编译原理

第五章 优先分析法

方徽星
扬州大学 信息工程学院(505)
fanghuixing@yzu.edu.cn

2018年春季学期

本章主要内容

一．算符优先分析法

- 算符优先分析法的思想
- 求FIRSTVT和LASTVT
- 算符优先表的构造
- 算符优先分析算法

二．优先函数的构造

- 逐次加1法
- Bell有向图法
- 设计优先分析程序

第一节　算符优先分析法

1.1 算符优先分析法的思想

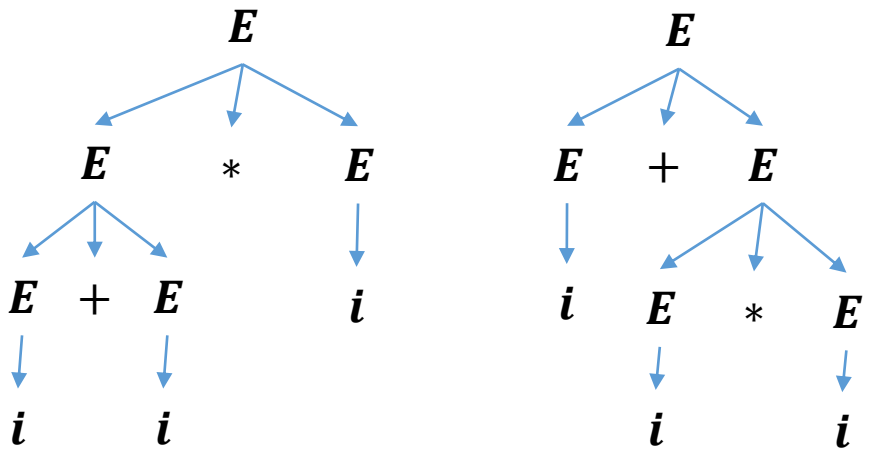
- 定义**终结符**之间的**优先关系**，借助优先关系寻找可**归约**串，进行**自下而上**分析
- 算符优先分析法不一定遵照规范归约过程，**不是**一种规范归约法

1.1 算符优先分析法的思想

- 计算**终结符**之间的**优先关系**，借助优先关系寻找可**归约**串，进行**自下而上**分析
- 算符优先分析法不一定遵照规范归约过程，**不是**一种规范归约法

例：考虑文法产生式：$E \to E+E \mid E*E \mid (E) \mid i$，分析句子$i+i*i$

算符优先分析法根据文法计算优先关系，并未反映附加条件，无法解决二义问题



1.1 算符优先分析法的思想

- **关键：比较**两个相继出现的终结符的**优先级**，而后决定应采取的动作
  - 先**定义**各种可能相继出现的终结符的**优先级**，表示成**矩阵**形式
  - 之后通过**查询**矩阵元素而获得终结符间的优先关系

1.1 算符优先分析法的思想

- 定义a和b之间的三种关系：
  - $a \lessdot b$：a的优先级**低**于b
  - $a \doteq b$：a的优先级**等**于b
  - $a \gtrdot b$：a的优先级**高**于b
  - 如果a和b不可能相继出现，则a和b之间没关系

算符优先表

左终结符 \ 右终结符	+	*
+	$\gtrdot$	$\lessdot$
*	$\gtrdot$	$\gtrdot$

1.1 算符优先分析法的思想

- **算符文法**：上下文无关文法G中**所有**产生式**右部**都**不含**有两个相继的非终结符，即**没有**如下形式的产生式体：

  …QR…

  则称G为算符文法，其中Q、R表示非终结符

1.1 算符优先分析法的思想

- 令G为一个**不含ε-产生式的算符文法**，对于任何一对终结符a和b定义如下三种**可能**关系：
  1. $a \doteq b$ **iff** G有形如P→… ab…或P→… a Q b …的产生式
  2. $a \lessdot b$ **iff** G有形如P→… a R…的产生式，而R $\overset{+}{\Rightarrow} b$ …或R$\overset{+}{\Rightarrow}$Q b …
  3. $a \gtrdot b$ **iff** G有形如P→…R b …的产生式，而 R$\overset{+}{\Rightarrow}$…a或R$\overset{+}{\Rightarrow}$…aQ

