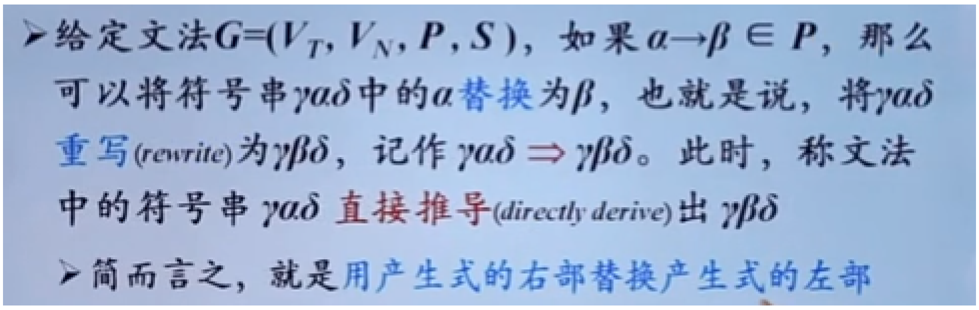
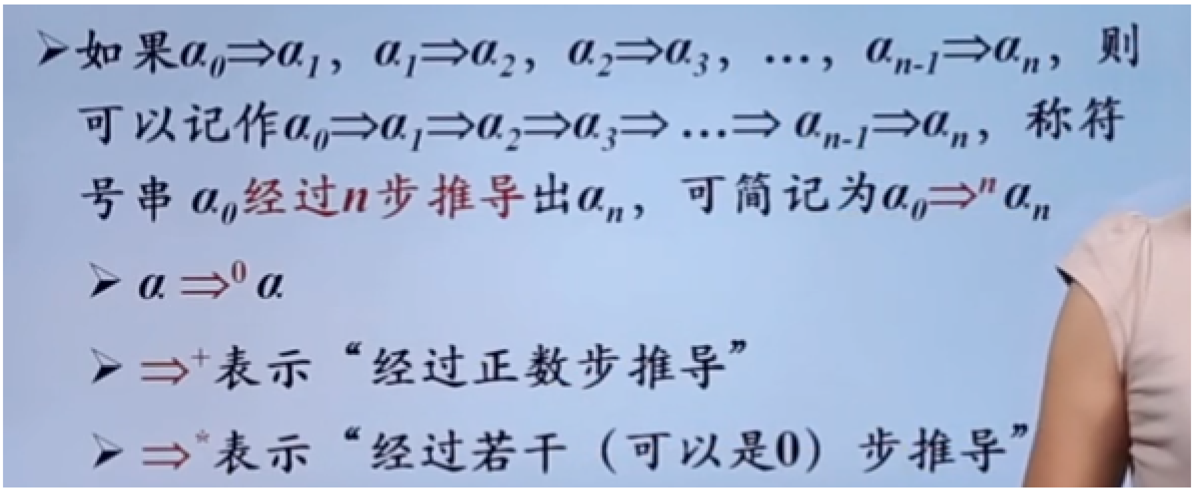
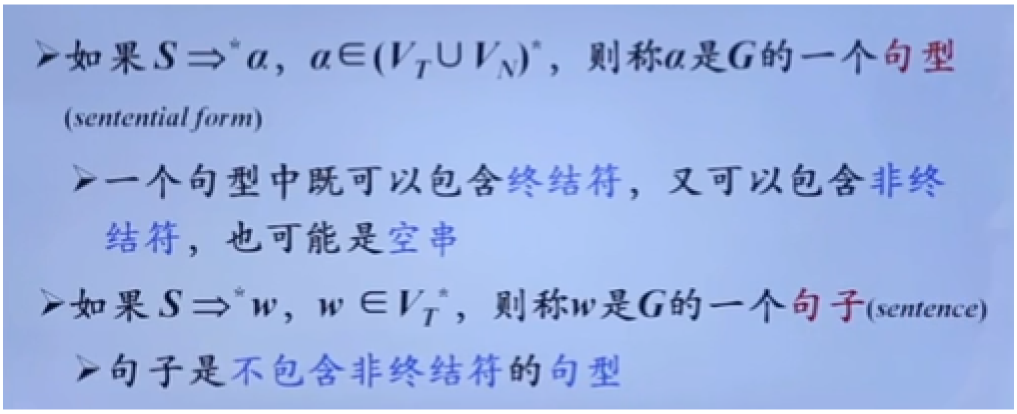
语言的定义

