
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
-

-
- 单词的描述工具
 - 正规文法（判断一个文法是否正规文法）
 - 正规式（给定字母表 Σ ，写出 Σ 上合法的正规式）、正规集（给定正规式，列出它对应的正规集；给定正规集，列出它对应的正规式）
 - 正规文法与正规式的相互转换（二者的相互转换）
 - 单词的识别机制
 - 确定有穷自动机
 - 不确定有穷自动机
-

自动识别单词的方法：

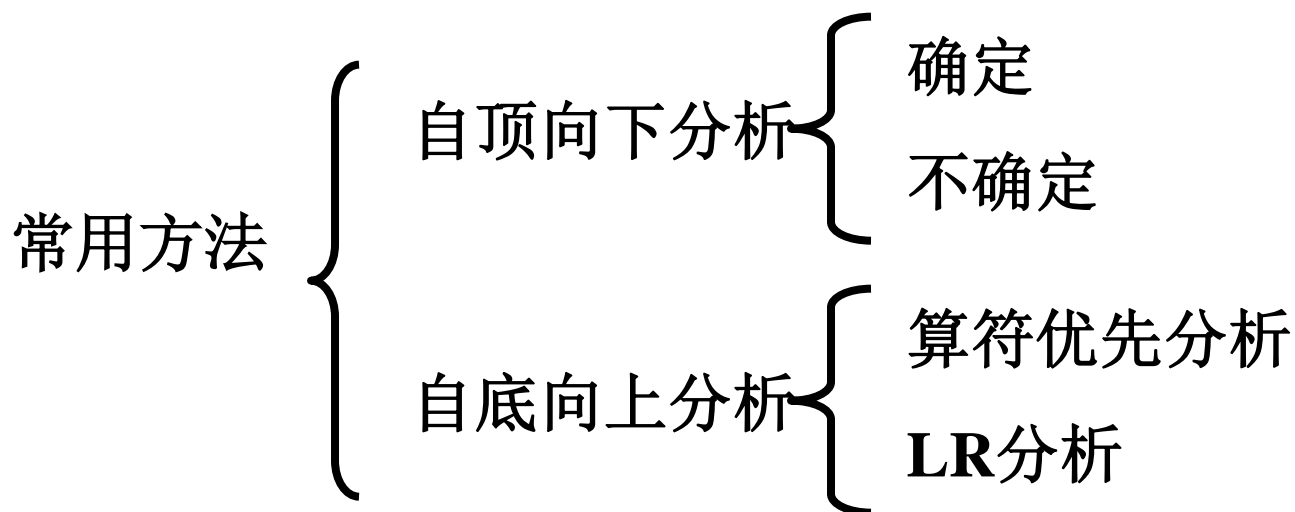
- (1) 把单词的结构用**正规式**描述；
 - (2) **把正规式转换为一个NFA**；
 - (3) **把NFA转换为相应的DFA**；
 - (4) **基于DFA构造词法分析程序。**
-

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
-

第5章 自顶向下语法分析方法

语法分析是编译程序的核心部分：在词法分析的基础上，识别单词符号序列是否是给定文法的正确句子（程序）。



自顶向下语法分析方法

- 自顶向下分析法就是从文法的**开始符号**出发，试图**推导出**与输入的单词串完全匹配的句子。
 - 如果能够推导出，则该输入串是给定文法的句子；
 - 如果不能推导出，则该输入串不是给定文法的句子。

自顶向下语法分析要解决的关键问题

- ▶ 假定要被代换的最左非终结符号是**B**，且有**n**条规则： $B \rightarrow A_1 | A_2 | \dots | A_n$ ，那么如何确定用哪条规则去替代**B**？

自顶向下语法分析方法

- 自顶向下分析法分**确定性和不确定性**两种。
 - **自顶向下的确定性分析法**对文法有一定的限制，但实现简单直观，便于手工或自动构造；
 - **自顶向下的不确定性分析法**是带有回溯的分析方法，效率低，代价高，极少使用。
-

