



第3讲

语法分析-自顶向下-递归下降

李 诚

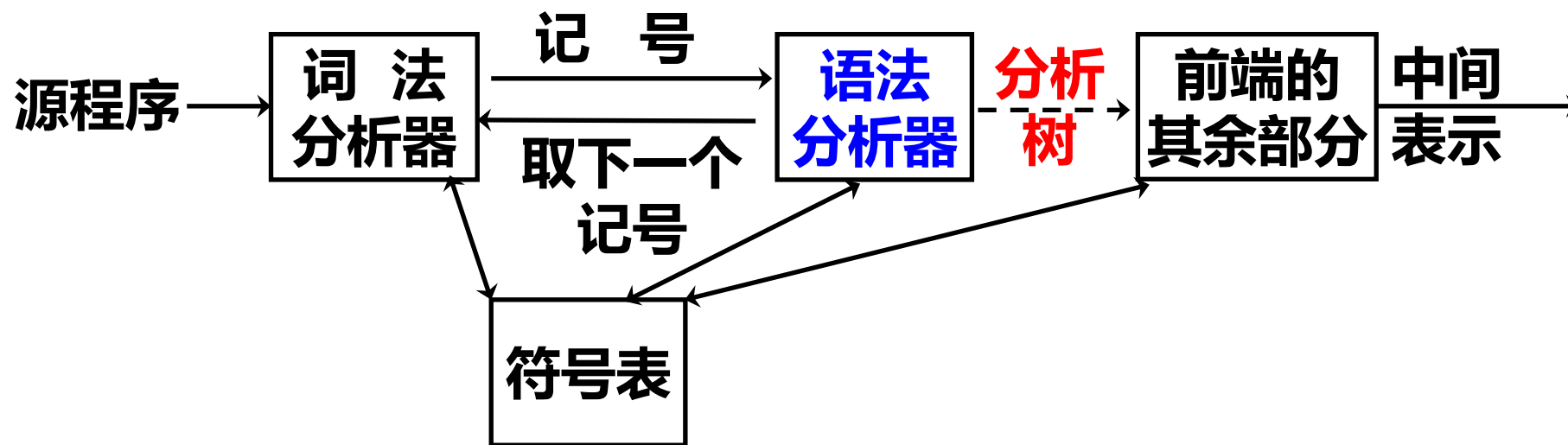
国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2023年09月13日



