

## 1. 预备知识

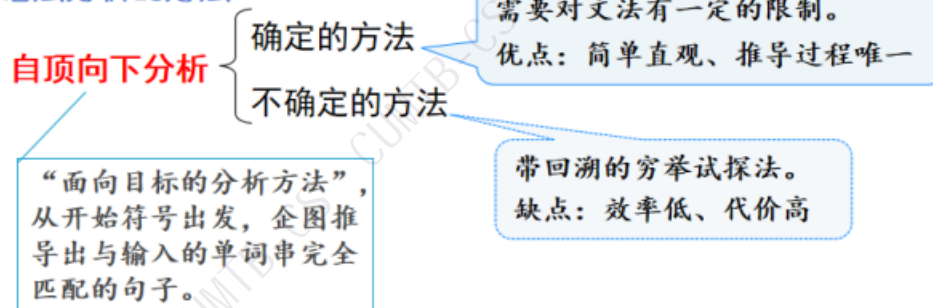
能使用自顶向下分析技术的文法必须是 **LL(1) 文法**

1. LL(1) 文法的充要条件
2. LL(1) 文法的判别方法
3. 某些 **非LL(1) 文法** 到 **LL(1) 文法** 的等价变换
  - 提取左公共因子
  - 消除左递归（直接左递归、间接左递归）
4. 不确定的自顶向下分析思想
5. 确定的自顶向下分析方法
  - 递归子程序法
  - **表驱动分析法（预测分析法）**

[①判别LL(1) 文法；②构造分析表；③分析输入串]

1. **语法分析的作用**：识别单词符号序列是否是给定文法的正确程序。

2. **语法分析的方法**：



**自底向上分析**

- 算符优先分析
- LR分析

2. 确定的自顶向下分析的主要思想：

- (1) 从文法开始符号出发，如何根据当前的单词符号，
- (2) 唯一地确定选用哪个产生式来替换相应的非终结符向下推导。

3. 确定的自顶向下分析文法的特点

- (1) 模式一：为保证推导过程唯一，该文法的特点是：
  - ① 每个产生式的右部都由终结符号开始。
  - ② 左部相同的产生式，它们的右部由不同的终结符开始。

【举例】

例1: 文法

$G[S]: S \rightarrow pA$

$S \rightarrow qB$

$A \rightarrow cAd$

$A \rightarrow a$

$B \rightarrow dB$

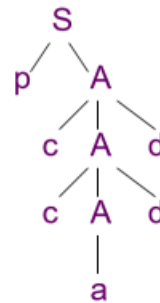
$B \rightarrow b$

分析句子 pccadd

推导过程:

$S \Rightarrow pA \Rightarrow pcAd \Rightarrow pccAdd \Rightarrow pccadd$

对应的语法树:



(1) 模式二. 为保证推导过程唯一, 该文法的特点是:

① 无空产生式 ( $\rightarrow \epsilon$ )

② 左部相同的产生式, 其右部的 **First 集** 不相交。

【举例】

例2:

文法  $G[S]$ :

$S \rightarrow Ap$

$S \rightarrow Bp$

$A \rightarrow a$

$A \rightarrow cA$

$B \rightarrow b$

$B \rightarrow dB$

分析句子 ccap

推导过程:

$S \Rightarrow \overset{\star}{A}p \Rightarrow cAp \Rightarrow ccAp \Rightarrow ccap$

语法树:



#### ● First 集合

定义: 设  $G = (V_T, V_N, S, P)$  是上下文无关文法,

$\text{First}(\alpha) = \{a \mid \alpha \xRightarrow{*} a\beta, a \in V_T, \alpha, \beta \in V^*\}$

若  $\alpha \xRightarrow{*} \epsilon$ , 则规定  $\epsilon \in \text{FIRST}(\alpha)$

【举例】对例2文法产生式的右部求 First 集

$\text{First}(Ap) = \{a, c\}$

$\text{First}(Bp) = \{b, d\}$

$\text{First}(a) = \{a\}$

$\text{First}(cA) = \{c\}$

$\text{First}(b) = \{b\}$

$\text{First}(dB) = \{d\}$

(2) 模式三. 为保证推导过程唯一, 该文法的特点是:

- ① 文法中包含空产生式。( $\rightarrow \epsilon$ )
- ② 任意两个左部相同的产生式  $A \rightarrow \alpha$  和  $A \rightarrow \beta$  ( $A \in V_N$ ), 满足  $SELECT(A \rightarrow \alpha) \cap SELECT(A \rightarrow \beta) = \Phi$

【举例】可以使用“确定自顶向下推导”的文法

例3: 文法G[S]:

$S \rightarrow aA$

$S \rightarrow d$

$A \rightarrow bAS$

$A \rightarrow \epsilon$

分析句子 abd

推导过程:

$S \Rightarrow aA \Rightarrow abAS \Rightarrow abS \Rightarrow abd$

例3 自顶向下推导过程是唯一的, 也就是确定的。

引出两个定义 Follow 和 Select 集合

● Follow

定义: 设G是上下文无关文法,  $A \in V_N$ , S是开始符号。

$FOLLOW(A) = \{a | S \xRightarrow{*} \mu A \beta \text{ 且 } a \in FIRST(\beta), \mu \in V_T^*, \beta \in V^*\}$

规定  $\# \in FOLLOW(S)$

求Follow集的方法:

1.  $\# \in FOLLOW(S)$

2. 对于产生式  $A \rightarrow \alpha B \beta$

● 若  $\beta \xRightarrow{*} \epsilon$ , 则  $FIRST(\beta) \in FOLLOW(B)$

● 若  $\beta \xRightarrow{*} \epsilon$ , 则  $(FIRST(\beta) - \{\epsilon\}) \cup FOLLOW(A) \in FOLLOW(B)$

【举例】对例3中的非终结符求Follow集

第一次迭代:

$FOLLOW(S) = \{\#\} \cup FOLLOW(A)$

$= \{\#\} \cup \Phi$

$= \{\#\}$

$FOLLOW(A) = FOLLOW(S) \cup FIRST(S)$

$= \{\#\} \cup \{a, d\}$

$= \{\#, a, d\}$

第二次迭代:

$FOLLOW(S) = \{\#\} \cup FOLLOW(A)$

$= \{\#\} \cup \{\#, a, d\}$

$= \{\#, a, d\}$

$FOLLOW(A) = FOLLOW(S) \cup FIRST(S)$

$= \{\#\} \cup \{a, d\}$

$= \{\#, a, d\}$

- Select

定义：SELECT集 定义：

给定上下文无关文法的产生式  $A \rightarrow \alpha$ ,  $A \in V_N$ ,  $\alpha \in V^*$ ,

① 若  $\alpha \xRightarrow{*} \epsilon$ , 则  $\text{SELECT}(A \rightarrow \alpha) = \text{First}(\alpha)$

② 若  $\alpha \xRightarrow{*} \epsilon$ , 则  $\text{SELECT}(A \rightarrow \alpha) = (\text{First}(\alpha) - \{\epsilon\}) \cup \text{Follow}(A)$

【举例】对例 3 中的产生式求 Select 集

$\text{Select}(S \rightarrow aA) = \text{First}(aA) = \{a\}$

$\text{Select}(S \rightarrow d) = \text{First}(d) = \{d\}$

$\text{Select}(A \rightarrow bAS) = \text{First}(bAS) = \{b\}$

$\text{Select}(A \rightarrow \epsilon) = \text{Follow}(A) = \{\#, a, d\}$

- 确定的自顶向下分析文法的判定

① 文法中包含空产生式。 ( $\rightarrow \epsilon$ )

② 任意两个左部相同的产生式  $A \rightarrow \alpha$  和  $A \rightarrow \beta$  ( $A \in V_N$ ),  
满足  $\text{SELECT}(A \rightarrow \alpha) \cap \text{SELECT}(A \rightarrow \beta) = \Phi$

【举例】例 3: 文法  $G[S]$ :  $S \rightarrow aA$

$S \rightarrow d$

$A \rightarrow bAS$

$A \rightarrow \epsilon$

$\therefore \text{Select}(S \rightarrow aA) \cap \text{Select}(S \rightarrow d) = \{a\} \cap \{d\} = \Phi$

$\text{Select}(A \rightarrow bAS) \cap \text{Select}(A \rightarrow \epsilon) = \{b\} \cap \{\#, a, d\} = \Phi$

$\therefore$  该文法是确定的自顶向下分析文法

4. LL(1)文法: 可使用“自顶向下分析”的文法称为 LL(1)文法。

(1) 一个上下文无关文法是 LL(1)文法的充要条件:

对每个非终结符  $A$  的两个不同产生式  $A \rightarrow \alpha$  和  $A \rightarrow \beta$ , 满足  $\text{SELECT}(A \rightarrow \alpha) \cap \text{SELECT}(A \rightarrow \beta) = \Phi$

其中  $\alpha, \beta$  不能同时推导出  $\epsilon$

(2) LL(1)的含义

L: scan from Left 从左向右扫描输入串

L: analyze from Left: 分析过程是最左推导

1: 只需向右看一个符号便可以决定选择哪个产生式进行推导。

(3) 【举例】例 4: 判断文法  $G[S]$  是否为 LL(1)文法?

$G[S]$ :  $S \rightarrow aAS$

$S \rightarrow b$

$A \rightarrow bA$

$A \rightarrow \epsilon$

解:  $\text{Follow}(S) = \{\#\} \cup \text{Follow}(S) = \{\#\}$

$\text{Follow}(A) = \text{First}(S) \cup \text{Follow}(A) = \{a, b\}$

$\text{Select}(S \rightarrow aAS) = \{a\}$

$\text{Select}(S \rightarrow b) = \{b\}$

$\text{Select}(A \rightarrow bA) = \{b\}$

$\text{Select}(A \rightarrow \epsilon) = (\text{First}(\epsilon) - \{\epsilon\}) \cup \text{Follow}(A) = \{a, b\}$

由于

$\text{Select}(S \rightarrow aAS) \cap \text{Select}(S \rightarrow b) = \Phi$

$\text{Select}(A \rightarrow bA) \cap \text{Select}(A \rightarrow \epsilon) \neq \Phi$

故该文法不是 LL(1) 文法，不能用自顶向下分析技术。

【讨论】对输入串 ab 进行推导，可能出现什么情况？为什么？

## 5. LL(1)文法的判别

判别步骤：

1. 求出能推出  $\epsilon$  的非终结符
2. 计算 FIRST 集
3. 计算 FOLLOW 集
4. 计算 SELECT 集
5. 判别是否是 LL(1) 文法

【举例】例 5：设文法  $G[S]$  为：

$S \rightarrow AB$

$S \rightarrow bC$

$A \rightarrow \epsilon$

$A \rightarrow b$

$B \rightarrow \epsilon$

$B \rightarrow aD$

$C \rightarrow AD$

$C \rightarrow b$

$D \rightarrow aS$

$D \rightarrow c$

判断它是否是 LL(1) 文法。

解：

Step 1. 求出能推出  $\epsilon$  的非终结符

- 1) 建立一个以非终结符的个数为上限的一维数组  $X[]$ ，数组元素为非终结符，对应每个元素有一标志位；（该标志位记录能否推出  $\epsilon$ ，其值为：“未定”、“是”、“否”）
- 2) 置初值——将数组  $X[]$  中对应的每个非终结符的标记置为“未定”；
- 3) 扫描文法中的产生式
  - ① 删除所有右部含终结符的产生式，若某一非终结符为左部的所有产生式都被删除，则将数组中对应的标记值改为“否”；
  - ② 若某一非终结符的某产生式右部为  $\epsilon$ ，则数组中对应的标记置为“是”，并删除该非终结符为左部的所有产生式。

4) 扫描产生式右部的每个非终结符

- ① 若该符号在数组中对应标志为“是”，则删去该符号；若删去后，产生式右部为空，则将该产生式左部的非终结符在数组中对应的标志改为“是”，并删去该非终结符为左部的所有产生式；
- ② 若该符号在数组中对应标志为“否”，则删去该产生式；若这使该产生式左部非终结符的有关产生式都被删去，则把数组中该非终结符对应的标志改为“否”；

