

# 编译原理

北方工业大学信息学院  
School of Information Science and Technology,  
North China University of Technology  
束劼  
[shujie@ncut.edu.cn](mailto:shujie@ncut.edu.cn)  
瀚学楼1122, 88801615

## 第五章 语法分析 -自下而上分析

## 第五章 语法分析-自下而上分析

- 本章目录
  - 5.1 自下而上分析基本问题
  - 5.2 算符优先分析
  - 5.3 LR分析法
  - 5.4 语法分析器的自动产生工具YACC

3

## 第五章 语法分析-自下而上分析

- 大纲要求
  - 掌握：归约，规范归约，算符优先分析法（算符优先文法、优先表的构造、算符优先分析算法、优先函数的构造）。
  - 理解：符号栈的使用方法。
  - 了解：自下而上语法分析的基本原理和工作方法，语法分析器的自动产生工具YACC的基本作用。

4

## 5.1 自下而上分析基本问题

### 5.1 自下而上分析基本问题

- 5.1.1 归约Reductions
- 5.1.2 规范归约Handle Pruning
- 5.1.3 符号栈的使用与语法树的表示 Shift-Reduce Parsing

## 5.1.1 归约

## 5.1.1 归约

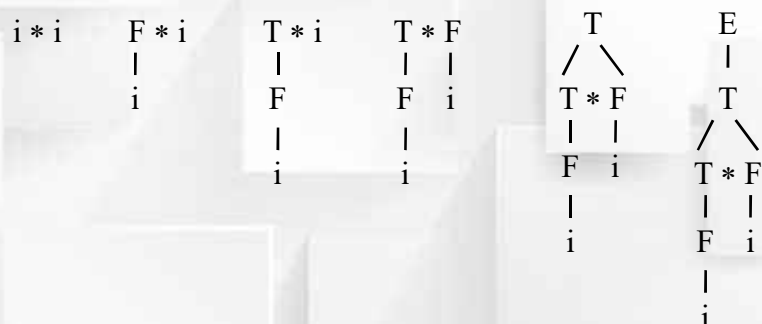
- 自下而上分析

从输入串开始，逐步进行“归约”，直到文法的开始符号。即从树末端开始，构造语法树。

所谓归约，是指根据文法的产生式规则，把产生式的右部替换成左部符号。

### 5.1.1 归约

例如：对输入字符串 $i * i$ 按语法 $G$ 归约  $E \rightarrow E + T \mid T$

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


9

### 5.1.1 归约

- 归约的目的  
是建立推导的逆过程

$$E \Rightarrow T \Rightarrow T * F \Rightarrow T * i \Rightarrow F * i \Rightarrow i * i$$

### 最右推导

10

## 5.1.1 归约

- 移进-归约的方法 例：设文法 $G(S)$ :

(1)  $S \rightarrow aAcBe$  (2)  $A \rightarrow b$

(3)  $A \rightarrow Ab$  (4)  $B \rightarrow d$

试对 $abbcd$ 进行“移进-归约”分析。

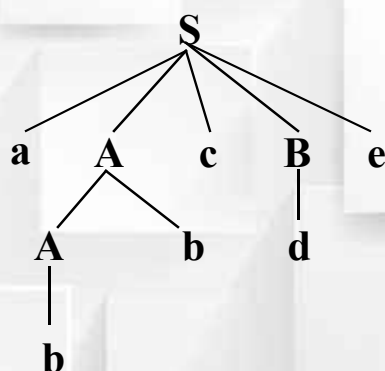
步骤:	1	2	3	4	5	6	7	8	9	10
动作:	进a	进b	归(2)	进b	归(3)	进c	进d	归(4)	进e	归(1)
									e	
							d	B	B	
				b		c	c	c	c	
	b	A	A	A	A	A	A	A	A	
	a	a	a	a	a	a	a	a	a	S

如果选文法(2)会如何?

