第 2 章　形式语言基础（3）

【内容提要】

2.1　形式语言是字符串集合
2.2　形式语言是由文法定义的
2.3　主要语法成分的定义
2.4　两类特性文法
2.5　文法变换方法
2.6　关于形式语言的分类问题

※ 上节课主要内容回顾：

❖ 文法的运算与主要语法成分的定义！

1. 文法的运算：推导和归约(二者互为逆运算)

(1) 推导 ：

直接推导($\Rightarrow$), 加推导($\overset{+}{\Rightarrow}$)，最左推导($\overset{+}{\underset{l}{\Rightarrow}}$)；

(2) 归约 ：

直接归约($\dot{\Rightarrow}$), 加归约($\overset{+}{\dot{\Rightarrow}}$)，最左归约($\overset{+}{\underset{l}{\dot{\Rightarrow}}}$)；

2. 主要语法成分的定义

句型，句子，语法树， 短语，简单短语，句柄

※从语法树上看 句型(句子)、短语、简单短语和句柄：

- 语法树的树叶全体--句型(句子)；
- 语法树的子树树叶全体—短语(简单短语)；
- 语法树的最左简单子树树叶全体— 句柄！

3 两种特性文法1

设有文法：$G(Z)=(V_N, V_T, Z, P)$

递归文法

【定义】 设 $A\in V_N$， x， $y\in(V_N+V_T)^*$，则；

若 $A \overset{+}{\Longrightarrow} xAy$， ：称文法具有递归性；

特别： 若 $A \rightarrow A\alpha$ ，称文法具有直接左递归性；

$A \rightarrow \alpha A$ ，称文法具有直接右递归性。

如：G1(S)：$\boxed{S \rightarrow S\ b \mid a}$ --- 直接左递归文法；

G2(S)：$\boxed{S \rightarrow b\ S \mid a}$ --- 直接右递归文法。

※ 递归文法是定义无限语言的工具（递归文法定义的语言有无限个句子）！！

4 两种特性文法2

二义性文法

【定义】 若文法中存在这样的句型，它具有两棵不同的语法树，则称该文法是二义性文法。

【例2.14】 算术表达式的另一种文法：

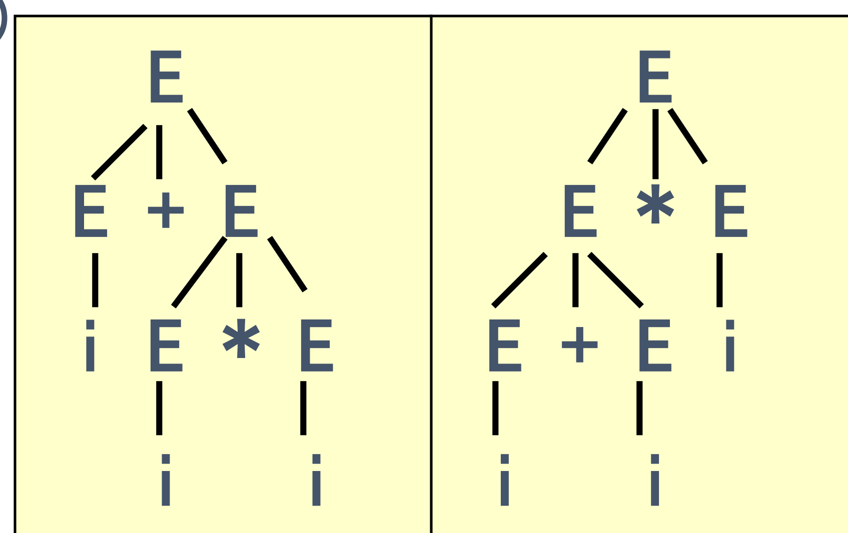
G′(E)：E -> E+E|E-E|E*E|E/E|(E)|i ;

其中：i(变量或常数)

∵ 句型 i+i*i 有两棵不同的语法树(右图)：

∴ G`(E) 是二义性文法。

二义性文法会引起歧义，应尽量避免之！



2.5 文法的等价变换

2.5.1 文法的等价性

【定义】 设G_1、G_2是两个文法，若$L(G_1)=L(G_2)$，则称G_1与G_2等价，记作$G_1 \equiv G_2$。

即：文法的等价性是指他们所定义的语言是一样的。

【例2.15】： G_1: $S \rightarrow aS|a$;

