource)
            {
                fprintf(listing, "%4d:%s", lineno, lineBuf);
            }
            bufsize = strlen(lineBuf);
            linepos = 0;
            return lineBuf[linepos++];
        }
        else
        {
            EOF_flag = true;
            return EOF;
        }
    }
    else return lineBuf[linepos++];
}

static void ungetNextChar(void)    //回退一个字符
{
    if (!EOF_flag)
    {
        linepos--;
    }

                                    }

static TokensType reservedLookup(char* s)       //查找关键字函数
{
```

```
    int i;
    for (int i = 0; i < MAXRESERVED; i++)          //遍历关键字数组
    {
        if (!strcmp(s, reservedWords[i].str))           //strcmp（S1,S2）
S1 > S2 返回值大于0,S1 <S2 返回值小于0，相等返回0
        {
            return reservedWords[i].tok;        //匹配到返回关键字
        }
    }
    return ID;                                  //没有找到，返回ID

                                    }

TokensType getToken(void)                  //获取Token函数
{
    int tokenStringIndex = 0;
    TokensType currentToken;
    StateType state = START;
    int save;
    while (state != DONE)                  //初始状态
    {
        int c = getNextChar();
        save = true;
        switch (state)
        {
        case START:
            if (isdigit(c))
            {
                state = INNUM;
            }
            else if (isalpha(c))
            {
                state = INID;
            }
            else if (c == ':')
            {
                state = INASSIGN;
            }
            else if ((c == ' ') || (c == '\t') || (c == '\n'))
            {
```

```
                save = false;
            }
            else if (c == '/')        //可能是注释或者除号
            {
                int next = getNextChar();   //获取下一个字符，用于判断
                if (next == '*')
                {
                    save = false;           //进入注释，不要保存
                    state = INCOMMENT;
                }
                else
                {
                    currentToken = OVER;    //进入除法
                }
                ungetNextChar();            //注意回退一个字符！！！

                                        }

                        ……省略部分代码

case '<':
        save = false;
            if (getNextChar() == '=')     //判断是否为==
                {
                    tokenString[tokenStringIndex++] = '<';
                    tokenString[tokenStringIndex++] = '=';
                    currentToken = LTP;
                                          //注意tokenString的添加
                }
            else
                {
                    currentToken = LT;
                    tokenString[tokenStringIndex++] = '<';
                }
    break;


    //打印记号函数
void printToken(TokensType token, char *tokenString)
{
    switch (token)
```

```
    {
    case IF:
    case INT:
    case ELSE:
    case VOID:
    case RETURN:
    case WHILE:
        fprintf(listing, "\treserved word:%s\n", tokenString);
        break;
    case ID:
        fprintf(listing, "\tID, name = %s\n", tokenString);
        break;
    case NUM:
        fprintf(listing, "\tNUM, val = %s\n", tokenString);
        break;
    case ASSIGN:
    case PLUS:
    case MINUS:
    case TIMES:
    case OVER:
    case LT:
    case LTP:
    case RT:
    case RTP:
    case EQUAL:
    case NEQUAL:
    case EQ:
    case SEMI:
    case COMMA:
    case LFKH:
    case RFLH:
    case LPAREN:
    case RPAREN:
    case LHKH:
    case RHKH:
    case LZS:
    case RLS:
        fprintf(listing, "\t%s\n", tokenString);
    }
}
```

结　论（运行结果实例、分析）

示例输入

example.c- - 记事本

文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

```
/* A program to perform finding the maximum number of the three. */

int max(int x, int y, int z)
{
        int result;
        if (x > y)
        {
                result = x;
                if (z > result)
                {
                        result = z;
                }
        }
        else
        {
                result = y;
                if (z > result)
                {
                        result = z;
                }
        }
        return result;
}

