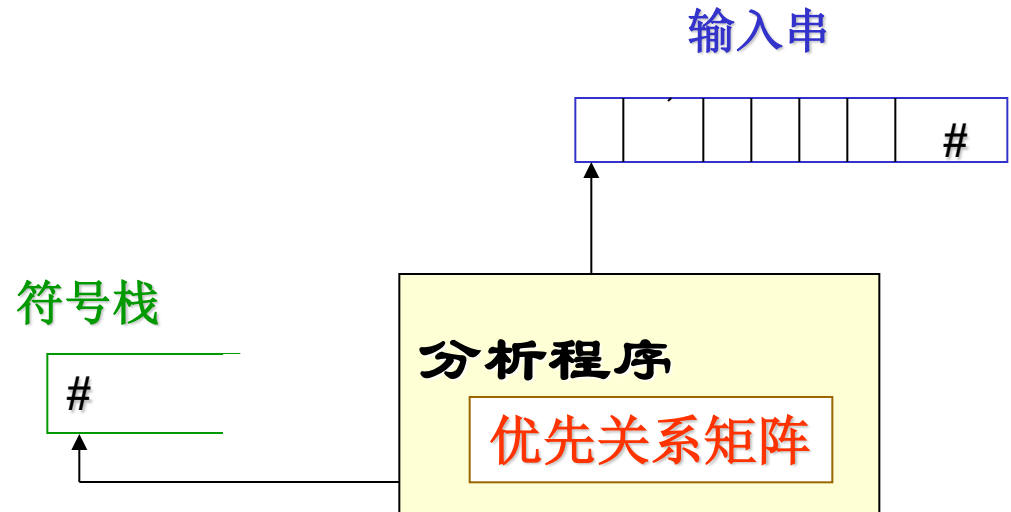


# 编译原理

- 第一章 编译程序概述
- 第二章 PL/0编译程序的实现
- 第三章 文法和语言
- 第四章 词法分析
- 第五章 自顶向下语法分析方法
- 第六章 自底向上优先分析方法
- 第七章 LR分析方法
- 第八章 语法制导翻译和中间代码生成
- 第九章 符号表
- 第一〇章 代码优化
- 第一一章 代码生成

# 自底向上优先分析

自底向上分析也称**移进归约分析**（是推导的逆过程）



**简单优先文法：** 确定所有符号优先关系的规则

**算符优先文法：** 确定算符优先关系的规则

$\text{firstVT}()$ ,  $\text{lastVT}()$

---

## 自下而上分析的关键问题： 如何确定可归约串？

- 简单优先分析法：寻找句柄(最左直接短语)
- 算符优先分析法：寻找最左素短语

## 二、简单优先分析法

### 1、优先关系的表示

$x = \cdot y$  表示 $x$ 与 $y$ 的优先关系相等

$x \cdot > y$  表示 $x$ 的优先性大于 $y$

$x < \cdot y$  表示 $x$ 的优先性小于 $y$

对任意两个文法符号 $X$ 、 $Y$ 按其在句型中可能会出现相邻关系来确定优先关系：

# 确定优先关系的规则

- (1)  $X =. Y$  当且仅当G中存在产生式A  
 $\rightarrow \dots XY \dots$  (在语法树的同一层)
- (2)  $X <. Y$  当且仅当G中存在产生式A  
 $\rightarrow \dots XB \dots$ , 且  $B \overset{+}{\Rightarrow} Y \dots$  (Y 在 X 的下一层)
- (3)  $X >. Y$  当且仅当G中存在产生式A  
 $\rightarrow \dots BD \dots$ , 且  $B \overset{+}{\Rightarrow} \dots X$  和  $D \overset{*}{\Rightarrow} Y \dots$  (X在 Y 的下一层或X比 Y先归约——规范归约/最左归约)

# 算符优先文法的定义

## 【算符文法定义】

设有一文法G，如果G中**没有**形如 $A \rightarrow \dots BC \dots$ 的产生式，其中B和C为**非终结符**，则称G为**算符文法** (或称OG文法)。

【要点】任何一个产生式中都不包含两个非终结符相邻的情况，就是算符文法。或：两个非终结符之间一定通过1个或多个终结符相连。

性质1：在算符文法中任何句型都不包含两个相邻的非终结符。

性质2：如果 $Ab$ 或 $(bA)$ 出现在算符文法的句型 $\gamma$ 中，其中 $A \in V_N$ ， $b \in V_T$ ，则 $\gamma$ 中任何含 $b$ 的短语必含有 $A$ 。

（含 $b$ 的短语必含 $A$ ，含 $A$ 的短语不一定含 $b$ ）

P.108 证明：反证法，如果 $b$ 不和 $A$ 一起规约。

## 【算符优先关系的定义】

设 $G$ 是一个算符文法， $a$ 和 $b$ 是任意两个终结符， $A$ ， $B$ ， $C$ 是非终结符，算符优先关系如下：

(1)  $a = .b$  当且仅当  $G$  中含有形如  $A \rightarrow \dots ab \dots$  或  $A \rightarrow \dots aBb \dots$  的产生式；