本节提纲



- 自顶向下与自底向上方法的区别
- 自顶向下分析方法
 - 递归下降分析方法
 - 消除左递归、提取左公因子



- **自顶向下 (Top-down)**

- 针对输入串，从文法的开始符号出发，尝试根据产生式规则 **推导 (derive)** 出该输入串。

- **自底向上 (Bottom-up)**

- 针对输入串，尝试根据产生式规则 **归约 (reduce)** 到文法的开始符号。



- **自顶向下 (Top-down)**

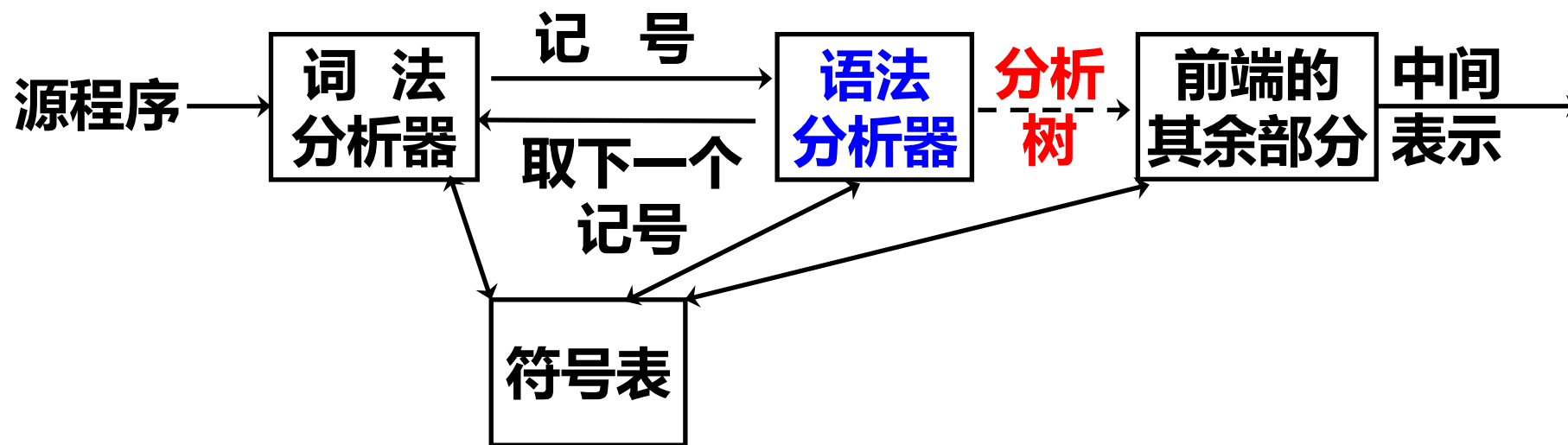
- 针对输入串，从文法的开始符号出发，尝试根据产生式规则 **推导 (derive)** 出该输入串。
- **分析树的构造方法**
 - 从根部开始

- **自底向上 (Bottom-up)**

- 针对输入串，尝试根据产生式规则 **归约 (reduce)** 到文法的开始符号。
- **分析树的构造方法：**
 - 从叶子开始



本节提纲



- 自顶向下与自底向上方法的区别
- 自顶向下分析方法
 - 递归下降分析方法
 - 消除左递归、提取左公因子



- **数据结构**

- 一个输入缓冲区和向前看指针 *lookahead*

- **分析过程**

- 自左向右扫描输入串
 - 设计一个辅助过程 *match()*，将 *lookahead* 指向的位置与产生式迭代生成的终结符进行匹配，如匹配，将 *lookahead* 挪到下一个位置
 - 为每一个非终结符写一个分析过程
 - 该过程可以调用其他非终结符的过程及 *match*
 - 这些过程可能是递归的



• 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

• 分析过程:

- 从左到右扫描输入串
- 开始符号: *expr*
- 按顺序尝试产生式

```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```

```
void term(){  
    if (lookahead is num){  
        match(lookahead);  
    } else{ if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");}  
}
```

- 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```

```
void term(){  
    if (lookahead is num){  
        match(lookahead);  
    } else{ if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");}  
}
```


- 考虑以下文法:

expr \rightarrow *term*

/ term + expr

/ term - expr

term \rightarrow num / (*expr*)

expr

(2	-	3)
---	---	---	---	---



```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```



```
void term() {  
    if (lookahead is num) {  
        match(lookahead);  
    } else { if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");  
    }  
}
```

• 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

expr
|
term

(2	-	3)
---	---	---	---	---



```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```

```
void term(){  
    if (lookahead is num){  
        match(lookahead);  
    } else{ if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");  
}
```

• 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

expr
|
term

(2	-	3)
---	---	---	---	---



```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```

```
void term(){  
    if (lookahead is num){  
        match(lookahead);  
    } else{ if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");  
}
```

• 考虑以下文法:

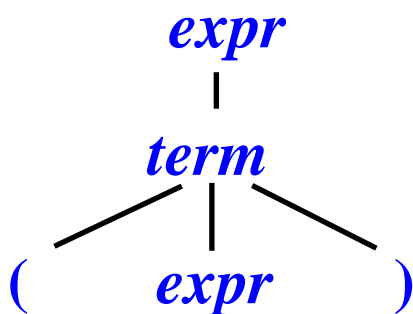
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```



• 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

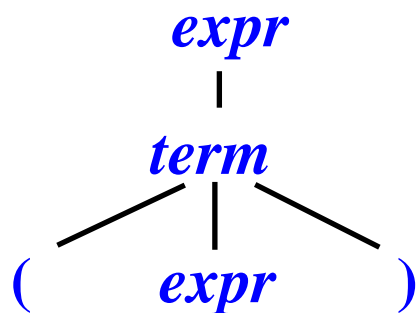
$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