1.1 算符优先分析法的思想

- 令G为一个**不含ε-产生式的算符文法**，对于任何一对终结符a和b定义如下三种**可能**关系：
  1. $a \doteq b$ **iff** G有形如P→…ab…或P→…a Q b …的产生式
  2. $a \lessdot b$ **iff** G有形如P→…a R…的产生式，而R $\overset{+}{\Rightarrow} b$ …或R$\overset{+}{\Rightarrow}$Q b …
  3. $a \gtrdot b$ **iff** G有形如P→…R b …的产生式，而 R$\overset{+}{\Rightarrow}$…a或R$\overset{+}{\Rightarrow}$…aQ
- 若算符文法G中任何终结符对(a,b)**至多只**满足下述关系之一：$a \doteq b$，$a \lessdot b$，$a \gtrdot b$，则称G为**算符优先文法**

1.1 算符优先分析法的思想

- 例：考虑如下文法G：
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$
  - **由$P \to (E)$，有($\doteq$)**

1.1 算符优先分析法的思想

- 例：考虑如下文法G：
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$
  - 由$P \to (E)$，有$(\doteq)$
  - **由$E \to E + T$和$T \overset{+}{\Rightarrow} T * F$，有$+ \lessdot *$**

1.1 算符优先分析法的思想

- 例：考虑如下文法G：
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$

  ➢由$P \to (E)$，有$(\doteq)$

  ➢由$E \to E + T$和$T \overset{+}{\Rightarrow} T * F$，有$+ \lessdot *$

  ➢**由$T \to T * F$和$F \to P \uparrow F$，有$* \lessdot \uparrow$**

1.1 算符优先分析法的思想

- 例：考虑如下文法G：
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$

  ➢由$P \to (E)$，有$(\doteq)$

  ➢由$E \to E + T$和$T \overset{+}{\Rightarrow} T * F$，有$+ \lessdot *$

  ➢由$T \to T * F$和$F \to P \uparrow F$，有$* \lessdot \uparrow$

  ➢**由$E \to E + T$和$E \overset{+}{\Rightarrow} E + T$，有$+ \gtrdot +$**

1.1 算符优先分析法的思想

- 例：考虑如下文法G：
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$

  ➢由$P \to (E)$，有$(\doteq)$

  ➢由$E \to E + T$和$T \overset{+}{\Rightarrow} T * F$，有$+ \lessdot *$

  ➢由$T \to T * F$和$F \to P \uparrow F$，有$* \lessdot \uparrow$

  ➢由$E \to E + T$和$E \overset{+}{\Rightarrow} E + T$，有$+ \gtrdot +$

  ➢**由$F \to P \uparrow F$和$F \overset{+}{\Rightarrow} P \uparrow F$，有$\uparrow \lessdot \uparrow$**

1.1 算符优先分析法的思想

- 例：考虑如下文法G：
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$

  ➢ 由$P \to (E)$，有$(\doteq)$

  ➢ 由$E \to E + T$和$T \overset{+}{\Rightarrow} T * F$，有$+ \lessdot *$

  ➢ 由$T \to T * F$和$F \to P \uparrow F$，有$* \lessdot \uparrow$

  ➢ 由$E \to E + T$和$E \overset{+}{\Rightarrow} E + T$，有$+ \gtrdot +$

  ➢ 由$F \to P \uparrow F$和$F \overset{+}{\Rightarrow} P \uparrow F$，有$\uparrow \lessdot \uparrow$

  ➢ **由$P \to (E)$和$E \Rightarrow E + T \Rightarrow T + T \Rightarrow T * F + T \Rightarrow F * F + T \Rightarrow P \uparrow F * F + T \Rightarrow i \uparrow F * F + T$，有$\lessdot \{+, *, \uparrow, i\}$**

1.1 算符优先分析法的思想

左终结符 \ 右终结符	+	*	↑	i	(	)	$
+	>	<	<	<	<	>	>
*	>	>	<	<	<	>	>
↑	>	>	<	<	<	>	>
i	>	>	>	—	—	>	>
(	<	<	<	<	<	=	—
)	>	>	>	—	—	>	>
$	<	<	<	<	<	—	=