第5章 自顶向下语法分析方法

- 一、确定的自顶向下分析思想
 - 二、LL(1)文法的判别
 - 三、某些非LL(1)文法到LL(1)文法等价变换
 - 四、不确定的自顶向下分析思想
 - 五、确定的自顶向下分析方法
-

自顶向下语法分析要解决的关键问题

- ▶ 假定要被代换的最左非终结符号是**B**，且有**n**条规则： $B \rightarrow A_1 | A_2 | \dots | A_n$ ，那么如何确定用哪个右部去替代**B**？

一、确定的自顶向下分析思想

1、方法：从开始符号出发，不断替换非终结符，根据当前的单词符号就可以唯一选定要替换的产生式。

例1：文法G(S)： $S \rightarrow pA$

$S \rightarrow qB$

$A \rightarrow cAd$

$A \rightarrow a$

输入串 $W = pccadd$

自顶向下的推导过程为：

例1：文法 $G(S)$ ： $S \rightarrow pA$

$S \rightarrow qB$

$A \rightarrow cAd$

$A \rightarrow a$

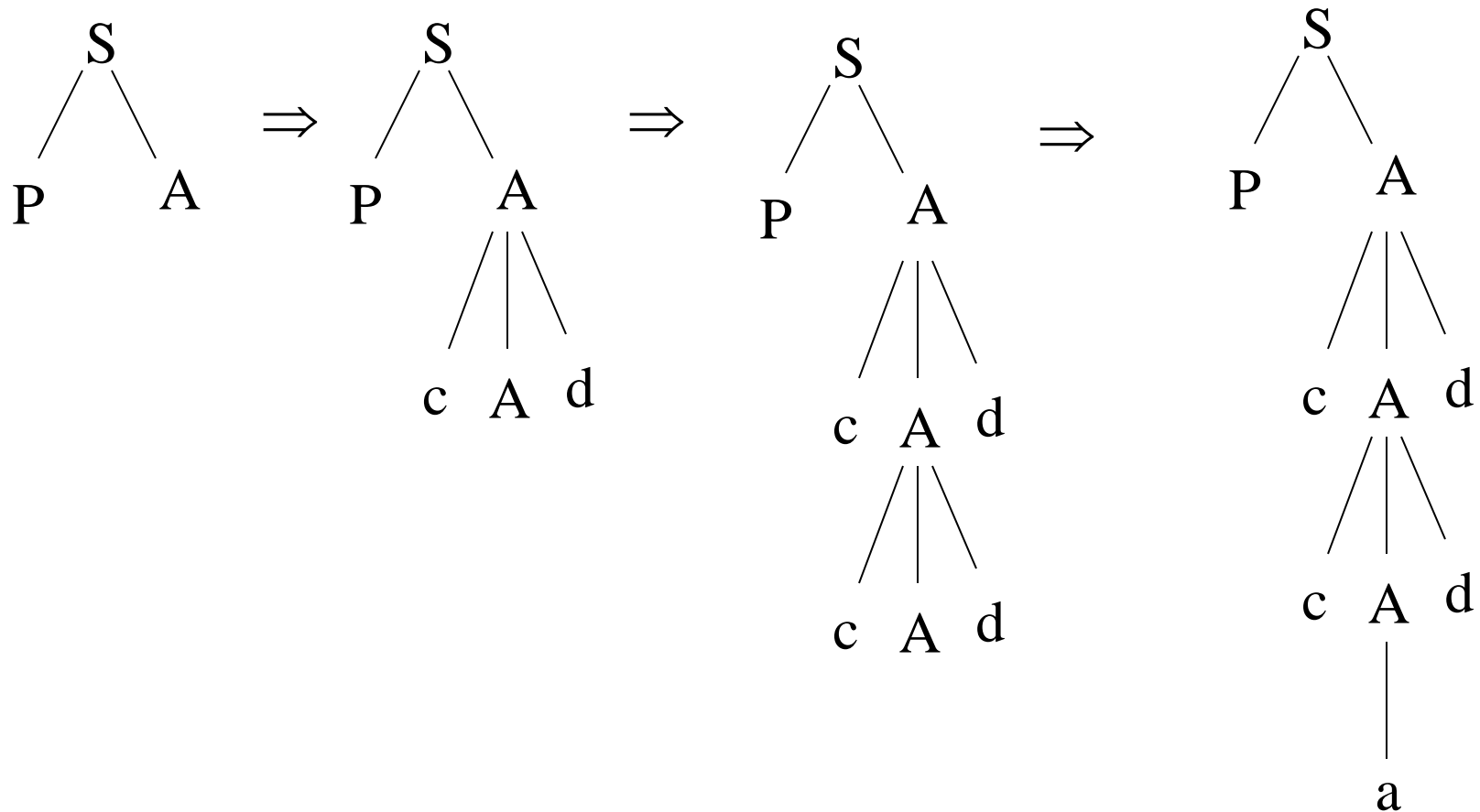
该文法的特点：

- (1) 每个产生式的右部都由终结符号开始；
- (2) 如果两个产生式有相同的左部，则它们的右部由不同的终结符开始。

对于这样的文法，其推导过程可以根据当前的输入符号决定选择哪个产生式往下推导，因此，分析过程是唯一确定的。

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

相应的语法树:



例2: 文法G(S)为: $S \rightarrow Ap$

$S \rightarrow Bq$

$A \rightarrow a \mid cA$

$B \rightarrow b \mid dB$

该文法的特点:

- (1) 产生式的右部不全是由终结符号开始;
- (2) 如果两个产生式有相同的左部, 则它们的右部由不同的终结符或非终结符开始。
- (3) 无空产生式。

ccap