11

## 5.1.1 归约

- 移进-归约的方法



12

### 5.1.1 归约

- 移进-归约的问题

是“可规约串”问题。对可规约串的刻画不同，分析的方法也不同。

- 解决方法

“直接短语”和“句柄(Handle)”两种可规约串刻画方法。

13

### 5.1.2 规范归约

## 5.1.2 规范归约

- 定义：令G是一个文法，S是文法的开始符号，假定 $\alpha\beta\delta$ 是文法G的一个句型，如果有

$$\begin{array}{ccc} S \xRightarrow{*} \alpha A \delta \xRightarrow{+} \alpha \beta \delta & A \xRightarrow{+} \beta \\ \text{rm} & \text{rm} & \end{array}$$

则称 $\beta$ 是句型 $\alpha\beta\delta$ 相对于非终结符A的**短语**。 $\delta$ 是终结符

$\beta$ 是句型 $\alpha\beta\delta$ 相对于规则 $A \rightarrow \beta$ 的**直接短语**。一个句型的最左直接短语称为该句型的**句柄**。

15

## 5.1.2 规范归约

- 短语、直接短语、句柄

考虑文法G(E)      短语的关键有两点，一是可以由某个非终结符推导出，二是可以由开始符推导出

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

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

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

和句型 $i_1 * i_2 + i_3$

$$E \Rightarrow T \Rightarrow F \Rightarrow i$$

$i_1, i_2, i_3$ 是短语

$$E \Rightarrow T \Rightarrow T * F \Rightarrow$$

$$T * i \Rightarrow F * i \Rightarrow i * i$$

$i_1 * i_2$ 是短语

**$i_2 + i_3$  是短语吗?**

$$E \Rightarrow E + T \Rightarrow E + F$$

$$\Rightarrow E + i \Rightarrow T + i \Rightarrow$$

$$T * F + i \Rightarrow T * i + i$$

$$\Rightarrow F * i + i \Rightarrow i * i + i$$

$i_1 * i_2 + i_3$ 是短语

16



## 5.1.2 规范归约

### • 短语、直接短语、句柄

考虑文法G(E)

$E \rightarrow T \mid E+T$

$T \rightarrow F \mid T * F$

$F \rightarrow (E) \mid i$

和句型  $i_1 * i_2 + i_3$

直接短语属于短语，直接短语是由非终结符直接推导的短语

直接短语:  $i_1, i_2, i_3$

句柄:  $i_1$ (最左直接短语)

$E \Rightarrow T \Rightarrow F \Rightarrow i$

$F \Rightarrow i$ 中间没有其他推导  $F \rightarrow i$

$E \Rightarrow T \Rightarrow T * F \Rightarrow$

$T * i \Rightarrow F * i \Rightarrow i * i$

$T * F \xRightarrow{+} i * i$ 中间有其他推导，没有直接产生式可以得到  $A \rightarrow i * i$

17

## 5.1.2 规范归约

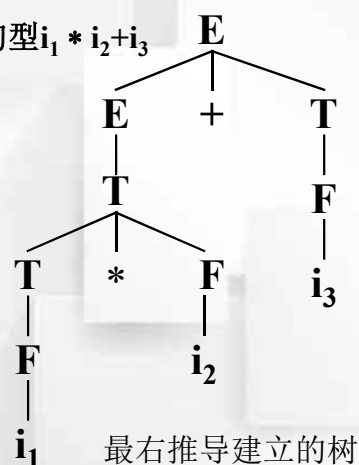
### • 短语、直接短语、句柄 句型 $i_1 * i_2 + i_3$

在一个句型对应的语法树中，以某非终结符为根的两代以上的子树的所有末端结点从左到右排列就是相对于该非终结符的一个短语，如果子树只有两代，则该短语就是直接短语。

短语:  $i_1, i_2, i_3, i_1 * i_2, i_1 * i_2 + i_3$

直接短语:  $i_1, i_2, i_3$

句柄:  $i_1$ (最左直接短语)



