

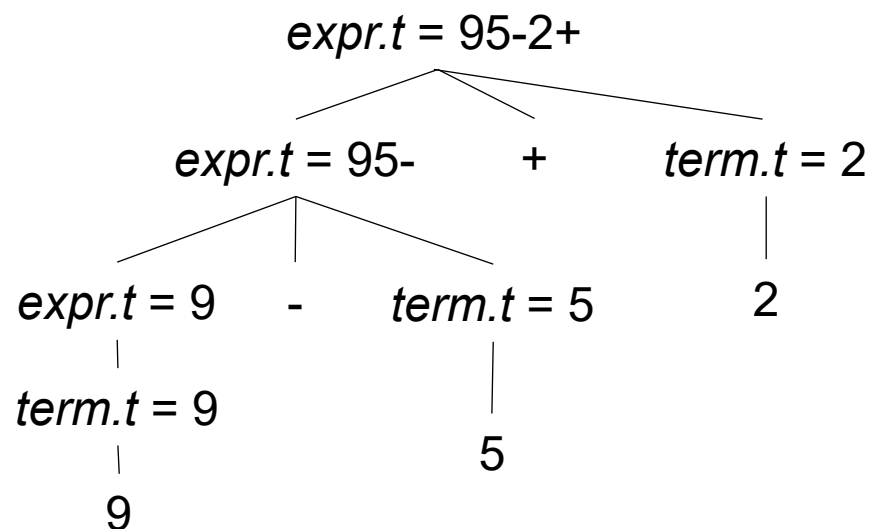
中缀表达式转后缀表达式

- 方法
 - 语法制导的定义
 - 语法制导的翻译

中缀表达式转后缀表达式

- 个位数加减法
 - 语法制导的定义

产生式	语义规则
$expr \rightarrow expr_1 + term$	$expr.t = expr_1.t \ \ term.t \ \ '+'$
$expr \rightarrow expr_1 - term$	$expr.t = expr_1.t \ \ term.t \ \ '-'$
$expr \rightarrow term$	$expr.t = term.t$
$term \rightarrow 0$	$term.t = '0'$
$term \rightarrow 1$	$term.t = '1'$
...	...
$term \rightarrow 9$	$term.t = '9'$



例： 9 - 5 + 2

中缀表达式转后缀表达式

- 给定一棵语法树，形式如下：
- (expr
 (expr
 (expr (term (9)))
 (-)
 (term (5)))
 (+)
 (term (2)))
- 要求输出：9 5 - 2 +

中缀表达式转后缀表达式

- 给定一棵语法树，形式如下：

- (expr

- (expr

- (expr (term (9)))

- (-)

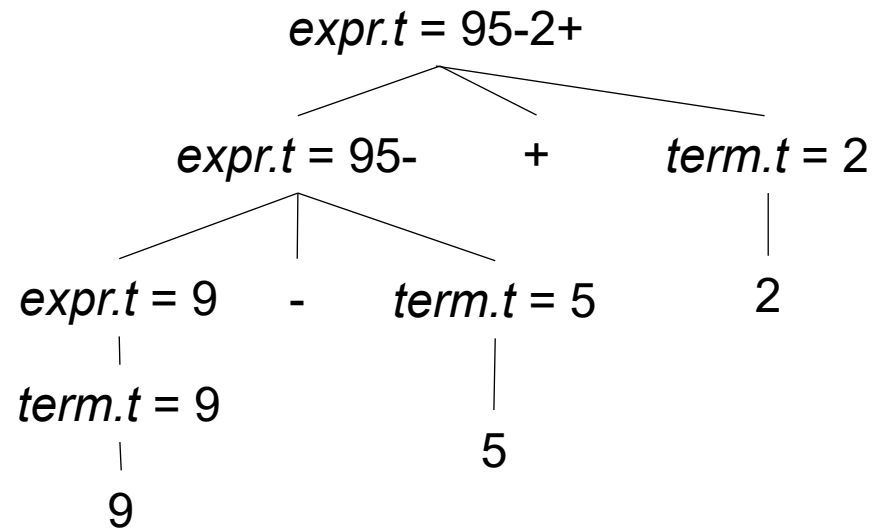
- (term (5)))

- (+)

- (term (2)))

- 或 (expr (expr (expr (term (9))) (-) (term (5))) (+) (term (2))))

