

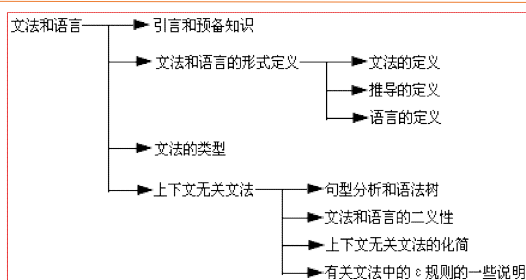
## 第二章 文法和语言

### 2.1 文法的直观概念

- 语言是由句子组成的集合，是由一组记号所构成的集合。
  - 汉语--所有符合汉语语法的句子的全体
  - 英语--所有符合英语语法的句子的全体
  - 程序设计语言--所有该语言的程序的全体

一个语言的**语法**是指一组规则，用它可以形成和产生一个合适的程序

### 结构框架



### 文法

- 文法是描述语言的语法结构的形式规则
- 以**有穷**的集合刻画**无穷**集合的工具
- 文法描述语言时不考虑语言的含义

### 形式语言

程序的含义以及与使用者的关系

- 若不考虑语义和语用，即只从语法这一侧面来看语言，这种意义下的语言称作**形式语言**。
- 形式语言可抽象定义为一个数学系统。
- 形式语言理论是对符号串集合的表示法、结构及其特性的研究，是程序设计语言语法分析研究的基础。
- 构造编译程序的算法是从研究源程序到目标程序产生，首先找到源语言的形式描述，根据这种描述，构造出相应的分析加工程序。

### 文法的非形式定义

句子：“我是大学生。”

如何定义句子的合法性？

- ✓ 有穷语言：只需逐一列举句子
- ✓ 无穷语言：使用文法定义句子的结构，用适当条数的规则把语言的全部句子描述出来。

## 1、建立语法规则

南京中医药大学 | 智信学院

通过建立一组规则，来描述句子的语法结构。

<句子> ::= <主语> <谓语>  
<主语> ::= <代词> | <名词>  
<代词> ::= 你|我|他  
<名词> ::= 王民|大学生|工人|英语  
<谓语> ::= <动词> <直接宾语>  
<动词> ::= 是|学习  
<直接宾语> ::= <代词> | <名词>

BNF表示

"::="表示"由.....组成"

"|"表示"或"

"< >"表示某个语法成分

## 2.2 符号和符号串

南京中医药大学 | 智信学院

字母表  $\Sigma$ ：一个有穷符号集合。例： $\Sigma = \{0, 1\}$

C语言的字母表

$A = \{a, b, \dots, 0, 1, \dots, 9, +, -, \times, /, (, ), =, \dots, \text{if, else, for}, \dots\}$

符号：字母表中的元素，字母、数字、标点符号等。

符号串：由字母表中的符号组成的任何有穷序列，例：ab, 0+1, 3/1, ...

空符号串：无任何符号的符号串即长度为0，用  $\epsilon$  (epsilon) 表示

符号串集合：若集合A中的一切元素都是某字母表上的符号串，则称A为该字母表上的符号串集合。例： $\Sigma = \{a, b, c\}$   $A = \{a, aa, ac\}$

## 2、由规则推导句子

南京中医药大学 | 智信学院

按照一定的方式用规则去推导或产生句子。

推导方法：从一个要识别的符号开始推导，即用相应规则的右部来替代规则的左部，每次仅用一条规则去进行推导。

<句子>  $\Rightarrow$  <主语> <谓语>  
 $\Rightarrow$  <代词> <谓语>

.....

这种**推导**一直进行下去，直到所有带< >的符号都由终结符号替代为止。

## 约定

南京中医药大学 | 智信学院

- 小写字母 a,b,c, ..., r 表示符号
- 小写字母 s,t,u, ..., z 表示符号串
- 大写字母 A,B,C, ..., Z 表示符号串集合



推导方法：从一个要识别的符号开始推导，即用相应规则的右部来替代规则的左部，每次仅用一条规则去进行推导。

<句子>  $\Rightarrow$  <主语> <谓语>  
 $\Rightarrow$  <代词> <谓语>

$\Rightarrow$  我 <谓语>

$\Rightarrow$  我 <动词> <直接宾语>

$\Rightarrow$  我是 <直接宾语>

$\Rightarrow$  我是 <名词>

$\Rightarrow$  我是大学生

<句子> ::= <主语> <谓语>  
<主语> ::= <代词> | <名词>  
<代词> ::= 你|我|他  
<名词> ::= 王民|大学生|工人|英语  
<谓语> ::= <动词> <直接宾语>  
<动词> ::= 是|学习  
<直接宾语> ::= <代词> | <名词>

<句子>  $\Rightarrow$  我是大学生

最左推导

最右推导

## 符号串的运算

南京中医药大学 | 智信学院

### 符号串的长度

符号串中符号的个数，记为  $|s|$

$|\epsilon| = 0$

### 符号串的连接

把 y 的符号写在 x 的符号之后得到的新符号串 xy

如  $x=00$ ,  $y=11$ , 则  $xy=0011$

对于任意一个符号串 s, 有  $|\epsilon s| = |s \epsilon| = |s|$

## 符号串的计算

### 符号串的方幂运算

符号串 $s$ 自身连接 $n$ 次得到符号串 $s^n$  定义为  $ss \dots ss$  , 包括 $n$ 个 $s$

$$s^0 = \varepsilon, s^1 = s, s^2 = ss, \dots$$

设 $s=01$ , 则

$$s^0 = \varepsilon$$

$$s^1 = 01$$

$$s^2 = 0101$$

.....

