

第九章 句法分析

目录

- 9.1 概述
- 9.2 短语结构分析
- 9.3 线图分析法
- 9.4 CYK分析算法
- 9.5 概率上下文无关文法
- 9.6 依存句法分析
- 9.7 短语结构与依存结构的关系

9.1 概述

9.1 概述

◆ 任务：

句法分析(syntactic parsing)的任务就是识别句子的句法结构(syntactic structure)。

◆ 类型：

- 短语结构分析 (Phrase parsing)
 - 完全句法分析 (Full parsing)
 - 局部句法分析 (Partial parsing)
- 依存句法分析 (Dependency parsing)

9.2 短语结构分析

9.2 短语结构分析

◆ 句法分析的例子（参见前面第4章）

例：

他还提出一系列具体措施的政策要点。

他/PN 还/AD 提出/VV 一/CD 系列/M 具体/JJ 措施
/NN 和/CC 政策/NN 要点/NN 。 /PU

9.2 短语结构分析

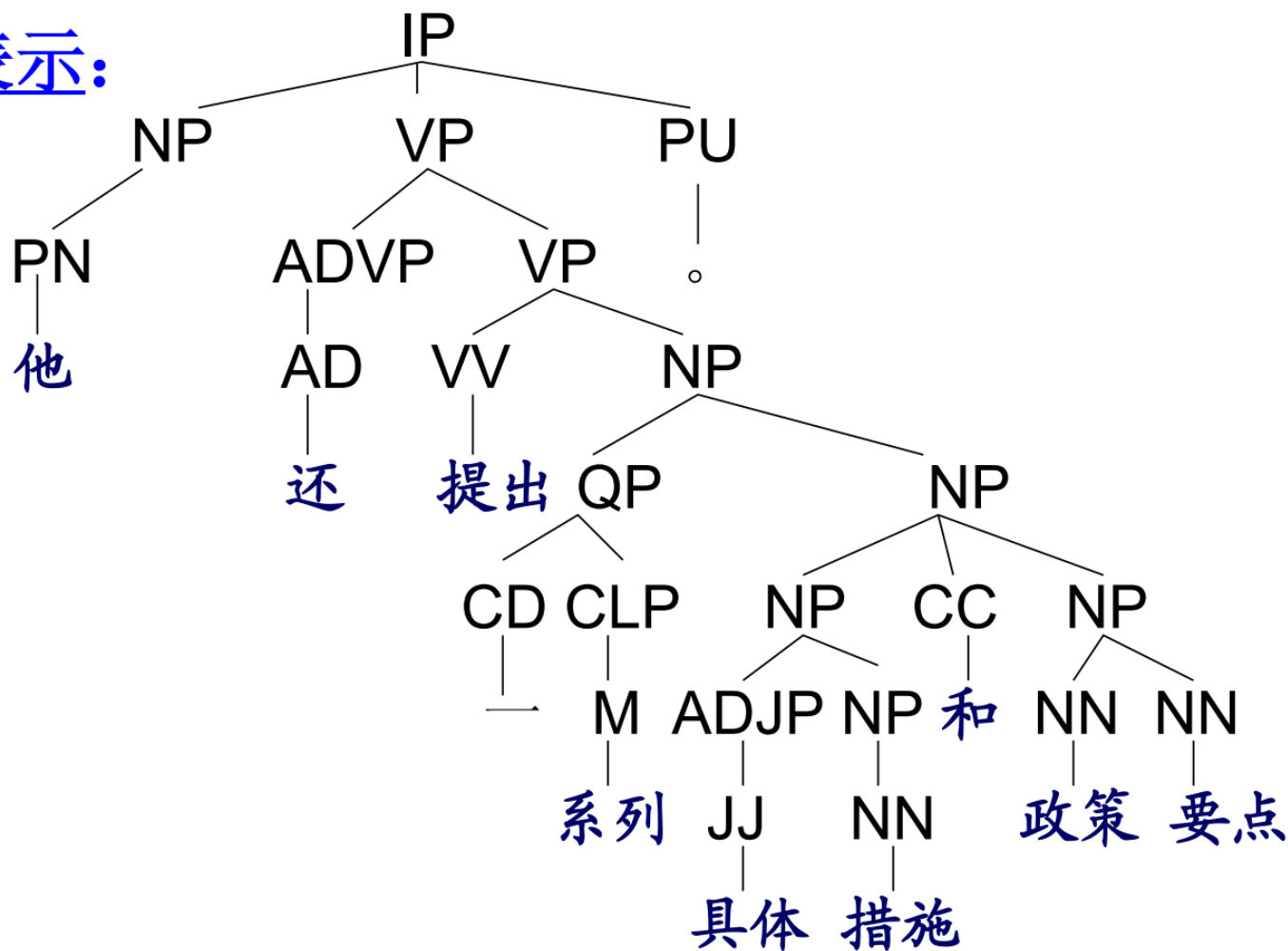
◆ 句法分析的例子（参见前面第4章）

(IP (NP-SBJ (PN 他))
 (VP (ADVP (AD 还))
 (VP (VV 提出)
 (NP-OBJ(QP (CD 一)
 (CLP (M 系列)))
 (NP (NP(ADJP (JJ 具体)
 (NP (NN 措施))))
 (CC 和)
 (NP (NN 政策)
 (NN 要点))))))))
 (PU 。))

9.2 短语结构分析

◆ 句法分析的例子（参见前面第4章）

树状表示：



9.2 短语结构分析

◆ 目标:

实现高正确率、高鲁棒性(robustness)、高速度的自动句法分析过程。

◆ 困难:

自然语言中存在大量的复杂的结构歧义(structural ambiguity)。

9.2 短语结构分析

◆ 结构歧义

例如：

(1) I saw a boy **in the park**.

[I saw a boy] in the park.

I saw a [boy in the park].

(2) I saw a boy **in the park** **with a telescope**.

(3) I saw a boy swimming **on the bridge**.

(4) 关于鲁迅的文章。

(5) 把重要的书籍和手稿带走了。

9.2 短语结构分析

◆ 结构歧义

英语中的结构歧义随介词短语组合个数的增加而不断加深的，这个组合个数我们称之为开塔兰数(Catalan number, 记作 C_N)。

如果句子中存在这样 n (n 为自然数)个介词短语, C_N 可由下式获得 [Samuelsson, 2000]:

$$C_N = \binom{2n}{n} \frac{1}{n+1} = \frac{(2n)!}{(n!)^2(n+1)}$$

9.2 短语结构分析

◆ 基本方法和开源的句法分析器

➤ 基于CFG规则的分析方法:

- 线图分析法 (chart parsing)
- CYK 算法
- Earley (厄尔利)算法
- LR 算法 / Tomita 算法
 - Top-down: Depth-first/ Breadth-first
 - Bottom-up

9.2 短语结构分析

◆ 基本方法和开源的句法分析器

➤ 基于PCFG的分析方法

PCFG: Probabilistic Context-Free Grammar (有时也写作 Stochastic CFG, SCFG)

➤ 其他统计模型

➤ 部分开源的句法分析器

9.3 线图分析法

9.3 线图分析法

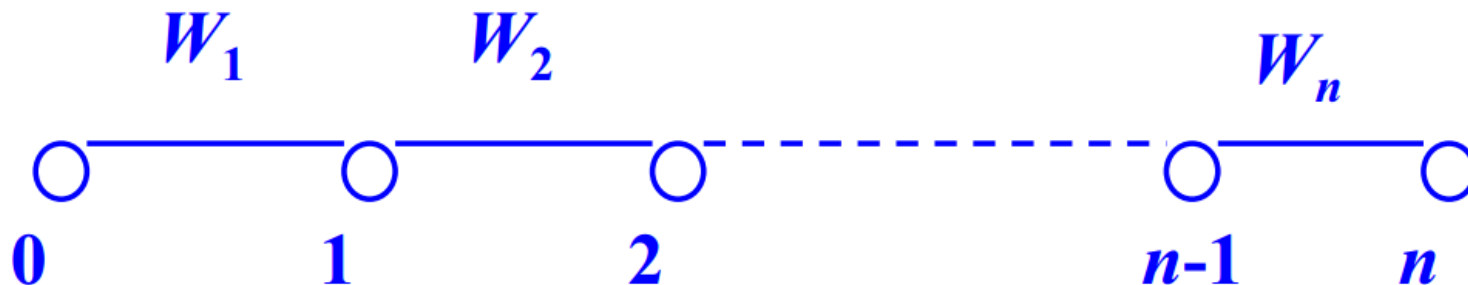
◆ 三种策略

- 自底向上 (Bottom-up)
- 从上到下 (Top-down)
- 从上到下和从下到上结合

9.3 线图分析法

◆ 自底向上的 Chart 分析算法

- 给定一组 CFG 规则: $XP \rightarrow \alpha_1 \cdots \alpha_n (n \geq 1)$
- 给定一个句子的词性序列: $S = W_1 W_2 \cdots W_n$
- 构造一个线图: 一组结点和边的集合;

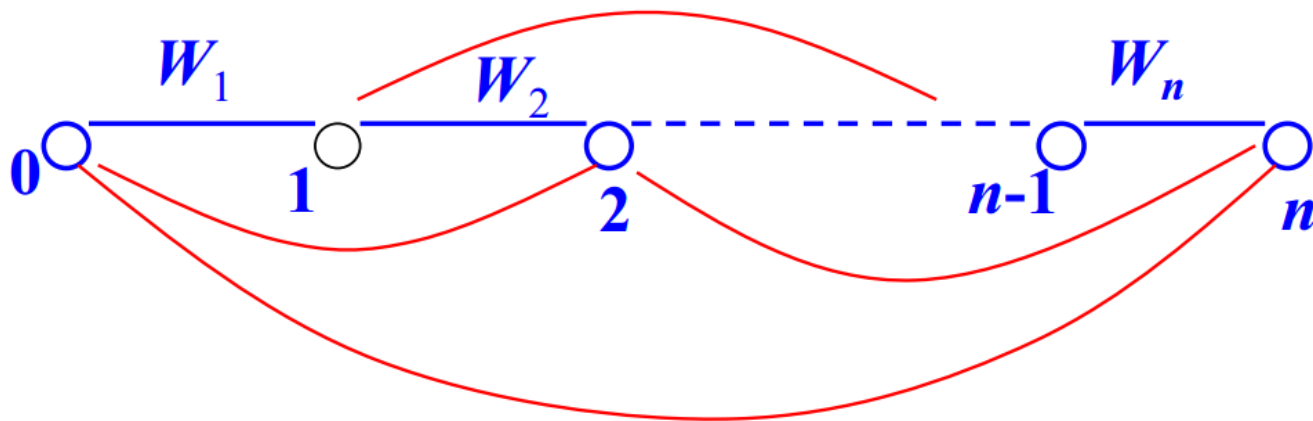


- 建立一个二维表: 记录每一条边的起始位置和终止位置。

9.3 线图分析法

◆ 自底向上的 Chart 分析算法

执行操作： 查看任意相邻几条边上的词性串是否与某条重写规则的右部相同，如果相同，则增加一条新的边跨越原来相应的边，新增加边上的标记为这条重写规则的头(左部)。重复这个过程，直到没有新的边产生。