18

## 5.1.2 规范归约

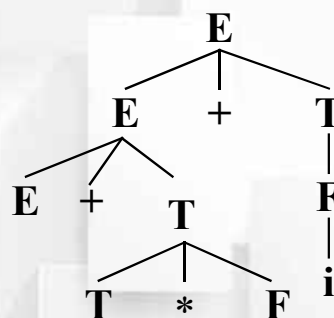
- 短语、直接短语、句柄

句型  $E + T * F + i$

短语:  $E + T * F + i$ ,  $E + T * F$ ,  $T * F$ ,  $i$

直接短语:  $T * F$ ,  $i$

句柄:  $T * F$



如何使用句柄帮助归约?

19

## 5.1.2 规范归约

- 句柄归约

句型 归约规则

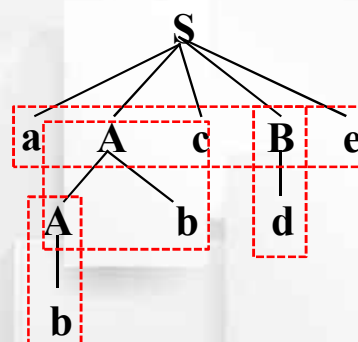
abbcde (2)  $A \rightarrow b$

aAbcde (3)  $A \rightarrow Ab$

aAcde (4)  $B \rightarrow d$

aAcBe (1)  $S \rightarrow aAcBe$

S



句柄是最左直接短语

20

## 5.1.2 规范归约

### • 规范归约总结

假定 $\alpha$ 是文法G的一个句子，我们称序列

$$\alpha_n, \alpha_{n-1}, \dots, \alpha_0$$

是 $\alpha$ 的一个**规范归约**，如果此序列满足：

- (1)  $\alpha_n = \alpha$
- (2)  $\alpha_0$ 为文法的开始符号，即 $\alpha_0 = S$
- (3) 对任何 $i$ ,  $0 \leq i \leq n$ ,  $\alpha_{i-1}$ 是从 $\alpha_i$ 经把句柄替换成为相应产生式左部符号而得到的。

21

## 5.1.2 规范归约

### • 规范归约总结

句型      归约规则

abbcde (2)  $A \rightarrow b$

aAbcde (3)  $A \rightarrow Ab$

aAcde (4)  $B \rightarrow d$

aAcBe (1)  $S \rightarrow aAcBe$

S

**规范归约**是一个最右推导的**逆过程**，也称**最左归约**

**规范推导**是**最右推导**

最右推导  $S \Rightarrow aAcBe \Rightarrow aAcde \Rightarrow aAbcde \Rightarrow abbcde$

最左推导  $S \Rightarrow aAcBe \Rightarrow aAbcBe \Rightarrow abbcBe \Rightarrow abbcde$

22

### 5.1.3 符号栈的使用与语法树的表示

### 5.1.3 符号栈的使用与语法树的表示

- 规范归约的实际操作

采用栈作为语法分析的基本数据结构

<u>符号栈</u>	<u>输入串</u>
栈底符号 #	w#
...	...
#S	# 栈底符号

语法分析对符号栈的使用有四类操作：“移进”、“归约”、“接受”和“出错处理”。

任何可规约串的出现必须在栈顶。

5.1.3 符号栈的使用与语法树的表示

• 规范归约的实际操作

例：考虑文法G(E):

$E \rightarrow T \mid E+T$   
 $T \rightarrow F \mid T * F$   
 $F \rightarrow (E) \mid i$

输入串为 $i_1 * i_2 + i_3$ ，写出规范归约步骤。

规范归约是个最右推导的逆过程，也称最左归约

规范推导是最右推导

5.1.3 符号栈的使用与语法树的表示

• 规范归约的实际操作 例：考虑文法G(E):