## 符号串集合的运算

### 符号串集合的闭包运算

$$\text{集合A的正闭包 } A^+ = A^1 \cup A^2 \cup A^3 \cup \dots \cup A^n \cup \dots$$

$$\text{集合A的闭包 } A^* = A^0 \cup A^+$$

$$\{0,1\}^+ = \{0, 1, 00, 11, 01, 10, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$$

$$\{0,1\}^* = \{\varepsilon, 0, 1, 00, 11, 01, 10, 000, 001, 010, 011, 100, 101, 110, 111, \dots\}$$

## 符号串集合的运算

### 符号串集合的乘积

设 $A$ 、 $B$ 为符号串集合, 则 $A$ 和 $B$ 的乘积定义为:

$$AB = \{xy \mid x \in A, y \in B\}$$

例如,  $A = \{a, b\}, B = \{0, 1\}$

$$AB = \{a0, a1, b0, b1\}$$

$$BA = \{0a, 1a, 0b, 1b\}$$

注意  
先后  
顺序

### $\Sigma$ 的闭包 $\Sigma^*$

表示 $\Sigma$ 上的一切符号串 (包括 $\varepsilon$ )组成的集合

### $\Sigma$ 的正闭包 $\Sigma^+$

表示 $\Sigma$ 上的除 $\varepsilon$ 外的所有符号串组成的集合

$$\Sigma^* = \{\varepsilon\} \cup \Sigma \cup \Sigma^2 \cup \dots$$

$$\Sigma^+ = \Sigma^* - \{\varepsilon\} = \Sigma \Sigma^* = \Sigma \cup \Sigma^2 \cup \Sigma^3 \cup \dots$$

## 符号串集合的运算

### 符号串集合的幂运算

$$A^0 = \{\varepsilon\}$$

$$A_1 = A$$

$$A^2 = AA$$

$$A^3 = AAA$$

.....

$$A_n = A^{n-1}A = AA^{n-1}$$

$$\{0,1\}^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$$

长度为 $n$ 的符号串构成的集合

### ★ 为什么对符号、符号串、符号串集合以及它们的运算感兴趣?

若 $A$ 为某语言的字母表

$$A = \{a, b, \dots, 0, 1, \dots, 9, +, -, \times, /, (, ), =, \dots \text{if, else, for} \dots\}$$

$B$ 为单词集

$$B = \{\text{if, else, for}, \dots, \langle \text{标识符} \rangle, \langle \text{常量} \rangle, \dots\}$$

则 $B \subseteq A^*$ 。

语言的句子是定义在 $B$ 上的符号串。

若令 $C$ 为句子集合, 则 $C \subseteq B^*$ , 程序 $C \subseteq A^*$

## 2.3 文法的形式定义

定义2.1 文法  $G = (V_N, V_T, P, S)$

$V_T$ : 终结符号集

**终结符 (terminal symbol)** 是文法所定义的语言的**基本符号**,  
有时也称为**token**

例:  $V_T = \{0, 1, 2, \dots\}$

<无符号整数>  $\rightarrow$  <数字串>  
<数字串>  $\rightarrow$  <数字串> <数字> | <数字>  
<数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

## 2.3 文法的形式定义

定义 文法  $G = (V_N, V_T, P, S)$

$V_T$ : 终结符号集

$V_N$ : 非终结符号集

$V = V_N \cup V_T$   
文法的字母表

$V_N \cap V_T = \emptyset$

★  $P$ : 产生式或规则的集合

$S$ : 开始符号

**开始符号 (start symbol)** 表示文法中最大的语法成分

$S \in V_N$

例:  $S = \text{<无符号整数>}$

<无符号整数>  $\rightarrow$  <数字串>  
<数字串>  $\rightarrow$  <数字串> <数字> | <数字>  
<数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

## 2.3 文法的形式定义

定义2.1 文法  $G = (V_N, V_T, P, S)$

$V_T$ : 终结符号集

$V_N$ : 非终结符号集

$V = V_N \cup V_T$   
文法的字母表

$V_N \cap V_T = \emptyset$

**非终结符 (nonterminal)** 是用来表示语法成分的**符号**,  
有时也称为“语法变量”

例:  $V_N = \{\text{<数字>, <数字串>, <无符号整数>}\}$

<无符号整数>  $\rightarrow$  <数字串>  
<数字串>  $\rightarrow$  <数字串> <数字> | <数字>  
<数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