**匹配
箭头前进**



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```





递归下降语法分析——演示过程



• 考虑以下文法:

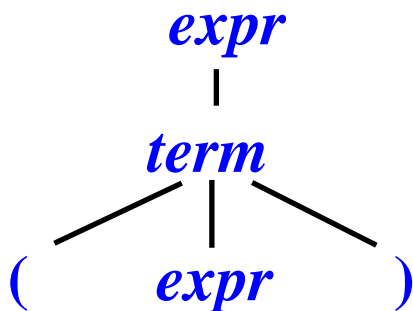
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```



• 考虑以下文法:

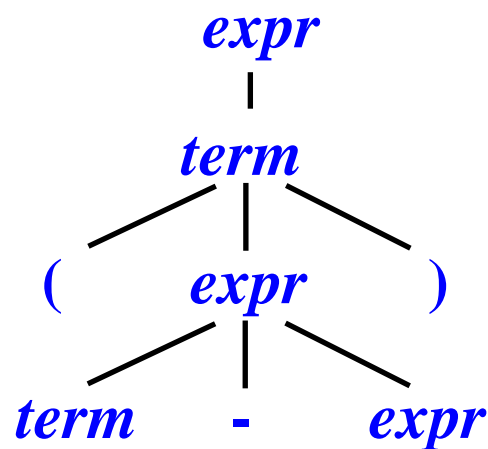
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```



```
void term(){  
    if (lookahead is num){  
        match(lookahead);  
    } else{ if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");}  
}
```

• 考虑以下文法:

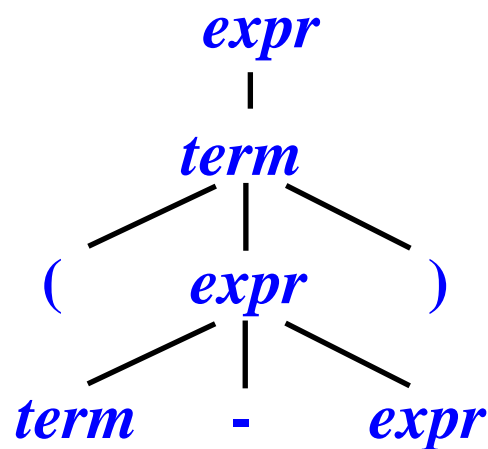
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == '+ / - ') {
        match('+ / - ');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```


• 考虑以下文法:

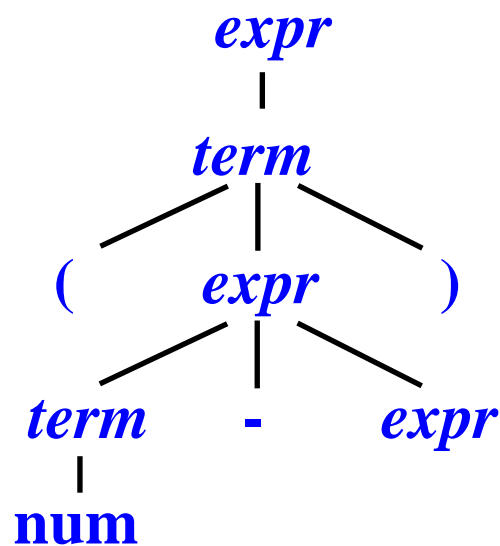
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```



• 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

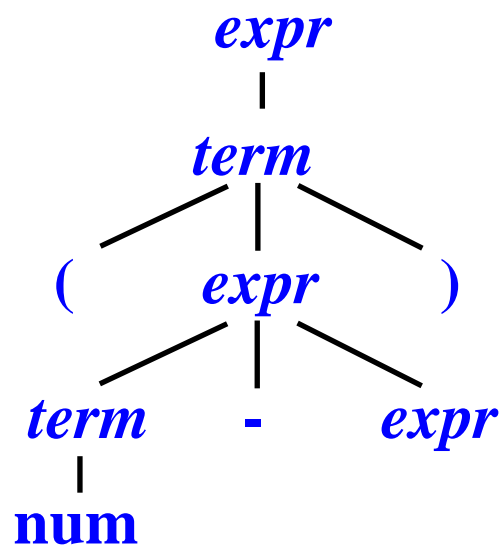
$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---