5) 重复4)，直至扫描完一遍文法的产生式，数组中的标志不再改变为止。

非终结符	S	A	B	C	D
初值	未定	未定	未定	未定	未定
第1次扫描		是	是		否
第2次扫描	是			否	

Step 2. 计算 FIRST 集

定义：

$\text{First}(\alpha) = \{a \mid \alpha \xRightarrow{*} a\beta, a \in V_T, \alpha, \beta \in V^*\}$

若  $\alpha \xRightarrow{*} \epsilon$ ，则  $\epsilon \in \text{First}(\alpha)$

求  $\text{First}(X)$  的算法：

- 1) 若  $X \in V_T$ ，则  $\text{first}(X) = \{X\}$
- 2) 若  $X \in V_N$ ，且有产生式  $X \rightarrow a...$ ， $a \in V_T$ ，则  $a \in \text{first}(X)$
- 3) 若  $X \in V_N$ ， $X \rightarrow \epsilon$ ，则  $\epsilon \in \text{first}(X)$
- 4) 若  $X \in V_N$ ， $X \rightarrow Y_1 Y_2 \dots Y_n$ ，其中  $Y_1, Y_2, \dots, Y_n \in V_N$ 
  - 当  $Y_1, Y_2, \dots, Y_{i-1}$  都能推导出  $\epsilon$  时 ( $1 \leq i \leq n$ )，  
则  $\text{first}(Y_i) - \{\epsilon\} \in \text{first}(X)$
  - ...
  - 当  $Y_1, Y_2, \dots, Y_n$  都能推导出  $\epsilon$  时，  
则  $\text{first}(X) = \text{first}(Y_1) \cup \text{first}(Y_2) \cup \dots \cup \text{first}(Y_n) \cup \{\epsilon\}$

$\text{First}(S) = (\text{First}(A) - \{\epsilon\}) \cup (\text{First}(B) - \{\epsilon\}) \cup \{\epsilon\} \cup \{b\}$

$= \{a, b, \epsilon\}$

$\text{First}(A) = \{b, \epsilon\}$

$\text{First}(B) = \{a, \epsilon\}$

$\text{First}(C) = \{a, b, c\}$

$\text{First}(D) = \{a, c\}$



$\text{First}(AB) = \{a, b, \epsilon\}$	$\text{First}(bC) = \{b\}$
$\text{First}(\epsilon) = \{\epsilon\}$	$\text{First}(b) = \{b\}$
$\text{First}(aD) = \{a\}$	$\text{First}(AD) = \{a, b, c\}$
$\text{First}(aS) = \{a\}$	$\text{First}(c) = \{c\}$

Step 3. 计算 FOLLOW 集

- 1) 设S为开始符号, 把  $\{\#\}$  加入  $\text{Follow}(S)$  中 ( $\#$ 为句子括号)
- 2) 若  $A \rightarrow \alpha B \beta$ , 则把  $\text{First}(\beta) - \{\epsilon\}$  加入  $\text{Follow}(B)$  中,  
如果  $\beta \xRightarrow{*} \epsilon$ , 则把  $\text{Follow}(A)$  也加入  $\text{Follow}(B)$  中。
- 3) 反复2, 直到每个  $V_N$  的  $\text{Follow}$  集不再增大为止。

$$\begin{aligned}\text{Follow}(S) &= \{\#\} \cup \text{Follow}(D) &&= \{\#\} \\ \text{Follow}(A) &= \{a\} \cup \{a, c\} \cup \text{Follow}(S) &&= \{a, c, \#\} \\ \text{Follow}(B) &= \text{Follow}(S) &&= \{\#\} \\ \text{Follow}(C) &= \text{Follow}(S) &&= \{\#\} \\ \text{Follow}(D) &= \text{Follow}(B) \cup \text{Follow}(C) &&= \{\#\}\end{aligned}$$

Step 4. 计算 SELECT 集

定义:

对于产生式  $A \rightarrow \alpha$ ,  $A \in V_N$ ,  $\alpha \in V^*$

若  $\alpha \not\xRightarrow{*} \epsilon$ , 则  $\text{Select}(A \rightarrow \alpha) = \text{First}(\alpha)$

若  $\alpha \xRightarrow{*} \epsilon$ , 则  $\text{Select}(A \rightarrow \alpha) = (\text{First}(\alpha) - \{\epsilon\}) \cup \text{Follow}(A)$

$$\begin{aligned}\text{Select}(S \rightarrow AB) &= (\text{First}(AB) - \{\epsilon\}) \cup \text{Follow}(S) = \{a, b, \#\} \\ \text{Select}(S \rightarrow bC) &= \text{First}(bC) = \{b\} \\ \text{Select}(A \rightarrow \epsilon) &= (\text{First}(\epsilon) - \{\epsilon\}) \cup \text{Follow}(A) = \{a, c, \#\} \\ \text{Select}(A \rightarrow b) &= \text{First}(b) = \{b\} \\ \text{Select}(B \rightarrow \epsilon) &= (\text{First}(\epsilon) - \{\epsilon\}) \cup \text{Follow}(B) = \{\#\} \\ \text{Select}(B \rightarrow aD) &= \text{First}(aD) = \{a\} \\ \text{Select}(C \rightarrow AD) &= \text{First}(AD) = \{a, b, c\} \\ \text{Select}(C \rightarrow b) &= \text{First}(b) = \{b\} \\ \text{Select}(D \rightarrow aS) &= \text{First}(aS) = \{a\} \\ \text{Select}(D \rightarrow c) &= \text{First}(c) = \{c\}\end{aligned}$$

Step 5. 判别是否是 LL(1)文法

左部相同的产生式的 SELECT 集的交集均为空。

$$\begin{aligned}\text{Select}(S \rightarrow AB) \cap \text{Select}(S \rightarrow bC) &= \{b\} \neq \phi \\ \text{Select}(A \rightarrow \epsilon) \cap \text{Select}(A \rightarrow b) &= \phi\end{aligned}$$

$$\text{Select}(B \rightarrow \epsilon) \cap \text{Select}(B \rightarrow aD) = \phi$$

$$\text{Select}(C \rightarrow AD) \cap \text{Select}(C \rightarrow b) = \{b\} \neq \phi$$