## 【例】只含+、\*运算的算术表达式的文法定义

$G = (V_T, V_N, S, P)$ , 其中:

$V_N = \{\text{表达式}\}$

$V_T = \{+, *, (, ), i\}$

$S = \{\text{表达式}\}$

$P = \{ \begin{aligned} &\langle \text{表达式} \rangle \rightarrow \langle \text{表达式} \rangle + \langle \text{表达式} \rangle \\ &\langle \text{表达式} \rangle \rightarrow \langle \text{表达式} \rangle * \langle \text{表达式} \rangle \\ &\langle \text{表达式} \rangle \rightarrow ( \langle \text{表达式} \rangle ) \\ &\langle \text{表达式} \rangle \rightarrow i \end{aligned} \}$

$E \rightarrow E + E$   
 $E \rightarrow E * E$   
 $E \rightarrow (E)$   
 $E \rightarrow i$

## 2.3 文法的形式定义

定义2.1 文法  $G = (V_N, V_T, P, S)$

$V_T$ : 终结符号集

$V_N$ : 非终结符号集

$P$ : 产生式或规则的集合

$V = V_N \cup V_T$   
文法的字母表

$V_N \cap V_T = \emptyset$

**产生式 (production)** 描述了将终结符和非终结符组合成串的方法

一般形式:  $\alpha \rightarrow \beta$

例:  $\langle \text{数字} \rangle \rightarrow 0$

$\langle \text{数字串} \rangle \rightarrow \langle \text{数字串} \rangle \langle \text{数字} \rangle$

<无符号整数>  $\rightarrow$  <数字串>  
<数字串>  $\rightarrow$  <数字串> <数字> | <数字>  
<数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

## 产生式的简写

$\alpha \rightarrow \beta_1$   
 $\alpha \rightarrow \beta_2$   
 $\dots$   
 $\alpha \rightarrow \beta_n$



$\alpha \rightarrow \beta_1 | \beta_2 | \dots | \beta_n$

$\alpha$  的**候选式 (Candidate)**

【例】该文法中，S代表什么？

G[S]:

$S \rightarrow L|SL|SD$

$L \rightarrow a|b|\dots|z$

$D \rightarrow 0|1|\dots|9$

描述了标识符的词法规则

S 表示以字母开头的字母数字串

推导

定义 如果  $\alpha \rightarrow \beta$  是文法 G[S] 的一个产生式， $\gamma, \delta \in V^*$ ，有符号串  $x, y$  满足：

$x = \gamma\alpha\delta, y = \gamma\beta\delta$

则称 x 直接产生了 y

y 是 x 的直接推导

y 直接归约到 x

记作  $x \rightarrow y$

简言之，即用产生式的右部替换产生式的左部

★说明：

描述词法规则的文法

G[S]:

$S \rightarrow L|SL|SD$

$L \rightarrow a|b|\dots|z$

$D \rightarrow 0|1|\dots|9$

终结符集是输入字符集，它是构成单词的最基本元素

描述语法规则的文法

G[E]:

$E \rightarrow E+E|E*(E)|i$

终结符集是经词法分析识别后的单词集，如变量 i, 运算符 +、\* 和分界符 (、)，它们被视为语法分析中最基本元素

例如：G[<无符号整数>]

(1) <无符号整数>  $\rightarrow$  <数字串>

(2) <数字串>  $\rightarrow$  <数字串> <数字>

(3) <数字串>  $\rightarrow$  <数字>

(4) <数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

<无符号整数>  $\Rightarrow$  <数字串>

$\Rightarrow$  <数字串> <数字>

$\Rightarrow$  <数字> <数字>

$\Rightarrow$  1 <数字>

$\Rightarrow$  1 0

当符号串已没有非终结符号时，推导就该终止了。

最左推导

<无符号整数>  $\xrightarrow{+}$  10

思考题

已知文法 G[E]:

$E \rightarrow T|E+T|E-T$

$T \rightarrow F|T*F|T/F$

$F \rightarrow (E)|i$

写出该文法的  $V_T$ 、 $V_N$ 、开始符号。

推导

定义 若有  $x \xrightarrow{+} y$ ，或  $x = y$ ，则记作  $x \xRightarrow{*} y$ 。

G[S]:

$S \rightarrow L|SL|SD$

$L \rightarrow a|b|\dots|z$

$D \rightarrow 0|1|\dots|9$

例:  $S \Rightarrow SD \Rightarrow LD \Rightarrow aD \Rightarrow a1$

记作  $S \xRightarrow{+} a1$

## 语言的形式定义

南京中医药大学 智信学院

定义 文法G[S]:

- (1) 句型:  $\alpha$ 是句型  $\Leftrightarrow S \Rightarrow^* \alpha$ , 且  $\alpha \in V_T^*$ ;
- (2) 句子:  $\alpha$ 是句子  $\Leftrightarrow S \Rightarrow^* \alpha$ , 且  $\alpha \in V_T^*$ ;
- (3) 语言:  $L(G[S]) = \{\alpha \mid \alpha \in V_T^*, S \Rightarrow^* \alpha\}$ ;

$\langle \text{无符号整数} \rangle \Rightarrow \langle \text{数字串} \rangle$   
 $\Rightarrow \langle \text{数字串} \rangle \langle \text{数字} \rangle$   
 $\Rightarrow \langle \text{数字} \rangle \langle \text{数字} \rangle$   
 $\Rightarrow 1 \langle \text{数字} \rangle$   
 $\Rightarrow 10 \rightarrow \text{句子}$

例:  $\{ab^na \mid n \geq 1\}$ , 构造其文法

南京中医药大学 智信学院

$G_1[S]: S \rightarrow aBa$   
 $B \rightarrow b|bB$

$G_2[S]: S \rightarrow aB$   
 $B \rightarrow bB|ba$

定义: G和G'是两个不同的文法, 若  $L(G) = L(G')$ , 则G和G'为等价文法。

例:  $\{a^n b^n c^i \mid n \geq 1, i \geq 0\}$ , 构造其文法

$G_2[S]: S \rightarrow aS \mid aA$   
 $A \rightarrow bA \mid bB$   
 $B \rightarrow cC \mid \epsilon$

$G_1[S]: S \rightarrow AB$   
 $A \rightarrow aAb \mid ab$   
 $B \rightarrow cB \mid \epsilon$

## 两条定理

南京中医药大学 智信学院

形式语言理论可以证明以下两点:

- (1)  $G \rightarrow L(G)$ ;
- (2)  $L(G) \rightarrow G_1, G_2, \dots, G_n$ ;

已知文法, 求语言, 通过推导;

已知语言, 构造文法, 无形式化方法, 更多是凭经验

## 课内练习

南京中医药大学 智信学院

$L(G) = \{a^n b^{n+m} \mid n \geq 1, m \geq 0\}$ , 试为该语言构造文法。

$a^n b^n \rightarrow E \rightarrow aEb \mid ab$

## 已知文法, 求语言, 通过推导

南京中医药大学 智信学院

例: G[S]

$S ::= aSb \mid ab$   
求其所产生的语言。

$L(G[S]) = \{a^n b^n \mid n \geq 1\}$

若  $S ::= aSb \mid \epsilon$  呢?

$L(G[S]) = \{a^n b^n \mid n \geq 0\}$

课堂练习1: G[S]

$S ::= bA$   
 $A ::= aA \mid a$

$L(G[S]) = \{ba^n \mid n \geq 1\}$

课堂练习2: G[S]

$S ::= AB$   
 $A ::= aA \mid a$   
 $B ::= bB \mid b$

$L(G[S]) = \{a^m b^n \mid m, n \geq 1\}$

## 2.4 文法和语言的分类

南京中医药大学 智信学院



**形式语言 (乔姆斯基)**: 通过抽象, 对语言进行形式化描述。  
用一组数学符号和规则来描述的语言称为形式语言。

$\Sigma^*$  的任何子集称作  $\Sigma$  上的形式语言

由文法定义语言  $L(G[S]) = \{\alpha \mid \alpha \in V_T^*, S \Rightarrow^* \alpha\}$

文法和语言分类: 0型、1型、2型、3型

## 0型文法——短语结构文法 *PSG*

定义:  $\forall \alpha \rightarrow \beta \in P$  满足:

$\alpha \in V^*$  且至少含有一个非终结符,  $\beta \in V^*$

- ☞ 0型文法称为无限制文法/短语结构文法。
- ☞ 规则的左部和右部都可以是符号串。
- ☞ 一个短语可以产生另一个短语。

0型语言:  $L_0$

## 1型文法举例

$G[S]:$

$S \rightarrow aSBE$

$S \rightarrow aBE$

$BE \rightarrow bE$

$aB \rightarrow ab$

$bE \rightarrow be$

$L(G[S]) = \{a^n(be)^n | n \geq 1\}$

## 0型文法举例

$G[S]:$

$S \rightarrow ABS|AB$

$AB \rightarrow BA$

$A \rightarrow 0$

$B \rightarrow 1$

➡ 该文法产生的语言是?