匹配
箭头前进



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '('){
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```

• 考虑以下文法:

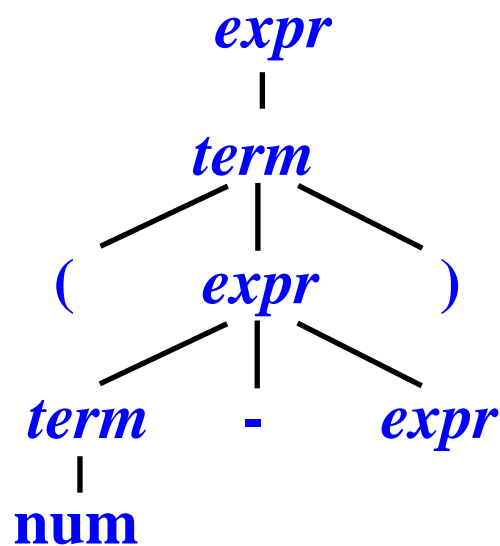
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == '+/-')
        match('+/-');
        expr();
    }
    report("语法正确");
}
```



```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```



递归下降语法分析——演示过程



• 考虑以下文法:

$expr \rightarrow term$

$/ term + expr$

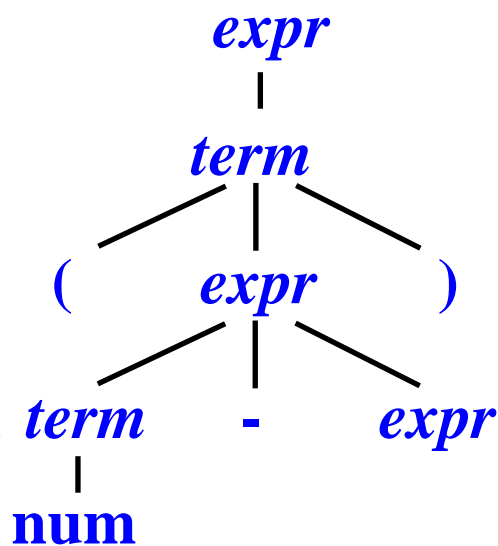
$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---