1.2 求FIRSTVT和LASTVT

- **算法实现**从算符优先文法G构造优先关系表
  - 检查G的每个产生式，可以找出所有满足$a \doteq b$的终结符对
  - 为了找出所有满足<和>关系的终结符对，首先需要对G的每个非终结符P构造两个集合$FIRSTVT(P)$和$LASTVT(P)$

$$\boldsymbol{FIRSTVT(P) = \{a \mid P \overset{+}{\Rightarrow} a\ldots, or\ P \overset{+}{\Rightarrow} Qa\ldots, a \in V_T, Q \in V_N\}}$$

$$\boldsymbol{LASTVT(P) = \{a \mid P \overset{+}{\Rightarrow} \ldots a\ \ or\ P \overset{+}{\Rightarrow} \ldots aQ, a \in V_T, Q \in V_N\}}$$

1.2 求FIRSTVT和LASTVT

- **算法实现**从算符优先文法G构造优先关系表
  - 检查G的每个产生式，可以找出所有满足$a \doteq b$的终结符对
  - 为了找出所有满足<和>关系的终结符对，首先需要对G的每个非终结符P构造两个集合$FIRSTVT(P)$和$LASTVT(P)$

$$\boldsymbol{FIRSTVT(P) = \{a \mid P \overset{+}{\Rightarrow} a\ldots, or\ P \overset{+}{\Rightarrow} Qa\ldots, a \in V_T, Q \in V_N\}}$$

$$\boldsymbol{LASTVT(P) = \{a \mid P \overset{+}{\Rightarrow} \ldots a\ \ or\ P \overset{+}{\Rightarrow} \ldots aQ, a \in V_T, Q \in V_N\}}$$

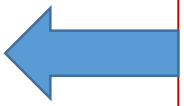
得到上述集合之后，通过检查每个产生式的候选式确定满足<和>关系的所有的终结符对

1.2 求FIRSTVT和LASTVT

- 如果有产生式的候选式形如

$$\dots aP \dots$$

则对任何$b \in FIRSTVT(P)$，$a \lessdot b$



$a \lessdot b$ iff G有形如

P→…a R…的产生式，

而R $\overset{+}{\Rightarrow} b$ …或R$\overset{+}{\Rightarrow}$Q b …

$$FIRSTVT(P) = \{a \mid P \overset{+}{\Rightarrow} a \dots, or\ P \overset{+}{\Rightarrow} Qa \dots, a \in V_T, Q \in V_N\}$$

$$LASTVT(P) = \{a \mid P \overset{+}{\Rightarrow} \dots a\ \ or\ P \overset{+}{\Rightarrow} \dots aQ, a \in V_T, Q \in V_N\}$$

1.2 求FIRSTVT和LASTVT

- 如果有产生式的候选式形如

$$\dots aP \dots$$

则对任何$b \in FIRSTVT(P)$，$a \lessdot b$

- 如果有产生式的候选式形如

$$\dots Pb \dots$$

则对任何$a \in LASTVT(P)$，$a \gtrdot b$

$a \lessdot b$ iff G有形如

P→…a R…的产生式，

而R $\overset{+}{\Rightarrow} b$ …或R$\overset{+}{\Rightarrow}$Q b …

$a \gtrdot b$ iff G有形如

P → …R b …的产生式，

而 R$\overset{+}{\Rightarrow}$…a或R$\overset{+}{\Rightarrow}$…aQ

$$FIRSTVT(P) = \{a \mid P \overset{+}{\Rightarrow} a \dots, or\ P \overset{+}{\Rightarrow} Qa \dots, a \in V_T, Q \in V_N\}$$

$$LASTVT(P) = \{a \mid P \overset{+}{\Rightarrow} \dots a\ \ or\ P \overset{+}{\Rightarrow} \dots aQ, a \in V_T, Q \in V_N\}$$

1.2 求FIRSTVT和LASTVT

- $\boldsymbol{FIRSTVT(P)}$构造规则：
  1. 若有产生式$\boldsymbol{P \to a \ldots}$或$\boldsymbol{P \to Qa \ldots}$，
     则$\boldsymbol{a \in FIRSTVT(P)}$
  2. 若$\boldsymbol{a \in FIRSTVT(Q)}$，且有产生式$\boldsymbol{P \to Q \ldots}$，
     则$\boldsymbol{a \in FIRSTVT(P)}$
- 建立一个布尔数组$\boldsymbol{F[P,a]}$使得

$$\boldsymbol{F[P,a]} = true \ \textbf{iff} \ \boldsymbol{a \in FIRSTVT(P)}$$

$$\boldsymbol{FIRSTVT(P) = \{a \mid P \overset{+}{\Rightarrow} a \ldots, or\ P \overset{+}{\Rightarrow} Qa \ldots, a \in V_T, Q \in V_N\}}$$