9.3 线图分析法

◆ 自底向上的 Chart 分析算法

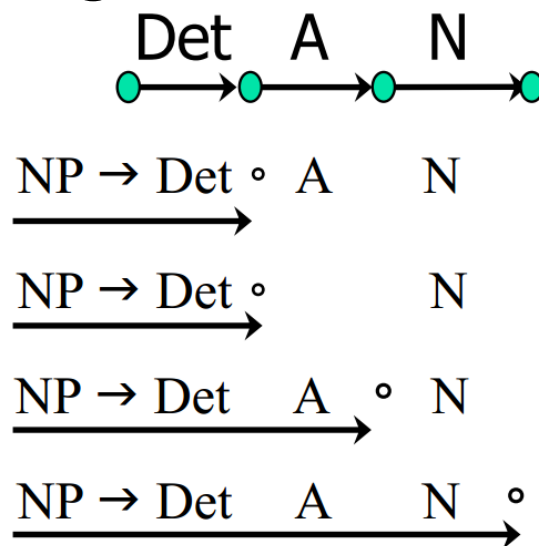
点规则：用于表示规则右部被归约(reduce)的程度。

设有规则：NP \rightarrow Det A N

NP \rightarrow Det N

NP \rightarrow A N

句子：The good book



9.3 线图分析法

◆ 自底向上的 Chart 分析算法

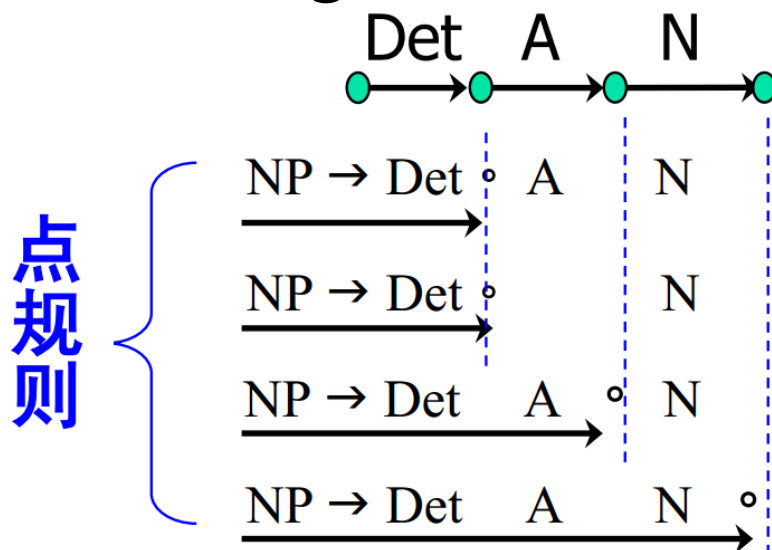
点规则：用于表示规则右部被归约(reduce)的程度。

设有规则：NP \rightarrow Det A N

NP \rightarrow Det N

NP \rightarrow A N

句子：The good book



9.3 线图分析法

◆ 自底向上的 Chart 分析算法

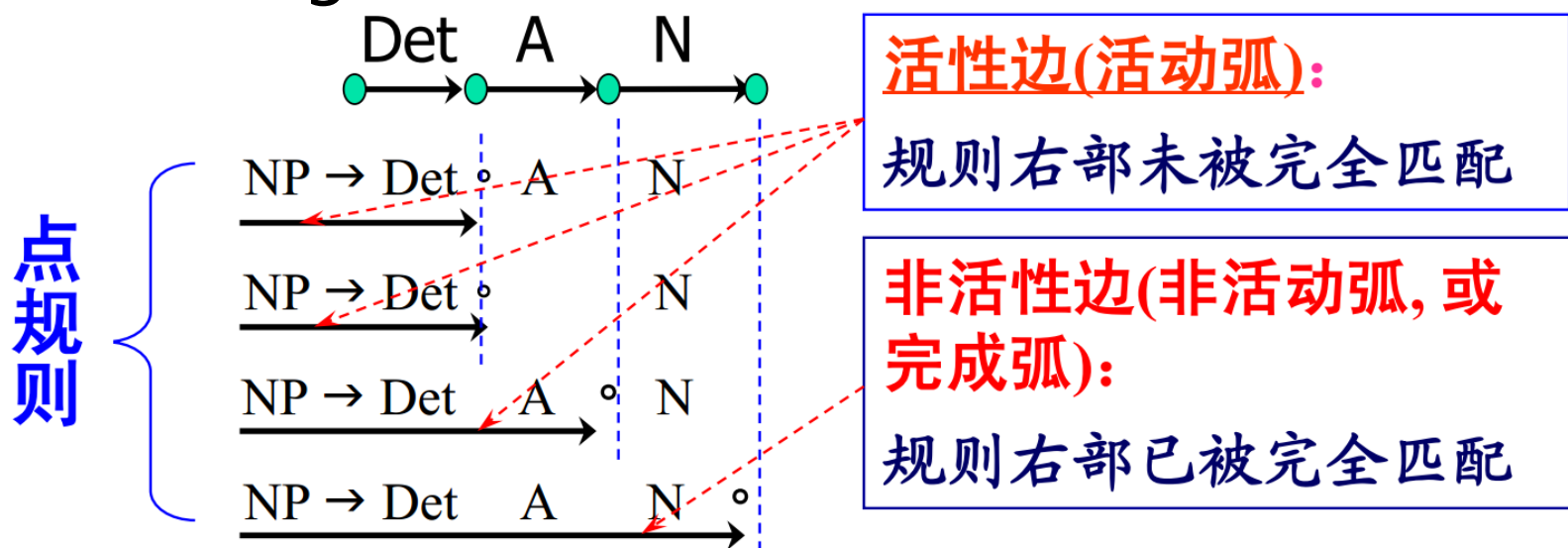
点规则： 用于表示规则右部被归约(reduce)的程度。

设有规则： $NP \rightarrow Det\ A\ N$

$NP \rightarrow Det\ N$

$NP \rightarrow A\ N$

句子： The good book



9.3 线图分析法

◆ 数据结构

- **线图(Chart)**: 保存分析过程中已经建立的成分(包括终结符和非终结符)、位置(包括起点和终点)。通常以 $n \times n$ 的数组表示(n 为句子包含的词数)。
- **代理表(待处理表)(Agenda)**: 记录刚得到的一些重写规则所代表的成分, 这些重写规则的右端符号串与输入词性串(或短语标志串)中的一段完全匹配, 通常以栈或线性队列表示。
- **活动边集(ActiveArc)**: 记录那些右端符号串与输入串的某一段相匹配, 但还未完全匹配的重写规则, 通常以数组或列表存储。

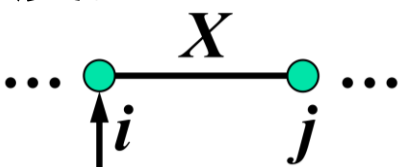
9.3 线图分析法

◆ 算法描述

从输入串的起始位置到最后位置，循环执行如下步骤：

(1) 如果待处理表(Agenda)为空，则找到下一个位置上的词，将该词对应的(所有)词类 X 附以 (i, j) 作为元素放到待处理表中，即 $X(i, j)$ 。其中， i, j 分别是该词的起始位置和终止位置， $j > i$ ， $j - i$ 为该词的长度。

(2) 从 Agenda 中取出一个元素 $X(i, j)$ 。



The diagram shows a horizontal line segment representing an arc. Above the line is the label X . Below the line, there are two green circular nodes. The left node is labeled i with an upward-pointing arrow, and the right node is labeled j . Ellipses (\dots) are placed at both ends of the line segment, indicating it is part of a larger structure.

(3) 对于每条规则 $A \rightarrow X\gamma$ ，将 $A \rightarrow X(i, j)$ 加入活动边集ActiveArc 中，然后调用**扩展弧子程序**。

9.3 线图分析法

◆ 算法描述

扩展弧子程序:

- 将 X 插入图表(Chart)的 (i, j) 位置中。
- 对于活动边集(ActiveArc)中每个位置为 (k, i) ($1 \leq k < i$) 的点规则, 如果该规则具有如下形式: $A \rightarrow \alpha X$, 如果 $A = S$, 则把 $S(1, n + 1)$ 加入到 Chart 中, 并给出一个完整的分析结果; 否则将 $A(k, j)$ 加入到 Agenda表中。
- 对于每个位置为 (k, i) 的点规则: $A \rightarrow \alpha X \beta$, 则将 $A \rightarrow \alpha X \beta(k, j)$ 加入到活动边集中。

9.3 线图分析法

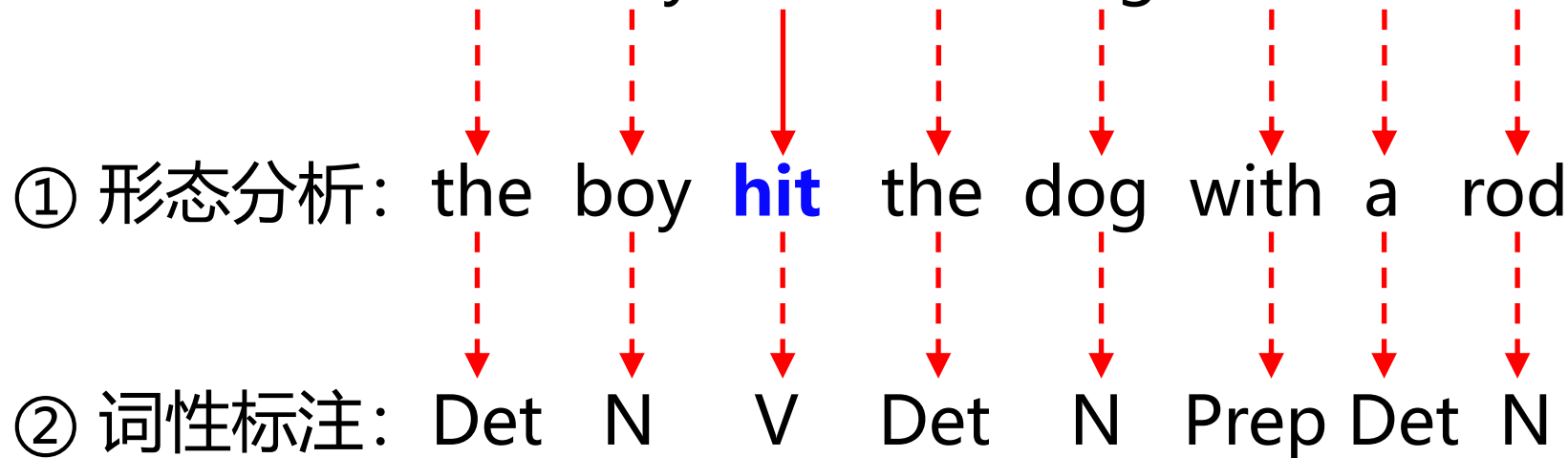
例:

G (S): $S \rightarrow NP VP$, $NP \rightarrow Det N$

$VP \rightarrow V NP$, $VP \rightarrow VP PP$

$PP \rightarrow Prep NP$

输入句子: the boy hits the dog with a rod



9.3 线图分析法

③ 句法分析

Agenda

ActiveArc

Chart

Acts

① Det (1, 2)

② NP \rightarrow Det \circ N(1, 2)

③ Det (1, 2)

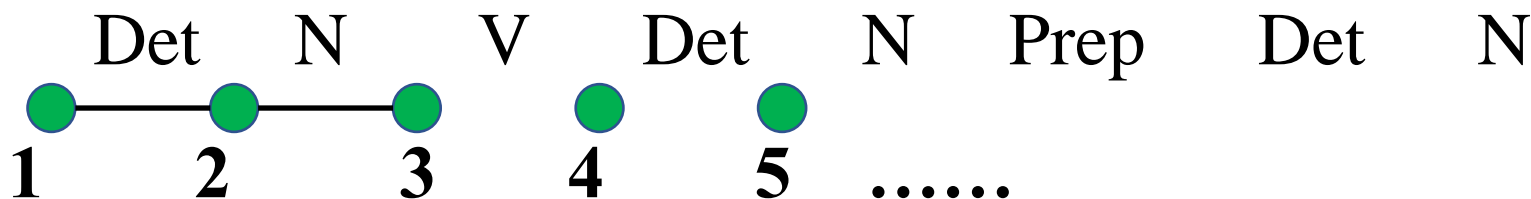
返回

④ N(2, 3)

无新的活动边加入

⑤ N(2, 3)

扩展



(1) $S \rightarrow NP VP$

(2) $NP \rightarrow Det N$

(3) $VP \rightarrow V NP$

(4) $VP \rightarrow VP PP$

(5) $PP \rightarrow Prep NP$

9.3 线图分析法

③ 句法分析

Agenda

ActiveArc

Chart

Acts

① Det (1, 2)

② $NP \rightarrow Det \circ N(1, 2)$

③ Det (1, 2)

返回

④ N(2, 3)

⑥ $NP \rightarrow Det N \circ (1, 3)$

⑤ N(2, 3)

扩展

⑦ NP(1, 3)

⑧ $S \rightarrow NP \circ VP(1, 3)$

⑨ NP(1, 3)

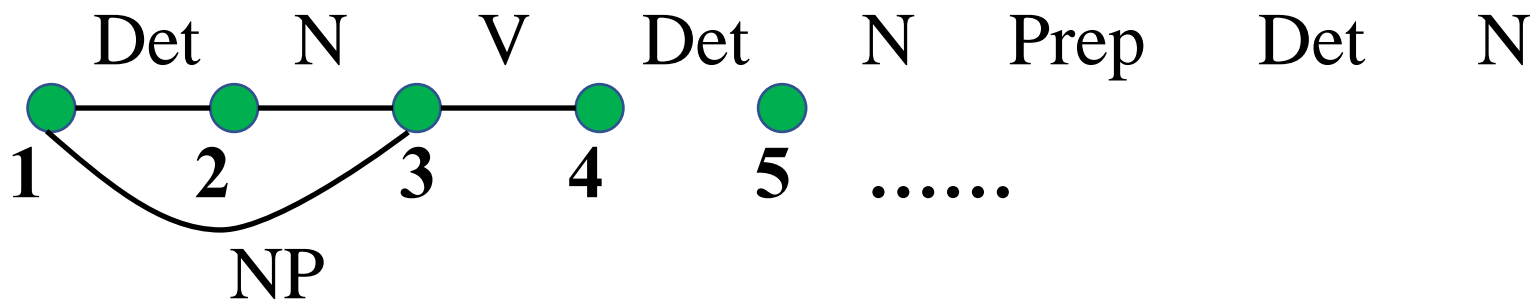
返回

⑩ V(3, 4)

⑪ $VP \rightarrow V \circ NP(3, 4)$

⑫ V(3, 4)

返回



(1) $S \rightarrow NP VP$

(2) $NP \rightarrow Det N$

(3) $VP \rightarrow V NP$

(4) $VP \rightarrow VP PP$

(5) $PP \rightarrow Prep NP$

9.3 线图分析法

③ 句法分析

Agenda

ActiveArc

Chart

Acts

⑬ Det (4, 5)

⑭ NP \rightarrow Det \circ N(4, 5)

⑮ Det (4, 5)