(2)  $a < .b$  当且仅当  $G$  中含有形如  $A \rightarrow \dots aB \dots$  的产生式，且  $B \xRightarrow{+} b \dots$  或  $B \xRightarrow{+} Cb \dots$ ；

(3)  $a > .b$  当且仅当  $G$  中含有形如  $A \rightarrow \dots Bb \dots$  的产生式，且  $B \xRightarrow{+} \dots a$  或  $B \xRightarrow{+} \dots aC$ 。

与简单优先关系区别：区分终结符与非终结符，非终结符忽略不计



### (3) 算符优先关系表

用表格形式来表示各终结符号的优先关系，这种表称为优先表。

构造优先关系表的方法：①按照定义来构造

②按关系图来构造

✓构造步骤：（根据算符优先关系的定义）

- 定义两个集合：firstVT集合lastVT集合。

$$\text{firstVT}(B) = \{ b | B \xRightarrow{+} b \dots \text{ 或 } B \xRightarrow{+} Cb \dots \}$$

$$\text{lastVT}(B) = \{ a | B \xRightarrow{+} \dots a \text{ 或 } B \xRightarrow{+} \dots aC \}$$

## 三种优先关系的计算:

a)  $=.$  关系:

$$\left. \begin{array}{l} A \rightarrow \dots ab \dots \\ A \rightarrow \dots aBb \dots \end{array} \right\} \text{则 } a =. b$$

b)  $<.$  关系:

对于每个非终结符B的 $\text{firstVT}(B)$ 有  
形如  $A \rightarrow \dots aB \dots$  中, 对每一个  
 $b \in \text{firstVT}(B)$ 。  $\left. \vphantom{\begin{array}{l} \text{对于每个非终结符B的} \\ \text{形如 } A \rightarrow \dots aB \dots \text{ 中, 对每一个} \\ \text{ } b \in \text{firstVT}(B) \end{array}} \right\} \text{则 } a <. b$

c)  $>.$ 关系:

每个非终结符B的 $\text{lastVT}(B)$ 有  
形如  $A \rightarrow \dots Bb \dots$  中, 对每一个  
 $a \in \text{lastVT}(B)$  } 则  $a >. b$

```
FOR 每条规则  $U \rightarrow x_1 x_2 \dots x_n$  DO
  FOR  $i := 1$  TO  $n-1$  DO
    BEGIN
      IF  $x_i$ 和 $x_{i+1}$ 均为终结符, THEN 置  $x_i \leftarrow x_{i+1}$ 
      IF  $i \leq n-2$ , 且 $x_i$ 和 $x_{i+2}$ 都为终结符号但
         $x_{i+1}$ 为非终结符号 THEN 置  $x_i \leftarrow x_{i+2}$ 
      IF  $x_i$ 为终结符号 $x_{i+1}$ 为非终结符号 THEN
        FOR  $\text{FIRSTVT}(x_{i+1})$ 中的每个 $b$  DO
          置 $x_i \leftarrow b$ 
      IF  $x_i$ 为非终结符号 $x_{i+1}$ 为终结符号 THEN
        FOR  $\text{LASTVT}(x_i)$ 中的每个 $a$  DO
          置 $a >. x_{i+1}$ 
    END
```

## 构造FIRSTVT(U)的算法

1)若有规则 $U \rightarrow b...$ 或 $U \rightarrow Vb...$ (存在 $U \xRightarrow{+} b...$ 或 $U \xRightarrow{+} Vb...$ )  
则 $b \in \text{FIRSTVT}(U)$

2)若有规则 $U \rightarrow V...$ 且 $b \in \text{FIRSTVT}(V)$ , 则 $b \in \text{FIRSTVT}(U)$

说明:因为 $V \xRightarrow{+} b...$ 或 $V \xRightarrow{+} Wb...$ ,所以有 $U \Rightarrow V... \xRightarrow{+} b...$ 或  
 $U \Rightarrow V... \xRightarrow{+} Wb...$

具体方法如下:

设一个栈S和一个二维布尔数组F

$F[U,b]=\text{TRUE}$  iff  $b \in \text{FIRSTVT}(U)$

**PROCEDURE INSERT(U,b)**

**IF NOT F[U,b] THEN**

**BEGIN**

$F[U,b] := \text{TRUE}$

把(U,b)推进S栈 /\*  $b \in \text{FIRSTVT}(U)$  \*/

**END**

**BEGIN {main}**

**FOR** 每个非终结符号U和终结符b **DO**

$F[U,b] := \text{FALSE}$  /\*赋初值\*/

**FOR** 每个形如 $U \rightarrow b \dots$ 或 $U \rightarrow Vb \dots$ 的规则 **DO**

**INSERT(U,b)**

---

```
WHILE S栈非空 DO
    BEIGN
        把S栈的顶项弹出,记为 (V,b)/*  $b \in \text{FIRSTVT}(V)$  */
        FOR 每条形如  $U \rightarrow V \dots$  的规则 DO
            INSTER(U,b); /*  $b \in \text{FIRSTVT}(U)$  */
        END OF WHILE
    END
```