1.2 求FIRSTVT和LASTVT

- 具体步骤
  - 按照规则1对每个数组元素$F[P,a]$赋初值，把所有初值为真的数组元素对应的符号对(P,a)放入栈K中，然后

1.2 求FIRSTVT和LASTVT

- 具体步骤
  - 按照规则1对每个数组元素$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]$赋初值，把所有初值为真的数组元素对应的符号对$(\boldsymbol{P},\boldsymbol{a})$放入栈K中，然后
  - 若栈K不空，则弹出栈顶元素，并记为$(Q,\boldsymbol{a})$，**对于每个形如**

$$\boldsymbol{P}\rightarrow Q\ldots$$

的产生式，若$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]=false$则设置$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]$为$true$，并把$(\boldsymbol{P},\boldsymbol{a})$放入栈K中

1.2 求FIRSTVT和LASTVT

- 具体步骤
  - 按照规则1对每个数组元素$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]$赋初值，把所有初值为真的数组元素对应的符号对$(\boldsymbol{P},\boldsymbol{a})$放入栈K中，然后
  - 若栈K不空，则弹出栈顶元素，并记为$(Q,\boldsymbol{a})$，**对于每个形如**

    $$\boldsymbol{P}\rightarrow Q\ldots$$

    的产生式，若$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]=false$则设置$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]$为$true$，并把$(\boldsymbol{P},\boldsymbol{a})$放入栈K中
  - 重复上述过程，直到栈空

1.2 求FIRSTVT和LASTVT

- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

  计算FIRSTVT：

  由$S \to if$ ...得$F(S, if) = true$

  由$E \to E + T$得$F(E, +) = true$

  由$T \to T\ *\ F$ 得$F(T, *) = true$

  由$F \to i$得$F(F, i) = true$

  由$C \to\ b$得$F(C, b) = true$

1.2 求FIRSTVT和LASTVT

- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

由$S \to if$ ...得$F(S, if) = true$

由$E \to E + T$得$F(E, +) = true$

由$T \to T\ *\ F$ 得$F(T, *) = true$

由$F \to i$得$F(F, i) = true$

由$C \to\ b$得$F(C, b) = true$

$(\boldsymbol{C}, \boldsymbol{b})$
$(\boldsymbol{F}, \boldsymbol{i})$
$(\boldsymbol{T}, *)$
$(\boldsymbol{E}, +)$
$(\boldsymbol{S}, \boldsymbol{if})$

1.2 求FIRSTVT和LASTVT

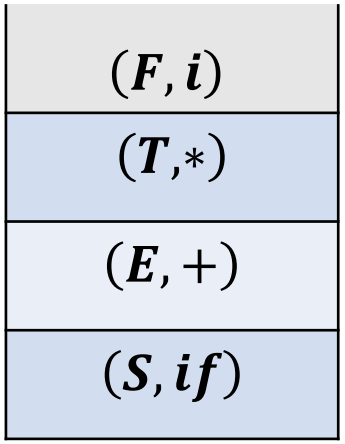
- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

将$(\boldsymbol{C}, \boldsymbol{b})$弹出栈，但是没有形如

$$P \to C \ldots$$

的产生式



1.2 求FIRSTVT和LASTVT

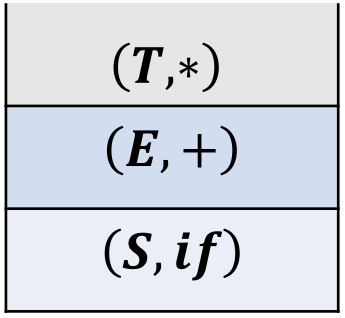
- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