推导和归约

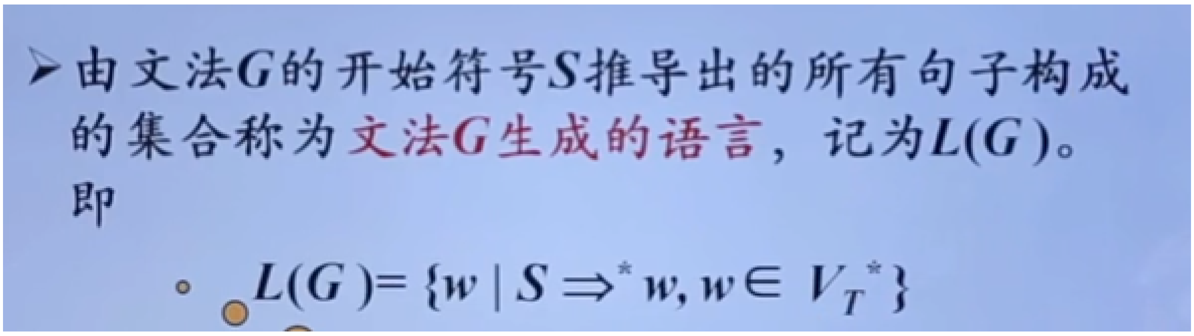




句型和句子



语言的形式化定义



语言上的运算

运算	定义和表示
L和M的并	$L \cup M = \{ s \mid s$属于L或者s属于$M \}$
L和M的连接	$LM = \{ st \mid s$属于L且t属于$M \}$
L的幂	$\begin{cases} L^0 = \{ \varepsilon \} \\ L^n = L^{n-1}L, n \geq 1 \end{cases}$
L的Kleene闭包	$L^* = \cup_{i=0}^{\infty} L^i$
L的正闭包	$L^+ = \cup_{i=1}^{\infty} L^i$

例：令$L=\{A, B, \cdots, Z, a, b, \cdots, z\}$，$D=\{0, 1, \cdots, 9\}$。则$L(L \cup D)^*$表示的语言是标识符

文法的分类

0 型文法（Type-0 Grammar）

$$\alpha \to \beta$$

- 无限制文法(Unrestricted Grammar)/短语结构文法(Phrase Structure Grammar, PSG)
  - $\forall \alpha \to \beta \in P$，$\alpha$中至少包含1个非终结符
- 0型语言
  - 由0型文法G生成的语言$L(G)$

1 型文法（Type-1 Grammar）

$$\alpha \to \beta$$

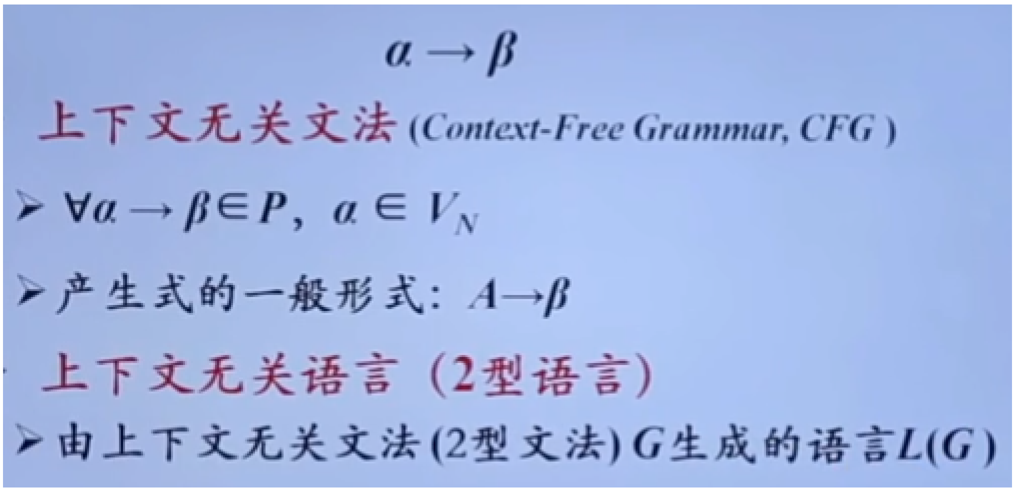
上下文有关文法(Context-Sensitive Grammar, CSG)

- $\forall \alpha \to \beta \in P$，$|\alpha| \leq |\beta|$
- 产生式的一般形式：$\alpha_1 A \alpha_2 \to \alpha_1 \beta \alpha_2 \ (\beta \neq \varepsilon)$