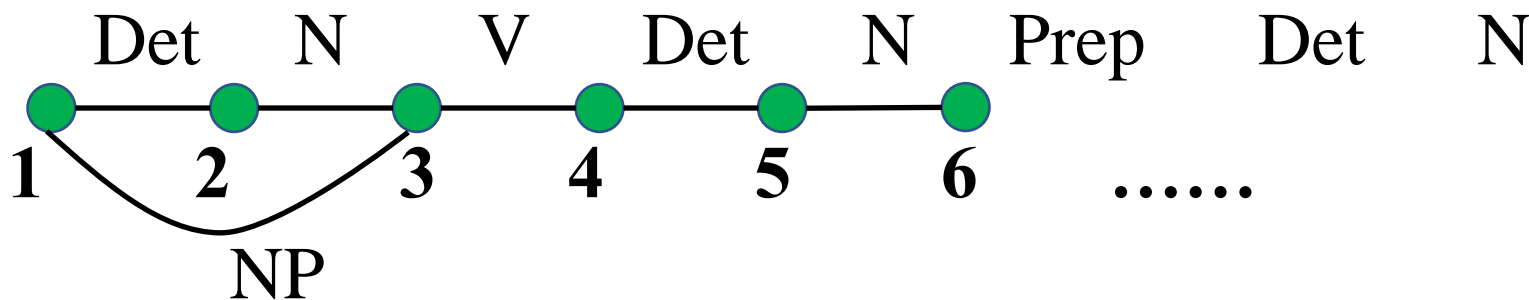
返回

⑯ N(5, 6)

无新的活动边加入

⑰ N(5, 6)

扩展



(1) $S \rightarrow NP VP$

(2) $NP \rightarrow Det N$

(3) $VP \rightarrow V NP$

(4) $VP \rightarrow VP PP$

(5) $PP \rightarrow Prep NP$

9.3 线图分析法

③
句
法
分
析

Agenda

ActiveArc

Chart

Acts

⑬ Det (4, 5)

⑭ $NP \rightarrow Det \circ N(4, 5)$

⑮ Det (4, 5)

返回

⑯ N(5, 6)

⑰ $NP \rightarrow Det N \circ (4, 6)$

⑱ N(5, 6)

扩展

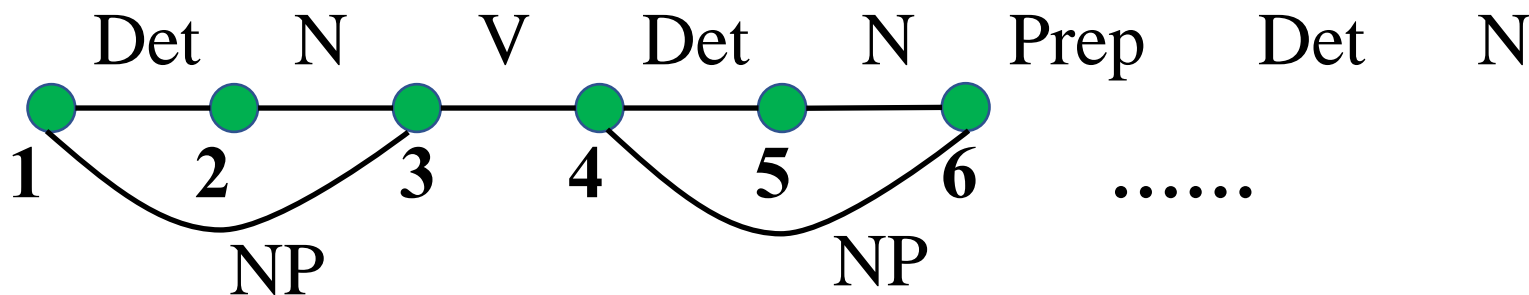
⑲ NP(4, 6)

⑳ $S \rightarrow NP \circ VP(4, 6)$

㉑ NP(4, 6)

扩展

将第11步的点规则 $VP \rightarrow V \circ NP(3, 4)$ 扩展



(1) $S \rightarrow NP VP$

(2) $NP \rightarrow Det N$

(3) $VP \rightarrow V NP$

(4) $VP \rightarrow VP PP$

(5) $PP \rightarrow Prep NP$

9.3 线图分析法

③ 句法分析

Agenda

ActiveArc

Chart

Acts

22 $VP \rightarrow V NP \circ (3, 6)$

23 $VP(3, 6)$

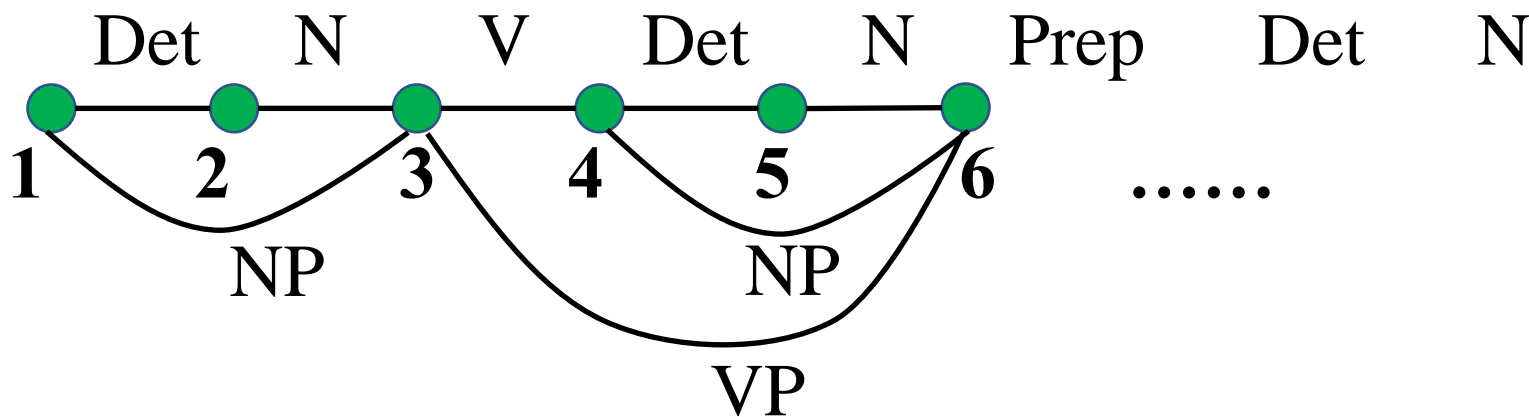
24 $VP \rightarrow VP \circ PP(3, 6)$

25 $VP(3, 6)$

扩展

.....

.....



(1) $S \rightarrow NP VP$

(2) $NP \rightarrow Det N$

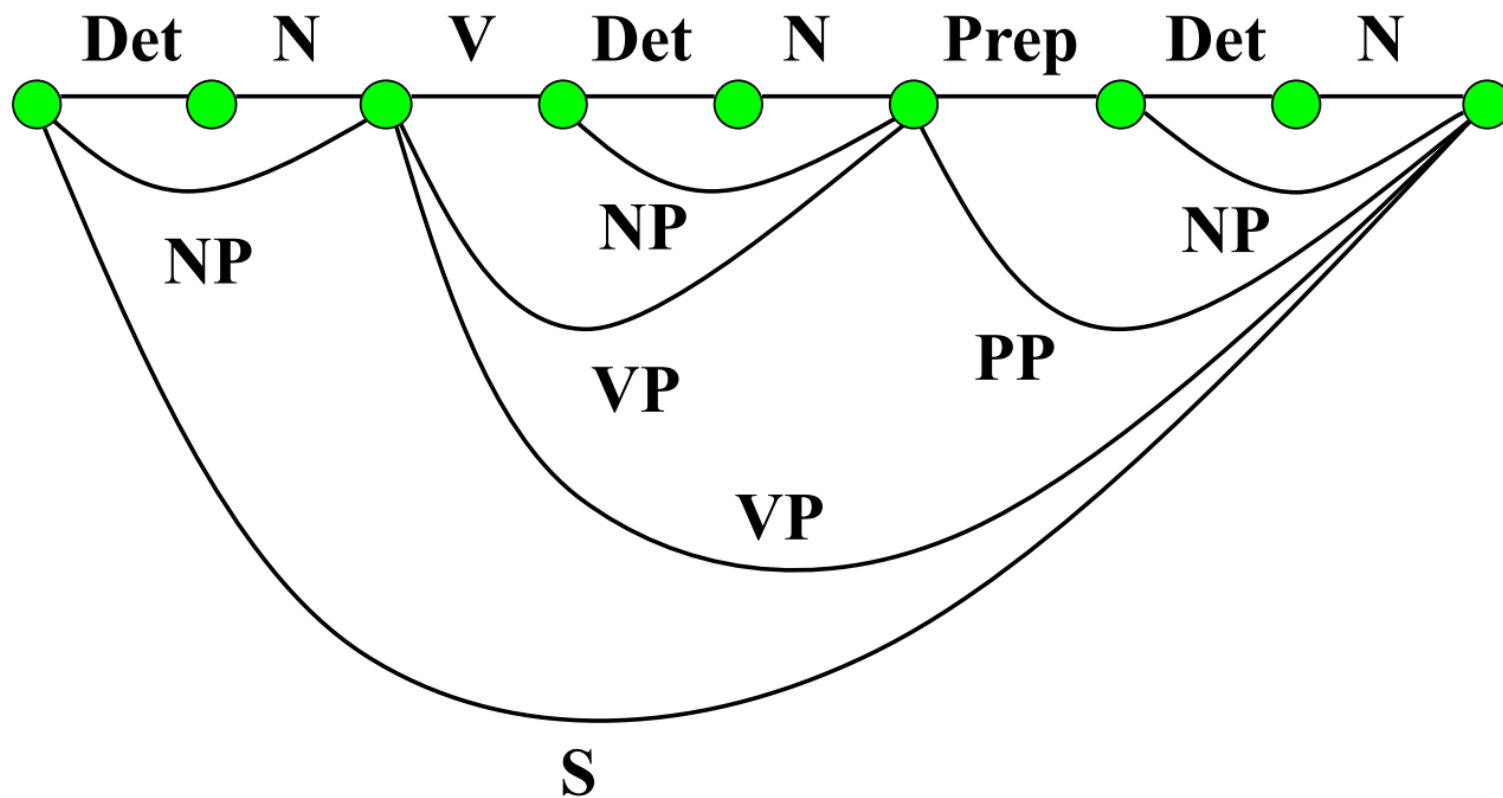
(3) $VP \rightarrow V NP$

(4) $VP \rightarrow VP PP$

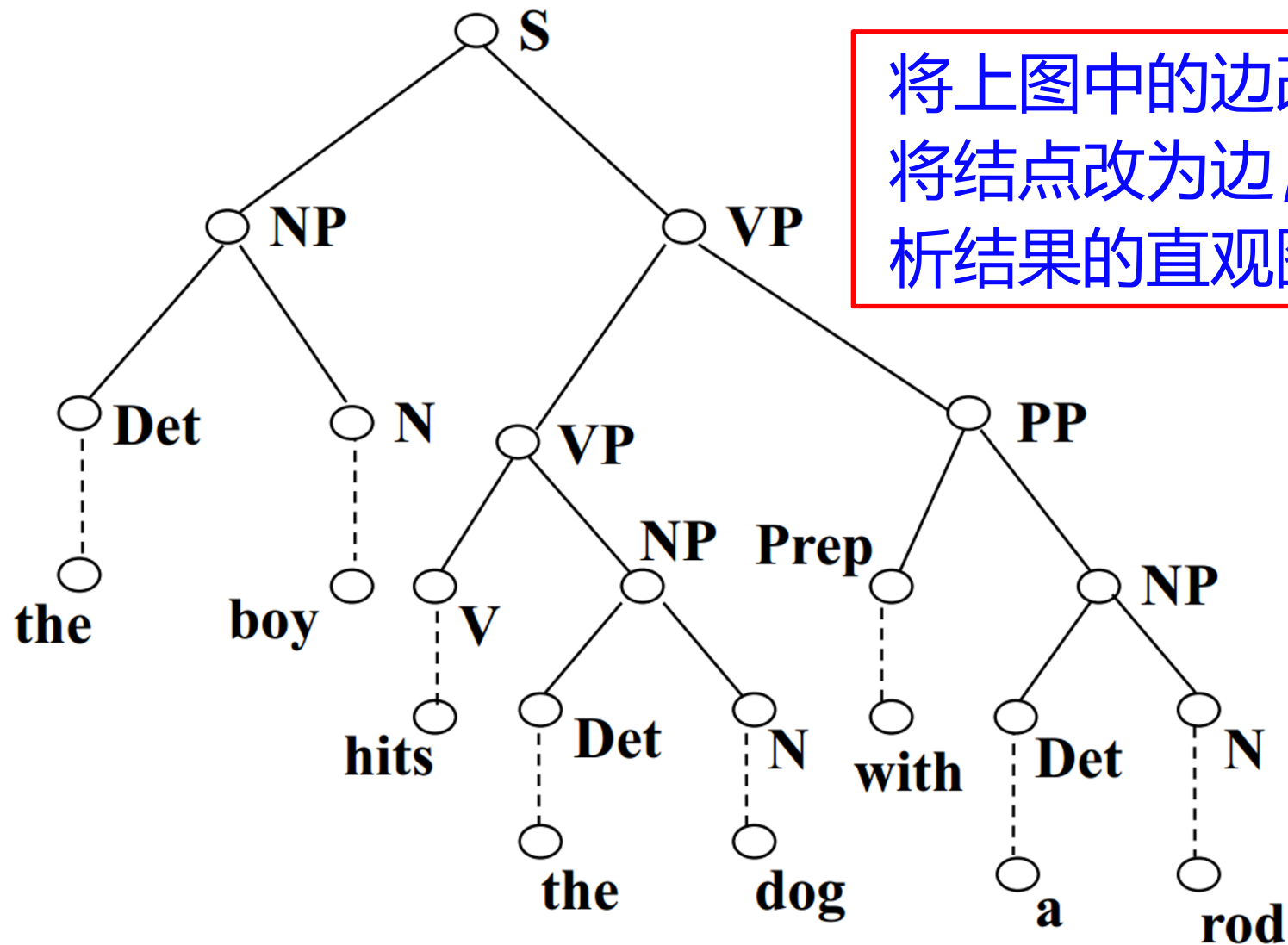
(5) $PP \rightarrow Prep NP$

9.3 线图分析法

最后分析结果：



9.3 线图分析法



将上图中的边改为结点，
将结点改为边，得到分
析结果的直观图。

9.3 线图分析法

◆ Chart parsing 算法评价

➤ 优点:

- 算法简单，容易实现，开发周期短。

➤ 弱点:

- 算法效率低，时间复杂度为 Kn^3 ;
- 需高质量的规则，分析结果与规则质量密切相关;
- 难以区分歧义结构。

9.4 CYK分析算法

9.4 CYK分析算法

◆ Cocke-Younger-Kasami (CYK) 算法

- 基于动态规划思想设计的自底向上语法分析算法
- 对 Chomsky 文法进行范式化:

$$A \rightarrow w \text{ 或 } A \rightarrow BC$$

$$A, B, C \in V_N, w \in V_T, G = (V_N, V_T, P, S)$$

- 构造 $(n + 1) \times (n + 1)$ 识别矩阵, n 为输入句子长度。
假设输入句子 $x = w_1 w_2 \cdots w_n$, w_i 为构成句子的单词, $n = |x|$ 。

9.4 CYK分析算法

◆ Cocke-Younger-Kasami (CYK) 算法

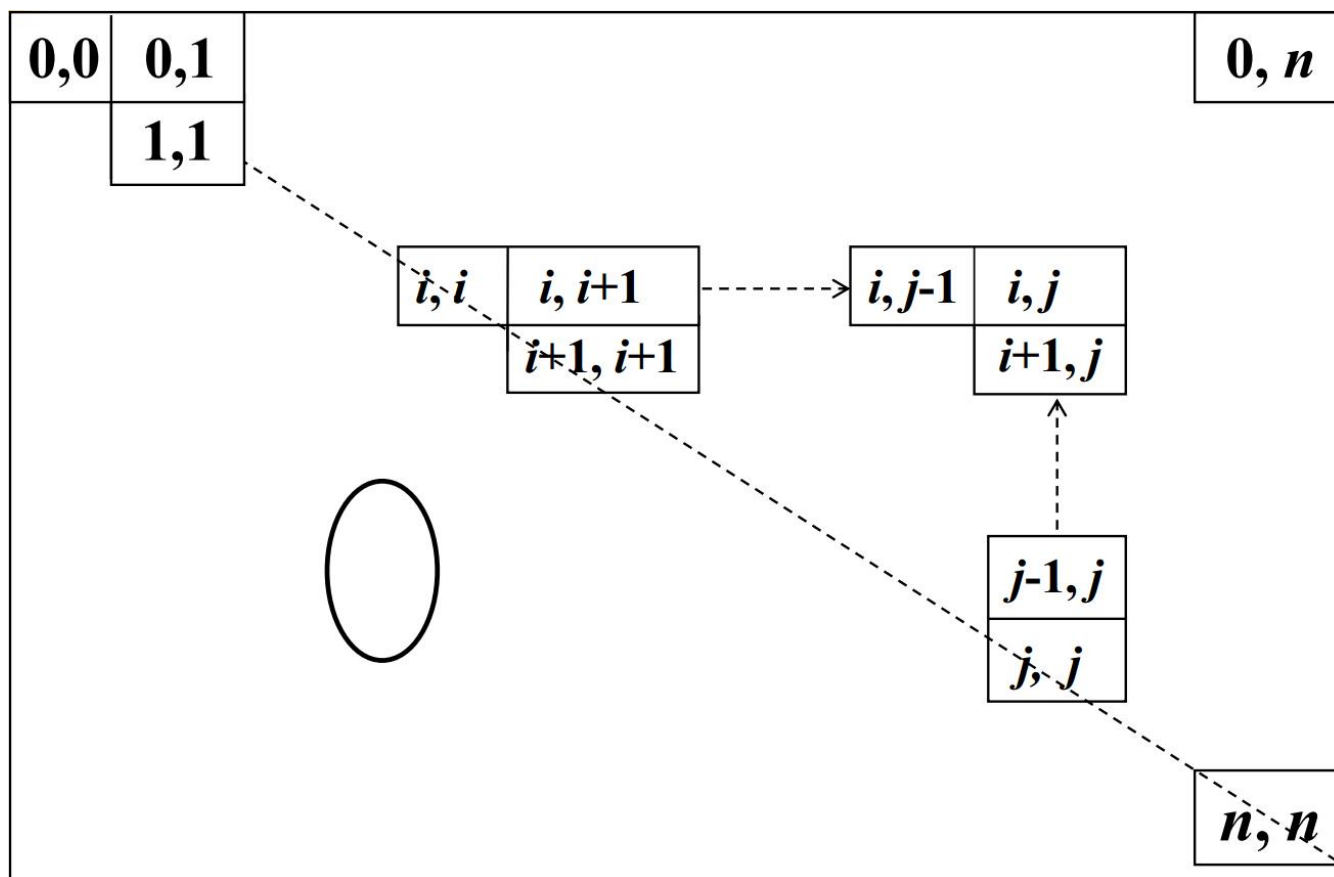
➤ 识别矩阵的构成:

- 方阵对角线以下全部为 0
- 主对角线以上的元素由文法 G 的非终结符构成
- 主对角线上的元素由输入句子的终结符号(单词)构成

9.4 CYK分析算法

◆ Cocke-Younger-Kasami (CYK) 算法

➤ 识别矩阵的构成:



9.4 CYK分析算法

◆ 识别矩阵构造步骤

- ① 首先构造主对角线, 令 $t_{0,0} = 0$, 然后, 从 $t_{1,1}$ 到 $t_{n,n}$ 在主对角线的位置上依次放入输入句子 x 的单词 w_i 。
- ② 构造主对角线以上紧靠主对角线的元素 $t_{i,i+1}$, 其中, $i = 0, 1, 2, \dots, n - 1$ 。对于输入句子 $x = w_1 w_2 \cdots w_n$, 从 w_1 开始分析。

9.4 CYK分析算法

◆ 识别矩阵构造步骤

如果在文法 G 的产生式集中有一条规则：

$$A \rightarrow w_1$$

则 $t_{0,1} = A$ 。

依此类推，如果有 $A \rightarrow w_{i+1}$ ，则 $t_{i,i+1} = A$ 。

即，对于主对角线上的每一个终结符 w_i ，所有可能推导出它的非终结符写在它的右边主对角线上方的位置上。

9.4 CYK分析算法

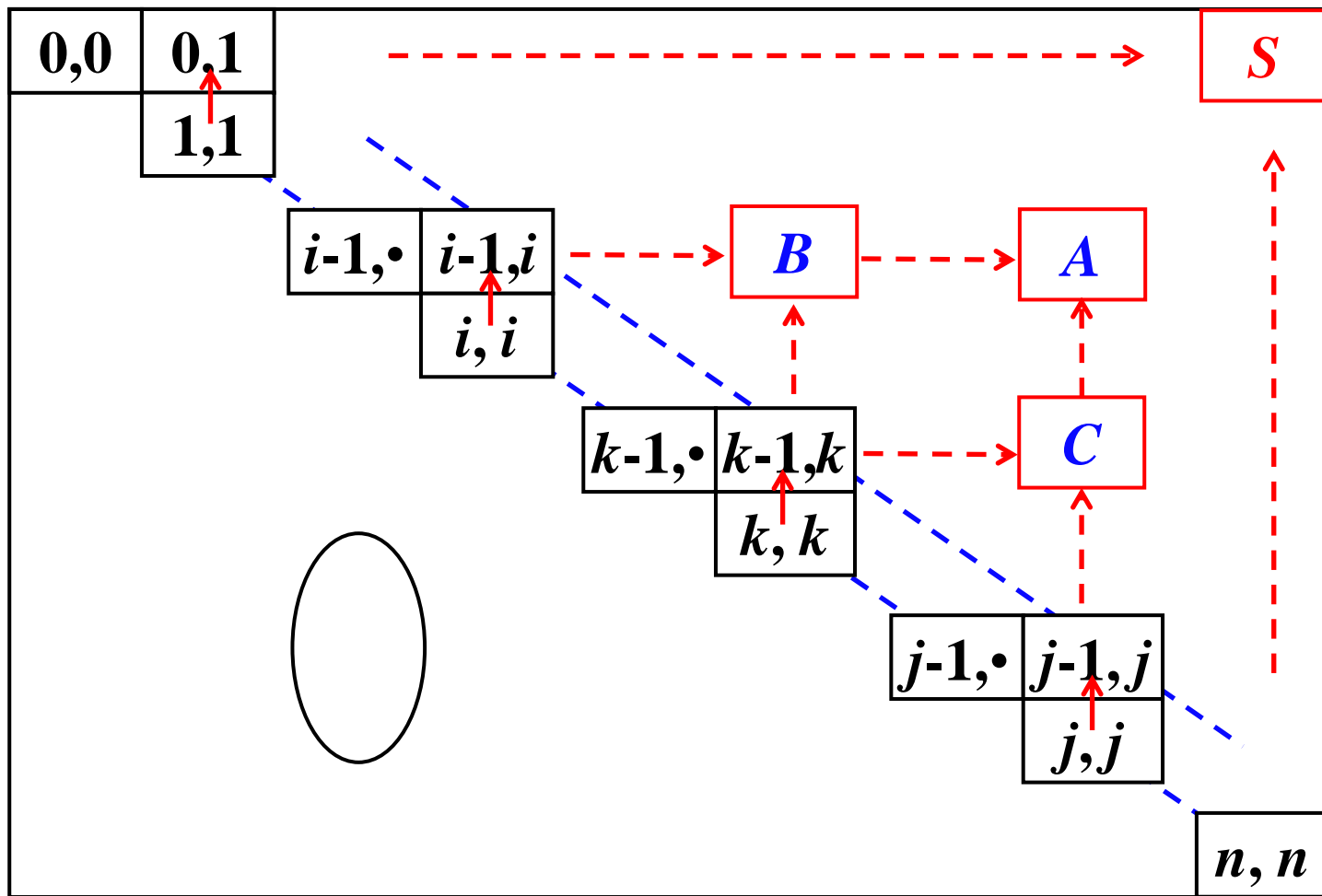
◆ 识别矩阵构造步骤

- ③ 按平行于主对角线的方向，一层一层地向上填写矩阵的各个元素 $t_{i,j}$ ，其中， $i = 0, 1, \dots, n - d, j = d + i, d = 2, 3, \dots, n$ 。如果存在一个正整数 $k, i + 1 \leq k \leq j - 1$ ，在文法 G 的规则集中有产生式 $A \rightarrow BC$ ，并且， $B \in t_{i,k}, C \in t_{k,j}$ ，那么，将 A 写到矩阵 $t_{i,j}$ 位置上。

判断句子 x 由文法 G 所产生的充要条件是: $t_{0,n} = S$ 。

9.4 CYK分析算法

◆ 识别矩阵构造步骤



9.4 CYK分析算法

◆ 例子

给定文法 $G(S)$:

(1) $S \rightarrow P VP$ (2) $VP \rightarrow V V$

(3) $VP \rightarrow VP N$ (4) $P \rightarrow \text{他}$

(5) $V \rightarrow \text{喜欢}$ (6) $V \rightarrow \text{读}$

(7) $N \rightarrow \text{书}$

请用 CYK 算法分析句子: 他喜欢读书

9.4 CYK分析算法

◆ 例子

(1) 汉语分词和词性标注以后：

他/P 喜欢/V 读/V 书/N $n = 4$

(2) 构造识别矩阵：

(3) 执行分析过程。

(1) $S \rightarrow P VP$

(2) $VP \rightarrow V V$

(3) $VP \rightarrow VP N$

	0	1	2	3	4
0	0	P			
1		他	V → VP		
2			喜欢	V	
3				读	N
4					书

9.4 CYK分析算法

◆ 例子

(1) 汉语分词和词性标注以后：

他/P 喜欢/V 读/V 书/N $n = 4$

(2) 构造识别矩阵：

(3) 执行分析过程。

- (1) $S \rightarrow P VP$
- (2) $VP \rightarrow V V$
- (3) $VP \rightarrow VP N$