void main(void)
{
        int a;
        int b;
        int c;
        /* int d; */
```

输出结果

myresult.txt - 记事本

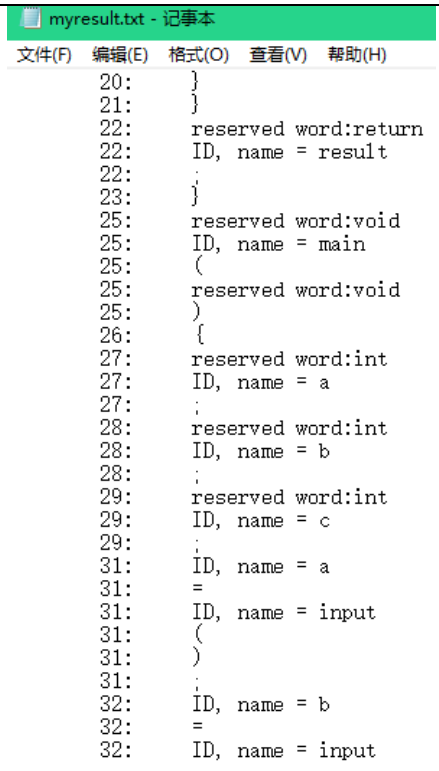
文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

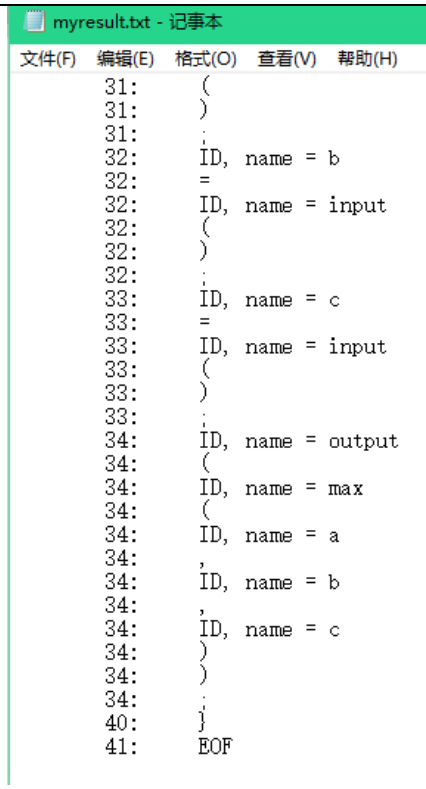
```
****************************************
C-Minus Lexical Analazer Result:
****************************************

        3:      reserved word:int
        3:      ID, name = max
        3:      (
        3:      reserved word:int
        3:      ID, name = x
        3:      ,
        3:      reserved word:int
        3:      ID, name = y
        3:      ,
        3:      reserved word:int
        3:      ID, name = z
        3:      )
        4:      {
        5:      reserved word:int
        5:      ID, name = result
        5:      ;
        6:      reserved word:if
        6:      (
        6:      ID, name = x
        6:      >
        6:      ID, name = y
        6:      )
        7:      {
        8:      ID, name = result
        8:      =
        8:      ID, name = x
```

myresult.txt - 记事本

文件(F)　编辑(E)　格式(O)　查看(V)　帮助(H)

```
        9:      reserved word:if
        9:      (
        9:      ID, name = z
        9:      >
        9:      ID, name = result
        9:      )
        10:     {
        11:     ID, name = result
        11:     =
        11:     ID, name = z
        11:     ;
        12:     }
        13:     }
        14:     reserved word:else
        15:     {
        16:     ID, name = result
        16:     =
        16:     ID, name = y
        16:     ;
        17:     reserved word:if
        17:     (
        17:     ID, name = z
        17:     >
        17:     ID, name = result
        17:     )
        18:     {
        19:     ID, name = result
        19:     =
        19:     ID, name = z
        19:     ;
```

<table>
<tr>
<td></td>
<td>




运行结果分析

实验结果达到了实验的要求，对于变量定义语句，IF 语句，关键字,RETURN 语句都得到了正确的分析结果，正确的输出了行号，并且实现了注释的去除，在读取完成后结束程序
</td>
</tr>
<tr>
<td>小　结</td>
<td>完成了实验，熟悉了 C-Minus 语言和其词法。根据语言和词法规则，顺利构造 DFA。成功用 C++语言，根据构造的 DFA，增强了自己的编程能力和水平技巧，使用了较大的数据结构，熟悉了 C++语言，实践了课本上 C-Minus 语言的语法分析，实现了 C-Minus 词法分析器，提高了对编译原理的兴趣，增强了对编译原理的认识</td>
</tr>
<tr>
<td>指导老师评　议</td>
<td>成绩评定：　　指导教师签名：</td>
</tr>
</table>