

自然语言处理

句法分析：形式语言

The path so far

- 最初，将语言视为由词构成的序列
 - n-gram (语言模型)
- 接着，引入词的句法属性
 - part-of-speech tagging (词性标注)
- 现在，考察词之间的句法关系
 - Syntactic parsing (句法分析)

语法和语义

- 大部分情况下，一个不合乎语法的句子也可以被理解
 - The boy quickly in the house the ball found
 - 看清楚路先兄弟
- 合乎语法的句子也可能无法理解
 - Are gyre and gimble in the wabe? (non-sense words)
 - 不会做饭的裁缝不是一个好司机
- 但句法规则可以传递如下信息：
 - 句子的语法
 - 词的顺序
 - 短语成分
 - 句子的层次结构
 - 句法关系，例如主语、宾语
 -

句法分析的应用

- 句法分析被广泛且成功地应用到NLP的各个方面：
 - **Meaning Representation** [Jeffrey Flanigan, et al., ACL 2014]
 - High precision **question answering** [Pasca and Harabagiu, SIGIR 2011]
 - Source sentence analysis for **machine translation** [Xu et al., 2009]
 - Syntactically based **sentence compression** [Lin and Wilbur, 2007]
 - **Extracting opinions** about products [Bloom et al., NAACL 2007]
 - **Relation extraction** systems [Fundel et al., *Bioinformatics* 2006]
 - Improved **interaction in computer games** [Gorniak and Roy, 2005]
 - Helping **linguists** find data [Resnik et al., BLS 2005]
 - Improving biological **named entity finding** [Finkel et al., JNLPBA 2004]

一个简单的句子

- I like the interesting lecture

- PRO VB DET JJ NN

- 除了以上的词性标注，往往还关心：

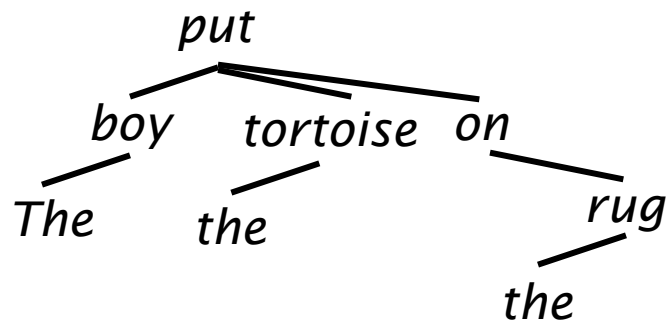
- 动词 *like* 的主语是代词 *I*，说明 who is doing the liking
- 动词 *like* 的宾语是名词 *lecture*，说明 what is being liked
- 定冠词 *the* 指出说明名词 *lecture*
- 形容词 *interesting* 给出更多关于名词 *lecture* 的信息

两种不同的句法结构

- 依存结构 (Dependency structure) :
 - 说明词和其它词之间的依赖关系 (从属关系、支配关系等)

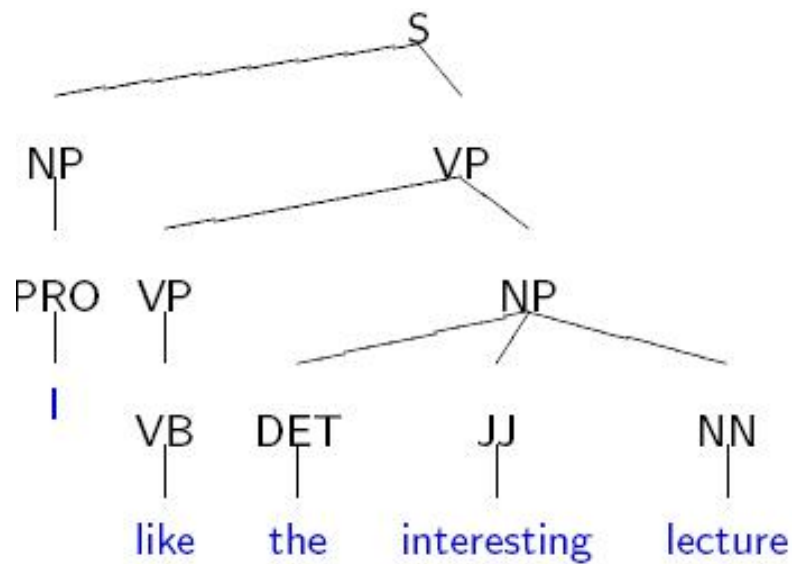


- 可以表示为一个依存树 (dependency tree) :



两种不同的句法结构

- 短语结构 (Phrase structure) :
 - 将句子表示成嵌套的短语成分
- 父节点将子节点组合成较大的短语单元
 - 例如将DET JJ NN组合成NP