$L(G[S]) = \{x | x \text{ 是由 } n \text{ 个 } 01 \text{ 或 } 10 \text{ 组成的二进制数字串, } n \geq 1\}$

## 2型文法——上下文无关文法 *CFG*

定义:  $\forall \alpha \rightarrow \beta \in P$  满足:  $\alpha \in V_N$

产生式的一般形式:  $A \rightarrow \beta$

2型语言:  $L_2$

## 1型文法——上下文有关文法 *CSG*

定义:  $\forall \alpha \rightarrow \beta \in P$  满足:  $|\beta| \geq |\alpha|$

产生式的一般形式:  $\alpha_1 A \alpha_2 \rightarrow \alpha_1 \beta \alpha_2 (\beta \neq \epsilon)$

☞ 1型文法中不包含  $\epsilon$ -产生式

1型语言:  $L_1$

## 2型文法举例

$G[S]:$

$S \rightarrow aSb$

$S \rightarrow ab$

$L(G[S]) = \{a^n b^n | n \geq 1\}$



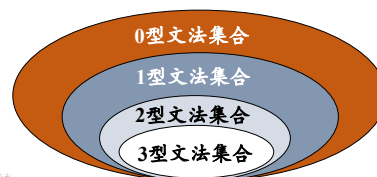
### 3型文法——正则文法 $RG$

相互等价 { 右线性文法:  $A \rightarrow wB$  或  $A \rightarrow w$   $A, B \in V_N$   
左线性文法:  $A \rightarrow Bw$  或  $A \rightarrow w$   $w \in V_T$

3型语言: L3 正则语言

正则文法能描述程序设计语言的多数单词

### 四种文法间的逐级“包含”关系



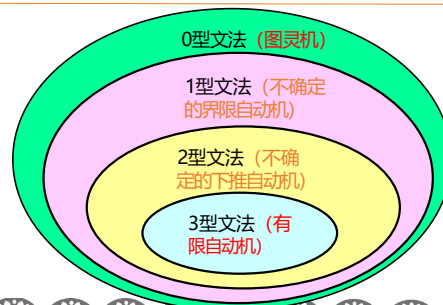
产生式约束越弱  
描述语言的能力越强  
通用性越强

### 3型文法举例

$G[S]$ :  
 $S \rightarrow 0A \mid 1B \mid 0$   
 $A \rightarrow 0A \mid 1B \mid 0S$   
 $B \rightarrow 1B \mid 1 \mid 0$

$G[I]$ :  
 $I \rightarrow aT \mid a$   
 $T \rightarrow aT \mid dT \mid a \mid d$

### 形式语言与自动机



例: 设  $L_2 = \{a^i b^j c^k \mid i, j, k \geq 1 \text{ 且 } a, b, c \in V_T\}$

试构造生成  $L_2$  的文法  $G_2$

$G_2[S]$ :  
 $S \rightarrow ABC$   
 $A \rightarrow aA \mid a$   
 $B \rightarrow bB \mid b$   
 $C \rightarrow cC \mid c$  2型文法

$G_2[A]$ :  
 $A \rightarrow aA \mid aB$   
 $B \rightarrow bB \mid bC$   
 $C \rightarrow cC \mid c$  3型文法

### 思考: 判断下列文法属于哪种类型

$G[E]: E \rightarrow abE \mid \varepsilon$

$G[S]: S \rightarrow BA$   
 $A \rightarrow BS \mid d$   
 $B \rightarrow aA \mid bS \mid c$



## 讨论

南京中医药大学 | 智信学院

$$L(G) = \{ a^n b^n \mid n \geq 1 \}$$

 $G[S] : S \rightarrow aSb \mid ab$  只能由上下文无关文法产生

$$L(G) = \{ a^n b^n c^n \mid n \geq 1 \}$$

 $G[S] : S \rightarrow aSBE \mid aBE$   
 $BE \rightarrow EB$   
 $aB \rightarrow ab$   
 $bB \rightarrow bb$   
 $bE \rightarrow be$   
 $eE \rightarrow ee$  只能由上下文有关文法产生

## 简化文法的步骤

南京中医药大学 | 智信学院

- 查找有无形如  $P \rightarrow P$  的产生式，有则删除；
- 若某个产生式在推导过程中永远不会被用到，删除它；
- 若某个产生式在推导过程中不能从中导出终结符，删除它。
- 最后，整理所有剩余产生式，就得到简化的文法。

## 去除文法中的有害规则和多余规则

南京中医药大学 | 智信学院

- 1、文法中形如  $U \rightarrow U$  的规则是有害规则，它会引起二义性，且无任何用处。

如有规则：  $U \rightarrow U \mid a \mid b$ ，则句子  $a$  有两棵语法树。



## 化简文法

南京中医药大学 | 智信学院

 $G[S]:$ 
 $S \rightarrow Be$ 
 $B \rightarrow Ce \mid Af$ 
 $A \rightarrow Ae \mid e$ 
 $C \rightarrow Cf$ 
 $D \rightarrow f$ 
 $G[S]:$ 
 $S \rightarrow Be$ 
 $B \rightarrow Af$ 
 $A \rightarrow Ae \mid e$ 

## 去除文法中的有害规则和多余规则

南京中医药大学 | 智信学院

## 2、多余规则：

- (1) 规则  $U \rightarrow u$  的左部非终结符  $U$  (非识别符) 不在任何其他规则右部出现，即所有的推导始终不会用到此规则。(不可到达)
- (2) 在推导句子的过程中，一旦使用了该规则，将推不出任何终结符号串。即该规则中含有推不出任何终结符号串的非终结符。(不可终止)

如文法对于  $U$  仅有如下规则：  $U \rightarrow xUy$ ，则该规则是多余的。

 $U \rightarrow xUy \mid a$ 

## 举例

南京中医药大学 | 智信学院

 $S \rightarrow CF$ 
 $C \rightarrow c$ 
 $S \rightarrow cF$ 

Wrong!