计算FIRSTVT：

将$(\boldsymbol{F}, \boldsymbol{i})$弹出栈，有产生式

$$T \rightarrow F$$

且$F(T, i) = false$，则将$F(T, i)$设置为真



1.2 求FIRSTVT和LASTVT

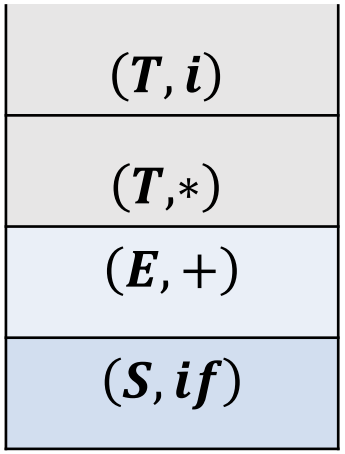
- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

将$(\boldsymbol{F}, \boldsymbol{i})$弹出栈，有产生式

$$T \to F$$

且$F(T, i) = false$，则将$F(T, i)$设置为真，把$(\boldsymbol{T}, \boldsymbol{i})$压栈



1.2 求FIRSTVT和LASTVT

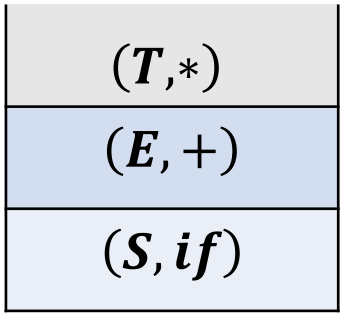
- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

计算FIRSTVT：

将$(\boldsymbol{T}, \boldsymbol{i})$弹出栈，有产生式

$$E \rightarrow T$$

且$F(E, i) = false$，则将$F(E, i)$设置为真



1.2 求FIRSTVT和LASTVT

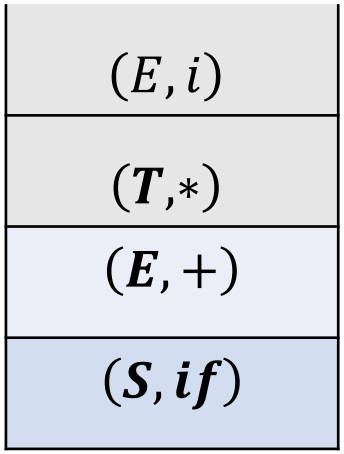
- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

计算FIRSTVT：

将$(\boldsymbol{T}, \boldsymbol{i})$弹出栈，有产生式

$$E \rightarrow T$$

且$F(E,i) = false$，则将$F(E,i)$设置为真，把(E,i)入栈



1.2 求FIRSTVT和LASTVT

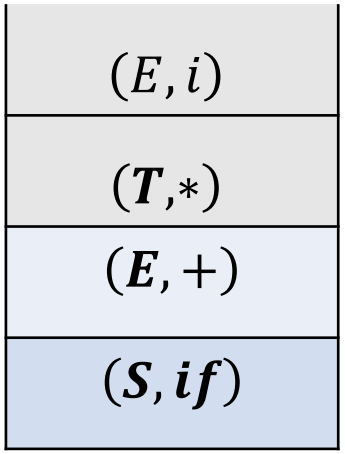
- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

又有产生式

$$T \to T\ *\ F$$

且$F(T, i) = true$，则不做处理



1.2 求FIRSTVT和LASTVT

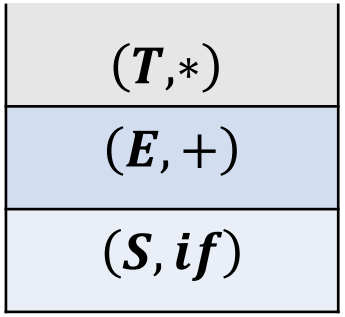
- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

计算FIRSTVT：

将(E, i)出栈，有产生式

$$E \rightarrow E + T$$

且$F(E, i) = true$，不做处理



1.2 求FIRSTVT和LASTVT

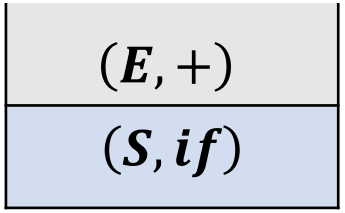
- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

将$(\boldsymbol{T},*)$出栈，有产生式

$$E \to T$$

且$F(E,*) = false$，设置$F(E,*)$为真，



1.2 求FIRSTVT和LASTVT

- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

计算FIRSTVT：

将$(E,*)$入栈

$(\boldsymbol{E},*)$
$(\boldsymbol{E},+)$
$(\boldsymbol{S},\boldsymbol{if})$

1.2 求FIRSTVT和LASTVT

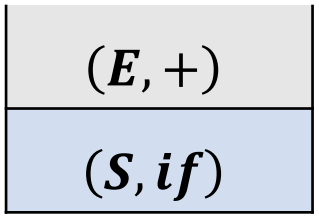
- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

