

# 实验—词法分析器

题目：（C语言、Java语言，语言不限）词法分析器

要求：1人一组

### 1. 单词的分类。

可将标识符（变量名、函数名、数组名等）各归一类；  
将常数归为一类；  
保留字和分隔符则采取一词一类。

### 2. 符号表的建立。

可事先建立一关键字表，以备在识别关键字时进行查询。变量名表及常数表则在词法分析过程中建立。

### 3. 单词串的输出形式。

所输出的每一单词，均按形如（**CLASS,VALUE**）的二元式编码。对于变量标识符和常数，**CLASS**字段为相应的类别码，**VALUE**字段则是该标识符、常数在其符号表中登记项的序号（要求在变量名表登记项中存放该标识符的字符串，其最大长度为四个字符；常数表登记项中则存放该整数的二进制形式。）。对于保留字和分隔号，由于采用一词一类的编码方式，所以仅需在二元式的**CLASS**字段上放置相应的单词的类别码，**VALUE**字段则为“空”。不过，为便于查看由词法分析程序所输出的单词串，也可以在**CLASS**字段上直接放置单词符号串本身。

可以仿照书上图3.3的实现程序的结构来编写上述词法分析程序，但其中的若干语义过程有待于具体编写。

包括：过滤无效字符、数值转换、宏展开、预包含处理等

## 实习报告内容：

- 1.目的要求
- 2.单词分类表
- 3.单词状态图
- 4.算法描述
- 5.程序结构
- 6.运行结果
- 7.调试情况
- 8.设计技巧及体会
- 9.源程序清单（电子版）

# 实验—语法分析

## 一、预测分析法基本要求:

- 1) 任意输入一个文法**G**;
- 2) 处理文法中可能存在的左递归和公共左因子问题;
- 3) 对文法中的每个非终结符自动生成并打印输出:  
    ① **FIRST**集;      ② **FOLLOW**集;
- 4) 判断处理后的文法是否为**LL(1)**文法,  
    如果是, 自动生成并打印输出其预测分析表;
- 5) 模拟分析过程。  
    如输入一个句子, 如果该句子合法则输出与句子对应的语法树; 能够输出分析过程中每一步符号栈的变化情况。如果该句子非法则进行相应的报错处理。

测试文法:

$$\textcircled{1} \quad S \rightarrow ABBA$$

$$A \rightarrow a \mid \varepsilon$$

$$B \rightarrow b \mid \varepsilon$$

$$\textcircled{2} \quad S \rightarrow aSe \mid B$$

$$B \rightarrow bBe \mid C$$

$$C \rightarrow cCe \mid d$$

$$\textcircled{3} \quad E \rightarrow E+T \mid T$$

$$T \rightarrow T^*F \mid F$$

$$F \rightarrow (E) \mid i$$

$$\textcircled{4} \quad S \rightarrow Qc \mid c$$

$$Q \rightarrow Rb \mid b$$

$$R \rightarrow Sa \mid a$$

## 二、算符优先分析法基本要求:

- 1) 任意输入一个文法**G**;
- 2) 判断该文法是否为算符文法;
- 3) 对文法中的每个非终结符自动生成并打印输出:  
① **FIRSTVT**集;      ② **LASTVT**集;
- 4) 判断该文法是否为算符优先文法,  
如果是自动生成并打印输出其算符优先矩阵;
- 5) 模拟分析过程。  
如输入一个句子, 如果该句子合法则输出与句子对应的语法树; 能够输出分析过程中每一步符号栈的变化情况以及根据当前最左素短语进行归约的过程。如果该句子非法则进行相应的报错处理。



测试文法:

①  $E \rightarrow TE'$   
 $E' \rightarrow +TE' | \epsilon$   
 $T \rightarrow FT'$   
 $T' \rightarrow *FT' | \epsilon$   
 $F \rightarrow (E) | i$

②  $E \rightarrow E+T | T$   
 $T \rightarrow T * F | F$   
 $F \rightarrow P \wedge F | P$   
 $P \rightarrow (E) | i$

### 三、SLR(1)分析法基本要求:

- 1) 任意输入一个文法**G**;
- 2) 打印输出该文法的所有项目;
- 3) 打印输出项目集;
- 4) 构造可以识别活前缀的**DFA**;
- 5) 利用**LR(0)**或**SLR(1)**方法构造分析表  
(**Action**和**Goto**表);
- 6) 模拟分析过程。

如输入一个句子，如果该句子合法输出分析过程中每一步符号栈及状态栈的变化情况；如果该句子非法则进行相应的报错处理。

测试文法:

①  $E \rightarrow aA | bB$

$A \rightarrow cA | d$

$B \rightarrow cB | d$

②  $S \rightarrow rD$

$D \rightarrow D, i \mid i$

## 实习报告内容:

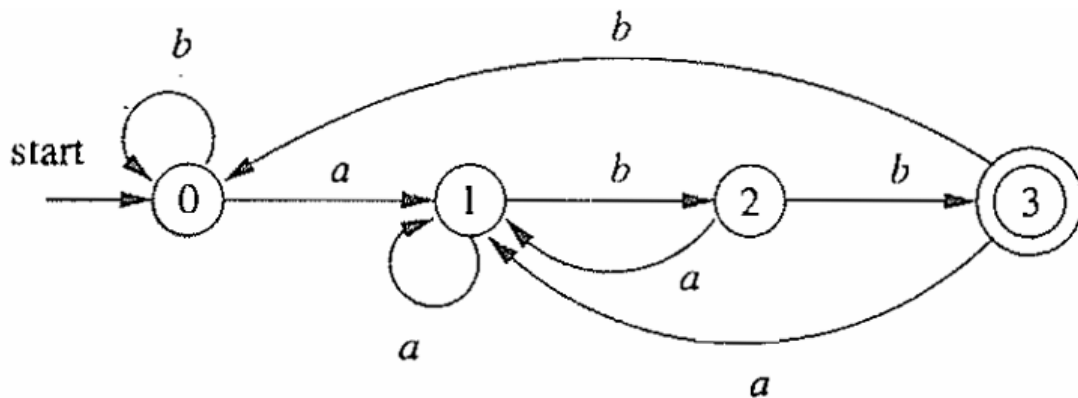
- 1.实验目的
- 2.实验要求
- 3.实验内容
- 4.采用的数据结构
- 5.算法描述
- 6.运行结果
- 7.调试情况
- 8.设计技巧及体会
- 9.源程序清单（电子版）

## 一、实验目的：

- 1、理解有穷自动机的作用；
- 2、利用五元组、状态图 and 矩阵表表示有穷自动机；
- 3、以程序实现有穷自动机的运行过程。

## 二、实验内容

所给自动机如图所示：



## 三、实验要求：

- 1、给出自动机的五元组形式及矩阵表示。
- 2、模拟 DFA 的识别算法
- 3、设计相应的例子进行测试。

功能要求：输入一个单行无空格的字符串（以“#”号结束），如果该字符串是一个合法的输入，则显示“接受”，否则显示“不接受”。

## 四、实验步骤

- 1、给出自动机的五元组形式及矩阵表示。
- 2、写出编程思路、源代码；
- 3、写出上机调试时发现的问题，以及解决的过程；

源代码

```
#include <stdio.h>
```

```
char str[30];
```

```
void move(int num, int id) {
```

```
    switch(num) {
```

```
        case 0:
```

```
            if(str[id] == 'a') {
```

```
                if(str[id+1] != '#')
```

```
                    move(1, id+1);
```

```
                else printf("不接受\n");
```

```
            }
```

```
            else if(str[id] == 'b') {
```

```
                if(str[id+1] != '#') {
```

```
                    move(0, id+1);
```

```
                }
```

```
                else printf("不接受\n");
```

```
            } else printf("不接受\n");
```

```
        break;
```

```
        case 1:
```

```
            if(str[id] == 'a') {
```

```
                if(str[id+1] != '#')
```

```
        move(1, id+1);

        else printf("不接受\n");

    } else if(str[id] == 'b') {

        if(str[id+1] != '#')

            move(2, id+1);

        else printf("不接受\n");

    } else printf("不接受\n");

    break;
```

case 2:

```
    if(str[id] == 'a') {

        if(str[id+1] != '#')

            move(1, id+1);

        else printf("不接受\n");

    } else if(str[id] == 'b') {

        if(str[id+1] != '#')

            move(3, id+1);

        else printf("接受\n");

    } else printf("不接受\n");

    break;
```

case 3:

```
    if(str[id] == 'a') {

        if(str[id+1] != '#')
```

```

        move(1, id+1);

        else printf("不接受\n");

    } else if(str[id] == 'b') {

        if(str[id+1] != '#')

            move(0, id+1);

        else printf("不接受\n");

    } else printf("不接受\n");

    break;

}

}

int main() {

    printf("自动机 M 五元组为 : \n");

    printf("{0,1,2,3},{a,b},f,0,{3}\n");

    printf("f(0,a)=1    f(0,b)=0,    f(1,a)=1,    f(1,b)=2,\n    f(2,a)=1,    f(2,b)=3, f(3,a)=1,

f(3,b)=0\n");

    printf("自动机的矩阵表示为:\n");

    printf("    状态\\符号    a        b\n");

    printf("    0        1        0\n");

    printf("    1        1        2\n");

    printf("    2        1        3\n");

    printf("    3        1        0\n");

    printf("请输入待检验字符串 : \n");

```



```
while(scanf("%s", str) != EOF) {  
  
    move(0, 0);  
  
    printf("请再来测试 : \n");  
  
}  
  
return 0;  
  
}
```