如何根据输入串的第1个符号来确定产生式呢?

例2: 文法G(S)为: $S \rightarrow Ap$

$S \rightarrow Bq$

$A \rightarrow a \mid cA$

$B \rightarrow b \mid dB$

当输入W=ccap推导:

自顶向下的推导过程为:



针对产生式规则的右部产生开始符号集合

2、开始符号集合的定义：

设 $G = (V_T, V_N, P, S)$ 是上下文无关文法，

开始符号集合为 $\text{First}(\alpha) = \{ a \mid \alpha \xRightarrow{*} a\beta, a \in V_T, \alpha, \beta \in V^* \}$

规则右部 α 的开始符号集包括所有终结符 a ，使得规则右部 α 经过若干推导后得到的字符串以 a 为起始。

若 $\alpha \xRightarrow{*} \varepsilon$ ，则规定 $\varepsilon \in \text{First}(\alpha)$ 。

例3：上例中文法是： $S \rightarrow Ap \mid Bq$

$$A \rightarrow a \mid cA$$
$$B \rightarrow b \mid dB$$

则： $\text{FIRST}(Ap) = ?$; $\text{FIRST}(Bq) = ?$

针对产生式规则的右部产生开始符号集合

2、开始符号集合的定义：

设 $G = (V_T, V_N, P, S)$ 是上下文无关文法，

开始符号集合为 $\text{First}(\alpha) = \{ a \mid \alpha \xRightarrow{*} a\beta, a \in V_T, \alpha, \beta \in V^* \}$ 规则右部 α 的开始符号集包括所有终结符 a ，使得规则右部 α 经过若干推导后得到的字符串以 a 为起始。

若 $\alpha \xRightarrow{*} \varepsilon$ ，则规定 $\varepsilon \in \text{First}(\alpha)$ 。