匹配
箭头前进



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```



```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```

• 考虑以下文法:

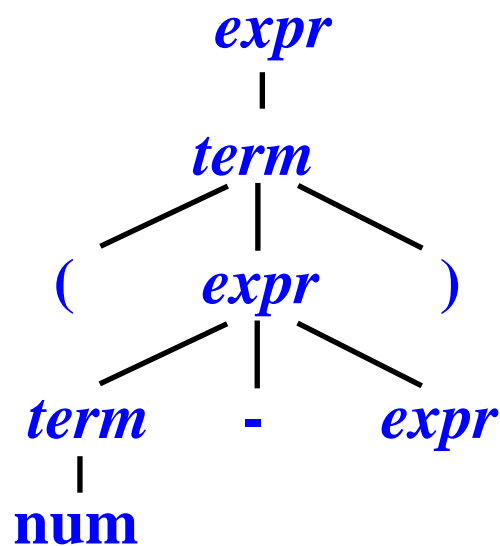
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```

• 考虑以下文法:

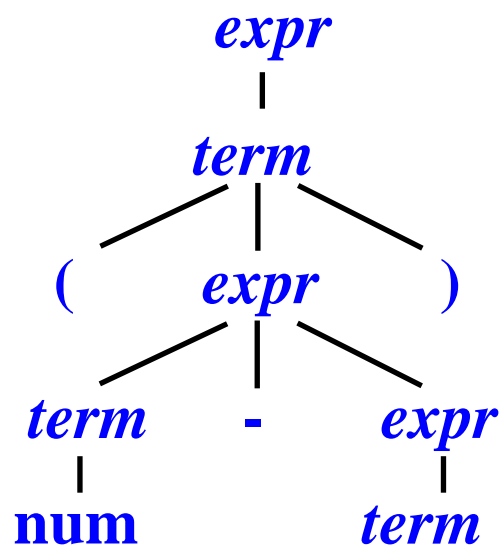
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow num / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {
    term();
    if (lookahead == ) {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```

• 考虑以下文法:

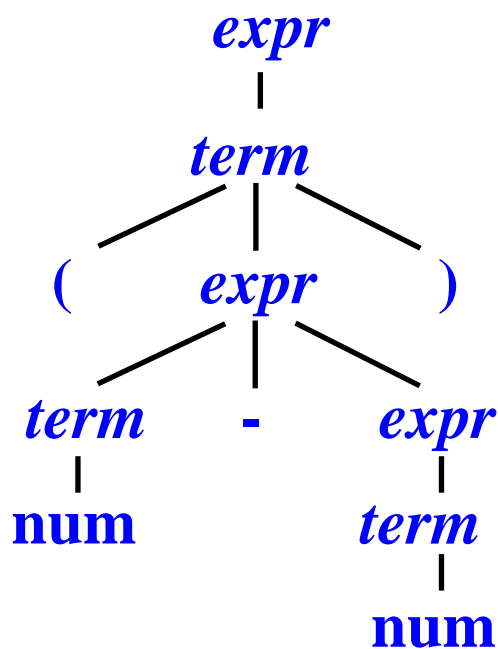
$expr \rightarrow term$

$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---



```
void expr() {  
    term();  
    if (lookahead == '+/-') {  
        match('+/-');  
        expr();  
    }  
    report("语法正确");  
}
```

```
void term(){  
    if (lookahead is num){  
        match(lookahead);  
    } else{ if (lookahead == '(') {  
        match('(');  
        expr();  
        match(')');  
    } else report("语法错误");  
}
```



• 考虑以下文法:

$expr \rightarrow term$

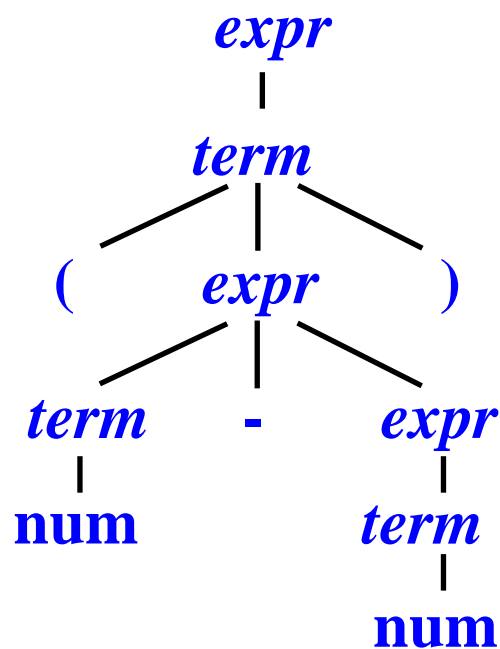
$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---

匹配
箭头前进



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```





递归下降语法分析——演示过程



• 考虑以下文法:

$expr \rightarrow term$

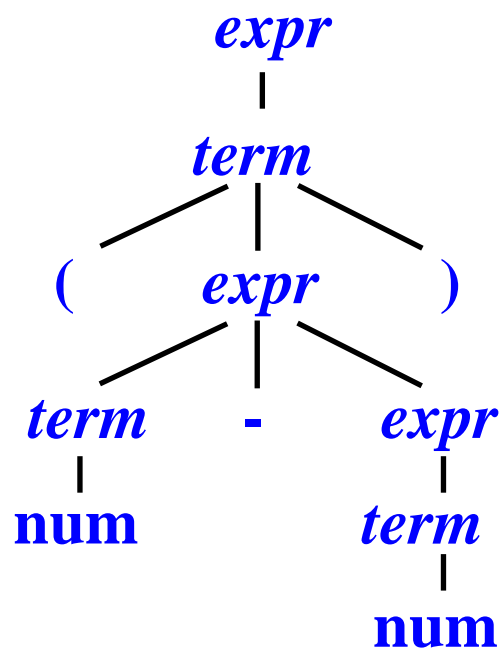
$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---

匹配
箭头前进