	0	1	2	3	4
0	0	P	P	S	
1		他	V	VP	?
2			喜欢	V	N
3				读	N
4					书

9.4 CYK分析算法

◆ 例子

(1) 汉语分词和词性标注以后：

他/P 喜欢/V 读/V 书/N $n = 4$

(2) 构造识别矩阵：

(3) 执行分析过程。

(1) $S \rightarrow P VP$

(2) $VP \rightarrow V V$

(3) $VP \rightarrow VP N$

	0	1	2	3	4
0	0	P → P			
1		他	V → VP → VP		
2			喜欢	V	N
3				读	N
4					书

9.4 CYK分析算法

◆ 例子

(1) 汉语分词和词性标注以后：

他/P 喜欢/V 读/V 书/N $n = 4$

(2) 构造识别矩阵：

(3) 执行分析过程。

- (1) $S \rightarrow P VP$

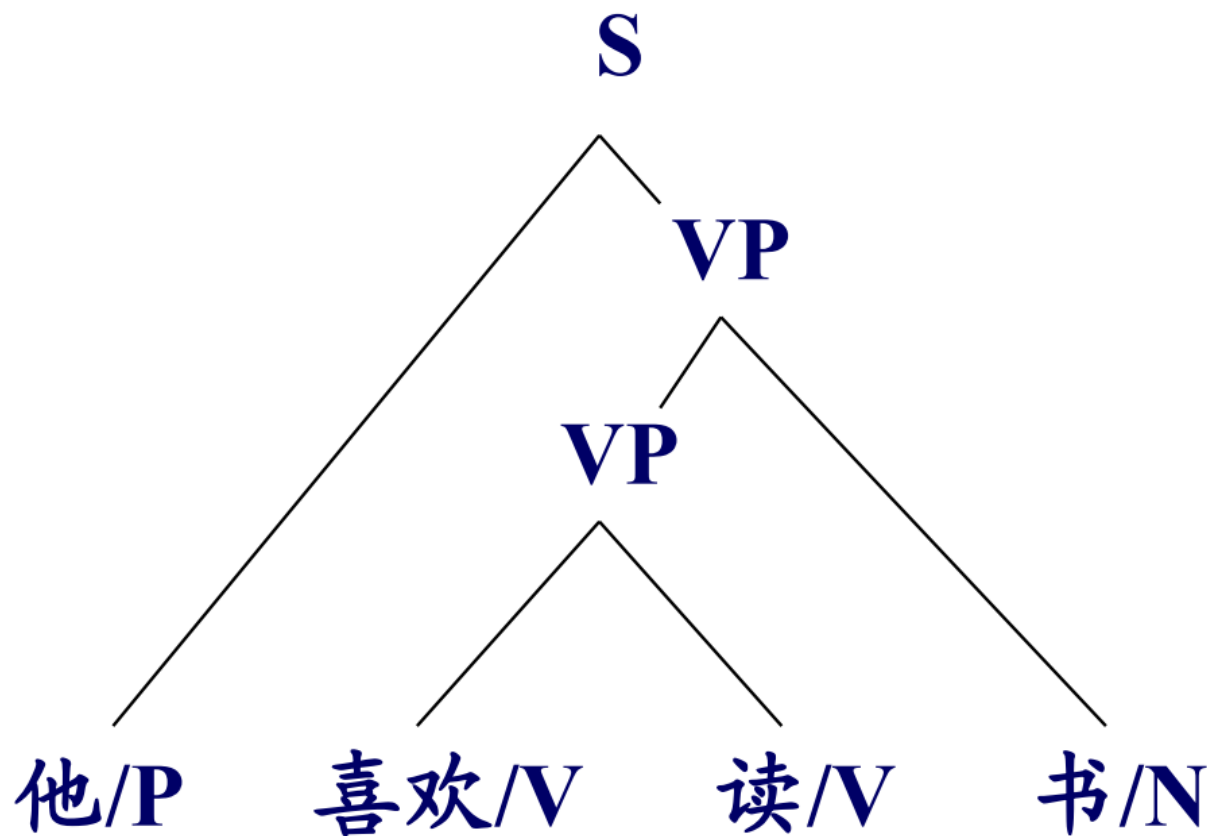
(2) $VP \rightarrow V V$

(3) $VP \rightarrow VP N$

	0	1	2	3	4
0	0	P →	P →	P →	S
1		他	V →	VP →	VP
2			喜欢	V →	N
3				读	N
4					书

9.4 CYK分析算法

◆ 例子



9.4 CYK分析算法

◆ CYK算法的评价

➤ 优点

- 简单易行，执行效率高

➤ 弱点

- 必须对文法进行范式化处理
- 无法区分歧义

9.5 概率上下文无关文法

9.5 概率上下文无关文法

◆ 概率上下文无关文法 (Probabilistic Context-Free Grammar)

由于语法的解析存在二义性，我们就需要找到一种方法从多种可能的语法树中找出最可能的一棵树。

一种常见的方法既是概率上下文无关文法 (Probabilistic Context-Free Grammar, PCFG)，通过将概率与语法中的每个规则相关联，来扩展上下文无关语法。

9.5 概率上下文无关文法

◆ PCFG 规则

形式: $A \rightarrow \alpha, p$

约束: $\sum_{\alpha} p(A \rightarrow \alpha) = 1$

例如:

$$\left. \begin{array}{l} NP \rightarrow NN \ NN, 0.60 \\ NP \rightarrow NN \ CC \ NN, 0.40 \end{array} \right\} \sum p = 1$$

$$\left. \begin{array}{l} CD \rightarrow QP, 0.99 \\ CD \rightarrow LST, 0.01 \end{array} \right\} \sum p = 1$$

9.5 概率上下文无关文法

◆ 例-1:

$S \rightarrow NP VP, 1.00$

$NP \rightarrow NP PP, 0.40$

$NP \rightarrow \text{astronomers}, 0.10$

$NP \rightarrow \text{ears}, 0.18$

$NP \rightarrow \text{saw}, 0.04$

$NP \rightarrow \text{stars}, 0.18$

$NP \rightarrow \text{telescopes}, 0.1$

$PP \rightarrow P NP, 1.00$

$P \rightarrow \text{with}, 1.00$

$VP \rightarrow V NP, 0.70$

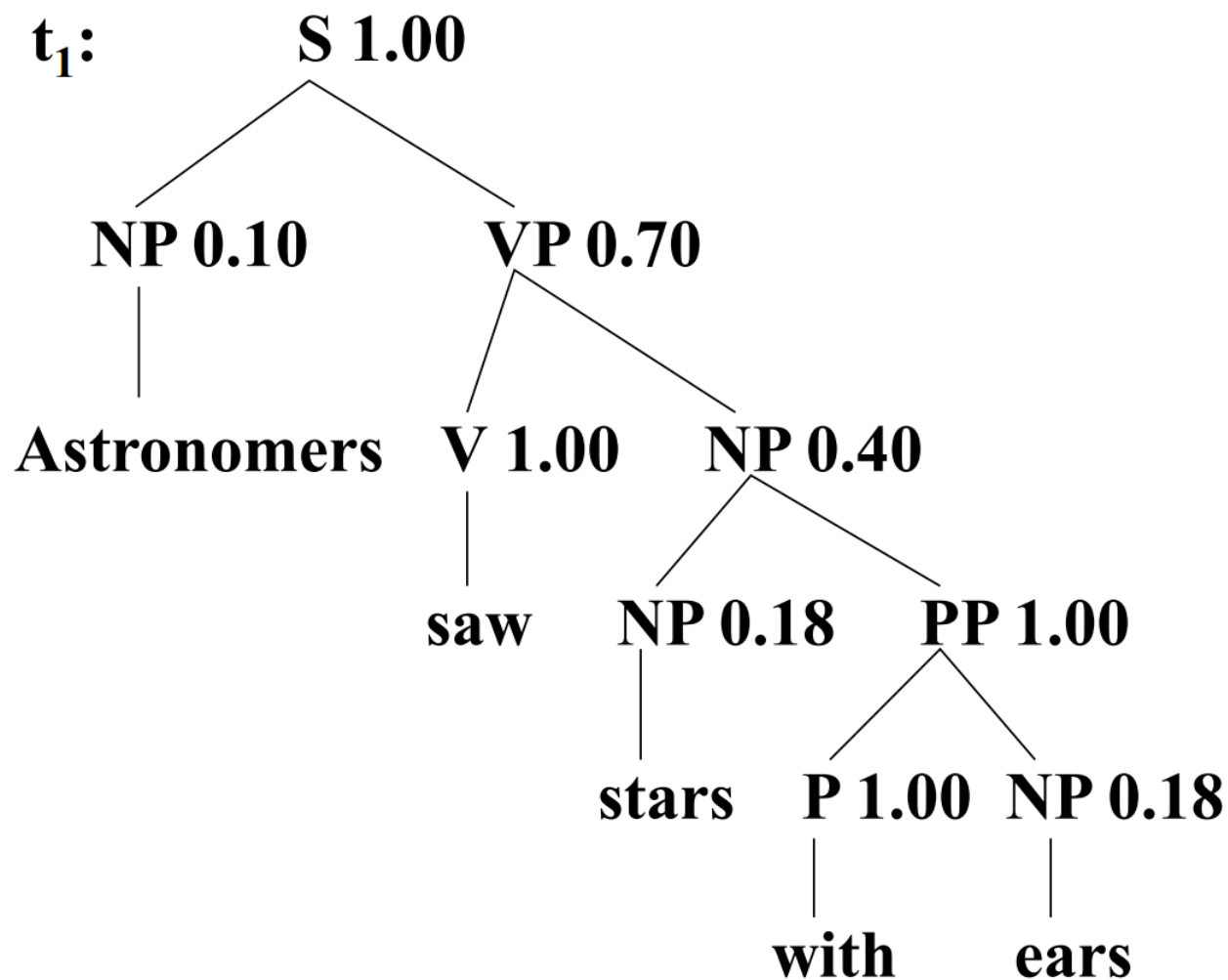
$VP \rightarrow VP PP, 0.30$

$V \rightarrow \text{saw}, 1.00$

给定句子 S: ***Astronomers saw stars with ears.***

9.5 概率上下文无关文法

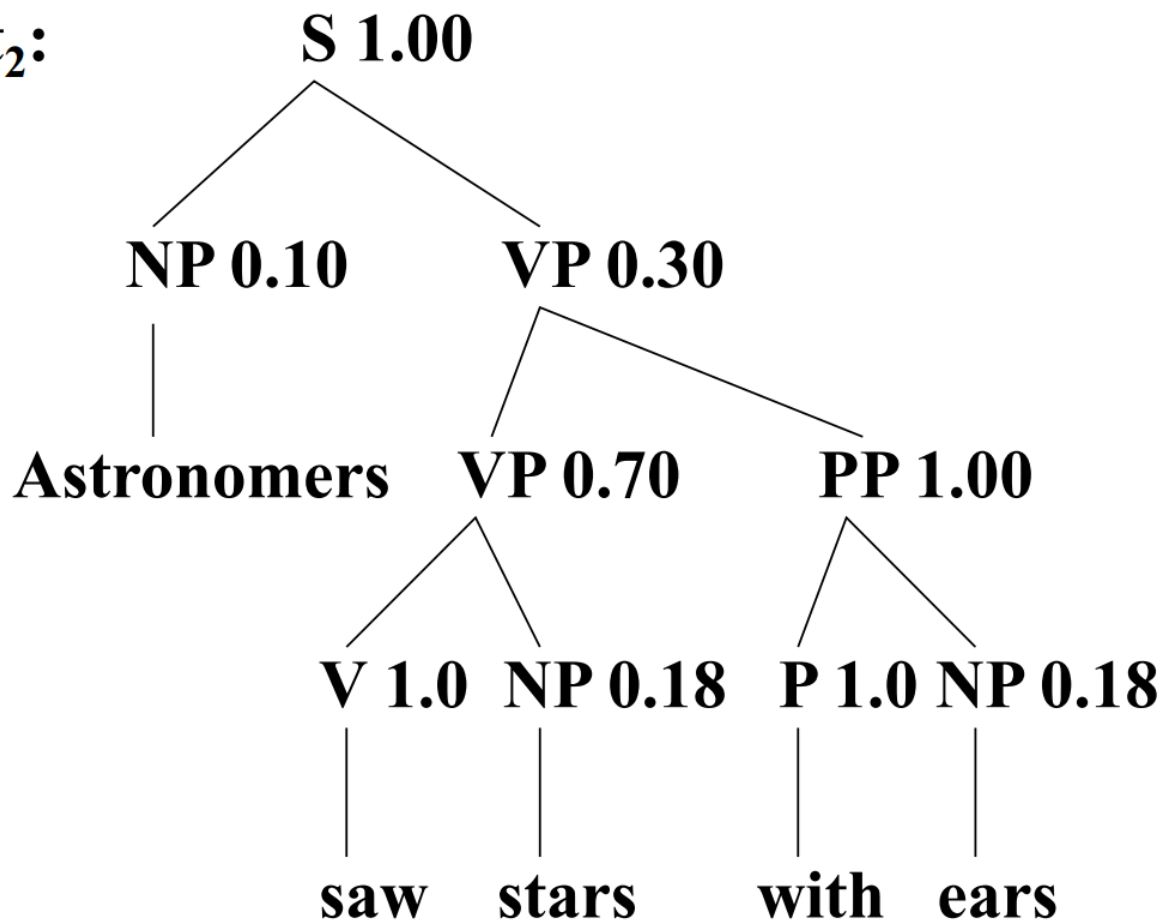
◆ 例-1:



9.5 概率上下文无关文法

◆ 例-1:

t_2 :