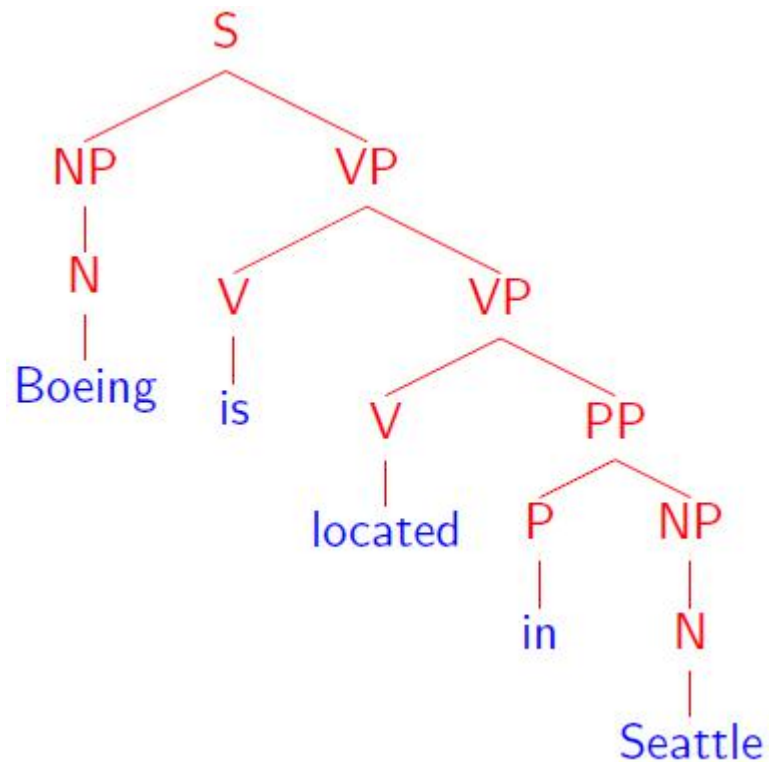
句法分析

- 这里的句法分析 (Parsing) : 专指短语结构分析

INPUT: 句子

Boeing is located in Seattle

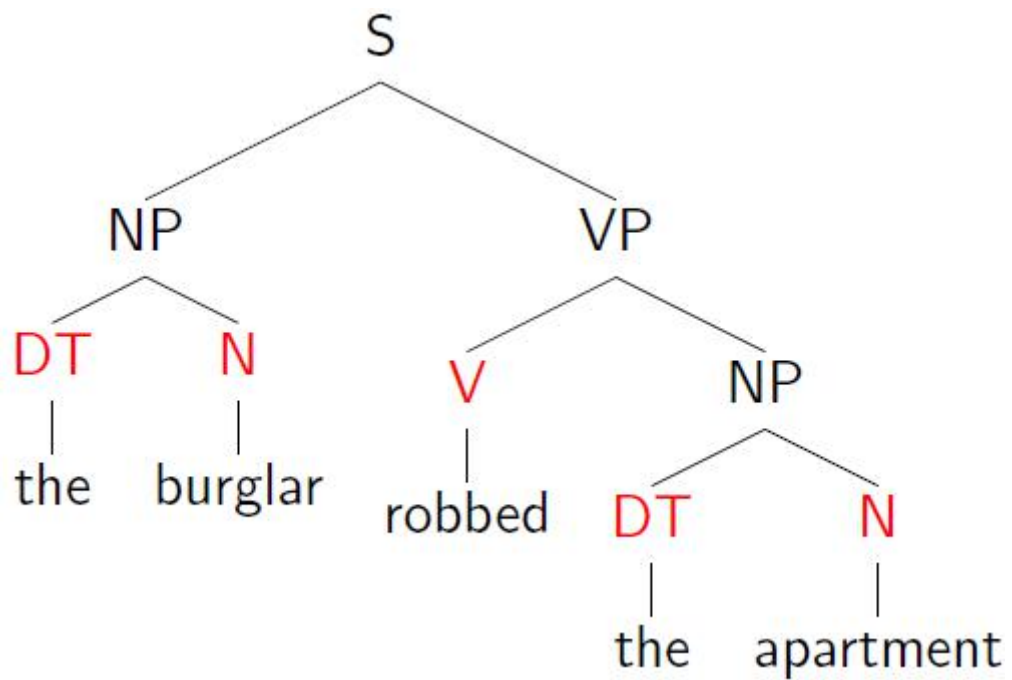
OUTPUT: 句法树



句法树中包含的信息

