北京郵電大學

实验报告

题目： C 语言词法分析程序的设计与实现

姓名: 谢蔚钦

班级: 2020211315

学号: 2020211451

学院: 计算机科学与技术学院

2020 年 9 月 26 号

一、实验内容及要求

1. 可以识别出用C语言编写的源程序中的每个单词符号,并以记号的形式输出每个单词符号。

2. 可以识别并跳过源程序中的注释。

3. 可以统计源程序中的语句行数、各类单词的个数、以及字符总数,并输出统计结果。

4. 检查源程序中存在的词法错误,并报告错误所在的位置。

5. 对源程序中出现的错误进行适当的恢复,使词法分析可以继续进行,对源程序进行一次扫描,即可检查并报告源程序中存在的所有词法错误。

二、实验说明

1. 操作系统: Ubuntu22.04

2. 编程语言: C

3. 编译器: GCC 11.2.0

编写的词法分析器支持ANCI C(32个关键词),代码目录结构如下:

main.c:主函数

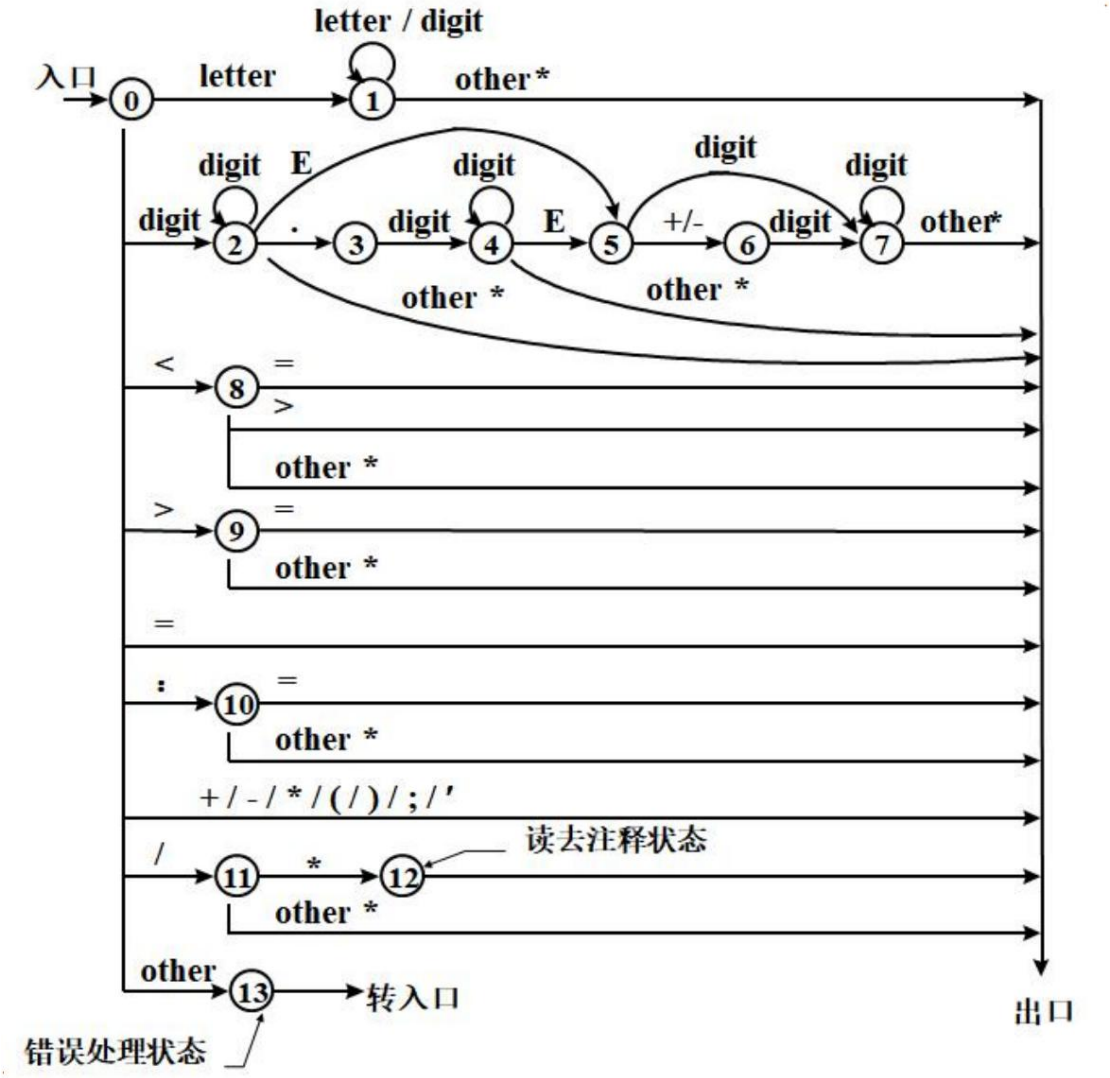
lexemer.h:包含了词法分析核心函数

inti.h: 定义了相关结构体和全局变量

tool.h:定义了一些辅助核心程序所使用的函数

三、实验分析

在本次实验中需要对 c 语言源代码进行分析，识别出其中的标识符保留字等,所以需要构造一个 DFA,能对 Token 中的字符串进行识别，识别流程图如下:



根据这个状态转移流程图，我们可以构造一个 DFA，并且通过 c 中的 switch 语句来实现这个词法分析器程序。头文件 init.h 中定义了实验所需结构体以及全局变量，如下所示:

```
#define BUFFER_SIZE 1024

typedef struct {
```

```
        char *type;
        int num;
    } Word;
    Word words[6] = {{"Keyword", 0}, {"Identifier",
0}, {"Integer", 0}, {"Float", 0},{"Punctuation", 0},
{"Unknown", 0}};
    typedef struct {
        char *type;
        char value[15];
        int line;
    }Symbol;
    char keywords[32][10] = {"auto", "break", "case",
"char","const","continue","default","do", "double",
"else",  "enum",  "extern","float","for",  "goto",
