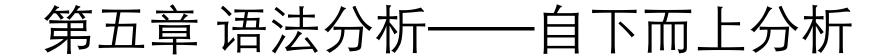
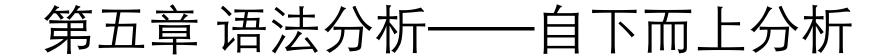
编译原理

第五章 语法分析——自下而上分析



- ■自下而上分析的基本问题
- ■算符优先分析算法
- ■LR 分析法



- ■自下而上分析的基本问题
- ■算符优先分析算法
- ■LR 分析法

算符优先文法

■ 一个文法,如果它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。

- 假定 G 是一个不含ε 产生式的算符文法。对于任何一对终结符 a 、 b ,我们说:
 - **1.** a♦b 当且仅当文法 **G** 中含有形如 **P**→····ab··· 或 **P**→····aQb··· 的产生式;
 - 2. a•b 当且仅当 G 中含有形如 P→…
 aR… 的产生式, → 面 R b → 或 R
 - 3. **②b** b ;当且仅当 G₊中含有形如 P→···
 Rb··· 的产生式, **→** R ··**→** 或 R
- 如果。个算符文法 G 中的任何终结符对 (a , b) 至多只满足下述三关系之一: a ◆ b , a b , a □ b 则称 G 是一个算符优先文法。

■ 通过检查 G 的每个产生式的每个候选式,可 找出所有满足 a ◆ b 的终结符对

```
1. a b 当且仅当文法 G 中含有形如 P→····ab··· 或 P→····aQb··· 的产生式
```

■ 确定满足关系 和 的所有终结符对

6

- 通过检查 G 的每个产生式的每个候选式,可 找出所有满足 a b 的终结符对。
- ■确定满足关系 和 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)

$$FIRSTVT(P) = \{a \mid P \stackrel{+}{\Rightarrow} a \cdots, \vec{\mathfrak{p}}P \stackrel{+}{\Rightarrow} Qa \cdots, a \in V_T \overrightarrow{\mathfrak{m}}Q \in V_N \}$$

 $\overline{\mathsf{Q}\mathsf{b}}$



- 通过检查 G 的每个产生式的每个候选式,可 找出所有满足 a b 的终结符对
- ■确定满足关系 和 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)

$$LASTVT(P) = \{a \mid P \stackrel{+}{\Rightarrow} \cdots a, \overrightarrow{\mathfrak{p}}P \stackrel{+}{\Rightarrow} \cdots aQ, a \in V_T \overrightarrow{\mathfrak{m}}Q \in V_N \}$$

- 通过检查 G 的每个产生式的每个候选式,可 找出所有满足 a ◆ b 的终结符对。
- ■确定满足关系 和 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):

$$FIRSTVT(P) = \{a | P \stackrel{+}{\Rightarrow} a \cdots, \vec{\boxtimes} P \stackrel{+}{\Rightarrow} Qa \cdots, a \in V_T \overrightarrow{\sqcap} Q \in V_N \}$$

**
$$FIRST(\alpha) = \{a \mid \alpha \Rightarrow a..., a \in V_T\}$$

$$LASTVT(P) = \{a \mid P \stackrel{+}{\Rightarrow} \cdots a, \mathfrak{R}P \stackrel{+}{\Rightarrow} \cdots aQ, a \in V_T \overline{m}Q \in V_N \}$$

*\mathbb{R} \mathbb{F} \mathbb{F}OLLOW(A) = \{a \ | S \Rightarrow ... Aa..., a \in V_T\}

- 有了这两个集合之后,就可以通过检查每个产生式的候选式确定满足关系 和 的 所有终结符对
 - □假定有个产生式的一个候选形为

...aP...

那么,对任何 b∈FIRSTVT(P),有 a b

□假定有个产生式的一个候选形为

...Pb...