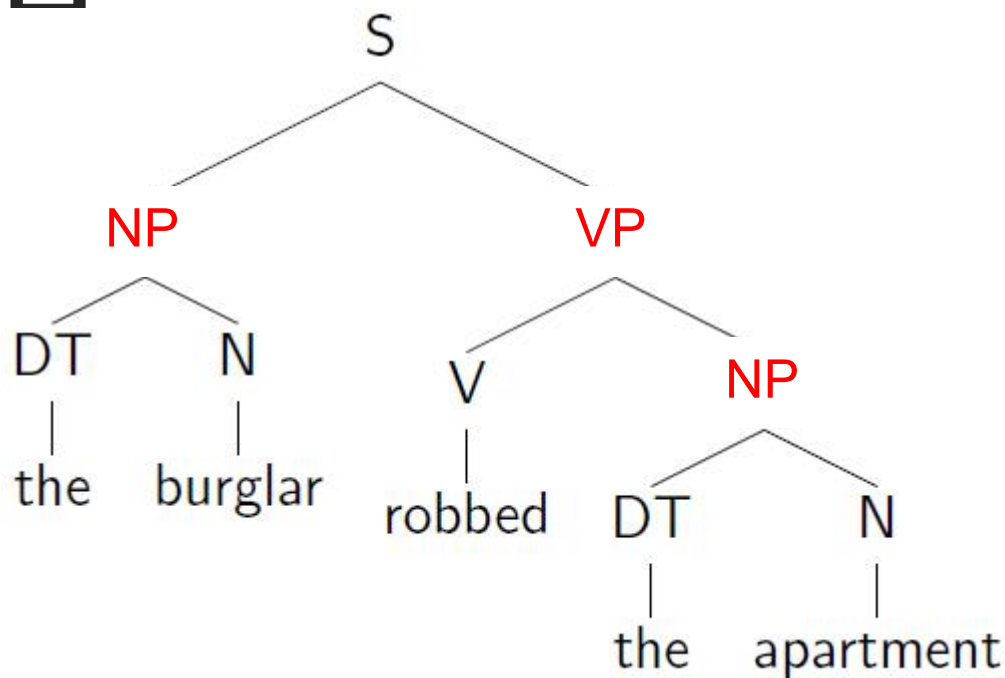
- (1) 词的词性类别

(N = noun, V = verb, DT = determiner)



句法树中包含的信息

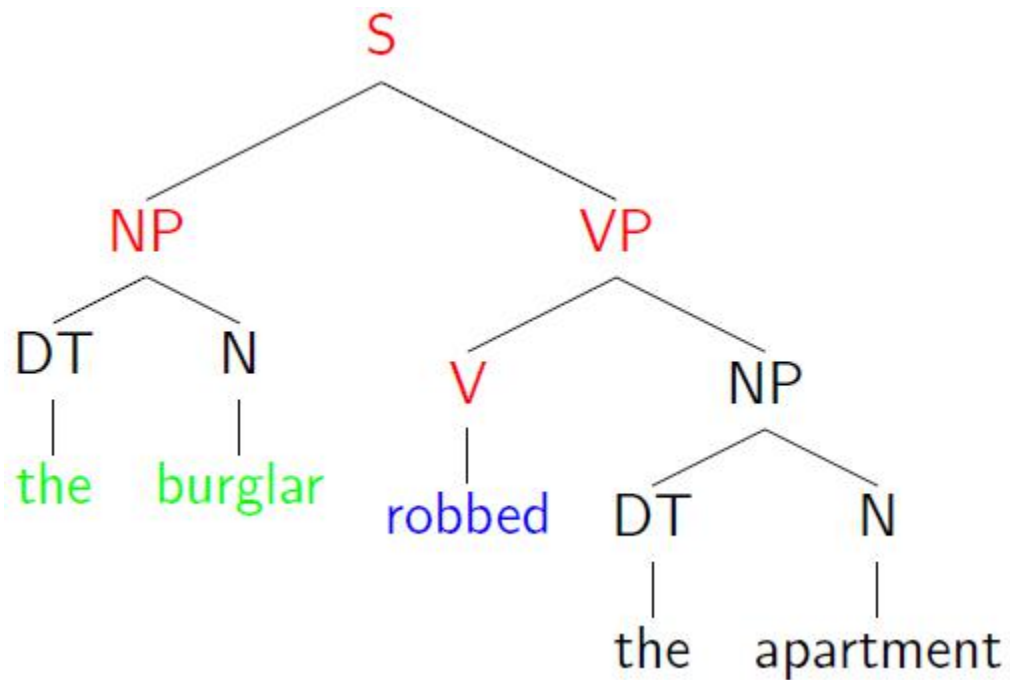
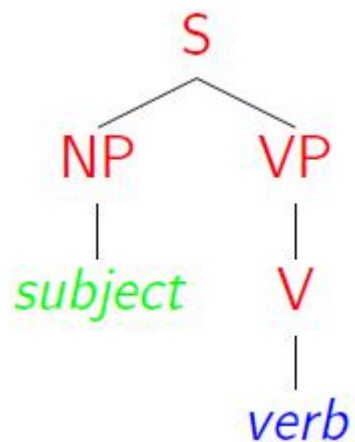
- (2) 短语