上述算法的工作结果是得到一个二维的布尔数组F,从F可以得到任何非终结符号U的FIRSTVT

$$\text{FIRSTVT}(U) = \{b | F[U,b] = \text{TRUE}\}$$

## 构造LASTVT(U)的算法

1.若有规则 $U \rightarrow \dots a$ 或 $U \rightarrow \dots aV$ ,则 $a \in \text{LASTVT}(U)$

2.若有规则 $U \rightarrow \dots V$ ,且 $a \in \text{LASTVT}(V)$ 则 $a \in \text{LASTVT}(U)$

设一个栈ST, 和一个布尔数组B

**PROCEDURE INSERT(U,a)**

**IF NOT B[U,a] THEN**

**BEGIN**

**B[U,a]  $\rightarrow$  TRUE; 把(U,a)推进ST栈;**

**END;**

---

**BEGIN**

**FOR 每个非终结符号U和终结符号a DO**

**B[U,a]:=FALSE;**

**FOR 每个形如 $U \rightarrow \dots a$ 或 $U \rightarrow \dots aV$ 的规则 DO**

**INSERT (U,a);**

**WHILE ST栈非空 DO**

**BEGIN**

**把ST栈的栈顶弹出,记为(V,a);**

**FOR 每条形如 $U \rightarrow \dots V$ 的规则 DO**

**INSERT(U,a);**

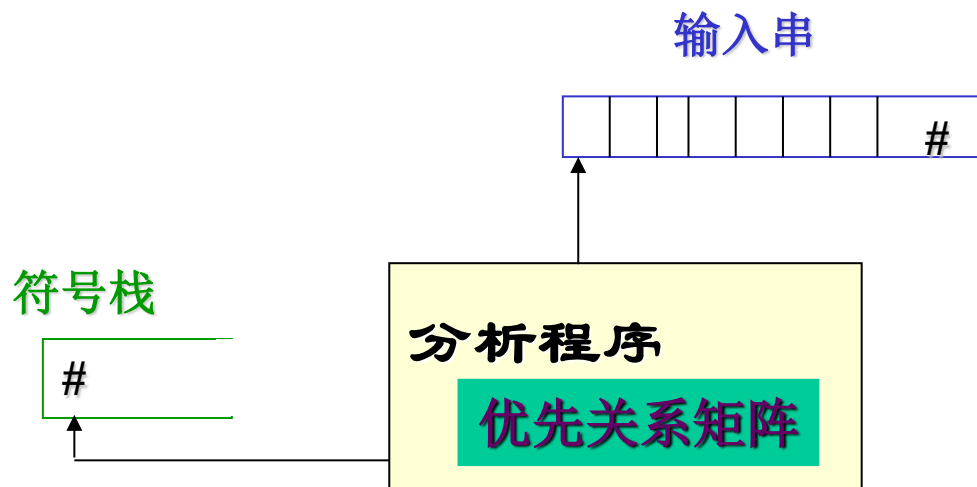
**END OF WHILE;**

**END;**



# 算符优先分析法的实现:

详见P117 图6.8



当栈内终结符的优先级  $\leq$  栈外的终结符的优先级时，移进；栈内终结符的优先级  $>$  栈外的终结符的优先级时，归约。表明找到了素短语的尾，再往前找其头，并进行规约。

# 算符优先分析句型的性质

- $\#N_1a_1N_2a_2\ldots N_na_nN_{n+1}\#$ 为句型
  - $N_i$ 为非终结符或空,  $a_i$ 为终结符
  - 若 $a_i\ldots N_ja_j$ 属于句柄, 则 $N_i$ 和 $N_{j+1}$ 也在句柄中
    - $a_{i-1} < \cdot a_i$
    - $a_i = \cdot a_{i+1} \ldots = \cdot a_j$
    - $a_j > \cdot a_{j+1}$
- 如果 $aNb$ (或 $ab$ )出现在句型 $r$ 中
  - $a < \cdot b$ , 则在 $r$ 中必含有 $b$ 而不含 $a$ 的短语存在
  - $a > b$ , 则在 $r$ 中必含有 $a$ 而不含 $b$ 的短语存在
  - $a = \cdot b$ , 则在 $r$ 中含有 $a$ 的短语必含有 $b$