句柄都在栈顶

$E \rightarrow T \mid E+T$   
 $T \rightarrow F \mid T * F$   
 $F \rightarrow (E) \mid i$   
输入串为 $i_1 * i_2 + i_3$ ，写出分析步骤。

步骤	符号栈	输入串	动作
0	#	$i_1 * i_2 + i_3 \#$	预备
1	$\#i_1$	$*i_2 + i_3 \#$	移进
2	$\#F$	$*i_2 + i_3 \#$	归约，用 $F \rightarrow i$
3	$\#T$	$*i_2 + i_3 \#$	归约，用 $T \rightarrow F$
4	$\#T*$	$i_2 + i_3 \#$	移进

## 5.1.3 符号栈的使用与语法树的表示

- 规范归约的实际操作 例：考虑文法G(E):

句柄都在栈顶

 $E \rightarrow T \mid E+T$  $T \rightarrow F \mid T^*F$  $F \rightarrow (E) \mid i$ 输入串为 $i_1*i_2+i_3$ ，写出分析步骤。

步骤	符号栈	输入串	动作
4	#T*	$i_2+i_3\#$	移进
5	#T* $i_2$	$+i_3\#$	移进
6	#T*F	$+i_3\#$	归约，用 $F \rightarrow i$
7	#T	$+i_3\#$	归约，用 $T \rightarrow T^*F$
8	#E	$+i_3\#$	归约，用 $E \rightarrow T$
9	#E+	$i_3\#$	移进

最左归约

27

## 5.1.3 符号栈的使用与语法树的表示

- 规范归约的实际操作 例：考虑文法G(E):

句柄都在栈顶

 $E \rightarrow T \mid E+T$  $T \rightarrow F \mid T^*F$  $F \rightarrow (E) \mid i$ 输入串为 $i_1*i_2+i_3$ ，写出分析步骤。

步骤	符号栈	输入串	动作
9	#E+	$i_3\#$	移进
10	#E+ $i_3$	#	移进
11	#E+F	#	归约，用 $F \rightarrow i$
12	#E+T	#	归约，用 $T \rightarrow F$
13	#E	#	归约，用 $E \rightarrow E+T$
14	#E	#	接受