- 名词短语(NP): "the burglar", "the apartment"
- 动词短语 (VP): "robbed the apartment"
- 句子 (S): "the burglar robbed the apartment"

句法树中包含的信息

- 有用的关系：



- “the burglar”是“robbed”的主语

句法分析

- 两个目的：
 - 判断输入句子是否合乎给定的语法
 - 识别句子各部分是如何依据语法规则组成合法句子，同时生成句法树

- 两个准备：
 - 语言的**形式化描述**
(规定该语言中允许出现的结构)
 - **句法分析**技术 (根据语法来分析句子并确定其结构)

◆ 关于语言的定义

人类所特有的用来表达意思、交流思想的工具，是一种特殊的社会现象，由语音、词汇和语法构成一定的系统。

— 商务印书馆，《现代汉语词典》，1996

语言可以被看成一个抽象的数学系统。

— 吴蔚天，1994

按照一定规律构成的句子和符号串的有限或无限的集合。

— N. Chomsky

形式语言

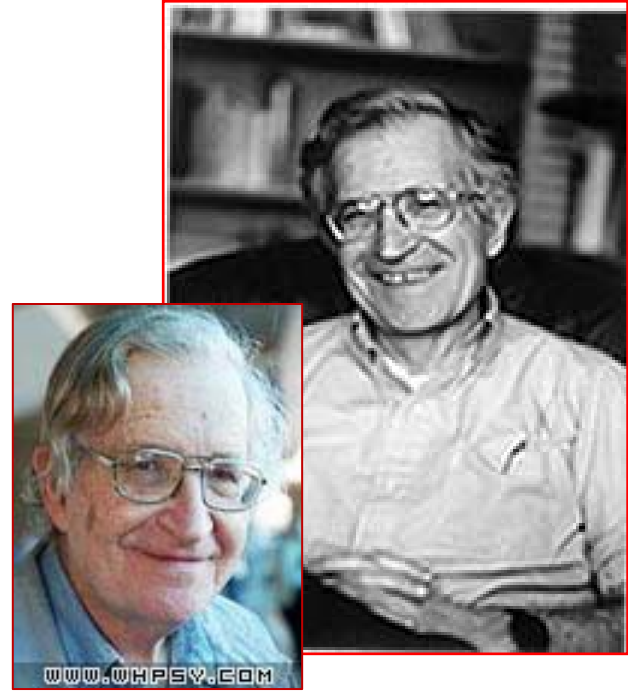
✧ 诺姆·乔姆斯基(Noam Chomsky)

—1928年12月生于美国费城

—1944年(16岁)进入UPenn 学习
哲学、语言学和数学

—1949年获学士学位、1951年获
硕士学位

—1952 起在哈佛认知研究中心研究员，后来获
UPenn博士学位；1957年(29岁) MIT副教授，
32岁成为现代语言学教授、47岁终生教授。



形式语言

◆语言描述的三种途径

- ❖ 穷举法 — 只适合句子数目有限的语言。
- ❖ 语法描述 — 生成语言中合格的句子。
- ❖ 自动机 — 对输入的句子进行检验，区别哪些是语言中的句子，哪些不是语言中的句子。

形式语言

◆形式语言的直观意义

形式语言是用来精确地描述语言（包括人工语言和自然语言）及其结构的手段。形式语言学 也称 代数语言学。

以重写规则 $\alpha \rightarrow \beta$ 的形式表示，其中， α ， β 均为字符串。顾名思义：字符串 α 可以被改写成 β 。一个初步的字符串通过不断地运用重写规则，就可以得到另一个字符串。通过选择不同的规则并以不同的顺序来运用这些规则，就可以得到不同的新字符串。

形式语言

◆形式语法的定义

形式语法是一个4元组 $G=(N, \Sigma, P, S)$, 其中 N 是非终结符的有限集合(有时也叫变量集或句法种类集); Σ 是终结符的有限集合, $N \cap \Sigma = \Phi$; $V=N \cup \Sigma$ 称总词汇表; P 是一组重写规则的有限集合: $P=\{ \alpha \rightarrow \beta \}$, 其中, α, β 是 V 中元素构成的串, 但 α 中至少应含有一个非终结符号; $S \in N$, 称为句子符或初始符。