# 实例比较

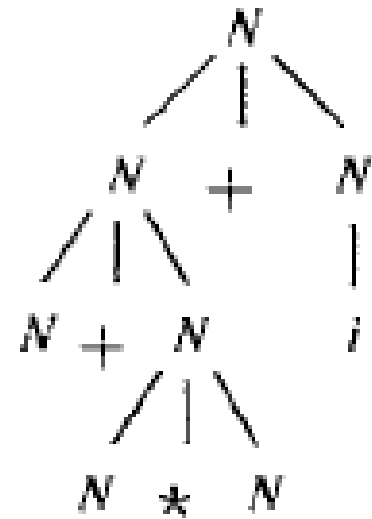
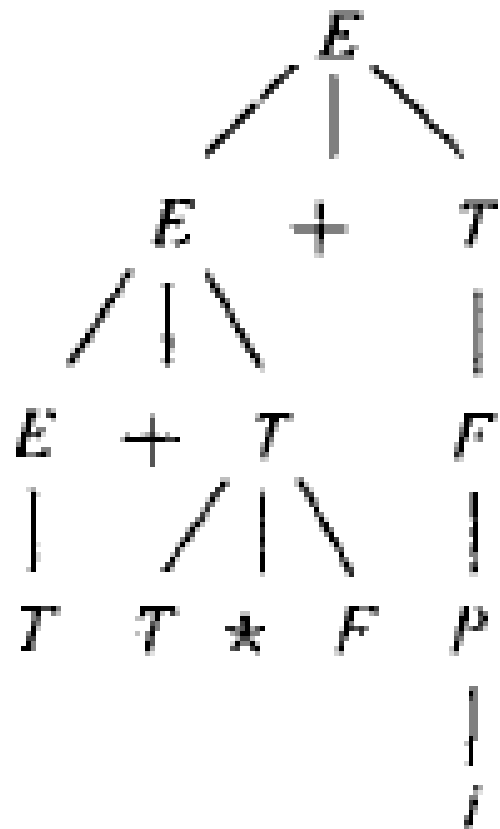
- 算符优先归约（P115表6.8）
- 规范归约（P115表6.7）
- 前者（1）去掉单非终结符的归约，（2）并且在归约时不考虑非终结符的名字，因此得到的不是真正的语法树，而是语法树的框架。（比较P116 图6.7与图6.6）
- 算符优先分析法的可归约串不是句柄而是最左素短语。
- P116 “算符优先分析的关键是如何找最左素短语……”

表 6.7 对输入串  $i+i\#$  的规范归约过程

步骤	栈	剩余输入串	句柄	归约用产生式
(1)	#	$i+i\#$		
(2)	# $i$	$+i\#$	$i$	$P \rightarrow i$
(3)	# $P$	$+i\#$	$P$	$F \rightarrow P$
(4)	# $F$	$+i\#$	$F$	$T \rightarrow F$
(5)	# $T$	$+i\#$	$T$	$E \rightarrow T$
(6)	# $E$	$+i\#$		
(7)	# $E+$	$i\#$		
(8)	# $E+i$	#	$i$	$P \rightarrow i$
(9)	# $E+P$	#	$P$	$F \rightarrow P$
(10)	# $E+F$	#	$F$	$T \rightarrow F$
(11)	# $E+T$	#	$E+T$	$E \rightarrow E+T$
(12)	# $E$	#		接受

表 6.8 对输入串  $i+i\#$  的算符优先归约过程

步骤	栈	优先关系	当前符号	剩余输入串	移进或归约
(1)	#	$<$	$i$	$+i\#$	移进
(2)	# $i$	$>$	$+$	$i\#$	归约
(3)	# $F$	$<$	$+$	$i\#$	移进
(4)	# $F+$	$<$	$i$	#	移进
(5)	# $F+i$	$>$	#		归约
(6)	# $F+F$	$>$	#		归约
(7)	# $F$	$=$	#		接受



## 五、最左素短语

定义：设有文法 $G[S]$ ，其句型的素短语是一个短语，它至少包含一个终结符，并除自身外不包含其它素短语，最左边的素短语称最左素短语。

与句柄的区别：至少包含一个终结符。（从而去掉了单非终结符的归约）

例：文法 $G[E]$ :  $E \rightarrow E+T | T$

$$T \rightarrow T * F | F$$
$$F \rightarrow P \uparrow F | P$$
$$P \rightarrow (E) | i$$