28

## 5.2 算符优先分析

## 5.2 算符优先分析

- 5.2.1 算符优先文法及优先表构造
- 5.2.2 算符优先分析算法
- 5.2.3 优先函数
- 5.2.4 算符优先分析中的出错处理

## 5.2.1 算符优先文法 及优先表构造

### 5.2.1算符优先文法及优先表构造

- 算符优先文法及优先表构造

有的文法形成的句子有几种不同的规范规约方式

**归约即计算表达式的值。**归约顺序不同，则计算的顺序也不同，结果也不一样。

如果规定算符的优先次序，并按这种规定进行归约，则归约过程是**唯一**的。

**如何使得计算过程唯一？**



## 5.2.1 算符优先文法及优先表构造

- 算符优先文法及优先表构造

四则运算的优先规则：

先乘除后加减，同级从左到右

考虑二义文法 $G(E)$ ：

$G(E): E \rightarrow i \mid E+E \mid E-E \mid E * E \mid E/E \mid (E)$

句子 $i+i*(i+i)$

33

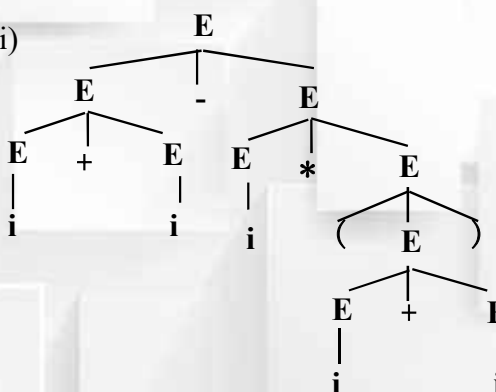
## 5.2.1 算符优先文法及优先表构造

- 算符优先文法及优先表构造

考虑二义文法 $G(E): E \rightarrow i \mid E+E \mid E-E \mid E * E \mid E/E \mid (E)$

句子 $i+i*(i+i)$

使用四则运算的优先级



34

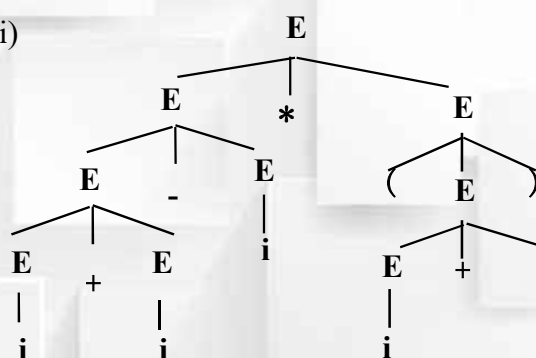
## 5.2.1 算符优先文法及优先表构造

- 算符优先文法及优先表构造

考虑二义文法 $G(E)$ :  $E \rightarrow i \mid E+E \mid E-E \mid E * E \mid E / E \mid (E)$

句子 $i+i-i*(i+i)$

不使用四则运算的优先级



35

## 5.2.1 算符优先文法及优先表构造

- 算符优先法则

- 1.乘除的优先级大于加减;
- 2.同优先级的运算符左大于右;
- 3.括号内的优先级大于括号外。

36

## 5.2.1 算符优先文法及优先表构造

- 算符优先法则

相继出现的终结符a与b

$a \leq b$  a的优先级低于b

$a = b$  a的优先级等于b

$a \geq b$  a的优先级高于b

37

## 5.2.1 算符优先文法及优先表构造

- 算符文法

文法G的任一产生式的右部，都不含两个相继(并列)的非终结符，即不含如下形式的产生式右部：

...QR...

则我们称该文法为**算符文法**。

a、b代表任意终结符；

P、Q、R代表任意非终结符；

‘...’代表由终结符和非终结符组成的任意序列，包括空字。

38

## 5.2.1 算符优先文法及优先表构造

### • 算符优先文法的优先性判断规则

假定G是一个不含 $\epsilon$ 产生式的算符文法。对于任何一对终结符a、b，我们说：

1.  $a \equiv b$  当且仅当文法G中含有形如  
 $R \rightarrow \dots ab \dots$  或  $P \rightarrow \dots aQb \dots$  的产生式；
2.  $a \leq b$  当且仅当G中含有形如  
 $R \rightarrow \dots aP \dots$  的产生式，而  $P \xRightarrow{+} b \dots$  或  $P \xRightarrow{+} Qb \dots$ ；
3.  $a \geq b$  当且仅当G中含有形如  
 $R \rightarrow \dots Pb \dots$  的产生式，而  $P \xRightarrow{+} \dots a$  或  $P \xRightarrow{+} \dots aQ$ 。

39

## 5.2.1 算符优先文法及优先表构造

### • 算符优先文法

一个算符文法G中的任何终结符对(a, b)至多只满足上述三关系之一，则称G是一个**算符优先文法**。

40

## 5.2.1 算符优先文法及优先表构造

### • 算符优先文法

- 例:考虑下面的文法G(E):

- (1)  $E \rightarrow E+T \mid T$
- (2)  $T \rightarrow T * F \mid F$
- (3)  $F \rightarrow P \uparrow F \mid P$
- (4)  $P \rightarrow (E) \mid i$

1.  $a \equiv b$  当且仅当文法G中含有形如  
 $R \rightarrow \dots ab \dots$  或  $R \rightarrow \dots aQb \dots$  的产生式;

( $\equiv$ )

41

## 5.2.1 算符优先文法及优先表构造

### • 算符优先文法

- 例:考虑下面的文法G(E):

- (1)  $E \rightarrow E+T \mid T$
- (2)  $T \rightarrow T * F \mid F$
- (3)  $F \rightarrow P \uparrow F \mid P$
- (4)  $P \rightarrow (E) \mid i$