9.5 概率上下文无关文法

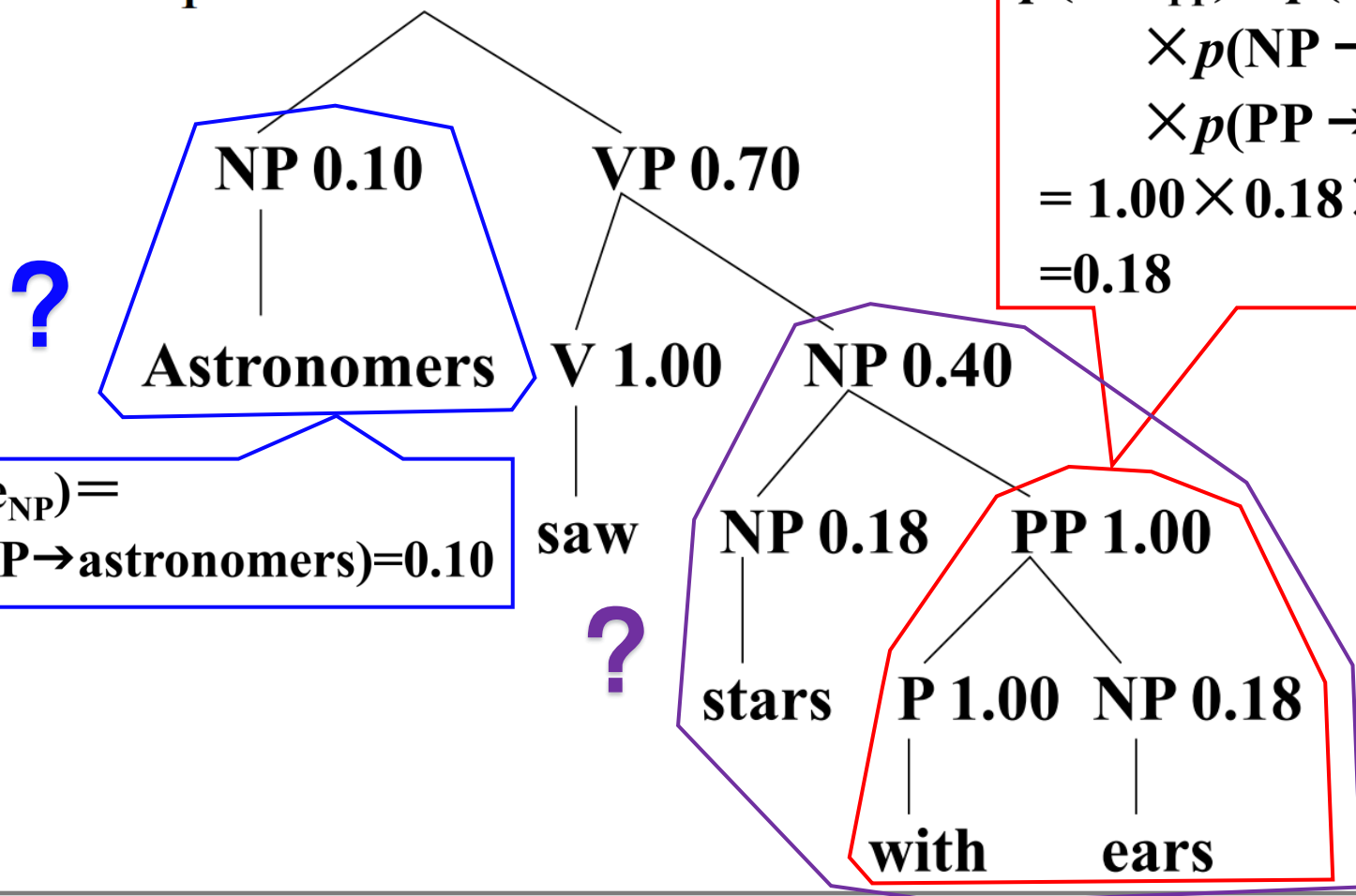
◆ 计算分析树概率的基本假设

- **位置不变性**：子树的概率与其管辖的词在整个句子中所处的位置无关，即对于任意的 $k, p(A_{k(k+c)} \rightarrow w)$ 一样。
- **上下文无关性**：子树的概率与子树管辖范围以外的词无关，即 $p(A_{kl} \rightarrow w | \text{任何超出 } k \sim l \text{ 范围的上下文}) = p(A_{kl} \rightarrow w)$ 。
- **祖先无关性**：子树的概率与推导出该子树的祖先结点无关，即 $p(A_{kl} \rightarrow w | \text{任何除 } A \text{ 以外的祖先结点}) = p(A_{kl} \rightarrow w)$ 。

9.5 概率上下文无关文法

◆ 计算分析树概率

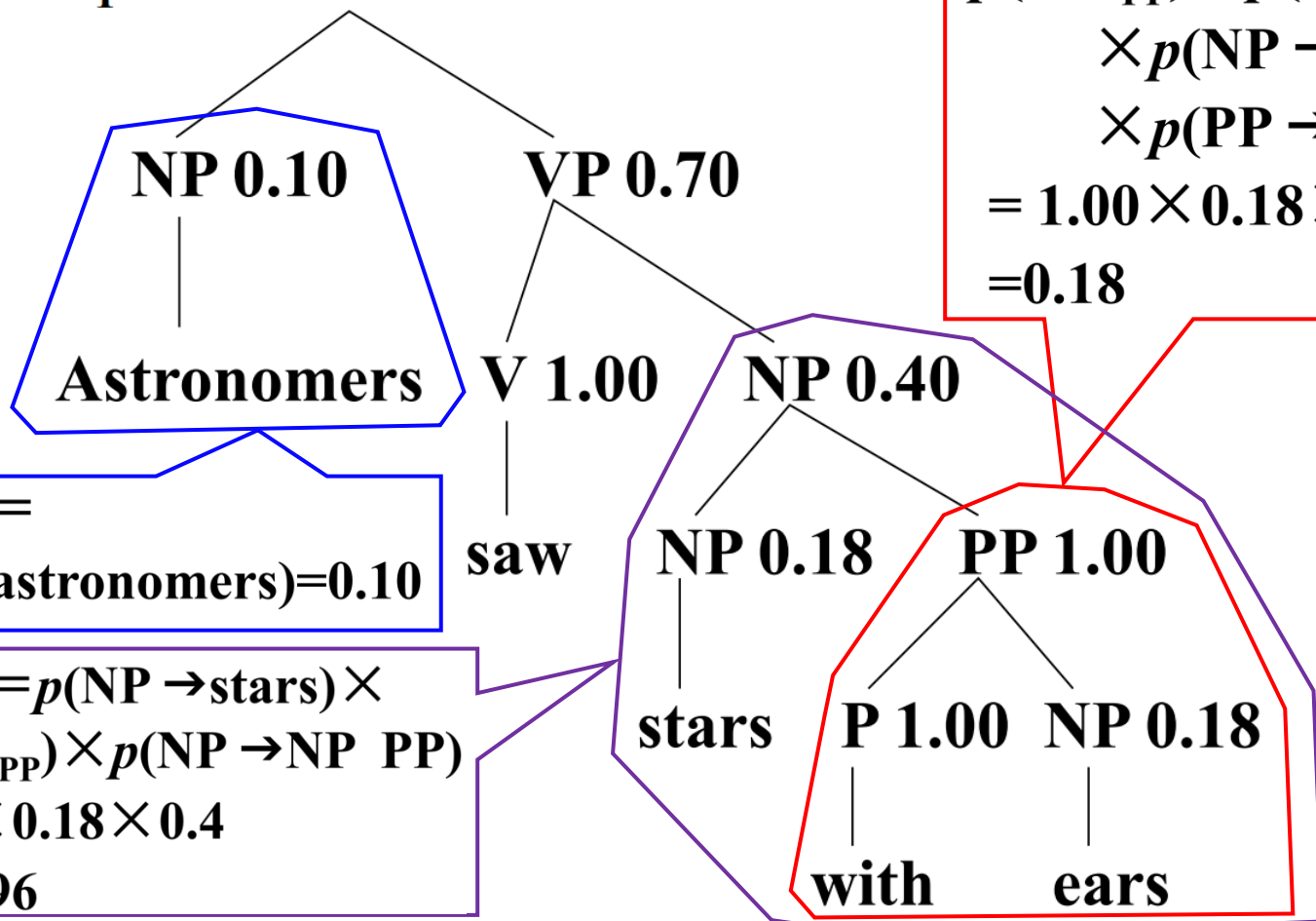
t_1 : S 1.00



9.5 概率上下文无关文法

◆ 计算分析树概率

t_1 : S 1.00



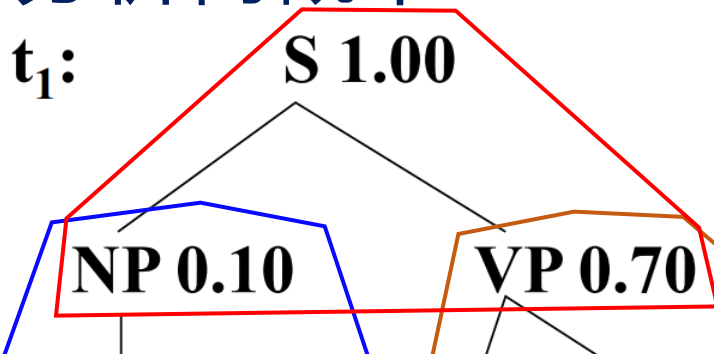
$$\begin{aligned}
 p(\text{tree}_{\text{PP}}) &= p(P \rightarrow \text{with}) \\
 &\quad \times p(\text{NP} \rightarrow \text{ears}) \\
 &\quad \times p(\text{PP} \rightarrow P \text{ NP}) \\
 &= 1.00 \times 0.18 \times 1.00 \\
 &= 0.18
 \end{aligned}$$

$$\begin{aligned}
 p(\text{tree}_{\text{NP}}) &= \\
 &p(\text{NP} \rightarrow \text{astronomers}) = 0.10
 \end{aligned}$$

$$\begin{aligned}
 p(\text{tree}_{\text{NP}}) &= p(\text{NP} \rightarrow \text{stars}) \times \\
 &\quad p(\text{tree}_{\text{PP}}) \times p(\text{NP} \rightarrow \text{NP PP}) \\
 &= 0.18 \times 0.18 \times 0.4 \\
 &= 0.01296
 \end{aligned}$$

9.5 概率上下文无关文法

◆ 计算分析树概率



$$p(\text{tree}_{\text{VP}}) = p(V \rightarrow \text{saw}) \times p(\text{tree}_{\text{NP}}) \times p(\text{VP} \rightarrow V \text{ NP})$$

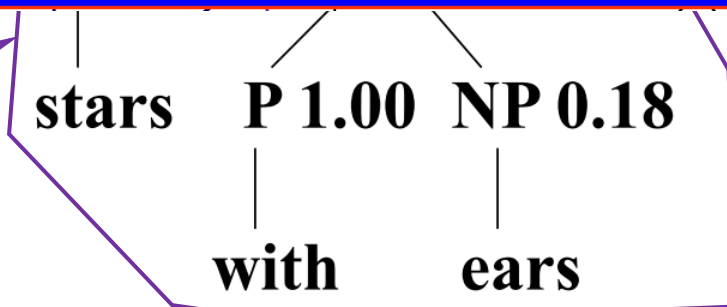
$$= 1.0 \times 0.01296 \times 0.70$$

$$= 0.009072$$

$$\begin{aligned} p(t_1) &= p(\text{tree}_{\text{NP}}) \times p(\text{VP}) \times p(\text{S} \rightarrow \text{NP VP}) \\ &= 0.10 \times 1.0 \times 0.009072 \\ &= 0.0009072 \end{aligned}$$