句型 $\#T+T*F+i\#$ 的语法树如下：

根据语法树可知：

句型 $\#T+T*F+i\#$ 的短语有：

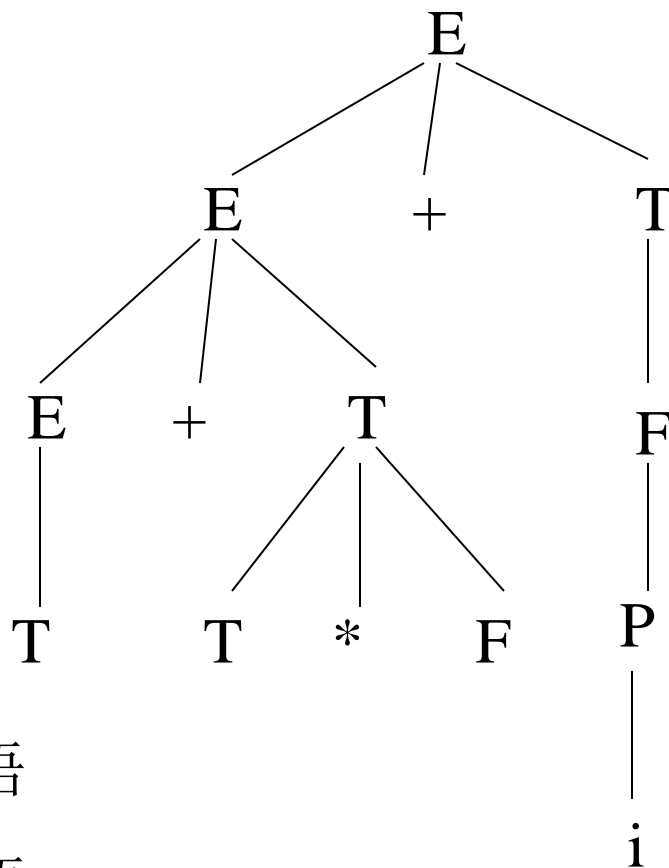
$T$  — 相对非终结符 $E$ 的短语

$T*F$  — 相对非终结符 $T$ 的短语

$T+T*F$  — 相对非终结符 $E$ 的短语

$i$  — 相对非终结符 $P$ 、 $F$ 、 $T$ 的短语

$T+T*F+i$  — 相对非终结符 $E$ 的短语



根据素短语的定义可知：

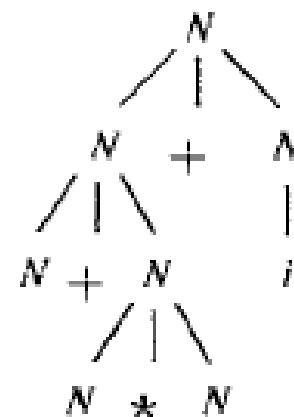
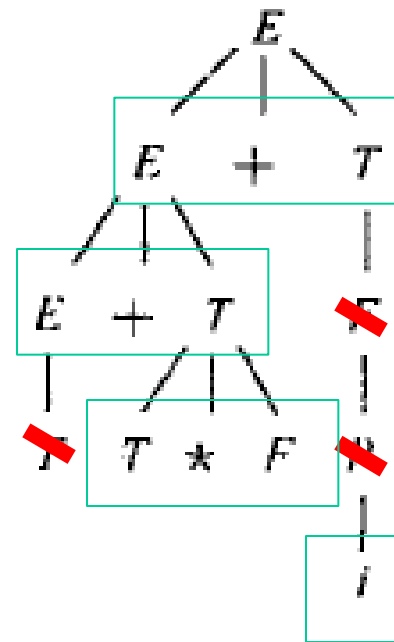
$i$ 和 $T * F$ 为素短语。

其中： $T + T * F$  (含其他 $T * F$ 素短语)和

$T + T * F + i$  不是素短语。

$T * F$ 为最左素短语。

$T$ 为句柄——最左直接短语





算符优先分析法的关键：

如何确定当前句型的最左素短语？

算符文法的任一句型形式为

$$\#N_1a_1N_2a_2\ldots N_na_nN_{n+1}\#$$

$$(N_i \in V_N \text{ 或 } N_i = \epsilon, a_i \in V_T)$$

**定理：** 一个OPG句型的最左素短语是满足下列条件的最左子串： $a_{j-1}N_ja_j\ldots N_ia_iN_{i+1}a_{i+1}$

其中  $a_{j-1} < .a_j$

$$a_j = .a_{j+1}, a_{j+1} = .a_{j+2}, \ldots, a_{i-2} = .a_{i-1}, a_{i-1} = .a_i$$