不存在  $P \rightarrow P$  的产生式

### 递归文法

**递归规则：**规则右部有与左部相同的符号

左递归规则： $A \rightarrow A \dots$

右递归规则： $A \rightarrow \dots A$

自嵌入递归规则： $A \rightarrow \dots A \dots$


**直接递归**

$A \rightarrow Ba$   
 $B \rightarrow Ab$

**间接递归**

$G[S]:$   
 $S \rightarrow L | SL | SD$   
 $L \rightarrow a | b | \dots | z$   
 $D \rightarrow 0 | 1 | \dots | 9$

含有递归规则的文法，称为**递归文法**



南京中医药大学 | 智信学院

### 语法树

$G[S]:$   
 $S \rightarrow aAS \mid a$   
 $A \rightarrow SbA \mid SS \mid ba$

对于文法G的任何句型，都能构造与之关联的语法树。  
试着画出句型aabbbaa的语法树。

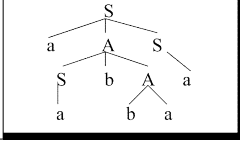
$S \Rightarrow aAS \Rightarrow aAa \Rightarrow aSbAa \Rightarrow aSbbAa \Rightarrow aabbbaa$  **最右推导**


$S \Rightarrow aAS \Rightarrow aSbAS \Rightarrow aabAS \Rightarrow aabbaS \Rightarrow aabbbaa$  **最左推导**

每个句子都有最左推导、最右推导

句子既是左句型，又是右句型

句型aabbbaa的语法树（推导树）





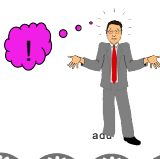
南京中医药大学 | 智信学院


### 递归文法

**优点：**用有穷条规则，定义无穷语言

例：对于前面给出的无符号整数的文法是有递归文法，用12条规则就可以定义出所有的无符号整数。若不用递归文法，那将要用多少条规则呢？

$G[S]:$   
 $S \rightarrow SD | D$   
 $D \rightarrow 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$





南京中医药大学 | 智信学院

### 句型


句子既是左句型，又是右句型


是否每个句型都有最左推导或最右推导呢？

$G[E]:$   
 $E \rightarrow E+T \mid T$   
 $T \rightarrow T * F \mid F$   
 $F \rightarrow (E) \mid i$

$E \Rightarrow E+T \Rightarrow T+T \Rightarrow T * F + T \Rightarrow T * i + T$

**既非左句型 又非右句型**





南京中医药大学 | 智信学院

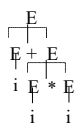
### 2.5 上下文无关文法及其语法树


CFG有足够的描述语言的语法结构。

$G[E]: E \rightarrow E+E \mid E * E \mid (E) \mid i$

该文法定义了由变量、+、\*、(、)组成的算术表达式。

描述上下文无关文法的句型推导的直观工具——**语法树（推导树）**






南京中医药大学 | 智信学院

### 文法二义性

**定义** 若一个文法的某句子存在两个不同的**最右(最左)推导**，则该文法是**二义性**的，否则是无二义性的。

$G[E]:$   
 $E \rightarrow E+E \mid E * E \mid (E) \mid i$

分析句子  $i + i * i$   
给出两种不同的规范推导，并画出语法树



南京中医药大学 | 智信学院

南京中医药大学 Nanjing University of Chinese Medicine 智信学院

$G[E]: E \rightarrow E+E \mid E * E \mid (E) \mid i$

两种不同的推导生成了两棵结构不同的语法树

(1)  $E \Rightarrow E+E \Rightarrow E+E * E \Rightarrow E+E * i \Rightarrow i + i * i$


```

graph TD
    E1[E] --- E2[E]
    E1 --- P1[+]
    E1 --- E3[E]
    E2 --- i1[i]
    E3 --- E4[E]
    E3 --- M1[*]
    E3 --- E5[E]
    E4 --- i2[i]
    E5 --- i3[i]
    
```

(2)  $E \Rightarrow E * E \Rightarrow E * i \Rightarrow E + E * i \Rightarrow i + i * i$

```

graph TD
    E1[E] --- E2[E]
    E1 --- M1[*]
    E1 --- E3[E]
    E2 --- E4[E]
    E2 --- P1[+]
    E2 --- E5[E]
    E3 --- i1[i]
    E4 --- i2[i]
    E5 --- i3[i]
    
```




南京中医药大学  
Nanjing University of Chinese Medicine

智信学院

## 改写二义性文法

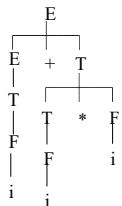
**G[E]:**

$E \rightarrow E+E \mid E * E \mid (E) \mid i$



$E \rightarrow E+T \mid T$   
 $T \rightarrow T * F \mid F$   
 $F \rightarrow (E) \mid i$