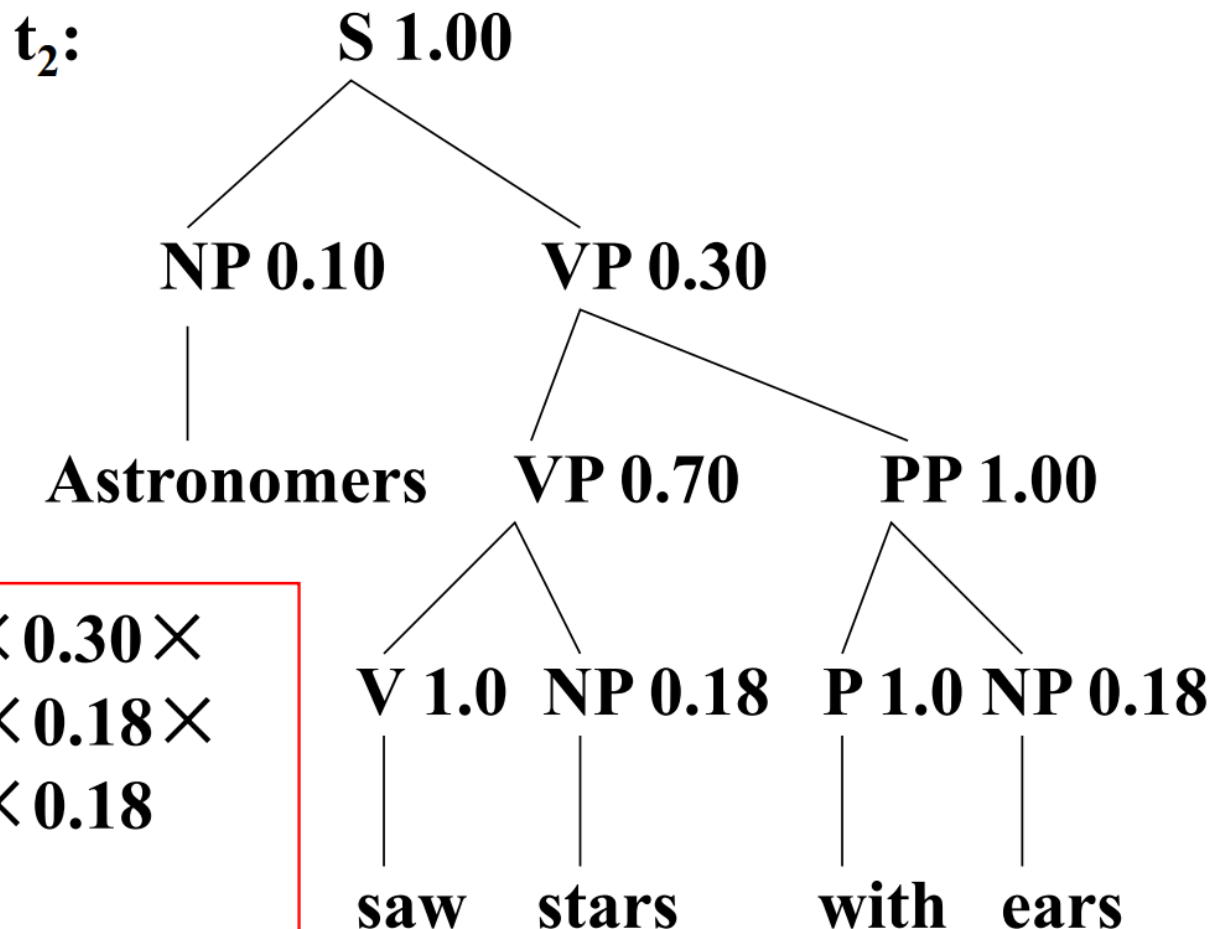
$p(\text{tree}_T)$
 $p(\text{NE})$

$$\begin{aligned} p(\text{tree}_{\text{NP}}) &= p(\text{NP} \rightarrow \text{stars}) \times \\ &= p(\text{tree}_{\text{PP}}) \times p(\text{NP} \rightarrow \text{NP PP}) \\ &= 0.18 \times 0.18 \times 0.4 \\ &= 0.01296 \end{aligned}$$



9.5 概率上下文无关文法

◆ 计算分析树概率



$$\begin{aligned} p(t_2) &= 1.00 \times 0.10 \times 0.30 \times \\ &\quad 0.70 \times 1.00 \times 0.18 \times \\ &\quad 1.00 \times 1.00 \times 0.18 \\ &= 0.0006804 \end{aligned}$$

9.5 概率上下文无关文法

对于给定的句子 S ，两棵句法分析树的概率不等， $P(t_1) > P(t_2)$ ，因此，可以得出结论：分析结果 t_1 正确的可能性大于 t_2 。

9.5 概率上下文无关文法

◆ PCFG的三个问题

1. 给定句子 $W = w_1 w_2 \cdots w_n$ 和 PCFG G , 如何快速计算 $p(W|G)$?
2. 给定句子 $W = w_1 w_2 \cdots w_n$ 和 PCFG G , 如何快速的选择最佳句法结构树?
3. 给定句子 $W = w_1 w_2 \cdots w_n$ 和 PCFG G , 如何调节 G 的参数, 使得 $p(W|G)$ 最大?

9.5 概率上下文无关文法

◆ PCFG算法的评价

➤ 优点

- 可以利用概率减少分析过程的搜索空间
- 可以利用概率对概率较小的子树剪枝，加快分析效率
- 可以定量的分析两个语法的性能

➤ 弱点

- 分析树的概率计算条件非常苛刻，甚至不够合理

9.6 依存句法分析

9.6 依存句法分析

◆ 依存句法理论

现代依存语法(dependency grammar)理论的创立者是法国语言学家**吕西安·泰尼埃** (其姓氏也被译作：特思尼耶尔、特尼耶尔等)(Lucien Tesnière, 1893-1954)。他的主要思想反映在1953年出版的专著《结构句法概要》(Esquisse d'une syntaxe structurale)中。

9.7 依存句法分析

◆ 依存句法理论

➤ L. Tesnière 的理论认为：

一切结构句法现象可以概括为关联(connexion)、组合(jonction)和转位(tanslation)这三大核心。句法关联建立起词与词之间的从属关系，这种从属关系是由支配词和从属词联结而成；**动词是句子的中心，并支配其他成分**，它本身不受其他任何成分的支配。

欧洲传统的语言学突出一个句子中主语的地位，句中其它成分称为“谓语”。依存语法打破了这种主谓关系，认为“谓语”中的动词是一个句子的中心，其他成分与动词直接或间接地产生联系。

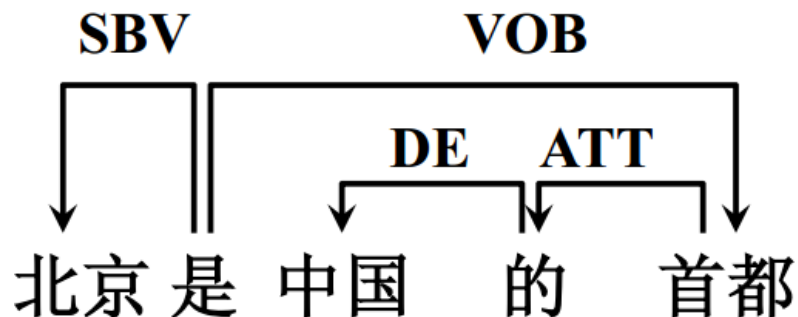
9.7 依存句法分析

◆ 依存句法理论

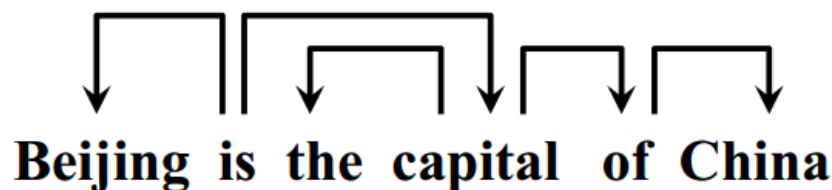
在依存语法理论中，“依存”就是指词与词之间支配与被支配的关系，这种关系不是对等的，而是有方向的。处于支配地位的成分称为支配者(governor, regent, head)，而处于被支配地位的成分称为从属者(modifier, subordinate, dependency)。

9.7 依存句法分析

◆ 依存句法理论



(e) 有向图-1

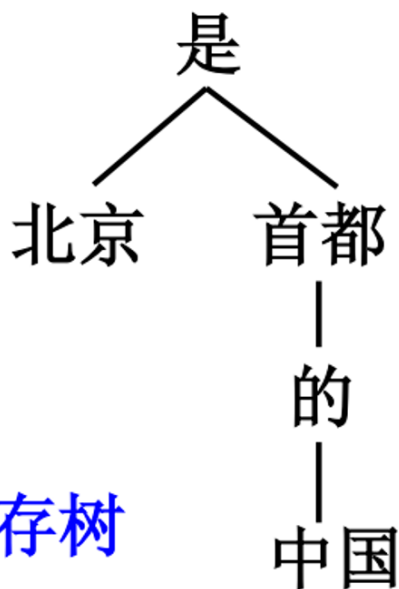


(e) 有向图-2

两个有向图用带有方向的弧(或称边, edge)来表示两个成分之间的依存关系, 支配者在有向弧的发出端, 被支配者在箭头端, 我们通常说被支配者依存于支配者。

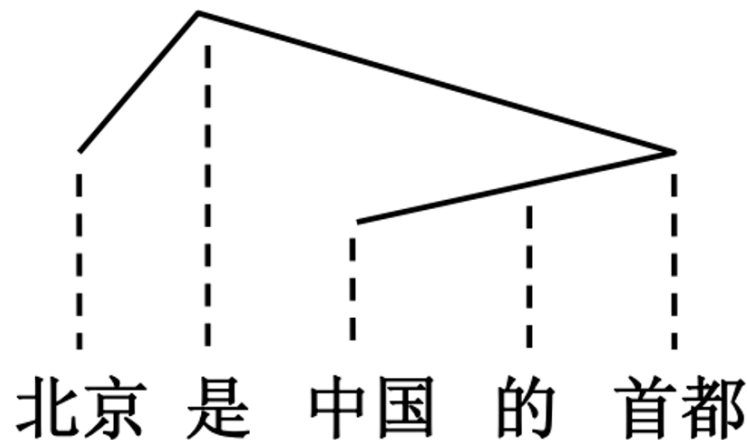
9.7 依存句法分析

◆ 依存句法理论



(f) 依存树

图(f)是用树表示的依存结构，树中子节点依存于该节点的父节点。



(g) 依存投射树

图(g)是带有投射线的树结构，实线表示依存联结关系，位置低的成份依存于位置高的成份，虚线为投射线

9.7 依存句法分析

◆ 依存句法理论

1970年计算语言学家 J. Robinson 在论文《依存结构和转换规则》中提出了依存语法的4条公理：

- (1) 一个句子只有一个独立的成分；
- (2) 句子的其他成分都从属于某一成分；
- (3) 任何一成分都不能依存于两个或多个成分；
- (4) 如果成分A直接从属于成分B，而成分C在句子中位于A和B之间，那么，成分C或者从属于A，或者从属于B，或者从属于A和B之间的某一成分。

9.7 依存句法分析

◆ 依存句法理论

这4条公理相当于对依存图和依存树的形式约束为：

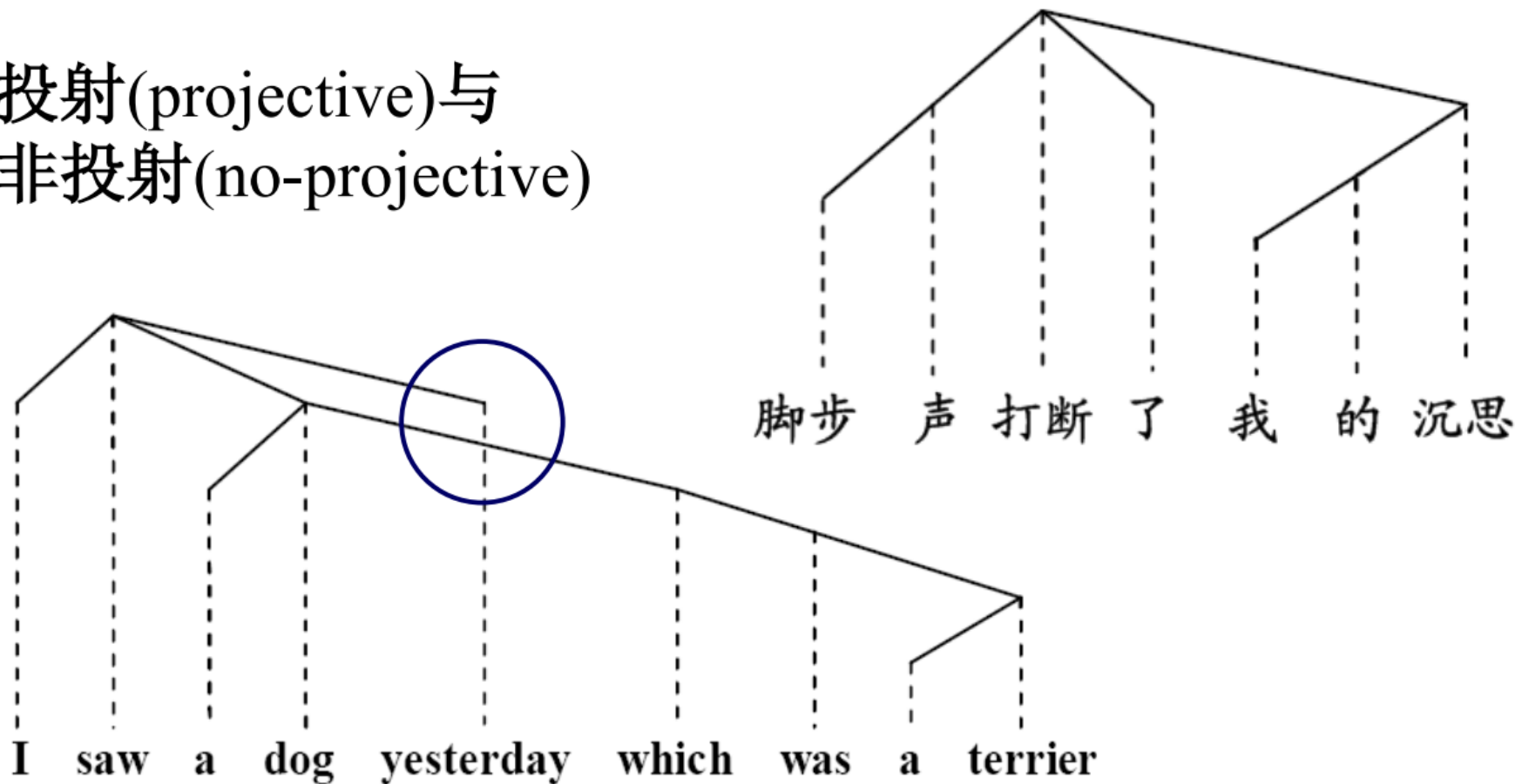
- 单一父结点(single headed)
- 连通(connective)
- 无环(acyclic)
- 可投射(projective)