$$a_i > .a_{i+1}$$

根据该定理,要找句型的最左素短语就是要找满足上述条件的最左子串。

## 算符优先分析法的实现:

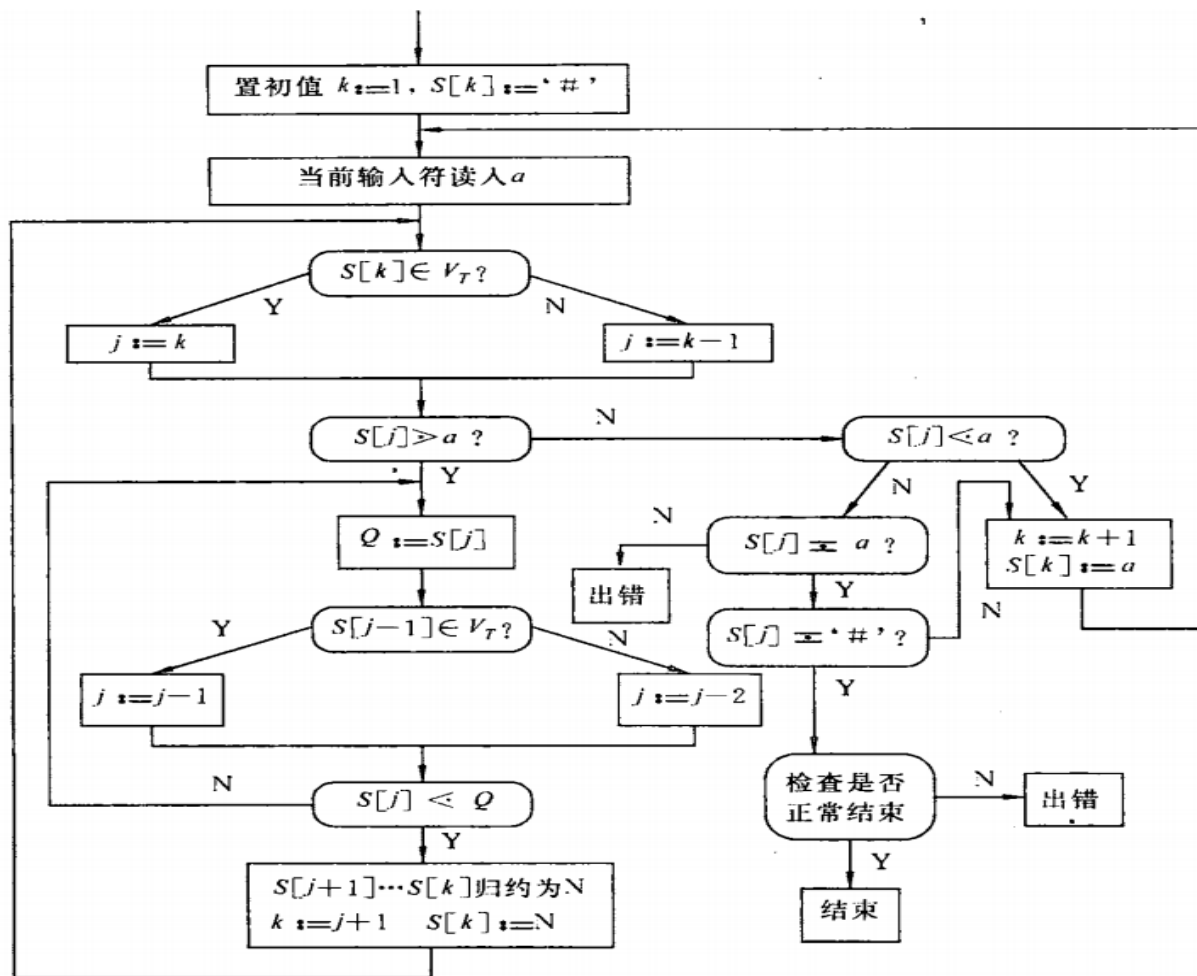


图 6.8 算符优先分析归约过程流程图

## 六、优先函数：

算符优先关系表示法 { 矩阵表示法(空间大)  
优先函数法

### 1、优先函数的定义：

当  $a =. b$ ，令  $f(a)=g(b)$

当  $a < . b$ ，令  $f(a)<g(b)$

当  $a > . b$ ，令  $f(a)>g(b)$

$f(a)$ 、 $g(b)$ 为优先函数，  
函数值用整数表示

## 2、优先函数的构造

### 构造规则

a) 对终结符  $a \in V_T$  (包括#号) 令  $f(a)=g(a)=1$  (初始化)

b) 如果  $a > .b$ , 而  $f(a) \leq g(b)$ , 则令  $f(a)=g(b)+1$

c) 如果  $a < .b$ , 而  $f(a) \geq g(b)$ , 则令  $g(b)=f(a)+1$

d) 如果  $a = .b$ , 而  $f(a) \neq g(b)$  则令  $\min\{f(a), g(b)\} = \max\{f(a), g(b)\} + 1$

重复b)~d)过程, 直到收敛。若重复过程中有一个值  $> 2n$  ( $n$ 为终结符个数), 则该文法不存在算符优先函数。

例1：有优先表如下，构造优先函数。

	+	*	↑
+	>.	<.	<.
*	>.	>.	<.
↑	>.	>.	<.

【解】(1) 赋初值

	+	*	↑
f	1	1	1
g	1	1	1

(2) 对 $a > .b$ 关系有:

$+ > . + \quad f(+)=g(+)+1$

	+	*	↑
f	2	1	1
g	1	1	1

$* > . + \quad * > . *$

	+	*	↑
f	2	2	1
g	1	1	1

	+	*	↑
+	$> .$	$< .$	$< .$
*	$> .$	$> .$	$< .$
↑	$> .$	$> .$	$< .$

$\uparrow > . + \quad \uparrow > . *$

	+	*	↑
f	2	2	2
g	1	1	1

(3) 对a <. b关系有

+ <. \*  $g(*) = f(+) + 1$

+ <. ↑  $g(↑) = f(+) + 1$

	+	*	↑
f	2	2	2
g	1	3	3

\* <. ↑    ↑ <. ↑

	+	*	↑
f	2	2	2
g	1	3	3

	+	*	↑
+		<.	<.
*	>.	>.	<.
↑	>.	>.	<.