E——表达式  
T——项  
F——因子



```

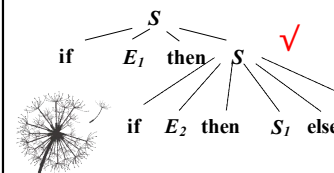
graph TD
    E1[E] --- E2[E]
    E1 --- P1[+]
    E1 --- T1[T]
    E2 --- T2[T]
    T2 --- F1[F]
    F1 --- i1[i]
    T1 --- P2[+]
    T1 --- M1[*]
    T1 --- F2[F]
    P2 --- T3[T]
    T3 --- F3[F]
    F3 --- i2[i]
    M1 --- F4[F]
    F4 --- i3[i]
    
```

## G[S]:

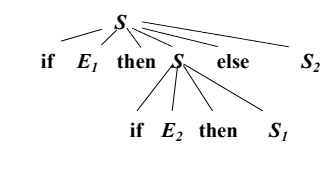
$S \rightarrow \text{if } E \text{ then } S \mid \text{if } E \text{ then } S \text{ else } S \mid \text{other}$

消歧规则：每个else和最近的尚未匹配的if匹配


### 分析句型：if $E_1$ then if $E_2$ then $S_1$ else $S_2$




if  $E_1$  then if  $E_2$  then  $S_1$  else  $S_2$



if  $E_1$  then if  $E_2$  then  $S_1$  else  $S_2$





Nanjing University of Chinese Medicine

南京中医药大学


智信学院


College of Wisdom and Faith

## 语言的二义性

找不到一个非二义性文法支持该语言，则该语言才是二义的。

对于语言  $L$ ，可能存在  $G$  和  $G'$ ，使得  $L(G)=L(G')$ ，而  $G$  是二义性文法，但  $G'$  是非二义性文法，故  $L$  是**非二义性语言**。




南京中医药大学  
Nanjing University of Chinese Medicine智信学院

## 文法二义性

若文法是二义性的，则在编译时就会产生**不确定性**。

**文法的二义性是不可判定的**，即不可能构造出一个算法，通过有限步骤来判定任一文法是否有二义性。



### 解决途径

- ✧ 不改变文法中原有的规则，仅加进一些语法的非形式规定。  
如规定 “\*” 运算优先级高于 “+” 运算，且服从左结合
- ✧ 改写文法，把排除二义性的规则合并到原有文法中。

课内练习 (多选)

下面的文法中，哪些是二义的？

A

文法G[S]:  
 $S \rightarrow SS \mid (S) \mid ()$

句子000

B

文法G[S]:  
 $S \rightarrow iSeS \mid iS \mid i$

句型iiSeS

C

文法G[S]:  
 $S \rightarrow SaS \mid SbS$   
 $S \rightarrow cSd \mid eS$   
 $S \rightarrow f$

句型SaSaS

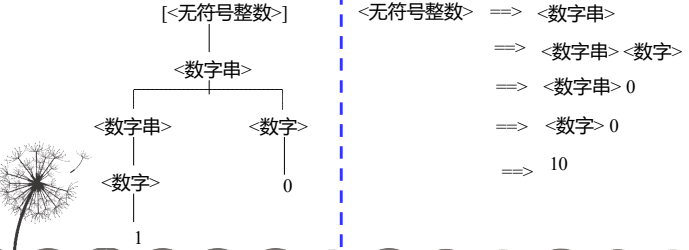
D

文法G[S]:  
 $S \rightarrow S+S \mid S*S$   
 $S \rightarrow i \mid (S)$

句子i+i

G[<无符号整数>]  
<无符号整数>  $\rightarrow$  <数字串>  
<数字串>  $\rightarrow$  <数字串> <数字> | <数字>  
<数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

句型10



2.6 句型的分析

句型的分析：识别输入字符串是否为某一文法的句型（或句子）的过程。

同一个句型或句子，可以通过不同的推导序列推导出来

推导过程中选择替换的非终结符的次序不同

句型的分析

句型推导过程  $\longleftrightarrow$  句型语法树的生长过程

自底向上分析方法

自下而上地修剪子树的末端结点，直至把整棵树剪掉（留根），每剪一次对应一次规约。

从句型开始，自右向左地逐步进行规约，建立推导序列。

句型的分析

句型推导过程  $\longleftrightarrow$  句型语法树的生长过程

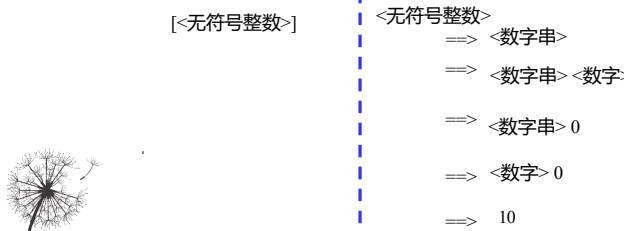
自顶向下分析方法

从识别符号开始，逐步建立推导序列。

由根结点开始，自上而下建立语法树。

G[<无符号整数>]  
<无符号整数>  $\rightarrow$  <数字串>  
<数字串>  $\rightarrow$  <数字串> <数字> | <数字>  
<数字>  $\rightarrow$  0 | 1 | 2 | 3 | ..... | 9