$E \rightarrow E+T \quad T \Rightarrow T * F \quad + \leq *$   
 $T \rightarrow T * F \quad F \Rightarrow P \uparrow F \quad * \leq \uparrow$   
 $F \rightarrow P \uparrow F \quad F \Rightarrow P \uparrow F \quad \uparrow \leq \uparrow$   
 $P \rightarrow (E) \mid i \quad E \overset{+}{\Rightarrow} +, *, \uparrow$   
 $( \leq +, ( \leq *, ( \leq \uparrow, ( \leq i$

2.  $a \leq b$  当且仅当G中含有形如  
 $R \rightarrow \dots aP \dots$  的产生式, 而  $P \overset{+}{\Rightarrow} b \dots$  或  $P \overset{+}{\Rightarrow} Qb \dots$ ;

42

## 5.2.1 算符优先文法及优先表构造

### • 算符优先文法

- 例:考虑下面的文法G(E):
 

$E \rightarrow E+T$	$E \Rightarrow E+T$	$+$	$\triangleright$	$+$	
(1) $E \rightarrow E+T \mid T$	$E \rightarrow E+T$	$E \xRightarrow{+} T * F$	$+$	$\triangleright$	$*$
(2) $T \rightarrow T * F \mid F$					
(3) $F \rightarrow P \uparrow F \mid P$	.....				
(4) $P \rightarrow (E) \mid i$					

- 3.  $a \triangleright b$  当且仅当G中含有形如  
 $R \rightarrow \dots Pb \dots$ 的产生式, 而  $P \xRightarrow{+} \dots a$  或  $P \xRightarrow{+} \dots aQ$ 。

43

## 5.2.1 算符优先文法及优先表构造

- 算符优先文法
- 例:考虑下面的文法G(E):
 

(1) $E \rightarrow E+T \mid T$
(2) $T \rightarrow T * F \mid F$
(3) $F \rightarrow P \uparrow F \mid P$
(4) $P \rightarrow (E) \mid i$

优先表

	+	*
+	$\triangleright$	$<$

44

## 5.2.1 算符优先文法及优先表构造

### • 优先表的构造

确定满足关系 $\leq$ 和 $\geq$ 的所有终结符对：

首先需要对G的每个非终结符P构造两个集合  
FIRSTVT(P)和LASTVT(P)：

$$\text{FIRSTVT}(P) = \{a \mid P \xRightarrow{+} a \cdots, \text{或} P \xRightarrow{+} Qa \cdots, a \in V_T \text{ 而 } Q \in V_N\}$$

$$\text{LASTVT}(P) = \{a \mid P \xRightarrow{+} \cdots a, \text{或} P \xRightarrow{+} \cdots aQ, a \in V_T \text{ 而 } Q \in V_N\}$$

45

## 5.2.1 算符优先文法及优先表构造

### • 优先表的构造

假定有个产生式的一个候选形为

...aP...

那么，对任何 $b \in \text{FIRSTVT}(P)$ ，有  $a \leq b$ 。

假定有个产生式的一个候选形为

...Pb...

那么，对任何 $a \in \text{LASTVT}(P)$ ，有  $a \geq b$ 。

FIRSTVT，导出式中的非终结符在b前；LASTVT，导出式中的非终结符在a后；

46

## 5.2.1 算符优先文法及优先表构造

- 优先表的构造

构造集合FIRSTVT(P)

$$\text{FIRSTVT}(P) = \{a \mid P \xRightarrow{+} a \cdots, \text{或} P \xRightarrow{+} Qa \cdots, a \in V_T \text{ 而 } Q \in V_N\}$$

若 $a \in \text{FIRSTVT}(Q)$ ，且有产生式 $P \rightarrow Q \dots$ ，则 $a \in \text{FIRSTVT}(P)$ 。