存在交集非空的 SELECT 集合，所以该文法不是 LL(1)文法。

## 6. 非 LL(1)文法

- ① 若文法含有左公共因子，一定不是 LL(1)文法
- ② 若文法含有直接或间接左递归，一定不是 LL(1)文法

## 7. 非 LL(1)文法到 LL(1)文法的等价变换

- ① 提取左公共因子
- ② 消除左递归

## 8. 提取左公共因子

### (1) 方法

对形如  $A \rightarrow \alpha\beta \mid \alpha\gamma$  进行等价变换为:  $A \rightarrow \alpha(\beta \mid \gamma)$

进一步变换:  $A \rightarrow \alpha A'$

$A' \rightarrow \beta \mid \gamma$

对形如  $A \rightarrow \alpha\beta_1 \mid \alpha\beta_2 \mid \dots \mid \alpha\beta_n$  进行等价变换为:

$A \rightarrow \alpha(\beta_1 \mid \beta_2 \mid \dots \mid \beta_n)$

进一步变换(引入新非终结符  $A'$ ):

$A \rightarrow \alpha A'$

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

注: 若  $A' \rightarrow \beta_1 \mid \beta_2 \mid \dots \mid \beta_n$  中仍含左公共因子, 再次提取, 直至所有的产生式不再有左公共因子。

### 【举例】

例 6: 为文法  $G[S]$  提取左公共因子

$S \rightarrow aSb$

$S \rightarrow aS$

$S \rightarrow \epsilon$

解:

$$\left. \begin{array}{l} S \rightarrow aSb \\ S \rightarrow aS \\ S \rightarrow \epsilon \end{array} \right\} S \rightarrow aS(b \mid \epsilon) \quad \left\{ \begin{array}{l} S \rightarrow aS \quad S' \\ S' \rightarrow b \mid \epsilon \\ S \rightarrow \epsilon \end{array} \right. \quad \begin{array}{l} S \rightarrow aS \quad S' \\ S \rightarrow \epsilon \\ S' \rightarrow b \\ S' \rightarrow \epsilon \end{array}$$

(2) 结论 1: 文法中不含左公共因子只是 LL(1)文法的必要条件。

即: LL(1)文法一定不含左公共因子

不含左公共因子的文法不一定是 LL(1)文法



(3) 提取隐含的左公共因子

- 1) 隐式变显式: 右部以非终结符开始的产生式, 用该非终结符对应的产生式进行相应替换。
- 2) 再用一般形式进行提取

【举例】例 7: 文法  $G[S]$  为:

$S \rightarrow aSd$

$S \rightarrow Ac$

$A \rightarrow aS$

$A \rightarrow b$

提取左公共因子。

解:

$S \rightarrow aSd$	$S \rightarrow aS(d c)$	$S \rightarrow aSA'$	$S \rightarrow aSA'$
$S \rightarrow aSc$	$S \rightarrow bc$	$S \rightarrow bc$	$S \rightarrow bc$
$S \rightarrow bc$	$A \rightarrow aS$	$A' \rightarrow d$	$A' \rightarrow d$
$A \rightarrow aS$	$A \rightarrow b$	$A' \rightarrow c$	$A' \rightarrow c$
$A \rightarrow b$		$A \rightarrow aS$	
		$A \rightarrow b$	

**A 是多余产生式!**

(4) 有些文法不能在有限步骤内提取完左公共因子

【举例】例 8: 文法  $G[S]$  为:

$S \rightarrow Ap$

$S \rightarrow Bq$

$A \rightarrow aAp$

$A \rightarrow d$

$B \rightarrow aBq$

$B \rightarrow e$

提取左公共因子。

解:

$S \rightarrow aApp$	$S \rightarrow a(App Bqq)$	$S \rightarrow aS'$
$S \rightarrow dp$	$S \rightarrow dp$	$S' \rightarrow App$
$S \rightarrow aBqq$	$S \rightarrow eq$	$S' \rightarrow Bqq$
$S \rightarrow eq$	$A \rightarrow aAp$	$S \rightarrow dp$
$A \rightarrow aAp$	$A \rightarrow d$	$S \rightarrow eq$
$A \rightarrow d$	$B \rightarrow aBq$	$A \rightarrow aAp$
$B \rightarrow aBq$	$B \rightarrow e$	$A \rightarrow d$
$B \rightarrow e$		$B \rightarrow aBq$
		$B \rightarrow e$

继续替换 $S'$ 产生式右部的 $A$ 和 $B$ ，只能使产生式无限地增加下去。

(5) 结论 2: 不是所有文法，都能在有限的步骤内提取完“左公共因子”。

## 9. 消除左递归

(1) 文法提取左公共因子后，并不一定是 LL(1)文法。

只有不含空产生式，且无左公共因子，且无左递归时，文法才是 LL(1)文法，否则需要进行判别。

(2) 左递归的形式

### ■ 直接左递归

$$A \rightarrow A\alpha \quad A \in V_N, \quad \alpha \in V^*$$

### ■ 间接左递归

$$A \rightarrow B\alpha$$

$$B \rightarrow A\beta \quad A, B \in V_N, \quad \alpha, \beta \in V^*$$

(3) 消除直接左递归的方法

把“直接左递归”改为“右递归”：

$$A \rightarrow A\alpha$$

$$A \rightarrow \beta$$

改写为右递归文法：

$$A \rightarrow \beta A'$$

$$A' \rightarrow \alpha A'$$

$$A' \rightarrow \epsilon$$

句子：

$\beta$

$\beta\alpha$

$\beta\alpha\alpha$

$\beta\alpha\alpha\alpha$

$\beta\alpha\alpha\alpha\dots\alpha$

左递归文法:

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

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

改写为右递归文法:

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

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

$$A' \rightarrow \epsilon$$

【举例】例 9: 消除文法  $G[S]$  的左递归。

$$S \rightarrow Sa$$

$$S \rightarrow b$$

解: 左递归文法改写为:

$$S \rightarrow bS'$$

$$S' \rightarrow aS'$$

$$S' \rightarrow \epsilon$$

(4) 消除间接左递归的方法

- 先把“间接左递归”变为“直接左递归”
- 再将“直接左递归”化为“右递归”