规范规约与规范推导互为逆过程



## 句型分析的有关问题

### 1) 如何选择使用哪个产生式进行推导?

假定非终结符  $V$  有  $n$  条规则:  $V \rightarrow A_1 | A_2 | \dots | A_n$ , 如何确定用右部哪个候选式替换  $V$ ?

### 2) 如何识别可归约的串?

在自下而上的分析方法中, 分析程序的每一步工作, 都是从当前串中选择一个子串, 将它归约到某个非终结符号, 该子串称为“可归约串”。

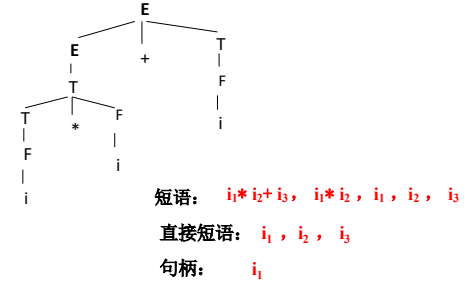
在“规范归约”的分析中, 这种“可归约串”称作“句柄”。

## 举例

$G[E]: E \rightarrow E+T \mid T$   
 $T \rightarrow T * F \mid F$   
 $F \rightarrow (E) \mid i$

句型:  $i * i + i$

请找出该句型的短语、直接短语和句柄



## 三个重要概念

定义 令文法  $G[S]$ ,  $\alpha \beta \delta$  是  $G$  的一个句型, 若:

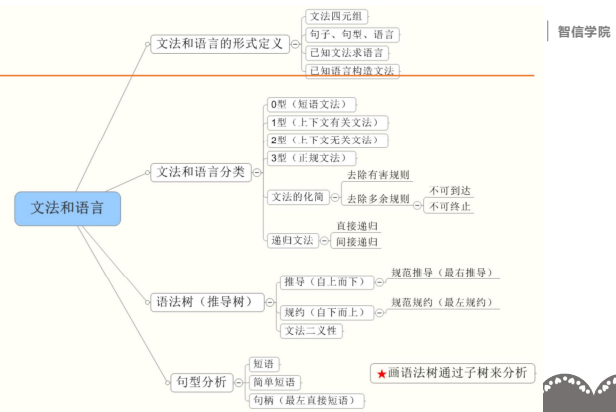
$$S \xRightarrow{*} \alpha A \delta \quad \text{且} \quad A \xRightarrow{+} \beta$$

则称  $\beta$  是句型  $\alpha \beta \delta$  相对于非终结符  $A$  的**短语**。

令  $A \Rightarrow \beta$ , 则称  $\beta$  是句型  $\alpha \beta \delta$  相对于  $A \rightarrow \beta$  的**直接短语**

令一个**右句型**的**最左直接短语**称为该句型的**句柄**。

## 小结



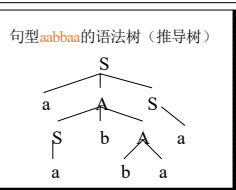
## 结合“子树”分析短语、直接短语、句柄

令**句型**: 在一棵语法树生长过程中的**任何时刻**, 所有那些叶子结点排列起来就是一个句型。

令**短语**: 一棵子树的所有叶子自左至右排列起来形成一个相对于子树根的短语。

令**直接短语**: 仅有父子两代的一棵子树, 它的所有叶子自左至右排列起来所形成的符号串。

令**句柄**: 一个句型的分析树中最左那棵只有父子两代的子树的所有叶子的自左至右排列。句柄是最左归约时要寻找的简单短语。



## CH2书面作业

1、为只包含数字、加号和减号的表达式构造一个文法, 如  $9-2+5, 3-1, 7$  等。

2、构造一个文法, 使其语言是偶正整数的集合(允许0打头)。

3、证明文法  $G = (\{E, O\}, \{(\,, +, *, v, d\}, P, E)$  是二义的, 其中  $P$  为  
 $E \rightarrow EOE \mid (E) \mid v \mid d$   
 $O \rightarrow + \mid *$

4、考虑下面的上下文无关文法:  $S \rightarrow SS^* \mid SS^+ \mid a$

(1) 通过此文法如何生成串  $aa+a^*$ , 并为该串构造语法树。

(2) 该文法生成的语言是什么?

5、一个上下文无关文法生成的句子  $abbbaa$  的唯一语法树如图:

(1) 给出该句子相应的最左推导和最右推导。

(2) 该文法的产生式集合  $P$  可能有哪些元素?

(3) 找出该句子的所有短语、简单短语、句柄。