G_2: $S \rightarrow Sa|a$

$L(G_1)=\{a, aa, aaa, \ldots\}=\{a^n \mid n \geqslant 1\}$

$L(G_2)=\{a, aa, aaa, \ldots\}=\{a^n \mid n \geqslant 1\}$

$\therefore L(G_1)=L(G_2)$ 即 $G_1 \equiv G_2$

【注】一个语言，其描述文法并不唯一。

2.5.2　文法变换方法

在实际工作中，人们总是希望定义一种语言的文法尽可能地简单。另外，某些常用的语法分析技术也会对文法提出一定的要求或限制；为了适应上述要求，有时需要对文法进行必要的改写。当然改写后的文法要与原文法等价—通常称为文法变换。

这里重点介绍三类变换：

(1)　删除无用的产生式（文法的化简）；

(2)　删除ε产生式；

(3)　常用的三种文法变换方法：

① 必选项法；

② 可选项法；

③ 重复可选项法。

2.5.2 文法变换方法1

Ⅰ. 文法的化简

文法化简是指消除如下无用产生式:

1. 删除 A->A 形式的产生式(自定己);
2. 删除不能从其推导出终结符串的产生式(不终结);
3. 删除在推导中永不使用的产生式(不可用)。

※ 第 2.步算法(删除不终结产生式):

(1) 构造能推导出终结符串的非终结符集 V_{VT}:

① 若有 $A \to \alpha$ 且 $\alpha \in V_T^*$;则令 $A \in V_{VT}$;

② 若有 $B \to \beta$ 且 $\beta \in (V_T + V_{VT})^*$;则令 $B \in V_{VT}$;

③ 重复步骤①②,直到V_{VT}不再扩大为止。

(2) 删除不在V_{VT}中的所有非终结符(连同其产生式)。

2.5.2 文法变换方法1(续1)

※ 第 3.步算法(删除不可用产生式):

(1) 构造可用的非终结符集 V_{US}:

① 首先令 $Z \in V_{US}$;(Z 为文法开始符号)

② 若有 $Z \stackrel{+}{\Rightarrow} \ldots A \ldots$,则 令 $A \in V_{US}$;

③ 重复步骤②,直到V_{US}不再扩大为止。

(2) 删除不在V_{US}中的所有非终结符(连同其产生式)。

【例2.16】化简下述文法:

G(S): S -> Be | Ec
A -> Ae | e | A
B -> C e | Af
C -> Cf ; D -> f ; G -> b

※ 文法化简示例：

- 化简步骤：

G(S)： S -> Be | ~~Ec~~
A -> Ae | e | ~~A~~
B -> ~~C e~~ | Af
~~C -> Cf~~ ; D -> f ; G -> b

1. 删除 A -> A ;

2. 删除不终结产生式：

∵ V_{VT}={ A, D, G, B, S }； ∴ 应删除 C, E(连同其产生式)

得：G`(S)： S -> Be ; A -> Ae | e ; B -> Af ;
~~D -> f~~ ; ~~G -> b~~ ;

3. 删除不可用产生式：

∵ V_{US}={ S, B, A }； ∴ 应删除 D, G(连同其产生式)

※ 整理后得：G``(S)：

S -> Be
A -> Ae | e
B -> Af

2.5.2 文法变换方法2

II 删除 ε 产生式

※假定 文法 G(Z) ; $\varepsilon \in L(G)$

【算法】

1. 首先构造可以推出空串的非终结符集：V_ε

① 若有 $A\to\varepsilon$；则 令 $A\in V_\varepsilon$;

② 若有 $B\to A_1 \ldots A_n$ 且全部 $A_i \in V_\varepsilon$;则令 $B\in V_\varepsilon$;

③ 重复步骤①②，直到V_ε不再扩大为止。

2. 删除 G(Z)中的 $A \to \varepsilon$ 形式的产生式;

3. 依次改写G(Z)中的产生式 $A \to X_1 X_2 \ldots X_n$:

若有 $X_i \in V_\varepsilon$ 则用 $(X_i|\varepsilon)$ 替换之(一个分裂为两个);

※ 若有 j 个 $X_i \in V_\varepsilon$ ，则一个产生式将分裂为2^j个!

※ 删除 ε 产生式示例：

【例2.17】G(S)：

$S \to aAbc \mid bS$
$A \to dABe \mid \cancel{\varepsilon}$ ；$B \to A \mid b$

(1) **求解** $V_\varepsilon = \{ A, B \}$

含有V_ε元素的产生式

(2) **删除** ε产生式　得：

$S \to aAbc \mid bS$ ；$A \to dABe$ ；$B \to A \mid b$

(3) **改写** 含有 V_ε 中元素的产生式：

∵ $S \to a(A \mid \varepsilon)bc$　　∴ $S \to aAbc \mid abc$

∵ $A \to d(A \mid \varepsilon)(B \mid \varepsilon)e$　　∴ $A \to dABe \mid dBe \mid dAe \mid de$