```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```

```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");
}
```





递归下降语法分析——演示过程



• 考虑以下文法:

$expr \rightarrow term$

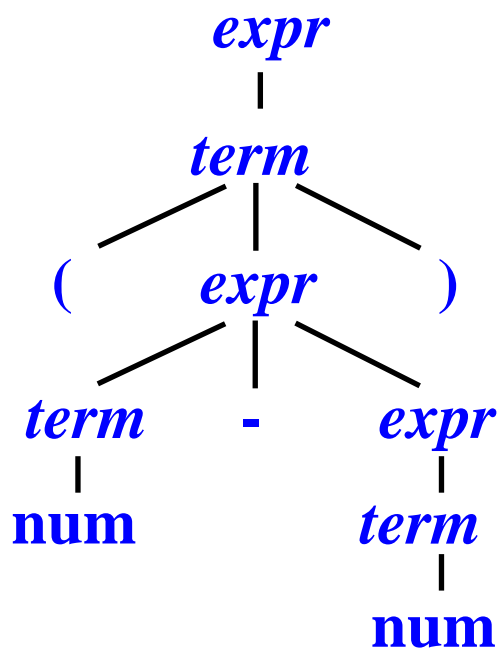
$/ term + expr$

$/ term - expr$

$term \rightarrow \text{num} / (expr)$

(2	-	3)
---	---	---	---	---

分析完毕
接受该串



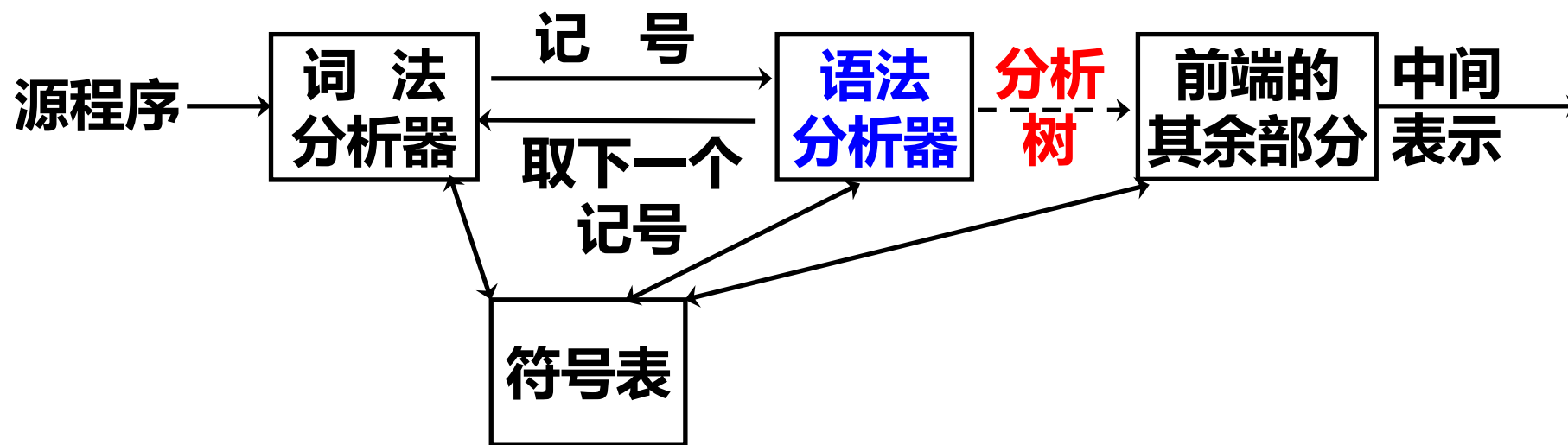
```
void expr() {
    term();
    if (lookahead == '+/-') {
        match('+/-');
        expr();
    }
    report("语法正确");
}
```



```
void term(){
    if (lookahead is num){
        match(lookahead);
    } else{ if (lookahead == '(') {
        match('(');
        expr();
        match(')');
    } else report("语法错误");}
}
```