(4) 对a =. b没有

重复过程(2)、(3)

#### (4) 对a>.b关系

+ >. +

	+	*	↑
f	2	2	2
g	1	3	3

\* >. +    \* >. \* f(\*)=g(\*)+1

	+	*	↑
f	2	4	2
g	1	3	3

	+	*	↑
+	>.	<.	<.
*	>.	>.	<.
↑	>.	>.	<.

↑ >. +    ↑ >. \* f(↑)=g(\*)+1

	+	*	↑
f	2	4	4
g	1	3	3



## (5) 对a <.b关系

+ <. \*    + <. ↑

	+	*	↑
f	2	4	4
g	1	3	3

	+	*	↑
+	>.	<.	<.
*	>.	>.	<.
↑	>.	>.	<.

\* <. ↑    ↑ <. ↑     $g(\uparrow) = f(\uparrow) + 1$

	+	*	↑
f	2	4	4
g	1	3	5

重复以上过程得：

	+	*	↑
f	2	4	4
g	1	3	5

所以存在优先函数为：

	+	*	↑
f	2	4	4
g	1	3	5

	+	*	↑
+	>.	<.	<.
*	>.	>.	<.
↑	>.	>.	<.

当  $a =. b$ ，令  $f(a)=g(b)$

当  $a <. b$ ，令  $f(a)<g(b)$

当  $a >. b$ ，令  $f(a)>g(b)$

结果同上步，因此收敛

例2：已知优先关系表，构造优先函数。

	+	*	↑	i	(	)	#
+	>.	<.	<.	<.	<.	>.	>.
*	>.	>.	<.	<.	<.	>.	>.
↑	>.	>.	<.	<.	<.	>.	>.
i	>.	>.	>.			>.	>.
(	<.	<.	<.	<.	<.	≤	
)	>.	>.	>.			>.	>.
#	<.	<.	<.	<.	<.		≤

# 构造过程

## (1) 初始化

	+	*	↑	i	(	)	#
f	1	1	1	1	1	1	1
g	1	1	1	1	1	1	1

## (2) 对a>.b关系

+>.+    +>.)    +>.#

	+	*	↑	i	(	)	#
f	2	1	1	1	1	1	1
g	1	1	1	1	1	1	1

$* > . +$     $* > . *$     $* > . )$     $* > . \#$

	+	*	↑	i	(	)	#
f	2	2	1	1	1	1	1
g	1	1	1	1	1	1	1

$\uparrow > . +$     $\uparrow > . *$     $\uparrow > . )$     $\uparrow > . \#$

	+	*	↑	i	(	)	#
f	2	2	2	1	1	1	1
g	1	1	1	1	1	1	1

$i > . +$     $i > . *$     $i > . \uparrow$     $i > . )$     $i > . \#$

	+	*	$\uparrow$	i	(	)	#
f	2	2	2	2	1	1	1
g	1	1	1	1	1	1	1

$) > . +$     $) > . *$     $) > . \uparrow$     $) > . )$     $)$

$> . \#$

	+	*	$\uparrow$	i	(	)	#
f	2	2	2	2	1	2	1
g	1	1	1	1	1	1	1

### (3) 对a<.b关系

+<.\*    +<. $\uparrow$     +<.*i*    +<.(

	+	*	$\uparrow$	i	(	)	#
f	2	2	2	2	1	2	1
g	1	3	3	3	3	1	1

\*<. $\uparrow$     \*<.*i*    \*<.(

	+	*	$\uparrow$	i	(	)	#
f	2	2	2	2	1	2	1
g	1	3	3	3	3	1	1

↑ < . ↑    ↑ < . i    ↑ < . (

	+	*	↑	i	(	)	#
f	2	2	2	2	1	2	1
g	1	3	3	3	3	1	1

( < . +    ( < . \*    ( < . ↑    ( < . i    ( < . (

	+	*	↑	i	(	)	#
f	2	2	2	2	1	2	1
g	2	3	3	3	3	1	1



#<.+ #<.\* #<. $\uparrow$  #<.i #<.(

	+	*	$\uparrow$	i	(	)	#
f	2	2	2	2	1	2	1
g	2	3	3	3	3	1	1

(4) 对 $a \leq b$ 关系

( $\leq$ ) #  $\leq$  #

	+	*	$\uparrow$	i	(	)	#
f	2	2	2	2	1	2	1
g	2	3	3	3	3	1	1

第二次重复以上过程(2,3,4步)