例3：上例中文法是： $S \rightarrow Ap \mid Bq$

$$A \rightarrow a \mid cA$$
$$B \rightarrow b \mid dB$$

则： $\text{FIRST}(Ap) = \{a, c\}$ ； $\text{FIRST}(Bq) = \{b, d\}$

-
- 如何根据输入串的第1个符号来确定产生式呢？
 - 根据当前输入符号属于哪个产生式右部的开始符号集合而决定选择相应产生式进行推导。
 - 再看一遍例2

例2: 文法G(S)为: $S \rightarrow Ap$

$S \rightarrow Bq$

$A \rightarrow a \mid cA$

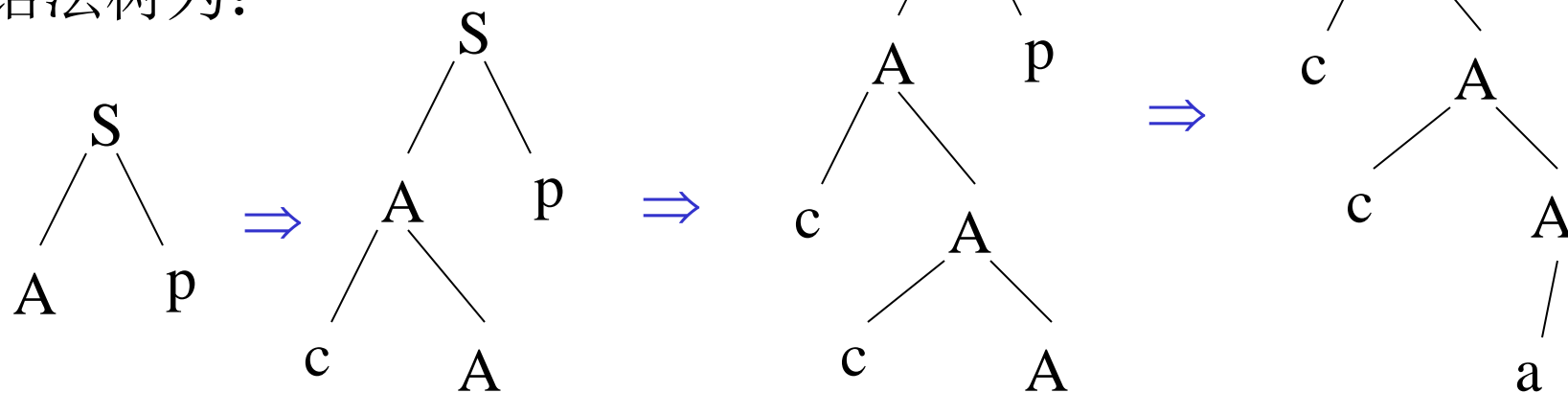
$B \rightarrow b \mid dB$

当输入W=ccap推导:

自顶向下的推导过程为:

$S \Rightarrow Ap \Rightarrow cAp \Rightarrow ccAp \Rightarrow ccap$

语法树为:



例3: 文法 $G[S]$: $S \rightarrow aA \mid d$

$A \rightarrow bAS \mid \varepsilon$

若输入 $W=abd$, 则推导过程为:



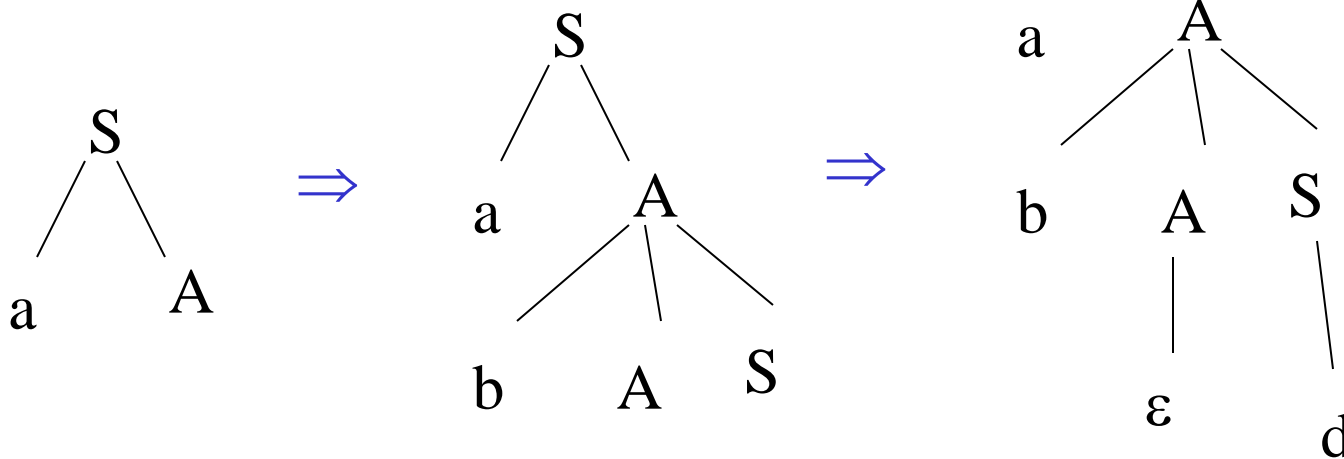
例3: 文法G[S] : $S \rightarrow aA \mid d$

$A \rightarrow bAS \mid \varepsilon$

若输入 $W=abd$, 则推导过程为:

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

语法树为:



当A的产生式右部的开始符号集合不含d, 但含有空时, 下一步推导的产生式依赖A后面的符号S确定

当某一非终结符的产生式中含有空产生式时，第二步推导的产生式该如何确定呢？根据后跟符号集合确定

3、后跟符号集合的定义：

设 $G = (V_T, V_N, P, S)$ 是上下文无关文法， $A \in V_N$ ， S 是开始符号，

$\text{Follow}(A) = \{a \mid S \xRightarrow{*} uA\beta \text{ 且 } a \in V_T, a \in \text{First}(\beta), u \in V_T^*, \beta \in V^+\}$ 。针对非终结符

若 $S \xRightarrow{*} uA\beta$ ，且 $\beta \xRightarrow{*} \varepsilon$ ，则 $\# \in \text{Follow}(A)$

($\#$ 表示输入串的结束符，或句子括号)

可写成为：

$\text{Follow}(A) = \{a \mid S \xRightarrow{*} \dots Aa \dots, a \in V_T\}$

若 $S \xRightarrow{*} \dots A$ ，则 $\# \in \text{Follow}(A)$ 。

Follow(A)是所有句型中出现在紧接A之后的终结符或“#”。

例4：在例2中文法G[S]为： $S \rightarrow Ap \mid Bq$

$A \rightarrow a \mid cA$

$B \rightarrow b \mid dB$

求Follow集。

解：Follow(S)={ ? }

Follow(A)={ ? }

Follow(B)={ ? }

换句话说：

$\text{Follow}(A)$ 是所有句型中出现在紧接A之后的终结符或“#”。

例4：在例2中文法G[S]为： $S \rightarrow Ap \mid Bq$

$$A \rightarrow a \mid cA$$

$$B \rightarrow b \mid dB$$

求Follow集。

解： $\text{Follow}(S) = \{\#\}$

$$\text{Follow}(A) = \{p\}$$

$$\text{Follow}(B) = \{q\}$$

求例3的Follow集

例3： $S \rightarrow aA \mid d ; A \rightarrow bAS \mid \varepsilon$

自上而下语法分析要解决的关键问题

- ▶ 假定要被代换的最左非终结符号是**B**，且有**n**条规则： $B \rightarrow A_1 | A_2 | \dots | A_n$ ，那么如何确定用哪个右部去替代**B**？
- ▶ 根据规则的选择集合来确定。

4、选择集合的定义

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

$\alpha \in V^*$, 若 $\alpha \xrightarrow{*} \varepsilon$, 则 $\text{Select}(A \rightarrow \alpha) = \text{First}(\alpha)$,

若 $\alpha \not\xrightarrow{*} \varepsilon$, 则 $\text{Select}(A \rightarrow \alpha) = (\text{First}(\alpha) - \{\varepsilon\}) \cup \text{Follow}(A)$

给定输入串, 根据产生式规则的选择集合选择产生式进行推导。

5、LL(1)文法的含义

- **LL(1)**的含义：从左**L**向右扫描输入串，分析过程中采用最左**L**推导，只需向右看**1**个符号就可确定如何推导（选择哪个产生式进行推导）。