∵ $B \to (A \mid \varepsilon)$　　∴ $B \to A$

※ **综合** G\`(S)：

$S \to aAbc \mid abc \mid bS$
$A \to dABe \mid dBe \mid dAe \mid de$
$B \to A \mid b$

2.5.2 文法变换方法3

Ⅲ 常用的三种文法变换方法：

※ **基本思想**：扩展文法，引进新的**描述符号**：

（） 圆括号；［ ］方括号；｛ ｝花括号。

1. 必选项法（园括号法）

必选其中之一！

令 $(\alpha|\beta) = \alpha$ 或者 β

例如：有 $A \to a\alpha|a\beta$

可变换成：$A \to a(\alpha|\beta)$

也可：$A \to aA`$ ； $A` \to \alpha|\beta$

【注】此法有称**提公因子法**，利用此法可以使文法：

具有相同左部的各产生式首符号不同！

2.5.2 文法变换方法3(续1)

Ⅲ 常用的三种文法变换方法：

2. 可选项法（方括号法）

可选也可不选！

令 〔α〕 = α 或者 ε

例如： S -> α|αβ

可变换成： S -> α〔β〕

也可： S -> α S\` ; S\` -> β | ε

例如 条件语句文法：

S -> if (B) S

S -> if (B) S else S

S(语句)，B(布尔表达式)

可变换成： S -> if (B) S 〔else S〕

或： S -> if (B) S S\` ; S\`-> else S| ε

2.5.2　文法变换方法3(续2)

Ⅲ 常用的三种文法变换方法：

3. 重复可选项法（花括号法）

令 $\{\alpha\} = \varepsilon$ 或 α 或 $\alpha\alpha$ 或 $\alpha\alpha\alpha$ ...

例如：　$A \to A\beta \mid \alpha$

可变换为：$A \to \alpha\{\beta\}$

也可：$A \to \alpha A'$ ； $A' \to \beta A' \mid \varepsilon$

※ 验证：

∵ 通过递推方法，可得：

$A \Rightarrow A\beta \Rightarrow A\beta\beta \Rightarrow A\beta\beta\beta \Rightarrow \ldots \Rightarrow \alpha\beta^*$

∴ 有 $A \to \alpha\{\beta\}$ ；或 $A \to \alpha A'$ ； $A' \to \beta A' \mid \varepsilon$ ；

【注】此方法常用来消除文法的直接左递归！

2.6 形式语言的分类

chomsky 把形式语言分为四类，分别由四类文法定义；四类文法的区别在于产生式的形式不同：

(1) 0 型语言 由 0型文法定义
- 产生式形式为：$\alpha \rightarrow \beta$

又称 无限制文法！

(2) 1 型语言 由 1型文法定义
- 产生式形式为：$xAy \rightarrow x\beta y$

又称 上下文有关文法！

(3) 2 型语言 由 2型文法定义
- 产生式形式为：$A \rightarrow \beta$

又称 上下文无关文法！

(4) 3 型语言 由 3型文法定义
- 产生式形式为：$A \rightarrow aB$ ，$A \rightarrow a$ ，$A \rightarrow \varepsilon$

又称 正规文法！

【注】 四类语言为 包含关系，且有 $L_0 \supset L_1 \supset L_2 \supset L_3$；编译处理中，主要应用后两种文法！

➢基本图形库

$\overset{+}{=>}$ $\overset{+}{\underset{\bullet}{=>}}$ $\overset{+}{\underset{\bullet}{=>_{\ell}}}$ $\overset{+}{=>_{\ell}}$ $A \longrightarrow \alpha\beta$

$$
\begin{aligned}
\textbf{P:}\quad E &\longrightarrow T \mid E + T \mid E - T \\
T &\longrightarrow F \mid T * F \mid T / F \\
F &\longrightarrow i \mid (\; \mathbf{E} \;)
\end{aligned}
$$

=>*, =>+ , =>.* , =>.+ , =>l* , =>l+ , =>.l+ , =>.l*

$=>_{\ell}$ $\underset{\bullet}{=>_{\ell}}$ $\bar{\in}$

※ 递归文法示例

【例2.15】 G(Z)：

$Z \to aAbB \mid cZ$

$A \to bBc \mid \varepsilon$

$B \to BbAc \mid a$

∵ $Z \to cZ$ ∴ 直接右递归性；

$B \to BbAc$ ∴ 直接左递归性；

$A \Rightarrow bBc \Rightarrow bBbAcc$

即 $A \Rightarrow^{+} \alpha A\beta$ ∴ 具有递归性

（ $\alpha \neq \varepsilon$ 且 $\beta \neq \varepsilon$ 又称为A具有自嵌套性）

∴ 可以统称文法G(Z)具有递归性。