例如: $G = (\{A, S\}, \{0, 1\}, P, S)$

$P: S \rightarrow 0 A 1 \quad 0 A \rightarrow 00A1 \quad A \rightarrow 1$

形式语言

◆推导的定义

设 $G=(N, \Sigma, P, S)$ 是一个文法, 在 $(N \cup \Sigma)^*$ 上定义关系 \Rightarrow_G (直接派生或推导)如下:

如果 $\alpha\beta\gamma$ 是 $(N \cup \Sigma)^*$ 中的符号串, 且 $\beta \rightarrow \delta$ 是 P 的产生式, 那么 $\alpha\beta\gamma \Rightarrow_G \alpha\delta\gamma$ 。

用 $\xRightarrow{+}_G$ (按非平凡方式派生) 表示 \Rightarrow_G 的传递闭包, 也就是 $(N \cup \Sigma)^*$ 上的符号串 ξ_i 到 ξ_{i+1} 的 n ($n \geq 1$) 步推导或派生。

用 $\xRightarrow{*}_G$ (派生) 表示 \Rightarrow_G 的自反和传递闭包, 即由 $(N \cup \Sigma)^*$ 上的符号串 ξ_i 到 ξ_{i+1} 经过 n ($n \geq 0$) 步的推导或派生。

如果清楚某个推导是文法 G 所产生的, 则符号 $\xRightarrow{*}_G$ 或 $\xRightarrow{+}_G$ 中的 G 可以省略不写。

形式语言

◆最左推导、最右推导和规范推导

约定每步推导中只改写最左边的那个非终结符，这种推导称为“最左推导”。

约定每步推导中只改写最右边的那个非终结符，这种推导称为“最右推导”。

最右推导也称规范推导。

形式语言

例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

$E \Rightarrow E+T$



形式语言


例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

$E \Rightarrow E+T \Rightarrow T+T$



形式语言


例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T$



形式语言


例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow a+T$



形式语言


例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow a+T \Rightarrow a+T^*F$



形式语言


例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

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



形式语言

例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

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

$\Rightarrow a+F*F \Rightarrow a+a*F$



形式语言

例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$


$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

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

$\Rightarrow a+F*F \Rightarrow a+a*F \Rightarrow a+a*a$ (最左推导)



形式语言

例3-1: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$

$P: E \rightarrow E + T \mid T \quad T \rightarrow T * F \mid F$

$F \rightarrow (E) \mid a$

字符串 $a+a*a$ 的两种推导过程:

$E \Rightarrow E+T \Rightarrow T+T \Rightarrow F+T \Rightarrow a+T \Rightarrow a+T*F$
 $\Rightarrow a+F*F \Rightarrow a+a*F \Rightarrow a+a*a$ (最左推导)

$E \Rightarrow E+T \Rightarrow E+T*F \Rightarrow E+T*a \Rightarrow E+F*a \Rightarrow E+a*a$
 $\Rightarrow T+a*a \Rightarrow F+a*a \Rightarrow a+a*a$ (最右推导)

形式语言

◆句型与句子

一些特殊类型的符号串为文法 $G=(N, \Sigma, P, S)$ 的句子形式(句型):

- (1) S 是一个句子形式;
- (2) 如果 $\alpha\beta\gamma$ 是一个句子形式, 且 $\beta \rightarrow \delta$ 是 P 的产生式, 则 $\alpha\delta\gamma$ 也是一个句子形式;

文法 G 的不含非终结符的句子形式称为 G 生成的句子。由文法 G 生成的语言, 记作 $L(G)$, 指 G 生成的所有句子的集合。即: $L(G)=\{x \mid x \in \Sigma, S \xrightarrow{+}_G x\}$

形式语言

◆正则文法

如果文法 $G=(N, \Sigma, P, S)$ 的 P 中的规则满足如下形式: $A \rightarrow Bx$, 或 $A \rightarrow x$, 其中 $A, B \in N$, $x \in \Sigma$, 则称该文法为正则文法 或称 3型文法。(左线性正则文法)

如果 $A \rightarrow xB$, 则该文法称为右线性正则文法。

形式语言

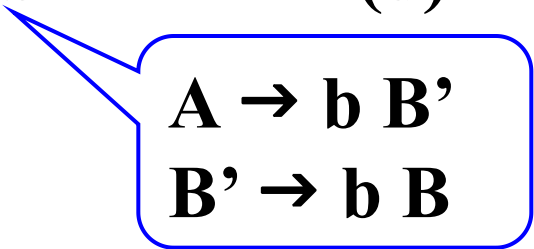
例3-2: $G = (N, \Sigma, P, S)$,

$$N = \{S, A, B\}, \quad \Sigma = \{a, b\},$$

$$P: (a) S \rightarrow a A \qquad (b) A \rightarrow a A$$

$$(c) A \rightarrow b b B \qquad (d) B \rightarrow b B$$

$$(e) B \rightarrow b$$


$$\begin{aligned} A &\rightarrow b B' \\ B' &\rightarrow b B \end{aligned}$$

$$L(G) = \{a^n b^m\}, n \geq 1, m \geq 3$$

形式语言

◆上下文无关文法 (context-free grammar, CFG)