计算FIRSTVT：

将$(E,*)$**出栈，有产生式**

$$E \rightarrow E + T$$

因$F(E,*) = true$**，不处理**



1.2 求FIRSTVT和LASTVT

- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

将$(\boldsymbol{E}, +)$**出栈，有产生式**

$$E \to E + T$$

因$F(E, +) = true$**，不处理**

$(\boldsymbol{S}, \boldsymbol{if})$

1.2 求FIRSTVT和LASTVT

- 例：考虑如下文法G
  1. $S \to if\ C\ then\ E\ else\ E$
  2. $E \to E + T \mid T$
  3. $T \to T\ *\ F \mid F$
  4. $F \to i$
  5. $C \to\ b$

计算FIRSTVT：

将$(\boldsymbol{S}, \boldsymbol{if})$**出栈，没有形如**

$$P \to S \ldots$$

的产生式

栈空！结束。

1.2 求FIRSTVT和LASTVT

非终结符 \ 终结符	if	then	else	+	*	i	b
S	True						
E				True	True	True	
T					True	True	
F						True	
C							True

FIRSTVT布尔数组

1.2 求FIRSTVT和LASTVT

- $\boldsymbol{LASTVT(P)}$构造规则：

$$\boldsymbol{LASTVT(P) = \{a \mid P \overset{+}{\Rightarrow} \ldots a \ \ or \ P \overset{+}{\Rightarrow} \ldots aQ, a \in V_T, Q \in V_N\}}$$

1.2 求FIRSTVT和LASTVT

- $\boldsymbol{LASTVT(P)}$构造规则：
  1. 若有产生式$\boldsymbol{P \to \cdots a}$**或**$\boldsymbol{P \to \cdots aQ}$，
     则$\boldsymbol{a \in LASTVT(P)}$
  2. 若$\boldsymbol{a \in LASTVT(Q)}$，且有产生式$\boldsymbol{P \to \cdots Q}$，
     则$\boldsymbol{a \in LASTVT(P)}$
- 建立一个布尔数组$\boldsymbol{F[P,a]}$使得

$$\boldsymbol{F[P,a]} = true \ \textbf{iff} \ \boldsymbol{a \in LASTVT(P)}$$

$$\boldsymbol{LASTVT(P)} = \{\boldsymbol{a} \mid \boldsymbol{P} \overset{+}{\Rightarrow} \ldots \boldsymbol{a} \ \ \boldsymbol{or} \ \boldsymbol{P} \overset{+}{\Rightarrow} \ldots \boldsymbol{aQ}, \boldsymbol{a \in V_T}, \boldsymbol{Q \in V_N}\}$$

1.2 求FIRSTVT和LASTVT

- 具体步骤
  - 按照规则1对每个数组元素$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]$赋初值，把所有初值为真的数组元素对应的符号对$(\boldsymbol{P},\boldsymbol{a})$放入栈K中，然后
  - 若栈K不空，则弹出栈顶元素，并记为$(Q,\boldsymbol{a})$，**对于每个形如**

    $$\boldsymbol{P}\rightarrow \ldots Q$$

    的产生式，若$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]=false$则设置$\boldsymbol{F}[\boldsymbol{P},\boldsymbol{a}]$为$true$，并把$(\boldsymbol{P},\boldsymbol{a})$放入栈K中
  - 重复上述过程，直到栈空

1.2 求FIRSTVT和LASTVT

- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

  计算LASTVT：

1.2 求FIRSTVT和LASTVT

非终结符 \ 终结符	if	then	else	+	*	i	b
S			True	True	True	True	
E				True	True	True	
T					True	True	
F						True	
C							True