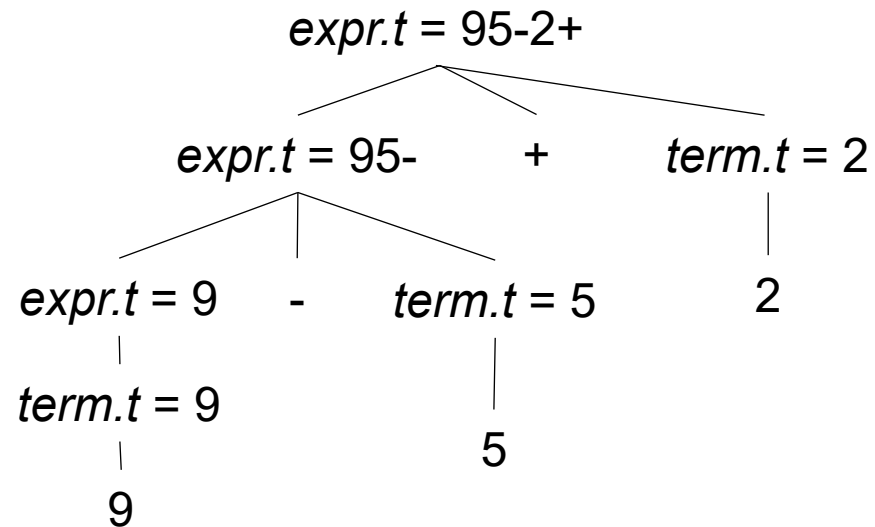
- 要求输出： 9 5 - 2 +



中缀表达式转后缀表达式

- 个位数加减法
 - 语法制导的定义

产生式	语义规则
$expr \rightarrow expr_1 + term$	$expr.t = expr_1.t \ \ term.t \ \ '+'$
$expr \rightarrow expr_1 - term$	$expr.t = expr_1.t \ \ term.t \ \ '-'$
$expr \rightarrow term$	$expr.t = term.t$
$term \rightarrow 0$	$term.t = '0'$
$term \rightarrow 1$	$term.t = '1'$
...	...
$term \rightarrow 9$	$term.t = '9'$



```
procedure visit ( node N) {
    for(从左到右遍历N的每个子节点C) {
        visit(C);
    }
    按照节点N上的语义规则求值;
}
```

中缀表达式转后缀表达式

- 方法
 - 语法制导的定义
 - 语法制导的翻译

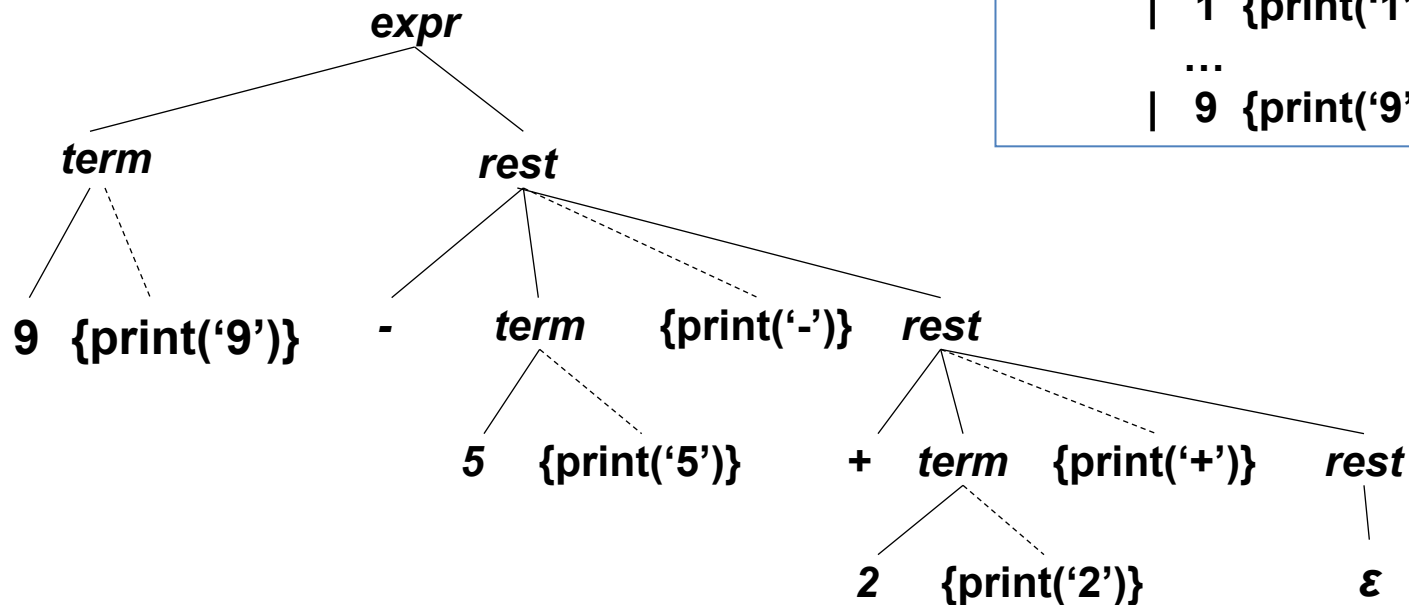
中缀表达式转后缀表达式

- 个位数加减法
– 语法制导的翻译

expr* \rightarrow *term rest

***rest* \rightarrow *+* *term* {print('+')} *rest*
 | *-* *term* {print('-')} *rest*
 | ϵ**

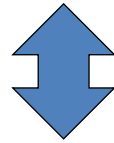
***term* \rightarrow 0 {print('0')}
 | 1 {print('1')}
 | ...
 | 9 {print('9')}**



中缀表达式转后缀表达式

- 非终结符的函数

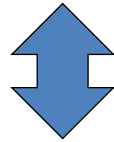
expr → *term rest*



```
void expr() {  
    term(); rest()  
}
```


中缀表达式转后缀表达式

- 非终结符的函数

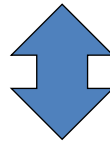
$$\begin{array}{l} rest \rightarrow + term \{print('+')\} rest \\ | - term \{print('-')\} rest \\ | \epsilon \end{array}$$


```
void rest () {  
    if( lookahead == '+' ) {  
        match('+'); term(); print('+'); rest();  
    }  
    else if( lookahead == '-' ) {  
        match('-'); term(); print('-'); rest();  
    }  
    else { } /*不对输入作任何处理*/  
}
```

中缀表达式转后缀表达式

- 非终结符的函数

```
term → 0 {print('0')}  
      | 1 {print('1')}  
      ...  
      | 9 {print('9')}
```



```
void term() {  
    if( lookahead是一个数位) {  
        t = lookahead; match(lookahead); print(t);  
    }  
    else {report(“语法错误”);}  
}
```

- 完整的程序：图2-27