如果 P 中的规则满足如下形式: $A \rightarrow \alpha$, 其中 $A \in N$, $\alpha \in (N \cup \Sigma)^*$, 则称该文法为上下文无关文法 (CFG) 或称 2 型文法。

形式语言

例3-3: $G = (N, \Sigma, P, S),$

$$N = \{S, A, B, C\}, \quad \Sigma = \{a, b, c\},$$

$$P: \text{(a) } S \rightarrow A B C \quad \text{(b) } A \rightarrow a A \mid a$$

$$\text{(c) } B \rightarrow b B \mid b \quad \text{(d) } C \rightarrow B A \mid c$$

$$L(G) = \{a^n b^m a^k c^\alpha\}, n \geq 1, m \geq 1, k \geq 0, \alpha \in \{0, 1\}$$

(如果 $k > 0$ 的话, $\alpha = 0$, 否则, $\alpha = 1$ 。)

形式语言

◆上下文有关文法

(context-sensitive grammar, CSG)

如果 P 中的规则满足如下形式: $\alpha A \beta \rightarrow \alpha \gamma \beta$, 其中 $A \in N$, $\alpha, \beta, \gamma \in (N \cup \Sigma)^*$, 且 γ 至少包含一个字符, 则称该文法为上下文有关文法(CSG) 或称 1 型文法。

另一种定义: if $x \rightarrow y$, $x \in (N \cup \Sigma)^+$,
 $y \in (N \cup \Sigma)^*$, 并且 $|y| \geq |x|$ 。

形式语言

例3-4: $G = (N, \Sigma, P, S)$

$$N = \{S, A, B, C\}, \quad \Sigma = \{a, b, c\},$$

$$P: \begin{array}{ll} \text{(a)} S \rightarrow A B C & \text{(b)} A \rightarrow a A \mid a \\ \text{(c)} B \rightarrow b B \mid b & \text{(d)} B C \rightarrow B c c \end{array}$$

$$L(G) = \{a^n b^m c^2\}, n \geq 1, m \geq 1$$

形式语言

◆ 无约束文法（无限制重写系统）

如果 P 中的规则满足如下形式： $\alpha \rightarrow \beta$ ， α ， β 是字符串，则称 G 为无约束文法，或称 0 型文法。

**显然，每一个正则文法都是上下文无关文法，
每一个上下文无关文法都是上下文有关文法，而
每一个上下文有关文法都是0型文法，即：**

$$L(G_0) \supseteq L(G_1) \supseteq L(G_2) \supseteq L(G_3)$$

形式语言

◆语言与文法类型的约定

如果一种语言能由几种文法所产生，则把这种语言称为在这几种文法中受限制最多的那种文法所产生的语言。

例3-5: $G = (\{S, A, B\}, \{a, b\}, P, S)$

$P: S \rightarrow aB \quad S \rightarrow bA \quad A \rightarrow aS \quad A \rightarrow bAA$

$A \rightarrow a \quad B \rightarrow bS \quad B \rightarrow aBB \quad B \rightarrow b$

G 为上下文无关文法。

$L(G) = \{ \text{等数量的 } a \text{ 和 } b \text{ 构成的链} \}$

形式语言

◆ CFG 产生的语言句子的派生树表示

CFG $G=(N, \Sigma, P, S)$ 产生一个句子的派生树由如下步骤构成:

- (1) 对于 $\forall x \in N \cup \Sigma$ 给一个标记作为节点, S 作为树的根节点。
- (2) 如果一个节点的标记为 A , 并且它至少有一个除它自身以外的后裔, 则 $A \in N$ 。
- (3) 如果一个节点的标记为 A , 它的 k ($k > 0$) 个直接后裔节点按从左到右的次序依次标记为 A_1, A_2, \dots, A_k , 则 $A \rightarrow A_1 A_2 \dots A_k$ 一定是 P 中的一个产生式。

