

# 对算术表达式构造递归下降翻译器

17计二许红凯 320170941570

## 实验任务

对算术表达式做递归下降分析，同时将其翻译为中间代码

## 输入

算术表达式

## 输出

四元式序列

## 题目

对实验四的程序进行升级改造，使得程序对于输入的任意一个算术表达式，在对其做递归下降分析的同时，生成等价的中间代码，一遍完成。

要求：

1. 基础文法：同实验四
2. 语法分析：沿用实验四的程序框架
3. 语义分析：生成四元式
4. 一遍处理：把语义处理的代码插入到语法分析中。设计要点有两个：
  - i. 语法分析走到哪里的是偶应该执行语义动作
  - ii. 语义动作（这里指的是生成四元式）应该怎么做
5. 为化简问题，不考虑输入有错误的情况，不考虑语义检查

## 程序功能说明

### 1. 词法分析

扫描输入的算术表达式，根据词法规则，识别单词。（实验要求默认不考虑输入有错误的情况，故不考虑语义检查，也不考虑词法错误）

## 2. 语法分析

根据LittleC文法对算术表达式进行语法分析

## 3. 语义分析并翻译为中间代码

利用递归下降对算术表达式做语义分析，同时将其翻译为中间代码（四元式序列）

# 代码思路

---

1. 每一个非终结符对应于一个函数（子过程），非终结符所对应的右侧产生式为函数体
2. 每遇到一个终结符，需要判断所输入的字符是否与之匹配。若匹配，则读取下一个字符；若不匹配，则进行出错处理
3. 对算术表达式的语法和语义分析同步进行，一遍完成
4. 对算术表达式的语法分析，自顶向下进行 <表达式> ——> <项> ——> <因子>的分析

过程如下：

```
PROCEDURE <表达式>:
BEGIN
    IF SYM='+' OR SYM='-' THEN
        BEGIN
            ADVANCE; <项>;
            WHILE SYM='+' OR SYM='-' DO
                BEGIN
                    ADVANCE; <项>;
                END
            END
        ELSE IF SYM=FIRST(<项>) THEN
            BEGIN
                <项>;
                WHILE SYM='+' OR SYM='-' DO
                    BEGIN
                        ADVANCE; <项>;
                    END
                END
            END
        ELSE ERROR
    END
```

```
-----
PROCEDURE <项>:
BEGIN
    IF SYM='*' OR SYM='/' THEN
        BEGIN
            ADVANCE; <因子>;
            WHILE SYM='*' OR SYM='/' DO
                BEGIN
                    ADVANCE; <因子>;
                END
            END
```

```

END
ELSE IF SYM=FIRST(<因子>) THEN
BEGIN
    <因子>;
    WHILE SYM='*' OR SYM='/' DO
    BEGIN
        ADVANCE; <因子>;
    END
END
ELSE ERROR
END
-----
PROCEDURE <项>:
BEGIN
    IF SYM='标识符' OR SYM=<无符号整数> THEN
    BEGIN
ADVANCE;
    END
    ELSE IF SYM='(' THEN
    BEGIN
        <表达式>
        IF SYM=')' THEN
        BEGIN
            ADVANCE;
        END
        ELSE ERROR
    END
    ELSE ERROR
END
-----
PROGRAM PAESER
BEGIN
    ADVANCE;
    <表达式>;
    IF SYM<>'#' THEN ERROR
END

```

## 代码测试

1. 文本输入:  $1+2*3+(4+5)*6+123+5*(6/3)$

1. 文本输入:  $a+b*c$

```

~~算术表达式:
1+2*3+(4+5)*6+123+5*(6/3)

~~算术表达式的中间代码表示:
(*,2,3,t1)
(+,1,t1,t2)
(+,4,5,t3)
(*,t3,6,t4)
(+,t2,t4,t5)
(+,t5,123,t6)
(/,6,3,t7)
(*,5,t7,t8)
(+,t6,t8,t9)

```

```

~~算术表达式:
a+b*c

~~算术表达式的中间代码表示:
(*,b,c,t1)
(+,a,t1,t2)

```