对于a>.b关系

	+	*	↑	i	(	)	#
f	3	4	4	4	1	4	1
g	2	3	3	3	3	1	1

对于a<.b关系

	+	*	↑	i	(	)	#
f	3	4	4	4	1	4	1
g	2	4	5	5	5	1	1

对于 $a \leq b$ 关系

	+	*	↑	i	(	)	#
f	3	4	4	4	1	4	1
g	2	4	5	5	5	1	1

第三次重复以上过程

对于 $a > b$ 关系

	+	*	↑	i	(	)	#
f	3	5	5	6	1	4	1
g	2	4	5	5	5	1	1

对于 $a < b$ 关系

	+	*	↑	i	(	)	#
f	3	5	5	6	1	4	1
g	2	4	6	6	6	1	1

对于 $a \leq b$ 关系

	+	*	↑	i	(	)	#
f	3	5	5	6	1	4	1
g	2	4	6	6	6	1	1

## 第四次重复过程

对于a  $\dot{>}$  b关系

	+	*	↑	i	(	)	#
f	3	5	5	7	1	7	1
g	2	4	6	6	6	1	1

对于a  $\dot{<}$  b关系

	+	*	↑	i	(	)	#
f	3	5	5	7	1	7	1
g	2	4	6	6	6	1	1

对于 $a \leq b$ 关系

	+	*	↑	i	(	)	#
f	3	5	5	7	1	7	1
g	2	4	6	6	6	1	1

第五次重复过程

	+	*	↑	i	(	)	#
f	3	5	5	7	1	7	1
g	2	4	6	6	6	1	1

第五次结果同第四次，表示收敛了。

因此优先函数为：

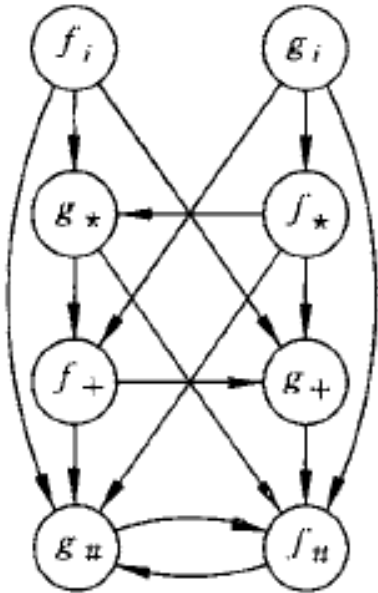
	+	*	↑	i	(	)	#
f	3	5	5	7	1	7	1
g	2	4	6	6	6	1	1

# 关系图法构建优先函数

- (1) 任一终结符 $a$ ，建立两个节点 $f_a$ 和 $g_a$
- (2)  $a > .a | a = .a$ ,  $f_a \rightarrow g_a$
- (3)  $a < .a | a = .a$ ,  $f_a \leftarrow g_a$
- (4) 优先函数按优先关系矩阵检查一遍，是否一致，否则存在3个或以上节点的回路。



# 例题



	<i>i</i>	*	+	#
<i>i</i>		>	>	>
*	<	>	>	>
+	<	<	>	>
#	<	<	<	=

表 6.11 优先函数关系表

	<i>i</i>	*	+	#
<i>f</i>	6	6	4	2
<i>g</i>	7	5	3	2

表 6.12 优先关系矩阵

	$a$	$b$
$a$	$\overline{\cdot}$	$\cdot >$
$b$	$\overline{\cdot}$	$\overline{\cdot}$

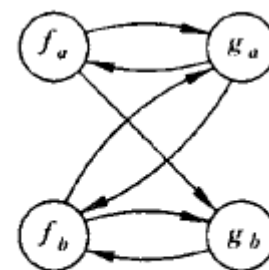


图 6.10 优先关系图

表 6.13 优先函数表

	$a$	$b$
$f$	4	4
$g$	4	4

# 局限性

$S \rightarrow S; D \mid D$

$D \rightarrow D(T) \mid H$

$H \rightarrow a \mid (S)$

$T \rightarrow T + S \mid S$

	:	(	)	a	+	#
:	>	<	>	<	>	>
(	<	<	=	<	<	
)	>	>	>		>	>
a	>	>	>		>	>
+	<	<	>	<	>	
#	<	<		<		=

- $(a+a)\#$
- 被成功规约
- 却不是合法的句子

# 例题分析

例题 1 已知布尔表达式文法  $G[B]$  为： $B \rightarrow B \circ T \mid T$   
 $T \rightarrow T \alpha F \mid F$   
 $F \rightarrow nF \mid (B) \mid t \mid f$

1.  $G[B]$  是算符优先文法吗？
2. 若  $G[B]$  是算符优先文法，请给出输入串  $ntofat\#$  的分析过程。

---

# 作业

- 计算 例6.1 (P102) 文法的简单优先关系矩阵
- P122 练习 1、2