

# 实验一 词法分析器设计与实现

## 一、实验目的

通过设计并实现的词法分析器，掌握对正规文法、正规式、有穷自动机、不确定的有穷自动机（NFA）到确定的有穷自动机（DFA）转换，DFA 最小化等单词符号分析工具的具体设计实现，更进一步加深对程序设计语言源程序进行词法分析方法和原理解，能初步达到设计编制和调试编译系统中词法分析器的能力，增强解决实际问题的实践能力。

## 二、实验内容

### 1、单词的 BNF 表示

〈标识符〉 $\rightarrow$  〈字母〉|〈字母〉〈字母数字〉

〈字母数字〉 $\rightarrow$  〈字母〉〈字母数字〉|〈数字〉〈字母数字〉|\_〈字母数字〉|  $\epsilon$

〈字母数字〉 $\rightarrow$  〈字母〉|〈数字〉

〈字母〉 $\rightarrow$  a|b|.....|z

〈数字〉 $\rightarrow$  0|1|.....|9

〈常量〉 $\rightarrow$  〈无符号整数〉

〈无符号整数〉 $\rightarrow$  〈数字〉〈数字串〉

〈数字串〉 $\rightarrow$  〈数字〉〈数字串〉|  $\epsilon$

〈运算符〉 $\rightarrow$  +|-|\*|/|=|<|==|<=|!=|>|>=

〈关键字〉 $\rightarrow$  main|if|then|while|do

〈界符〉 $\rightarrow$  { } | ( ) | ;

说明：产生式中〈，〉括起来的部分表示非终结符号。

2. 将以文法转换成有穷自动机。

3. 实现有穷自动机识别单词符号

## 三、实验步骤

1. 根据词法规则总结出每种单词符号的结构特征
2. 设计每种单词符号有穷自动机(状态转换图)
3. 将几种单词符号的状态转换图进行合并成唯一的初态, 构造一个能识别所有种类的单词符号 NFA
4. NFA 确定化 DFA, 再最小化 DFA
5. 将 DFA 构造状态矩阵;
6. 用状态矩阵设计出一个词法分析程序。

#### 四、实验要求

- 1、从输入的源程序中，识别出各个具有独立意义的单词符号（保留字、标识符、常数、运算符、界符）五大类。

单词符号	种别编码
关键字（保留字）	1
标识符	2
常数	3
运算符	4
界符	5

- 2、依次以二元式(单词种别码, 单词符号的属性值)输出各个单词符号，如遇到错误单词符号时可显示“Error, 出错的位置”，然后跳过错误部分继续显示）。

例：

如源程序为 C 语言。输入如下一段：

```
main()
{  int  a,b;

    a = 10;

    b = a + 20;

}
```

要求输出如下图。

(2, " main" )

(5, " ( " )

(5, " ) " )

(5, " { " )

(1, " int" )

(2, " a" )

(5, " , " )

(2, " b" )

(5, " ;" )

(2, " a" )

(4, " =" )

(3, " 10" )

(5, " ;" )

(2, " b" )

(4, " =" )

(2, " a" )

(4, " +" )

(3, " 20" )

(5, " ;" )

(5, " } " )

## 五、实验过程及学时安排

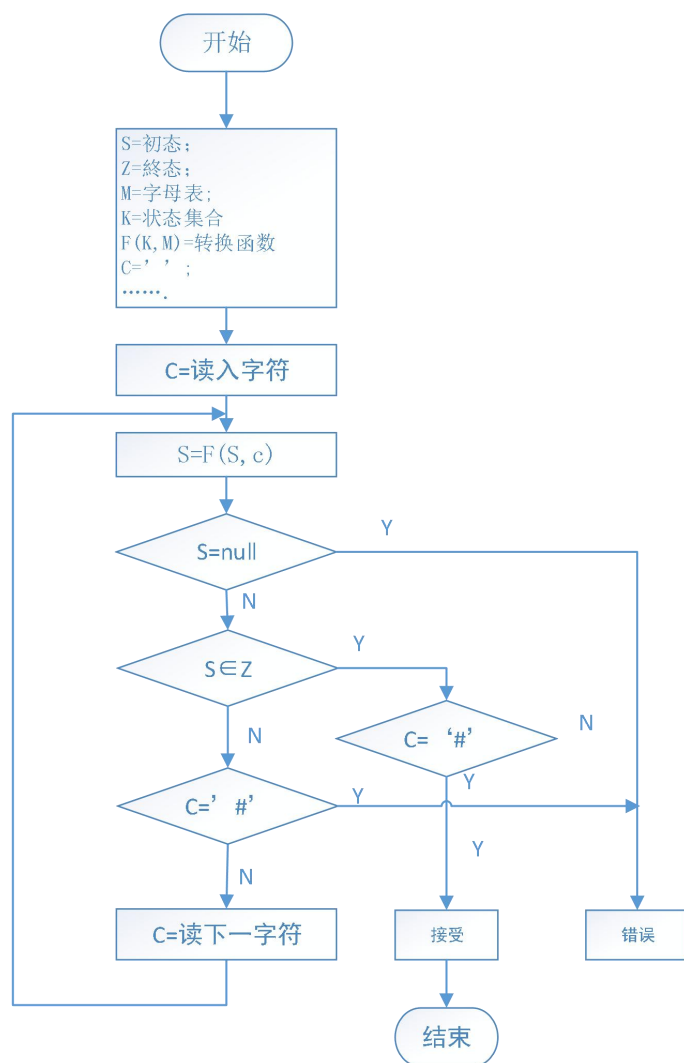
1. 做到实验前预习，阅读课本有关章节，明确语言的语法，写出基本保留字、标识符、常数、运算符、分隔符和程序例。（课外）

2. 课外初步编制好程序。（课外）

3. 准备好多组测试数据。（课外）
4. 将源代码上机调试，发现错误，再修改完善。（课内 4 学时）
5. 撰写并提交实验报告。（课外）

注：3-5 人一组

提示：



DFA 识别符号串流程图