【举例】例 10: 消除文法  $G[A]$  的间接左递归。

$$A \rightarrow aB$$

$$A \rightarrow Bb$$

$$B \rightarrow Ac$$

$$B \rightarrow d$$

解: 变为直接左递归

$$A \rightarrow aB$$

$$A \rightarrow Acb$$

$$A \rightarrow db$$

$$B \rightarrow Ac$$

$$B \rightarrow d$$

“左递归”化为“右递归”

$$A \rightarrow aBA' \mid dbA'$$

$$A' \rightarrow cbA' \mid \epsilon$$

$$B \rightarrow Ac$$

$$B \rightarrow d$$

(5) 消除文法中一切左递归

```

以某种顺序将非终结符的排序为:  $A_1, A_2, \dots, A_n$ 
for ( $i=1; i \leq n; i++$ )
{
    for ( $j=1; j \leq i-1; j++$ ) //以 $A_1, \dots, A_{i-1}$ 的产生式代入 $A_i$ 产生式
    {
        若 $A_j$ 的产生式为:  $A_j \rightarrow \delta_1 | \delta_2 | \dots | \delta_k$ 
        则形如 $A_i \rightarrow A_j \gamma$ 的产生式变为:  $A_i \rightarrow \delta_1 \gamma | \delta_2 \gamma | \dots | \delta_k \gamma$ 
    }
    消除 $A_i$ 中的一切直接左递归
}
删除多余产生式

```

【举例】例 11: 消除文法  $G[S]$  的一切左递归。

$S \rightarrow Qc | c$

$Q \rightarrow Rb | b$

$R \rightarrow Sa | a$

解: 非终结符排序为:  $S, Q, R$

(1) 对于 $S$ : 无直接左递归 (不用消除)

(2) 对于 $Q$ : 右部不含 $S$ 开头的产生式  
无直接左递归 (不用消除)

(3) 对于 $R$ : 右部含 $S$ 开头的产生式, 则:

$S \rightarrow Qc$

$S \rightarrow c$

$Q \rightarrow Rb$

$Q \rightarrow b$

$R \rightarrow Qca$

$R \rightarrow ca$

$R \rightarrow a$

右部含 $Q$ 开头的产生式:

$S \rightarrow Qc$

$S \rightarrow c$

$Q \rightarrow Rb$

$Q \rightarrow b$

$R \rightarrow Rbca$

$R \rightarrow bca$

$R \rightarrow ca$

$R \rightarrow a$

消除直接左递归:

$S \rightarrow Qc$

$S \rightarrow c$

$Q \rightarrow Rb$

$Q \rightarrow b$

$R \rightarrow (bca | ca | a) R'$

$R' \rightarrow bcaR' | \epsilon$

(4) 考察**是否存在无用产生式**: 没有“无用产生式”, 所以不用删除。

消除一切左递归以后的文法为:

$S \rightarrow Qc$

$S \rightarrow c$

$Q \rightarrow Rb$

$Q \rightarrow b$

$R \rightarrow bcaR'$

$R \rightarrow caR'$

$R \rightarrow aR'$

$R' \rightarrow bcaR'$

$R' \rightarrow \epsilon$

## 10. 确定的自顶向下分析方法

(1) 递归子程序法

(2) 预测分析法 (表驱动分析法)

## 11. 递归子程序法

- (1) 主要思想：对文法中每个非终结符编写一个递归过程，每个过程的功能是识别由该非终结符推出的串，当某非终结符的产生式有多个候选时能够按 LL(1)形式可唯一地确定选择某个候选进行推导。
- (2) 优点：简单、直观、易于构造。（PL/O 的语法分析）
- (3) 缺点：对文法要求高，必须满足 LL(1)文法；由于递归调用多，速度慢，占用空间多。

## 12. 预测分析法（表驱动分析法）

### (1) 分析器的组成：

预测分析程序、先进后出栈、预测分析表

### (2) 预测分析法的步骤：

- ① 提取左公共因子，消除左递归。
- ② 判断文法是否为 LL(1)文法。
- ③ 若是，构造预测分析表；  
否则，不能进行“确定的自顶向下”分析。
- ④ 预测分析程序根据“预测分析表”，利用“分析栈”，对输入串进行分析。

### (3) 【举例】例 12：文法 $G[E]$ ： $E \rightarrow E+T \mid T$ 构造预测分析表，并分析句子 $\#i+i*i\#$ 。

$T \rightarrow T * F \mid F$

$F \rightarrow i \mid (E)$

解：

#### (1) 消除左递归：

$V_N$ 排列为  $E, T, F$

消除E一切直接左递归：

$E \rightarrow TE'$

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

$T \rightarrow T * F$

$T \rightarrow F$

$F \rightarrow i$

$F \rightarrow (E)$

消除T的一切直接左递归：

$E \rightarrow TE'$

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

$T \rightarrow FT'$

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

$F \rightarrow i$

$F \rightarrow (E)$

F没有左递归。

文法无左公共因子。

所以，提取左公共因子和消除左递归后的文法为：

$E \rightarrow TE'$

$E' \rightarrow +TE'$

$E' \rightarrow \epsilon$

$T \rightarrow FT'$

$T' \rightarrow *FT'$

$T' \rightarrow \epsilon$

$F \rightarrow i$

$F \rightarrow (E)$

(2) 判断改写后的文法是否为LL(1)文法:

- a) 求非终结符可否推导出  $\epsilon$  :
- | E | E' | T | T' | F |
|---|----|---|----|---|
| 否 | 是  | 否 | 是  | 否 |

b) 求First集:

$\text{First}(E) = \{ i, ( \}$   
 $\text{First}(E') = \{ +, \epsilon \}$   
 $\text{First}(T) = \{ i, ( \}$   
 $\text{First}(T') = \{ *, \epsilon \}$   
 $\text{First}(F) = \{ i, ( \}$   
 $\text{First}(TE') = \text{First}(T) = \{ i, ( \}$   
 $\text{First}(FT') = \text{First}(F) = \{ i, ( \}$

c) 求Follow集:

$\text{Follow}(E) = \{ \#, ) \}$   
 $\text{Follow}(E') = \text{Follow}(E) \cup \text{Follow}(E') = \{ \#, ) \}$   
 $\text{Follow}(T) = (\text{First}(E') - \{ \epsilon \}) \cup \text{Follow}(E') = \{ +, \#, ) \}$   
 $\text{Follow}(T') = \text{Follow}(T) \cup \text{Follow}(T') = \{ +, \#, ) \}$   
 $\text{Follow}(F) = (\text{First}(T') - \{ \epsilon \}) \cup \text{Follow}(T) \cup \text{Follow}(T')$   
 $\quad = \{ *, +, \#, ) \}$

d) 求各产生式的SELECT集:

$\text{SELECT}(E \rightarrow TE') = \text{First}(TE') = \{ i, ( \}$   
 $\text{SELECT}(E' \rightarrow +TE') = \text{First}(+TE') = \{ + \}$   
 $\text{SELECT}(E' \rightarrow \epsilon) = \text{Follow}(E') = \{ \#, ) \}$   
 $\text{SELECT}(T \rightarrow FT') = \text{First}(FT') = \{ i, ( \}$   
 $\text{SELECT}(T' \rightarrow *FT') = \text{First}(*FT') = \{ * \}$   
 $\text{SELECT}(T' \rightarrow \epsilon) = \text{Follow}(T') = \{ +, \#, ) \}$   
 $\text{SELECT}(F \rightarrow (E)) = \text{First}((E)) = \{ ( \}$   
 $\text{SELECT}(F \rightarrow i) = \text{First}(i) = \{ i \}$

e) 判定:

$\text{SELECT}(E' \rightarrow +TE') \cap \text{SELECT}(E' \rightarrow \epsilon) = \phi$   
 $\text{SELECT}(T' \rightarrow *FT') \cap \text{SELECT}(T' \rightarrow \epsilon) = \phi$   
 $\text{SELECT}(F \rightarrow (E)) \cap \text{SELECT}(F \rightarrow i) = \phi$

所以该文法是LL(1)文法, 可以使用预测分析法。

(3) 构造预测分析表

构造预测分析表的方法:

若  $a \in \text{SELECT}(A \rightarrow \alpha)$ , 则把  $\alpha$  放入  $M[A, a]$  中。

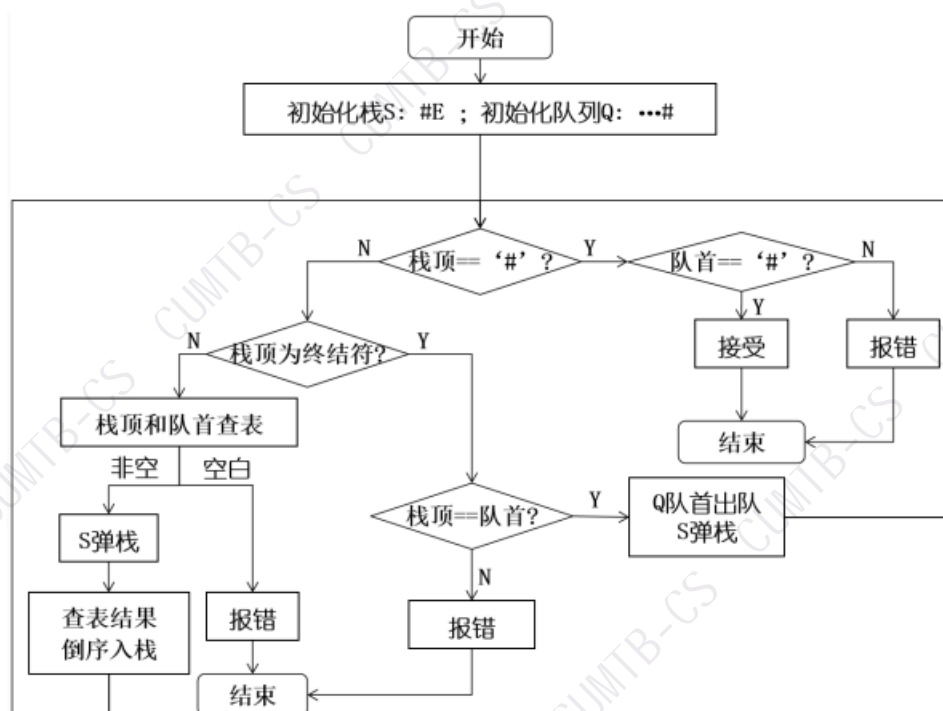
$\text{Select}(E \rightarrow TE') = \{ i, ( \}$   
 $\text{Select}(E' \rightarrow +TE') = \{ + \}$   
 $\text{Select}(E' \rightarrow \epsilon) = \{ \#, ) \}$   
 $\text{Select}(T \rightarrow FT') = \{ i, ( \}$   
 $\text{Select}(T' \rightarrow *FT') = \{ * \}$   
 $\text{Select}(T' \rightarrow \epsilon) = \{ +, \#, ) \}$   
 $\text{Select}(F \rightarrow (E)) = \{ ( \}$   
 $\text{Select}(F \rightarrow i) = \{ i \}$

	+	*	(	)	i	#
E			TE'		TE'	
E'	+TE'			$\epsilon$		$\epsilon$
T			FT'		FT'	
T'	$\epsilon$	*FT'		$\epsilon$		$\epsilon$
F			(E)		i	

注: 所有空白的单元格表示“报错”。



#### (4) 预测分析



#i+i\*i#

步骤	分析栈	剩余输入串	动作
1	#E	i+i*i#	弹栈E, 查表得到TE'倒序入栈
2	#E' T	i+i*i#	弹栈T, 查表得到FT'倒序入栈
3	#E' T' F	i+i*i#	弹栈F, 查表得到i倒序入栈
4	#E' T' i	i+i*i#	i匹配
5	#E' T'	+i*i#	弹栈T', 查表得到ε倒序入栈
6	#E'	+i*i#	弹栈E', 查表得到+TE'倒序入栈
7	#E' T +	+i*i#	+匹配
8	#E' T	i*i#	弹栈T, 查表得到FT'倒序入栈
9	#E' T' F	i*i#	弹栈F, 查表得到i倒序入栈
10	#E' T' i	i*i#	i匹配
11	#E' T'	*i#	弹栈T', 查表得到*FT'倒序入栈
12	#E' T' F *	*i#	*匹配
13	#E' T' F	i#	弹栈F, 查表得到i倒序入栈
14	#E' T' i	i#	i匹配
15	#E' T'	#	弹栈T', 查表得到ε倒序入栈
16	#E'	#	弹栈E', 查表得到ε倒序入栈
17	#	#	接受