"if",  "int",  "long","register","return","short",
"signed", "sizeof", "static","struct","switch",
"typedef","union","unsigned","void","volatile","w
hile"
    };
    int symbol_count = 0;
    int line = 0;
    int characters = 0;
```

在如上代码中，BUFF_SIZE 定义了缓冲区的最大值，Word 结构体用于记录各类型单词的数量，Symbol 结构体用于存储识别出的记号对，kewords 用于存储 c 的关键字，共 32 个，全局变量 symbol_count 用于记录记号对的数量，line 和 characters 分别用于记录代码行数以及代码字符数量。

词法分析程序的核心代码使用了 switch 和 if else 语句对流程转移进行判断，通过变量 c 读取当前的字符，然后根据不同的状态以及 token 当前的值进行判断，决定下一步的状态转换，在能够识别出当前的 Token 时，将其记录在表中，然后状态值重新设置为 0，照此不断循环，直到读取到文件末尾。（核心代码见附件）

三、运行结果及分析

对 check.c 文件进行词法分析，得到的结果如下（部分结果，详情见 result.txt）：

```
The number of rows in this source file : 278
The number of characters in this source file : 8404
[Keyword] : 95
[Identifier] : 300
[Integer] : 94
[Float] : 0
[Punctuation] : 818
[Unknown] : 4
```

该程序能够识别出代码总行数、代码字符总数量、关键词总数量、标识符总数量、整数总数量、浮点数总数量、标点符号总数量以及未知字符总数量。此为，还能对错误结果进行报错，显示出出错的行数，

并且能对其进行纠错。

四、实验心得

通过这次编写词法分析器的实验，让我对 c 语言的指针操作更加熟练，在进行记号存储的时候都需要用到指针对内存地址值进行改写，并且随着识别出的记号对增多，还要对存储数组的大小进行扩增，要避免内存泄漏以及指针无效。在设计自 DFA 的过程中，让我对自动机的状态转移过程有了更深的认识，明白了如何用编程语言来实现这一过程。总之，这次实验提高了我的代码编写能力，让我明白如何更好的去设计逻辑判断语句，来避免一些不容易发现的边界问题，同时也让我对词法分析过程有了更清晰明确的认识。