一个上下文无关文法是LL(1)文法的充分必要条件是：

对每个非终结符A的两个不同产生式：

$A \rightarrow \alpha$ ， $A \rightarrow \beta$ 满足 $\text{Select}(A \rightarrow \alpha) \cap \text{Select}(A \rightarrow \beta) = \emptyset$ ，
其中 α 、 β 不同时推出 ϵ

• 只有对满足LL(1)文法的句子，才能进行确定的自顶向下分析：

给定输入串，就可以根据产生式规则的选择集合确定唯一的产生式进行推导。

例5: 上例3文法: $S \rightarrow aA \mid d$

$A \rightarrow bAS \mid \varepsilon$

证明该文法为LL(1)文法。



例5：上例3文法： $S \rightarrow aA \mid d$

$A \rightarrow bAS \mid \varepsilon$

$S \rightarrow Aa$

~~$aA \Rightarrow \varepsilon$~~

证明该文法为LL(1)文法。

不难看出： $\text{Select}(S \rightarrow aA) = \text{First}(aA) = \{a\}$

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

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

$\text{Select}(A \rightarrow \varepsilon) = (\text{first}(\varepsilon) - \{\varepsilon\}) \cup \text{follow}(A) = \{a, d, \#\}$

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

$\text{Select}(A \rightarrow bAS) \cap \text{Select}(A \rightarrow \varepsilon)$

$= \{b\} \cap \{a, d, \#\} = \{ \phi \}$

所以上述文法是LL(1)文法。

$S \Rightarrow aA \Rightarrow abAS \Rightarrow abAd$
 $S \Rightarrow aA \Rightarrow abAS \Rightarrow abAaA$
所以 $\text{Follow}(A) = \{a, d, \#\}$

例：设文法G[S]为：

$$S \rightarrow aAS \mid b$$

$$A \rightarrow bA \mid \varepsilon$$

判别是否是LL(1)文法。



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

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

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

$\text{Select}(A \rightarrow \varepsilon) = (\text{first}(\varepsilon) - \{\varepsilon\}) \cup \text{follow}(A) = \{a, b\}$

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

$\text{Select}(A \rightarrow bA) \cap \text{Select}(A \rightarrow \varepsilon) = \{b\} \cap \{a, b\} \neq \emptyset$

因此, 该文法不是LL(1)文法, 因而也就不可能用确定的自顶向下分析。

-
- 当需要选用自顶向下分析技术时，首先必须判定所给的文法是否是LL(1)文法。



二、LL(1)文法的判别

例：若文法G[S]为： $S \rightarrow AB \mid bC$

$A \rightarrow \varepsilon \mid b$

$B \rightarrow \varepsilon \mid aD$

$C \rightarrow AD \mid b$

$D \rightarrow aS \mid c$

判别文法是否是LL(1)文法。

解： (1) 求出能推出 ε 的非终结符

(2) 计算first集：

$$S \rightarrow AB \mid bC$$

$$A \rightarrow \varepsilon \mid b$$

$$B \rightarrow \varepsilon \mid aD$$

$$C \rightarrow AD \mid b$$

$$D \rightarrow aS \mid c$$

解： (1) 求出能推出 ε 的非终结符

| 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 | |

(1) 将数组 $X[]$ 中对应每一非终结符的标记置初值为“未定”。

(2) 扫描语法中的产生式。

① 删除所有右部含有终结符的产生式,若这使得以某一非终结符为左部的所有产生式都被删除,则将数组中对应该非终结符的标记值改为“否”,说明该非终结符不能推出 ε 。

② 若某一非终结符的某一产生式右部为 ε ,则将数组中对应该非终结符的标志置为“是”,并从语法中删除该非终结符的所有产生式。例中对应非终结符 A、B 的标志改为“是”。

(3) 扫描产生式右部的每一符号。

① 若所扫描到的非终结符号在数组中对应的标志是“是”,则删去该非终结符,若这使产生式右部为空,则对产生式左部的非终结符在数组中对应的标志改为“是”,并删除该非终结符为左部的所有产生式。