LASTVT布尔数组

1.3 算符优先表构造算法

For (Each $P \to X_1X_2 \ldots X_n$) {
  For ($i := 1 : n-1$) {
    If ($X_i \in V_T \wedge X_{i+1} \in V_T$) $X_i \doteq X_{i+1}$;
    If ($i \le n-2 \wedge X_i \in V_T \wedge X_{i+1} \in V_N \wedge X_{i+2} \in V_T$)
      $X_i \doteq X_{i+2}$;
    If ($X_i \in V_T \wedge X_{i+1} \in V_N$) {
      For (Each $a \in FIRSTVT(X_{i+1})$) $\{X_i \lessdot a;\}$
    }
    If ($X_i \in V_N \wedge X_{i+1} \in V_T$) {
      For (Each $a \in LASTVT(X_i)$) $\{a \gtrdot X_{i+1};\}$
    }
  }
}

1.3 算符优先表构造算法

- **如果有产生式的候选式形如**

$$\dots aP \dots$$

则对任何$b \in FIRSTVT(P)$，$a \lessdot b$

```
If (X_i ∈ V_T ∧ X_{i+1} ∈ V_N) {
        For (Each a ∈ FIRSTVT(X_{i+1})) {
                X_i ⋖ a;
        }
}
```

1.3 算符优先表构造算法

- **如果有产生式的候选式形如**

$$\dots Pb \dots$$

则对任何$a \in LASTVT(P)$，$a \gtrdot b$

```
If (X_i ∈ V_N ∧ X_{i+1} ∈ V_T) {
        For (Each a ∈ LASTVT(X_i)) {
                a ⋗ X_{i+1};
        }
}
```

1.3 算符优先表构造算法

- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

  构造G的算符优先表

1.4 算符优先分析算法

- **短语：**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

$$\boldsymbol{S \overset{*}{\Rightarrow} \alpha A \delta}\text{且}\boldsymbol{A \overset{+}{\Rightarrow} \beta}$$

  则称β是句型$\alpha\beta\delta$相对于非终结符A的短语
- **直接短语：**如果有$A \Rightarrow \beta$，则称β是句型$\alpha\beta\delta$相对于规则$A \rightarrow \beta$的直接短语
- **句柄：**一个句型的**最左直接短语**称为该句型的句柄

1.4 算符优先分析算法

- 例：考虑文法G:
  - $E \to T \mid E + T$
  - $T \to F \mid T * F$
  - $F \to i \mid (E)$

  的一个句型$i * i + i$：

  ✓$i + i$不是该句型的一个短语，
  ✓但$i * i$是句型$i * i + i$相对于T的一个短语
  ✓i为句型$i * i + i$相对于规则$F \to i$ 的直接短语，也是最左直接短语，即句柄

1.4 算符优先分析算法

- **素短语：**是指**至少含有一个终结符**的短语，并且除自身外不再含有更小的素短语
- **最左素短语：**处于句型**最左边**的那个素短语
- 例：考虑如下文法
  1. $E \to E + T \mid T$
  2. $T \to T * F \mid F$
  3. $F \to P \uparrow F \mid P$
  4. $P \to (E) \mid i$

$P * P$和i是句型$P * P + i$的素短语，且$P * P$是该句型的最左素短语

1.4 算符优先分析算法

- 考虑算符优先文法，其句型（括在两个$之间）的一般形式表示如下：

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

其中，每个a_i都是终结符，N_i是可省略的非终结符

- 一个算符优先文法的任何句型的最左素短语是满足如下条件的最左子串$N_ja_j\ldots N_ia_iN_{i+1}$

1. $a_{j-1} \lessdot a_j$
2. $a_j \doteq a_{j+1} \doteq \ldots \doteq a_i$
3. $a_i \gtrdot a_{i+1}$

若最左素短语有两个以上终结符，
则不存在A_k，满足$A_k \to a_k$，
但有：$P \to \cdots a_kN_{k+1}a_{k+1}\cdots$
所以$a_j \doteq \ldots \doteq a_i$

1.4 算符优先分析算法

```
k:=1;  S[k]:='$';
Repeat
        a:=nextInput();
        If  (S[k]∈ V_T)  j:=k Else j:=k-1;
        While (S[j] ⋗a) {
                Reduce();
        }
        If (S[j] ⋖a or S[j] ≐a) {
            k:=k+1;
            S[k]:=a;
        }
        Else Error;
Until a='$'
```