#### 13. 不确定的自顶向下分析思想

- (1) 非 LL(1)文法不能用“确定的”自顶向下分析, 但可以使用“不确定的”自顶向下分析 (“带回溯的”自顶向下分析)

(2) 引起回溯的原因:

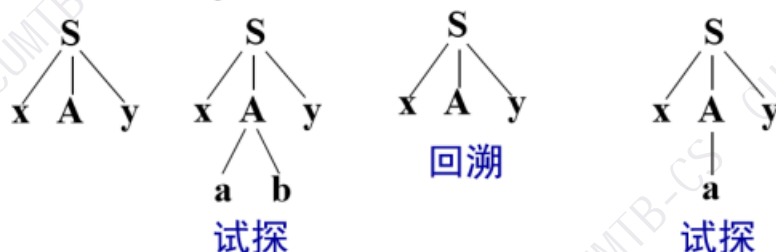
- 1) 由于左部相同的产生式的右部 First 集交集不为空。

例13: 文法G[S]:

$S \rightarrow xAy$

$A \rightarrow ab|a$

分析输入串xay是否为该文法接受。



- 2) 由于左部相同非终结符的右部能推导出 $\epsilon$ , 且该非终结符的 Follow 集中含有其右部 First 集的元素。

例14: 设文法G[S]为:

$S \rightarrow aAS$

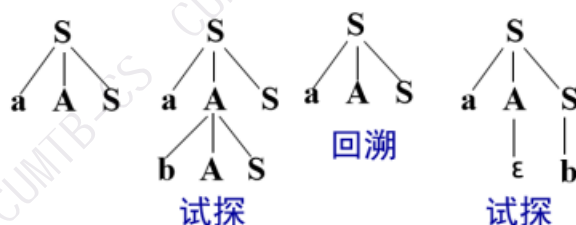
$S \rightarrow b$

$A \rightarrow bAS$

$A \rightarrow \epsilon$

分析输入串ab。

$\text{Follow}(A) = \{a, b\}$



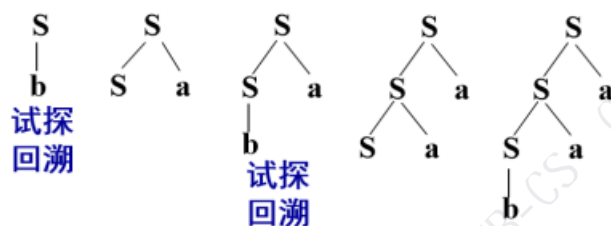
- 3) 由于文法中含有左递归。

例15: 设文法G[S]:

$S \rightarrow Sa$

$S \rightarrow b$

分析输入串baa。



【课后练习】 3(1)(2), 7(3)(4)(5)

3. 已知文法G[S]:

$S \rightarrow MH|a$

$H \rightarrow LSo|\epsilon$

$K \rightarrow dML|\epsilon$

$L \rightarrow eHf$

$M \rightarrow K|bLM$

- (1) 判断G是否是LL(1)文法, 如果是, 构造LL(1)分析表;
- (2) 分析句子 #befo#。

解: (1) ① 判断非终结符能否推导出

S H K L M  
是 是 是 否 是

② 求 First 集

$$\text{First}(S) = \{a, b, d, e, \epsilon\}$$

$$\text{First}(H) = \{e, \epsilon\}$$

$$\text{First}(K) = \{d, \epsilon\}$$

$$\text{First}(L) = \{e\}$$

$$\text{First}(M) = \{b, d, \epsilon\}$$

③ 求 Follow 集

$$\text{Follow}(S) = \{\#\} \cup \{o\} = \{\#, o\}$$

$$\text{Follow}(H) = \text{Follow}(S) \cup \{f\} = \{\#, o, f\}$$

$$\text{Follow}(K) = \text{Follow}(M) = \{\#, o, e\}$$

$$\begin{aligned} \text{Follow}(L) &= (\text{First}(S) - \{\epsilon\}) \cup \{o\} \cup \text{Follow}(K) \cup (\text{First}(M) - \{\epsilon\}) \cup \text{Follow}(M) \\ &= \{\#, a, b, e, o, d\} \end{aligned}$$

$$\text{Follow}(M) = (\text{First}(H) - \{\epsilon\}) \cup \text{Follow}(S) \cup \text{First}(L) \cup \text{Follow}(M) = \{e, \#, o\}$$

④ 求 Select 集

$$\text{Select}(S \rightarrow MH) = (\text{First}(MH) - \{\epsilon\}) \cup \text{Follow}(S) = \{b, d, e, \#, o\}$$

$$\text{Select}(S \rightarrow a) = \{a\}$$

$$\text{Select}(H \rightarrow LSo) = \text{First}(LSo) = \{e\}$$

$$\text{Select}(H \rightarrow \epsilon) = \text{Follow}(H) = \{\#, o, f\}$$

$$\text{Select}(K \rightarrow dML) = \text{First}(dML) = \{d\}$$

$$\text{Select}(K \rightarrow \epsilon) = \text{Follow}(K) = \{\#, o, e\}$$

$$\text{Select}(L \rightarrow eHf) = \text{First}(eHf) = \{e\}$$

$$\text{Select}(M \rightarrow K) = (\text{First}(K) - \{\epsilon\}) \cup \text{Follow}(M) = \{d, e, \#, o\}$$

$$\text{Select}(M \rightarrow bLM) = \text{First}(bLM) = \{b\}$$

⑤ 判别

$$\text{Select}(S \rightarrow MH) \cap \text{Select}(S \rightarrow a) = \{b, d, e, \#, o\} \cap \{a\} = \phi$$

$$\text{Select}(H \rightarrow LSo) \cap \text{Select}(H \rightarrow \epsilon) = \{e\} \cap \{\#, o, f\} = \phi$$

$$\text{Select}(K \rightarrow dML) \cap \text{Select}(K \rightarrow \epsilon) = \{d\} \cap \{\#, o, e\} = \phi$$

$$\text{Select}(M \rightarrow K) \cap \text{Select}(M \rightarrow bLM) = \{d, e, \#, o\} \cap \{b\} = \phi$$