那么,对任何 a∈LASTVT(P),有 a b

$$FIRSTVT(P) = \{a | P \stackrel{+}{\Rightarrow} a \cdots, \vec{\boxtimes} P \stackrel{+}{\Rightarrow} Qa \cdots, a \in V_T \overrightarrow{\sqcap} Q \in V_N \}$$

- 反复使用下面两条规则构造集合 FIRSTVT(P)
 - 1. 若有产生式 P→a···· 或 P→Qa···,则 a∈FIRSTVT(P)
 - 2. 若 $a \in FIRSTVT(Q)$,且有产生式 $P \rightarrow Q \cdots$, 将对推导的遍历转换成对产生式的反复遍历

- ■数据结构
 - □布尔数组 F[P, a],使得 F[P, a]为真的条件是,当且仅当 a∈FIRSTVT(P)。开始时,按上述的规则 (1) 对每个数组元素 F[P, a] 赋初值。
 - □栈 STACK ,把所有初值为真的数组元素 F[P
 - 1. 若有产生式 P→a··· 或 P→Qa···,则 a∈FIRSTVT(P)
 - 2. 若 a∈FIRSTVT(Q) ,且有产生式 P→Q···,则 a∈FIRSTVT(P)

- ■运算
 - □如果栈 STACK 不空,就将顶项逐出,记此 项为 (Q, a)。对于每个形如P→Q…

的产生式,若 F[P, a]为假,则变其值为 真且将 (P, a) 推进 STACK 栈

- □上述过程必须一直重复,直至栈 STACK 拆空为止
- 1. 若有产生式 P→a····或 P→Qa···,则 a∈FIRSTVT(P)
- 2. 若 a∈FIRSTVT(Q),且有产生式 P→Q···,则 a∈FIRSTVT(P)

■程序

```
PROCEDURE INSERT(P, a);
IF NOT F[P, a] THEN
BEGIN
F[P, a]:=TRUE;
把(P, a) 下推进 STACK 栈
END;
```

主程序:

BEGIN

```
1. 若有产生式 P→a··· 或 P→Qa··· , 则
a∈FIRSTVT(P)
```

2. 若 a∈FIRSTVT(Q), 且有产生式 P→Q···, |则 a∈FIRSTVT(P)

```
FOR 每个非终结符 P 和终结符 a DO
  F[P, a]:=FALSE;
FOR 每个形如 P→a···或 P→Qa···的产生式
DO
  INSERT(P, a);
WHILE STACK 非空 DO
BEGIN
  把 STACK 的顶项,记为 (Q, a),上托出去
  FOR 每条形如 P→Q··· 的产生式 DO
     INSERT(P, a);
```



■ 这个算法的工作结果得到一个二维数组 F,从它可得任何非终结符 P的 FIRSTVT。

 $FIRSTVT(P) = \{a \mid F[P, a]=TRUE\}$

■同理,可构造计算 LASTVT 的算法。

构造集合 LASTVT(P) 的算法

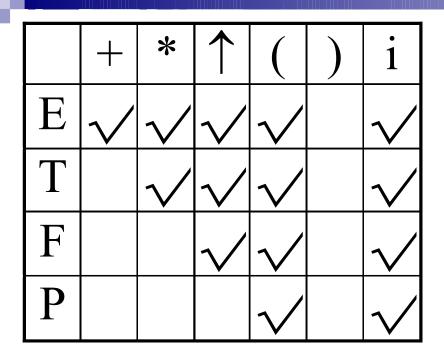
$$LASTVT(P) = \{a \mid P \stackrel{+}{\Rightarrow} \cdots a, \mathfrak{R}P \stackrel{+}{\Rightarrow} \cdots aQ, a \in V_T \overline{m}Q \in V_N \}$$

- 反复使用下面两条规则构造集合 LASTVT(P)
 - 1. 若有产生式 P→··· a 或 P→ ··· aQ ,则 a∈ LASTVT(P);
 - 2. 若 a∈ LASTVT(Q) ,且有产生式 P→··· Q ,则 a∈ LASTVT(P) 。

- 100
- 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) $P \rightarrow (E) \mid i$

计算文法 G 的 FIRSTVT 和 LASTVT.