本节提纲



• 自顶向下与自底向上方法的区别

• 自顶向下分析方法

- 递归下降预测分析方法
- 消除左递归、提取左公因子



递归下降的问题1



- 可能进入无限循环
- 考虑以下文法

$$S \rightarrow Sa / b$$

- 该文法是左递归的(left-recursive)



递归下降的问题1



- 可能进入无限循环
- 考虑以下文法

$$S \rightarrow Sa / b$$

- 该文法是左递归的(left-recursive)
- 自顶向下分析方法无法处理左递归
 - Why?



递归下降的问题1



- 可能进入无限循环
- 考虑以下文法

$$S \rightarrow Sa / b$$

- 该文法是左递归的(left-recursive)
- 自顶向下分析方法无法处理左递归
 - 考虑输入文法符号串为baaaaa
 - 最左推导如下：
 - $S \Rightarrow Sa \Rightarrow Saa \Rightarrow Saaa \Rightarrow Saaaa \dots$
 - 输入缓冲区lookahead指针纹丝未动



- 直接左递归

$A \rightarrow A\alpha \mid \beta$, 其中 α, β 不以 A 开头

- 串的特点 $\beta\alpha \dots \alpha$ ($A \Rightarrow^+ A\alpha$)

- 消除直接左递归

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \epsilon$$



- 直接左递归

$A \rightarrow A\alpha \mid \beta$, 其中 α, β 不以 A 开头

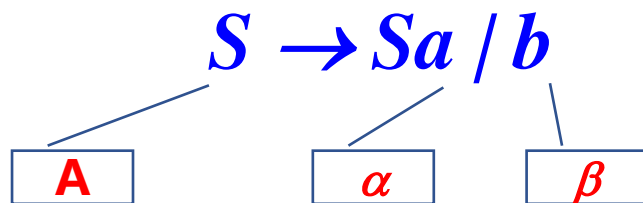
■ 串的特点 $\beta\alpha \dots \alpha$ ($A \Rightarrow^+ A\alpha$)

- 消除直接左递归

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \epsilon$$

- 考虑之前的文法





- 直接左递归

$A \rightarrow A\alpha | \beta$, 其中 α, β 不以 A 开头

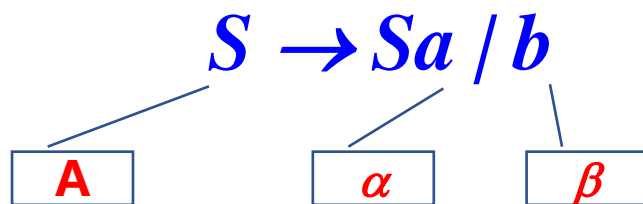
■ 串的特点 $\beta\alpha \dots \alpha$ ($A \Rightarrow^+ A\alpha$)

- 消除直接左递归

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' | \epsilon$$

- 考虑之前的文法



$$S \rightarrow bS'$$

$$S' \rightarrow aS' | \epsilon$$

baaaaaa推导:

$$S \Rightarrow bS' \Rightarrow baS' \Rightarrow baaS' \Rightarrow baaaS' \Rightarrow baaaaS' \Rightarrow baaaaaS'$$

输入缓冲区指针不停地移动



- 直接左递归

$A \rightarrow A\alpha \mid \beta$, 其中 α, β 不以 A 开头

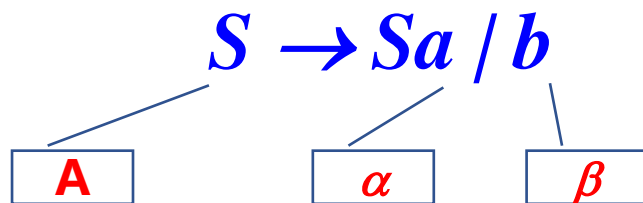
■ 串的特点 $\beta\alpha \dots \alpha$ ($A \Rightarrow^+ A\alpha$)