故该文法是 LL(1) 文法。

⑥ 构造该文法的预测分析表

	a	b	d	e	f	o	#
S	a	MH	MH	MH		MH	MH
H				LSo	$\epsilon$	$\epsilon$	$\epsilon$
K			dML	$\epsilon$		$\epsilon$	$\epsilon$
L				eHf			
M		bLM	K	K		K	K

(2) 分析句子#befo#

序号	分析栈	待输入串	动作
1	#S	befo#	弹栈 S, MH 倒序入栈
2	#HM	befo#	弹栈 M, bLM 倒序入栈
3	#HMLb	befo#	b 匹配
4	#HML	efo#	弹栈 L, eHf 倒序入栈
5	#HMfHe	efo#	e 匹配
6	#HMfH	fo#	弹栈 H, ε 倒序入栈
7	#HMf	fo#	f 匹配
8	#HM	o#	弹栈 M, K 倒序入栈
9	#HK	o#	弹栈 K, ε 倒序入栈
10	#H	o#	弹栈 H, ε 倒序入栈
11	#	o#	报错

## 7. 消除左递归, 提取左公共因子

(3)  $S \rightarrow Aa$      $S \rightarrow Saba$      $S \rightarrow bS'$   
 $S \rightarrow b$      $S \rightarrow b$      $S' \rightarrow abaS'$   
 $A \rightarrow SB$      $S' \rightarrow \epsilon$   
 $B \rightarrow ab$

(4)  $S \rightarrow AS$      $S \rightarrow SAS$      $S \rightarrow aSS'$      $S \rightarrow aSS'$      $S \rightarrow aSS'$   
 $S \rightarrow b$      $S \rightarrow aS$      $S \rightarrow bS'$      $S \rightarrow bS'$      $S \rightarrow bS'$   
 $A \rightarrow SA$      $A \rightarrow SA$      $S' \rightarrow ASS'$      $S' \rightarrow ASS'$      $S' \rightarrow ASS'$   
 $A \rightarrow a$      $A \rightarrow a$      $S' \rightarrow \epsilon$      $S' \rightarrow \epsilon$      $S' \rightarrow \epsilon$   
 $A \rightarrow SA$      $A \rightarrow a$      $A \rightarrow SA$      $A \rightarrow aSS'A$      $A \rightarrow aA'$   
 $A \rightarrow a$      $A \rightarrow a$      $A \rightarrow bS'A$      $A \rightarrow a$      $A \rightarrow bS'A$   
 $A' \rightarrow SS'A$   
 $A' \rightarrow \epsilon$

(5)  $S \rightarrow Ab$      $S \rightarrow aAb$      $S \rightarrow aS'$      $S \rightarrow aS'$      $S \rightarrow aS'$   
 $S \rightarrow Ba$      $S \rightarrow ab$      $S' \rightarrow Ab$      $S' \rightarrow aA'b$      $S' \rightarrow aS''$   
 $A \rightarrow aA$      $S \rightarrow aa$      $S' \rightarrow a$      $S' \rightarrow a$      $S' \rightarrow b$   
 $A \rightarrow a$      $A \rightarrow aA$      $S' \rightarrow b$      $S' \rightarrow b$      $S'' \rightarrow A'b$   
 $B \rightarrow a$      $A \rightarrow a$      $A \rightarrow aA'$      $A \rightarrow aA'$      $S'' \rightarrow \epsilon$   
 $A' \rightarrow A$      $A' \rightarrow A$      $A' \rightarrow \epsilon$      $A' \rightarrow A$      $A' \rightarrow aA'$   
 $A' \rightarrow \epsilon$      $A' \rightarrow \epsilon$      $A' \rightarrow \epsilon$

## 【本章小结】

### 1. “能够使用自顶向下分析技术的文法必须是LL(1)文法。”

LL(1)文法充要条件：对每个非终结符A的两个不同产生式  $A \rightarrow \alpha$  和  $A \rightarrow \beta$ ，则  $\text{Select}(A \rightarrow \alpha) \cap \text{Select}(A \rightarrow \beta) = \phi$

### 2. LL(1)文法的判别

(1) 求出能推导出 $\epsilon$ 的非终结符

(2) 求First集

(3) 求Follow集

$\# \in \text{Follow}(S)$

对于产生式  $A \rightarrow \alpha B \beta$

✓ 若  $\beta \not\Rightarrow \epsilon$ ,  $\text{First}(\beta) \in \text{Follow}(B)$

✓ 若  $\beta \Rightarrow \epsilon$ ,  $(\text{First}(\beta) - \{\epsilon\}) \cup \text{Follow}(A) \in \text{Follow}(B)$

(4) 计算SELECT集

对于产生式  $A \rightarrow \alpha$

✓ 若  $\alpha \not\Rightarrow \epsilon$ ,  $\text{Select}(A \rightarrow \alpha) = \text{First}(\alpha)$

✓ 若  $\alpha \Rightarrow \epsilon$ ,  $\text{Select}(A \rightarrow \alpha) = (\text{First}(\alpha) - \{\epsilon\}) \cup \text{Follow}(A)$

(5) 判别：左部相同的产生式，Select集两两相交为空。

### 3. 非LL(1)文法到LL(1)文法的等价变换

- 提取左公共因子的方法（显式、隐式）

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

转换为：

$A \rightarrow \alpha A'$

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

- 消除左递归的方法（消除直接左递归、消除间接左递归）

$A \rightarrow A \alpha_1 \mid A \alpha_2 \mid \dots$

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

转换为：

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

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

$A' \rightarrow \epsilon$

### 4. 确定的自顶向下分析方法

- 递归子程序法
- 预测分析法(表驱动分析法)
  - ① 判定LL(1)文法
  - ② 构造预测分析表
  - ③ 用预测分析程序分析

### 5. 不确定的自顶向下分析方法

- 带回溯的自顶向下分析