② 若所扫描到的非终结符号在数组中对应的标志是“否”,则删去该产生式,若这使产生式左部非终结符的有关产生式都被删去,则把在数组中该非终结符对应的标志改成“否”。

(4) 重复(3),直到扫描完一遍语法的产生式,数组中非终结符对应的特征再没有改变为止。

二、LL(1)文法的判别

例：若文法G[S]为： $S \rightarrow AB \mid bC$

$A \rightarrow \varepsilon \mid b$

$B \rightarrow \varepsilon \mid aD$

$C \rightarrow AD \mid b$

$D \rightarrow aS \mid c$

判别文法是否是LL(1)文法。

解： (1) 求出能推出 ε 的非终结符

(2) 计算first集：

方法一： 计算first集合的算法：

对于每一个文法符号 $x \in V$ ， 计算first(x)的方法如下：

a) 若 $x \in V_T$, 则 $\text{first}(x) = \{x\}$

b) 若 $x \in V_N$, 且有 $x \rightarrow a\dots$, $a \in V_T$, 则 $a \in \text{first}(x)$

c) 若 $x \in V_N$, $x \rightarrow \varepsilon$, 则 $\varepsilon \in \text{first}(x)$

d) 若 $x \in V_N$, y_1, y_2, \dots, y_i 都 $\in V_N$, 产生式
 $x \rightarrow y_1 y_2 \dots y_n$, 当 y_1, y_2, \dots, y_{i-1} 都 $\xRightarrow{*} \varepsilon$ 时 ($1 \leq i \leq n$),

则 $\text{first}(y_1) - \{\varepsilon\}$, $\text{first}(y_2) - \{\varepsilon\}$, \dots , $\text{first}(y_{i-1}) - \{\varepsilon\}$,
 $\text{first}(y_i)$ 都包含在 $\text{first}(x)$ 中。

e) 当上式中所有 $y_i \xRightarrow{*} \varepsilon$ ($1 \leq i \leq n$),

则 $\text{first}(x) = \text{first}(y_1) \cup \text{first}(y_2) \cup \dots \cup \text{first}(y_n) \cup \{\varepsilon\}$

按上面的规则可得上例文法中每个文法符号的
first集合如下：

first(S)=

first(A)=

first(B)=

first(C)=

first(D)=

$S \rightarrow AB \mid bC$

$A \rightarrow \varepsilon \mid b$

$B \rightarrow \varepsilon \mid aD$

$C \rightarrow AD \mid b$

$D \rightarrow aS \mid c$

按上面的规则可得上例文法中每个文法符号的 first 集合如下：

$$\begin{aligned} \text{first}(S) &= \{ \text{first}(A) - \{\varepsilon\} \} \cup \{ \text{first}(B) - \{\varepsilon\} \} \\ &\cup \{\varepsilon\} \cup \{b\} = \{a, b, \varepsilon\} \end{aligned}$$

$$\text{first}(A) = \{b\} \cup \{\varepsilon\} = \{b, \varepsilon\}$$

$$\text{first}(B) = \{\varepsilon\} \cup \{a\} = \{a, \varepsilon\}$$

$$\text{first}(C) = \{ \text{first}(A) - \{\varepsilon\} \} \cup \text{first}(D) \cup \text{first}(b) = \{a, b, c\}$$

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

$$S \rightarrow AB \mid bC$$

$$A \rightarrow \varepsilon \mid b$$

$$B \rightarrow \varepsilon \mid aD$$

$$C \rightarrow AD \mid b$$

$$D \rightarrow aS \mid c$$

一个文法符号串的first集合计算方法:

如果文法符号串 $\alpha \in V^*$, $\alpha = X_1 X_2 \dots X_n$,

1、当 $X_1 \xRightarrow{*} \varepsilon$, 则 $\text{first}(\alpha) = \text{first}(X_1)$

2、当对任何 j ($1 \leq j \leq i-1$, $2 \leq i \leq n$), $\varepsilon \in \text{first}(X_j)$