- 1. 若有产生式 P→··· a 或 P→ ··· aQ ,则 a∈ LASTVT(P);
- 2. 若 a∈ LASTVT(Q) , 且有产生式 P→··· Q , 则 a∈ LASTVT(P) 。



| | + | * | | (|) | i |
|---|---|----------|----------|---|----------|----------|
| E | | | | | | |
| T | | / | / | | / | / |
| F | | | | | | |
| P | | | | | / | |

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

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

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

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

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

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

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

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



■使用每个非终结符 P 的 FIRSTVT(P) 和 LASTVT(P),就能够构造文法 G 的优先 表



- 通过检查 G 的每个产生式的每个候选式,可找出 所有满足 a b 的终结符对
- 通过检查每个产生式的候选式确定满足关系 和 的所有终结符对
 - □假定有个产生式的一个候选形为

···aP···

那么,对任何 b∈FIRSTVT(P),有 a b

□假定有个产生式的一个候选形为

...Pb...

那么,对任何 a∈LASTVT(P) ,有 a b

M

FOR 每条产生式 P→X₁X₂···X_n DO FOR i:=1 TO n-1 DO BEGIN

= $\sim \square \times$

IF X_i和 X_{i+1} 均为终结符 THEN 置 X_i◆X_{i+1}

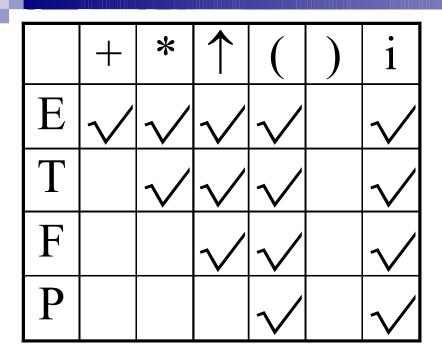
IF i≤n-2 且 X_i 和 X_{i+2} 都为终结符
但 X_{i+1} 为非终结符 THEN 置 X_i◆X_{i+2}

•

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•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

- 100
- 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. 计算文法 G 的 FIRSTVT 和 LASTVT;
 - 1. 若有产生式 P→··· a 或 P→ ··· aQ ,则 a∈ LASTVT(P);
 - 2. 若 a∈ LASTVT(Q) ,且有产生式 P→··· Q ,则 a∈ LASTVT(P) 。



| | + | * | ^ | (|) | i |
|---|---|----------|----------|---|----------|----------|
| E | | | | | | |
| T | | / | / | | / | / |
| F | | | | | / | |
| P | | | | | / | / |

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

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

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

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

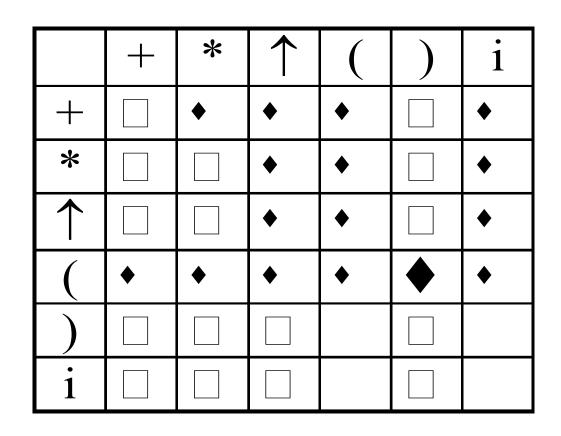
$$LASTVT(T) = \{*, \uparrow,), i\}$$

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

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

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

G的算符优先关系表



彎结论:G是算符优先文法



- 计算 FIRSTVT 和 LASTVT(P) 集合
- ■构造算符优先关系表