1.4 算符优先分析算法

```
k:=1;  S[k]:='$';
Repeat
        a:=nextInput();
        If  (S[k]∈ V_T)  j:=k Else j:=k-1;
        While (S[j] ⋗a) {
              Reduce();
        }
        If (S[j] ⋖a or S[j] ≐a) {
            k:=k+1;
            S[k]:=a;
        }
        Else Error;
Until a='$'
```

将a移进栈顶

```
Repeat
        Q:=S[j];
        If (S[j-1] ∈ V_T)
            j:=j-1;
        Else
            j:=j-2;
Until S[j] ⋖ Q;

把S[j+1]...S[k]归约为某个N;

k:=j+1;
S[k]:=N;
```

1.4 算符优先分析算法

- 例：考虑如下文法G
  1. $S \rightarrow if\ C\ then\ E\ else\ E$
  2. $E \rightarrow E + T \mid T$
  3. $T \rightarrow T\ *\ F \mid F$
  4. $F \rightarrow i$
  5. $C \rightarrow\ b$

使用算符优先分析算法分析语句$if\ b\ then\ i\ else\ i\$$

1.4 算符优先分析算法

步骤	栈	关系	输入串	动作
0	$	$\lessdot$	$if\ b\ then\ i\ else\ i\$$	
1	$\$\mathbf{if}$	$\lessdot$	$\mathbf{b}\ then\ i\ else\ i\$$	移进
2	$\$\mathbf{if\ b}$	$\gtrdot$	$\mathbf{then}\ i\ else\ i\$$	移进
3	$\$\mathbf{if\ N}$	?	$then\ i\ else\ i\$$	$N \rightarrow b$归约
4	?	?	?	?

1.4 算符优先分析算法

- 与规范归约的区别：
  - 可归约短语必须至少包含一个终结符，这与规范归约过程不同
  - 规范归约是严格按照句柄进行归约，是终结符和非终结符一起考虑，只要栈顶形成了句柄，不管该句柄是否包含终结符总要进行归约
- 缺点：
  - 有些文法未必满足算法优先文法的要求
  - 终结符数量多时，优先表占用空间较多

第二节 优先函数的构造

2.1 优先函数

- 优先表的一个缺点是占用太大的存储空间，使用优先函数可以克服这个缺点
- 将每个终结符a与一对整数f(a)，g(a)联系起来：
  - f(a)是终结符a在栈内的优先数
  - g(a)是终结符a在进栈前的优先数
  - f(a)和g(a)满足如下关系：
    - **若a⋖b，则f(a) < g(b)**
    - **若a≍b，则f(a) = g(b)**
    - **若a⋗b，则f(a) > g(b)**
- 存储空间：$2 * n < n * n$，其中n =终结符数量

2.1 优先函数

- 优先函数并不等价于优先表，原先不存在优先关系的两个终结符，由于与自然数相对应而变得可比较
- 有些优先表不存在对应的优先函数

	a	b
a	⋗	⋗
b	⋗	⋗

- 如果存在一对优先函数，则存在无穷多的优先函数，优先函数值都加上一个常数后仍然满足优先关系

2.2 逐次加1法

- 算法步骤：
  1. 对所有的终结符a(包括$)，令f(a)=g(a)=c（c为任意常数）
  2. 对所有终结符：
     - 若a≻b，f(a) ≤ g(b)，则f(a):=g(b)+1;
     - 若a≺b，f(a) ≥ g(b)，则g(b):=f(a)+1;
     - 若a≍b，f(a) ≠ g(b)，则f(a):=g(b):=max(f(a),g(b)).
  3. 重复步骤2，直到f(a)，g(b)不再变化

如果f(a)或g(b)>=2n+c，但步骤(2)未能结束，则优先函数不存在

2.3 Bell有向图法

- 算法步骤：
  - 对每个终结符a (包括$)，令其对应两个结点fa和ga，画一张以所有fa和ga为结点的有向图
    - 如果a⋗b或a≍b就从fa画一弧指向gb
    - 如果a⋖b或a≍b就从gb画一弧指向fa
  - f(a):=结点fa可达的结点数（包括fa自身）
  - g(a):=结点ga可达的结点数（包括ga自身）
  - 检查f(a)和g(a)是否与优先表符合

小结

- 算符优先分析法：
  - 计算FIRSTVT和LASTVT,
  - 构造优先表,
  - 算符优先分析算法
- 优先函数：
  - 逐次加1法,
  - Bell有向图法