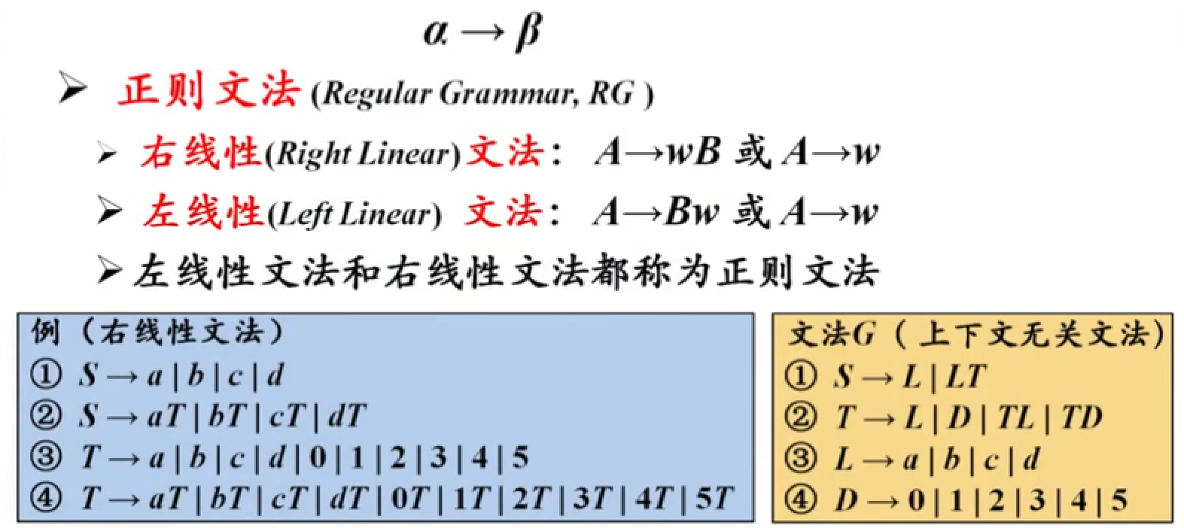
上下文有关语言（1型语言）

- 由上下文有关文法(1型文法)G生成的语言$L(G)$

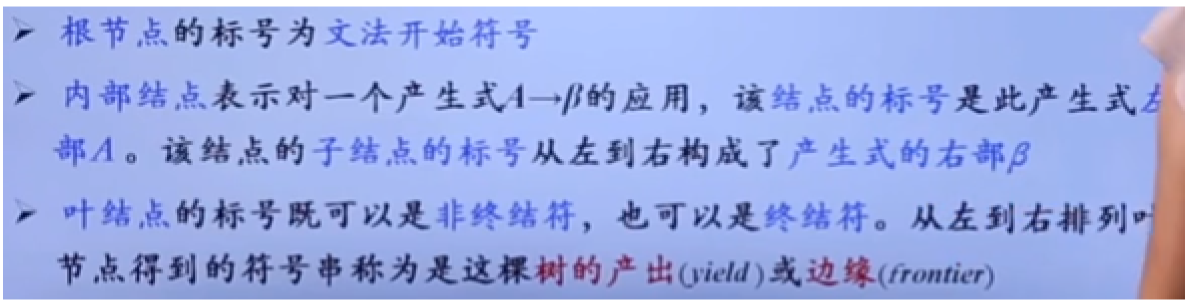
2 型文法（Type-2 Grammar）



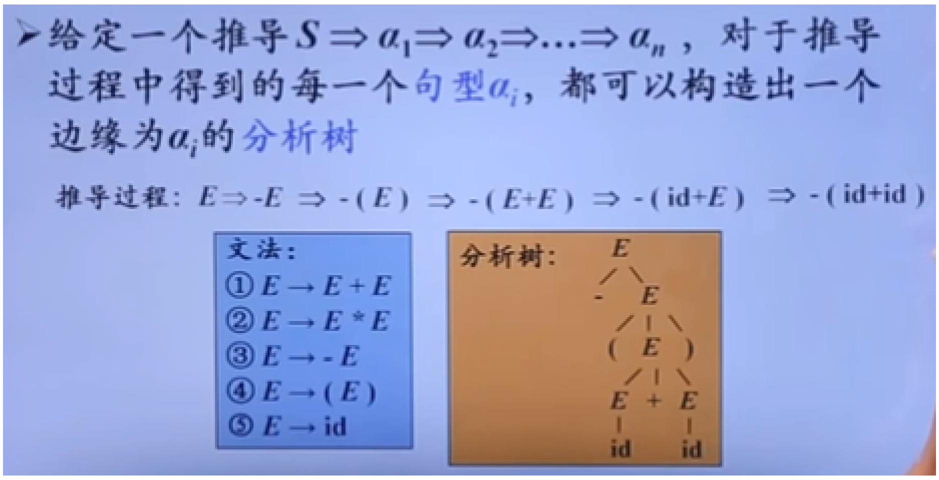
3 型文法（Type-3 Grammar）



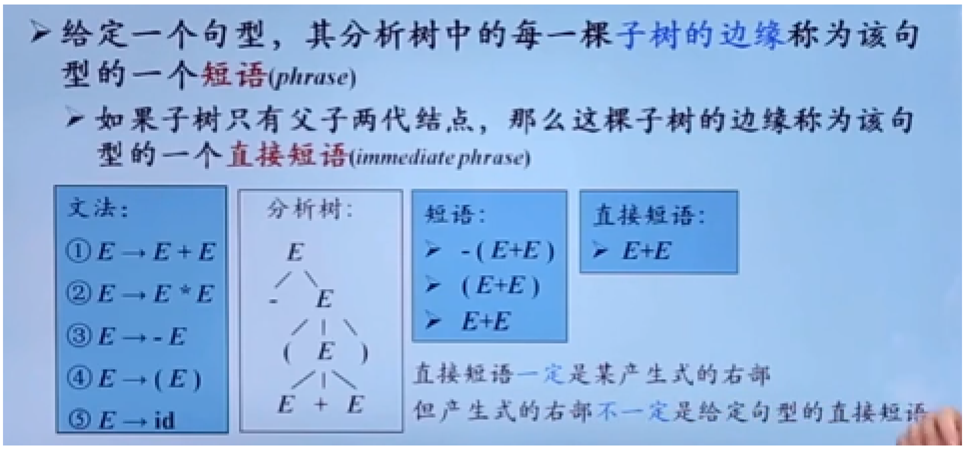