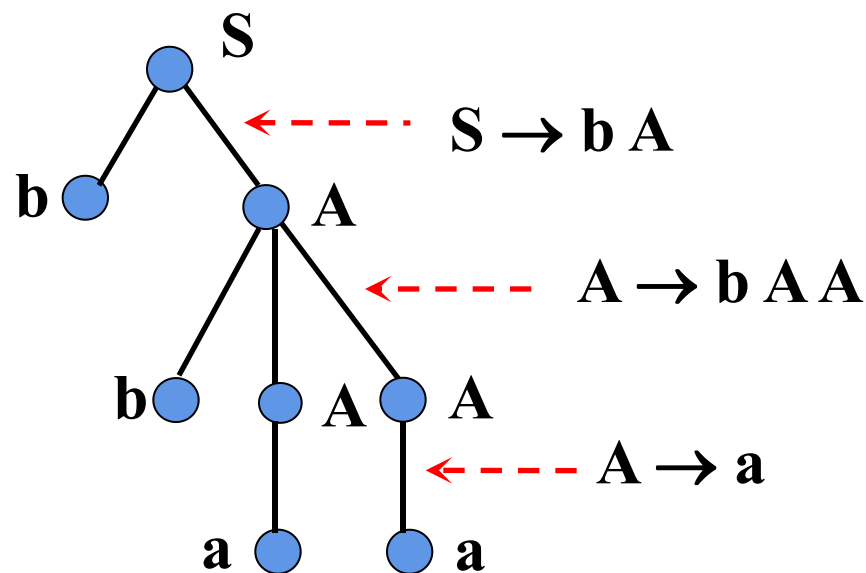
形式语言

例如, $G = (\{S, A\}, \{a, b\}, P, S)$

$P: S \rightarrow bA \quad A \rightarrow bAA \quad A \rightarrow a$

G 所产生的句子 $bbaa$ 可以由下面的生树表示:

$S \Rightarrow bA$
 $\Rightarrow bbAA$
 $\Rightarrow bbaA$
 $\Rightarrow bbaa$



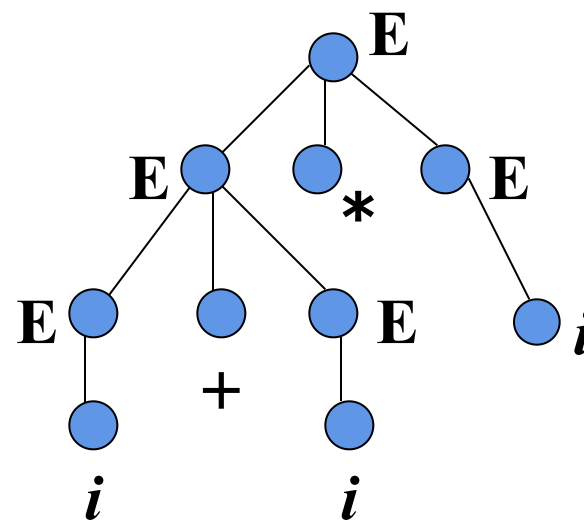
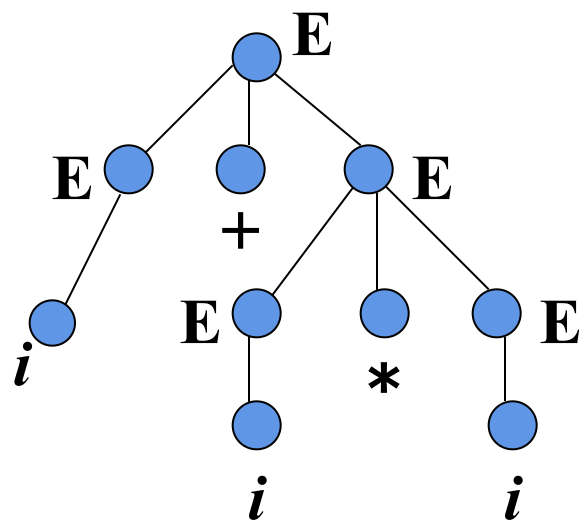
形式语言

◆ 上下文无关文法的二义性

一个文法 G ，如果存在某个句子有不只一棵分析树与之对应，那么称这个文法是二义的。

形式语言

例: $G(E): E \rightarrow E + E \mid E * E \mid (E) \mid E - E \mid i$
对于句子 $i + i * i$ 有两棵对应的分析树。



形式语言

例：给定文法 $G(S)$ ：

- ① $S \rightarrow P NP \mid PP Aux NP$
- ② $PP \rightarrow P NP$
- ③ $NP \rightarrow NN \mid NP Aux NP$
- ④ $P \rightarrow \text{关于}$
- ⑤ $NN \rightarrow \text{鲁迅} \mid \text{文章}$
- ⑥ $Aux \rightarrow \text{的}$

短语“关于鲁迅的文章”的推导。

$$\begin{aligned} S &\Rightarrow P NP \Rightarrow \text{关于 } NP \Rightarrow \text{关于 } NP Aux NP \\ &\Rightarrow \text{关于 } NN Aux NP \Rightarrow \text{关于鲁迅 } Aux NP \\ &\Rightarrow \text{关于鲁迅的 } NP \Rightarrow \text{关于鲁迅的 } NN \\ &\Rightarrow \text{关于鲁迅的文章} \end{aligned}$$