47

## 5.2.1 算符优先文法及优先表构造

- 优先表的构造

构造集合LASTVT(P)

$$\text{LASTVT}(P) = \{a \mid P \xRightarrow{+} \cdots a, \text{或} P \xRightarrow{+} \cdots aQ, a \in V_T \text{ 而 } Q \in V_N\}$$

若 $a \in \text{LASTVT}(Q)$ ，且有产生式 $P \rightarrow \dots Q$ ，则 $a \in \text{LASTVT}(P)$ 。

48



## 5.2.1 算符优先文法及优先表构造

### • 优先表的构造程序

```

 $P \rightarrow ()$        $P \rightarrow (E)$ 
FOR 每条产生式  $P \rightarrow X_1 X_2 \dots X_n$  DO
  FOR  $i := 1$  TO  $n-1$  DO
    BEGIN
      IF  $X_i$  和  $X_{i+1}$  均为终结符 THEN 置  $X_i \preceq X_{i+1}$ 
      IF  $i \leq n-2$  且  $X_i$  和  $X_{i+2}$  都为终结符
         但  $X_{i+1}$  为非终结符 THEN 置  $X_i \preceq X_{i+2}$ ;
    1.  $a \preceq b$  当且仅当文法  $G$  中含有形如
        $R \rightarrow \dots ab \dots$  或  $R \rightarrow \dots aQb \dots$  的产生式;
  
```

49

## 5.2.1 算符优先文法及优先表构造

### • 优先表的构造程序

```

  IF  $X_i$  为终结符而  $X_{i+1}$  为非终结符 THEN
    FOR FIRSTVT( $X_{i+1}$ ) 中的每个  $a$  DO
      置  $X_i \leq a$ ;
  IF  $X_i$  为非终结符而  $X_{i+1}$  为终结符 THEN
    FOR LASTVT( $X_i$ ) 中的每个  $a$  DO
      置  $a \geq X_{i+1}$ 
END

```

50

### 5.2.1算符优先文法及优先表构造

• 优先表的构造程序

例：考虑下面的文法G(E)：

$E \rightarrow E+T \mid T$

$T \rightarrow T * F \mid F$

$F \rightarrow P \uparrow F \mid P$

$P \rightarrow (E) \mid i$

$FIRSTVT(E) = \{+, *, \uparrow, (, i\}$

$FIRSTVT(T) = \{*, \uparrow, (, i\}$

$FIRSTVT(F) = \{\uparrow, (, i\}$

$FIRSTVT(P) = \{(, i\}$

(1)计算文法G的FIRSTVT和LASTVT;

$LASTVT(E) = \{+, *, \uparrow, ), i\}$

(2) 构造优先关系表;

$LASTVT(T) = \{*, \uparrow, ), i\}$



(3) G是算符优先文法吗?

$LASTVT(F) = \{\uparrow, ), i\}$

$LASTVT(P) = \{), i\}$

### 5.2.1算符优先文法及优先表构造

• 优先表的构造程序

	<div></div> <div><math>P \rightarrow \dots a</math></div> <div><math>P \rightarrow \dots a</math></div>
<div><math>(0) E' \rightarrow \# E \#</math></div>	<div><math>\#</math>  <math>\#</math></div>

5.2.1算符优先文法及优先表构造

FIRSTVT 标记行

LASTVT 标记列

	+	*
+	$\triangleright$	$\triangleleft$
*	$\triangleright$	$\triangleright$

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

$FIRSTVT(E) = \{+, *, \uparrow, (, i\}$

$FIRSTVT(T) = \{*, \uparrow, (, i\}$

$FIRSTVT(F) = \{\uparrow, (, i\}$

$FIRSTVT(P) = \{(, i\}$

$LASTVT(E) = \{+, *, \uparrow, ), i\}$

$LASTVT(T) = \{*, \uparrow, ), i\}$

$LASTVT(F) = \{\uparrow, ), i\}$

$LASTVT(P) = \{), i\}$

G是算符优先文法