- 消除直接左递归

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A' \mid \epsilon$$

- 考虑之前的文法



$$S \rightarrow bS'$$

$$S' \rightarrow aS' \mid \epsilon$$

baaaaaa推导:

$$S \Rightarrow bS' \Rightarrow baS' \Rightarrow baaS' \Rightarrow baaaS' \Rightarrow baaaaS' \Rightarrow baaaaaS'$$

输入缓冲区指针不停地移动



- 例 算术表达文法

$$E \rightarrow E + T \mid T$$

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

$$F \rightarrow (E) \mid \text{id}$$



- 例 算术表达文法

$$E \rightarrow E + T \mid T$$

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

$$F \rightarrow (E) \mid \text{id}$$

$$(T + T \dots + T)$$

$$(F * F \dots * F)$$



• 例 算术表达文法

$$E \rightarrow E + T \mid T$$

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

$$F \rightarrow (E) \mid \text{id}$$

$$(T + T \dots + T)$$

$$(F * F \dots * F)$$

• 消除左递归后文法

$$E \rightarrow TE'$$

$$E' \rightarrow +TE' \mid \varepsilon$$

$$T \rightarrow FT'$$

$$T' \rightarrow *FT' \mid \varepsilon$$

$$F \rightarrow (E) \mid \text{id}$$

注明：红色部分代表了 α ，蓝色部分代表了 β



- 处理任意数量的A产生式

$$A \rightarrow A\alpha_1 \mid A\alpha_2 \mid \dots \mid A\alpha_m \mid \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$$

其中 β_i 都不以A开头

改为:

$$A \rightarrow \beta_1 A' \mid \beta_2 A' \mid \dots \mid \beta_n A'$$

$$A' \rightarrow \alpha_1 A' \mid \alpha_2 A' \mid \dots \mid \alpha_m A' \mid \varepsilon$$



消除间接左递归



- 非直接左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Sd \mid \varepsilon$$



- 非直接左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Sd \mid \varepsilon$$

- 先变换成直接左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow Aad \mid bd \mid \varepsilon$$

- 再消除左递归

$$S \rightarrow Aa \mid b$$

$$A \rightarrow bd A' \mid A'$$

$$A' \rightarrow adA' \mid \varepsilon$$



- **有左公因子的(left -factored)文法:**

- $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2$

- **提左公因子(left factoring)**

- 推后选择产生式的时机, 以便获取更多信息

$$A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \text{ 等价于}$$

$$A \rightarrow \alpha A'$$

$$A' \rightarrow \beta_1 \mid \beta_2$$



- 例 悬空 $else$ 的文法

$$\begin{aligned} stmt &\rightarrow \text{if } expr \text{ then } stmt \text{ else } stmt \\ &\quad | \text{if } expr \text{ then } stmt \\ &\quad | \text{other} \end{aligned}$$

提左因子

$$\begin{aligned} stmt &\rightarrow \text{if } expr \text{ then } stmt \text{ optional_else_part} \\ &\quad | \text{other} \\ \text{optional_else_part} &\rightarrow \text{else } stmt \\ &\quad | \varepsilon \end{aligned}$$

算法仍然二义!!!



递归下降的问题3



- **复杂的回溯→代价太高**
 - 非终结符有可能有多个产生式
 - 由于信息缺失，无法准确预测选择哪一个
 - 考虑到往往需要对多个非终结符进行推导展开，因此尝试的路径可能呈指数级爆炸
- **其分析过程类似于NFA**
- **问题：是否可以构造一个类似于DFA的分析方法？**

2023年秋季学期 《编译原理和技术》



一起努力 打造国产基础软硬件体系！

李 诚

国家高性能计算中心(合肥)、信息与计算机国家级实验教学示范中心

计算机科学与技术学院

2023年09月13日