形式语言

例：给定文法 $G(S)$ ：

① $S \rightarrow P\ NP \mid PP\ Aux\ NP$ ② $PP \rightarrow P\ NP$

③ $NP \rightarrow NN \mid NP\ Aux\ NP$ ④ $P \rightarrow \text{关于}$

⑤ $NN \rightarrow \text{鲁迅} \mid \text{文章}$ ⑥ $Aux \rightarrow \text{的}$

短语“关于鲁迅的文章”的推导。

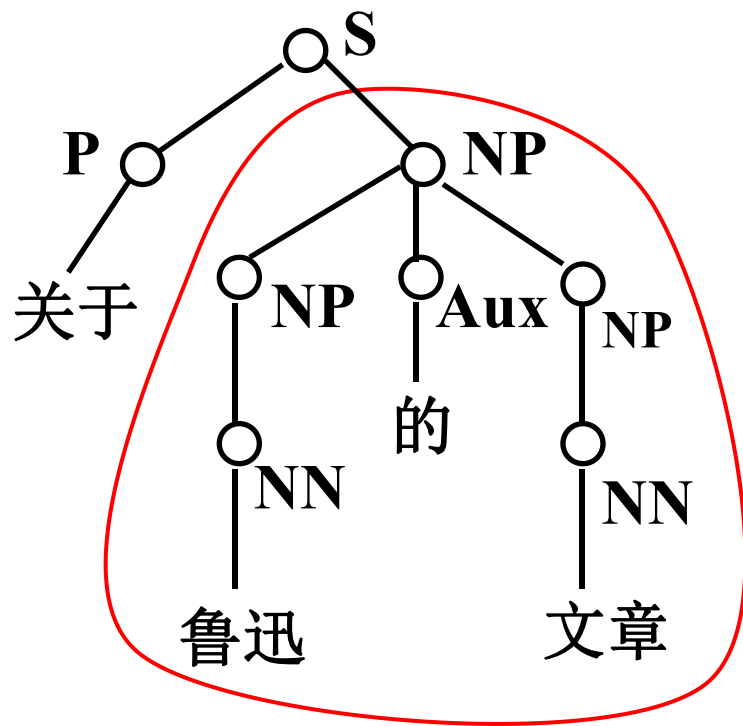
$S \Rightarrow PP\ Aux\ NP \Rightarrow P\ NP\ Aux\ NP \Rightarrow \text{关于}\ NP\ Aux\ NP$

$\Rightarrow \text{关于}\ NN\ Aux\ NP \Rightarrow \text{关于鲁迅}\ Aux\ NP$

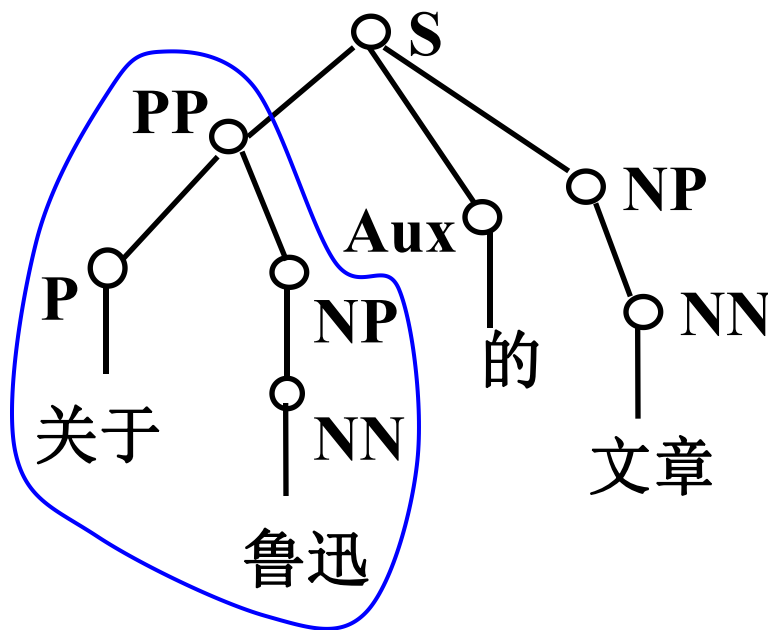
$\Rightarrow \text{关于鲁迅的}\ NP \Rightarrow \text{关于鲁迅的}\ NN$

$\Rightarrow \text{关于鲁迅的文章}$

形式语言



短语的派生树 - (1)



短语的派生树 - (2)