CFG的分析树



分析树是推导的图形化表示

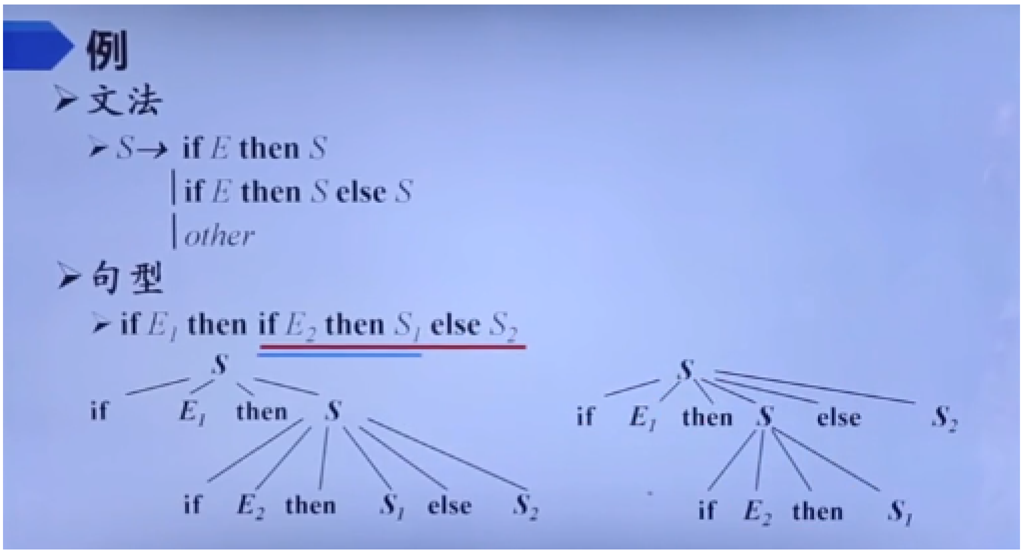


（句型的）短语



二义性文法（Ambiguous Grammar）

- 如果一个文法可以为某个句子生成多棵分析树，则称这个文法是二义性的



二义性文法的判定

- 对于任意一个上下文无关文法，不存在一个算法，判定它是无二义性的；但能给出一组**充分条件**，满足这组充分条件的文法是无二义性的
  - 满足，肯定无二义性
  - 不满足，也未必就是有二义性的