则 $\text{first}(\alpha) = (\text{first}(X_1) - \{\varepsilon\}) \cup (\text{first}(X_2) - \{\varepsilon\})$
 $\cup \dots \cup (\text{first}(X_{i-1}) - \{\varepsilon\}) \cup \text{first}(X_i)$

3、当 $\text{first}(X_j)$ 都含有 ε 时 ($1 \leq j \leq n$), 则

$\text{first}(\alpha) = \text{first}(X_1) \cup \text{first}(X_2) \cup \dots \cup \text{first}(X_j) \cup \{\varepsilon\}$

作业

- P100 第2题 (1)、(2)



内容回顾

- 自上而下语法分析要解决的关键问题是什么？
 - 假定要被代换的最左非终结符号是**B**，且有**n**条规则： $B \rightarrow A_1 | A_2 | \dots | A_n$ ，那么如何确定用哪个右部去替代**B**？
 - 确定的自顶向下分析思想是什么？
 - 从开始符号出发，不断替换非终结符，根据当前输入的单词符号就可以唯一选定要替换的产生式。
-

内容回顾

- 怎样根据当前输入的单词符号选定要替换的产生式？
- 根据当前输入符号落在哪个产生式规则的选择集合中，来唯一确定产生式进行推导。
- 确定的自顶向下分析方法中，产生式的选择集合是什么？
- 对于产生式 $A \rightarrow \alpha$ ， $A \in V_N$ ， $\alpha \in V^*$ ，且 α 不能推导出 ε ，则 $\text{select}(A \rightarrow \alpha) = \text{First}(\alpha)$ ；
- 对于产生式 $A \rightarrow \alpha$ ， $A \in V_N$ ， $\alpha \in V^*$ ，且 α 能推导出 ε ，则 $\text{select}(A \rightarrow \alpha) = (\text{First}(\alpha) - \{\varepsilon\}) \cup \text{Follow}(A)$ ；

内容回顾

- 哪类文法产生的句子才能用确定的自顶向下方法进行分析？
- LL(1)文法
- LL(1)文法的定义是什么？
- 对每个非终结符A的两个不同产生式：

$A \rightarrow \alpha, A \rightarrow \beta$ 满足 $\text{Select}(A \rightarrow \alpha) \cap \text{Select}(A \rightarrow \beta) = \emptyset$ ，其中 $\alpha、\beta$ 不同时推出 ϵ

- 怎样对LL(1)文法产生的句子用确定的自顶向下方法进行分析？
- 给定输入串，根据当前输入符号落在哪个产生式规则的选择集合中，来唯一确定产生式进行推导。
- **LL(1)**的含义是什么？

- **LL(1)**的含义：从左**L**向右扫描输入串，分析过程中采用最左**L**推导，只需向右看**1**个符号就可确定如何推导（选择哪个产生式进行推导）。
- 给定一个文法**G**，如何判断它是否属于**LL(1)**文法？
- 对每个非终结符**A**的两个不同产生式：

$A \rightarrow \alpha, A \rightarrow \beta$ 满足 $\text{Select}(A \rightarrow \alpha) \cap \text{Select}(A \rightarrow \beta) = \emptyset$ ，其中 α 、 β 不同时推出 ϵ

- 求每个产生式规则的**Select**集合；
- 求每个规则右部 α 的**First**集合；
- 求每个非终结符**A**的**Follow**集合；
- 对于产生式 $A \rightarrow \alpha$ ， $A \in V_N$ ， $\alpha \in V^*$ ，且 α 不能推导出 ϵ ，则 $\text{select}(A \rightarrow \alpha) = \text{First}(\alpha)$ ；
- 对于产生式 $A \rightarrow \alpha$ ， $A \in V_N$ ， $\alpha \in V^*$ ，且 α 能推导出 ϵ ，则 $\text{select}(A \rightarrow \alpha) = (\text{First}(\alpha) - \{\epsilon\}) \cup \text{Follow}(A)$ ；