由此来保证句子的依存分析结果是一棵有 “根(root) ” 的树结构。

9.7 依存句法分析

◆ 依存句法理论

投射(projective)与
非投射(no-projective)



9.7 依存句法分析

◆ 依存语法的优势

- 简单，直接按照词语之间的依存关系工作，是天然词汇化的；
- 不过多强调句子中的固定词序，对自由语序的语言分析更有优势；
- 受深层语义结构的驱动，词汇的依存本质是语义的；
- 形式化程度较短语结构语法浅，对句法结构的表述更为灵活。

9.7 依存句法分析

◆ 依存句法分析方法

依存句法结构描述一般采用有向图方法或依存树方法，所采用的句法分析算法可大致归为以下4类：

- 生成式的分析方法(generative parsing)
- 判别式的分析方法(discriminative parsing)
- 决策式的(确定性的)分析方法(deterministic parsing)
- 基于约束满足的分析方法(constraint satisfaction parsing)

9.8 短语结构与依存结构 的关系

9.8 短语结构与依存结构的关系

◆ 短语结构可转换为依存结构

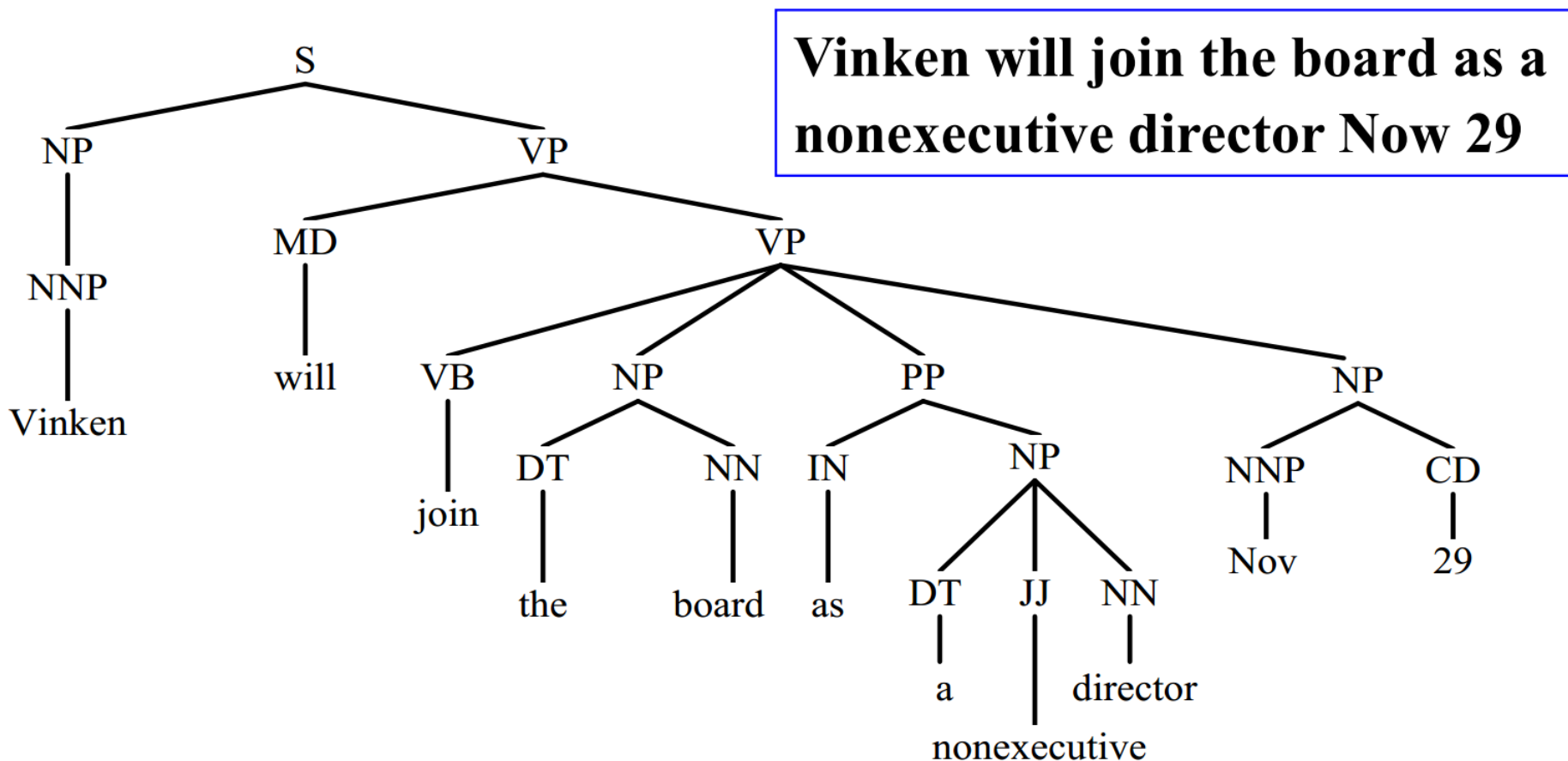
➤ 实现方法:

- (1) 定义中心词抽取规则，产生中心词表;
- (2) 根据中心词表，为句法树中每个节点选择中心子节点;
- (3) 将非中心子节点的中心词依存到中心子节点的中心词上，得到相应的依存结构。

9.8 短语结构与依存结构的关系

◆ 短语结构可转换为依存结构

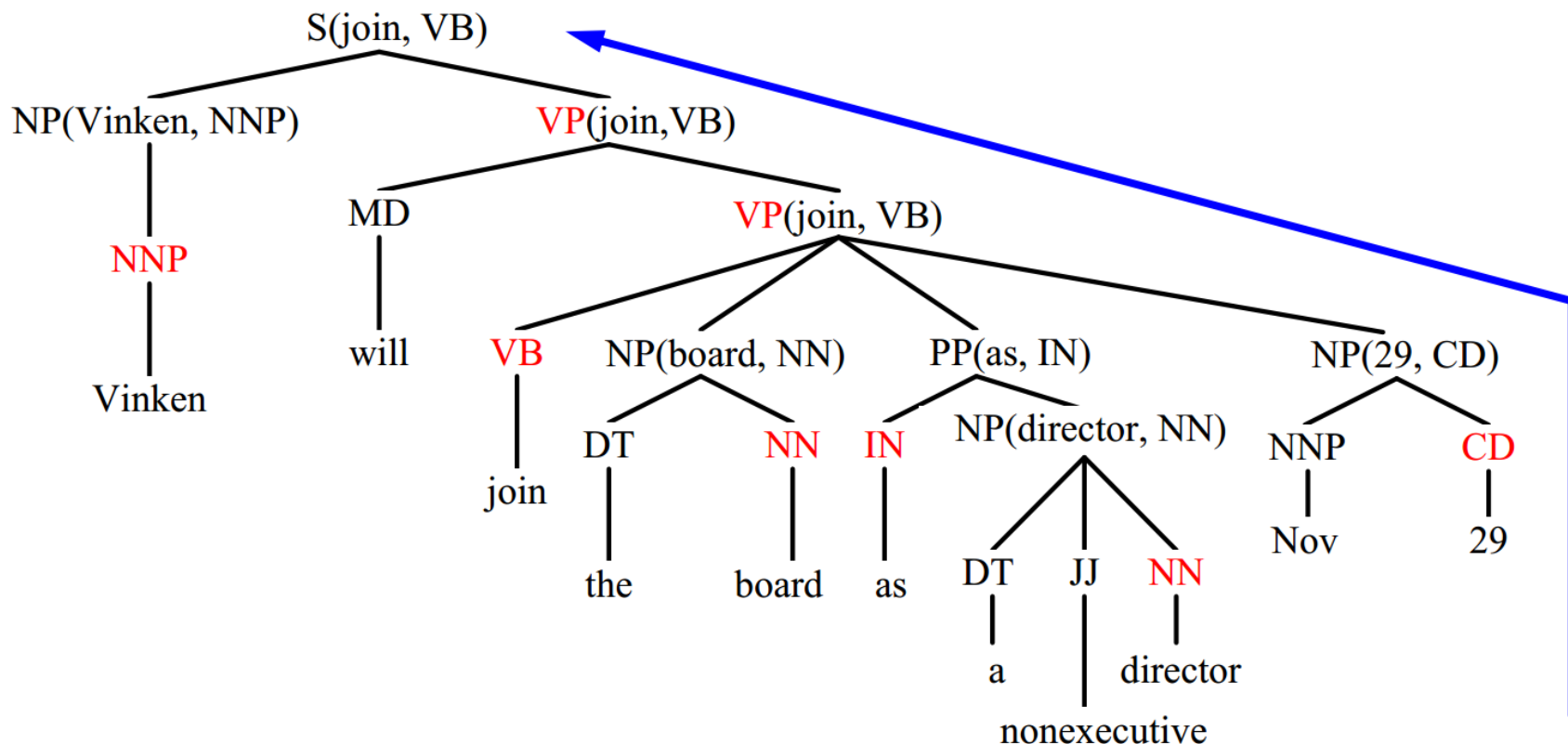
例如：给定如下短语结构树



9.8 短语结构与依存结构的关系

◆ 短语结构可转换为依存结构

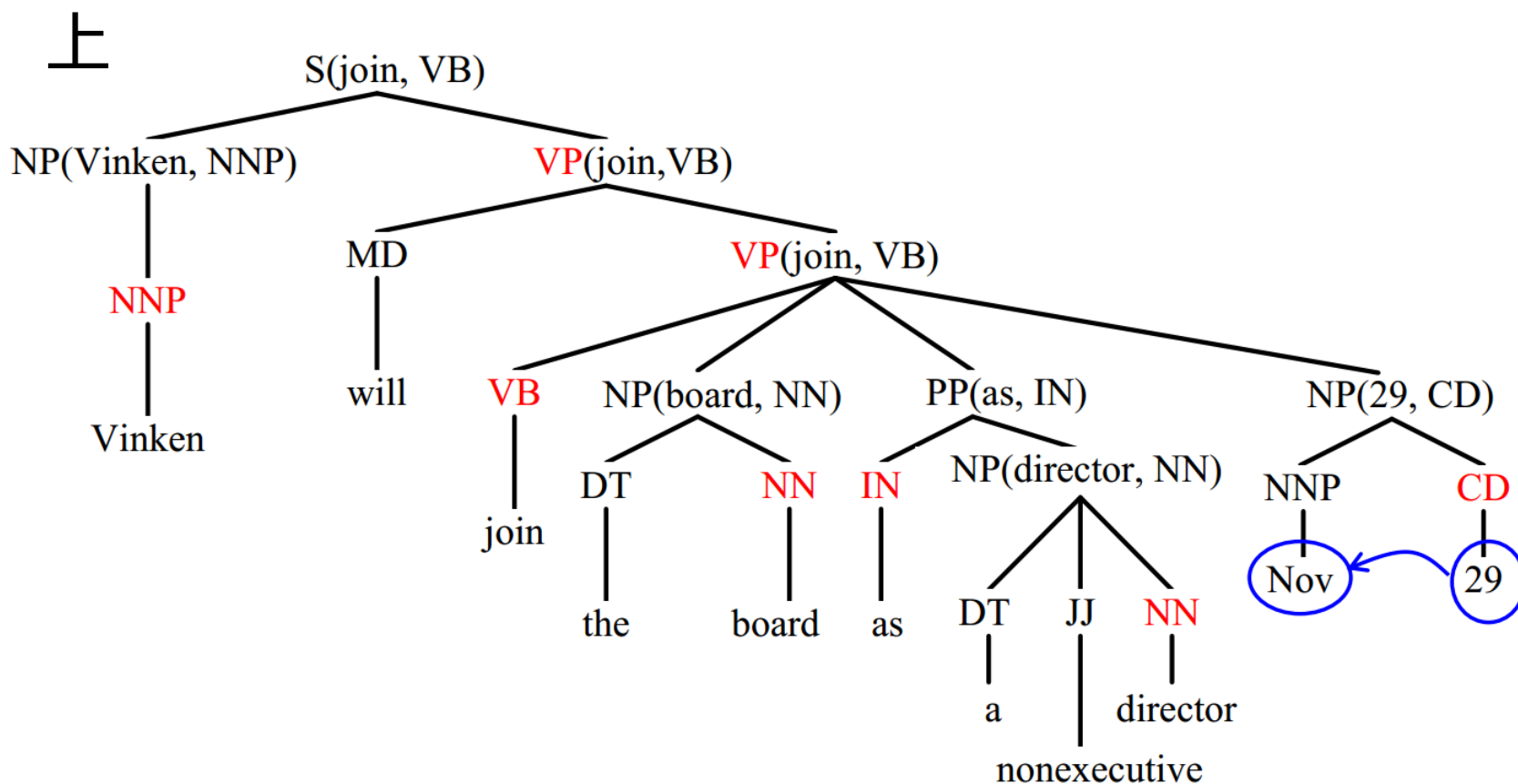
- 根据中心词表为每个节点选择中心子节点（中心词通过自底向上传递得到）



9.8 短语结构与依存结构的关系

◆ 短语结构可转换为依存结构

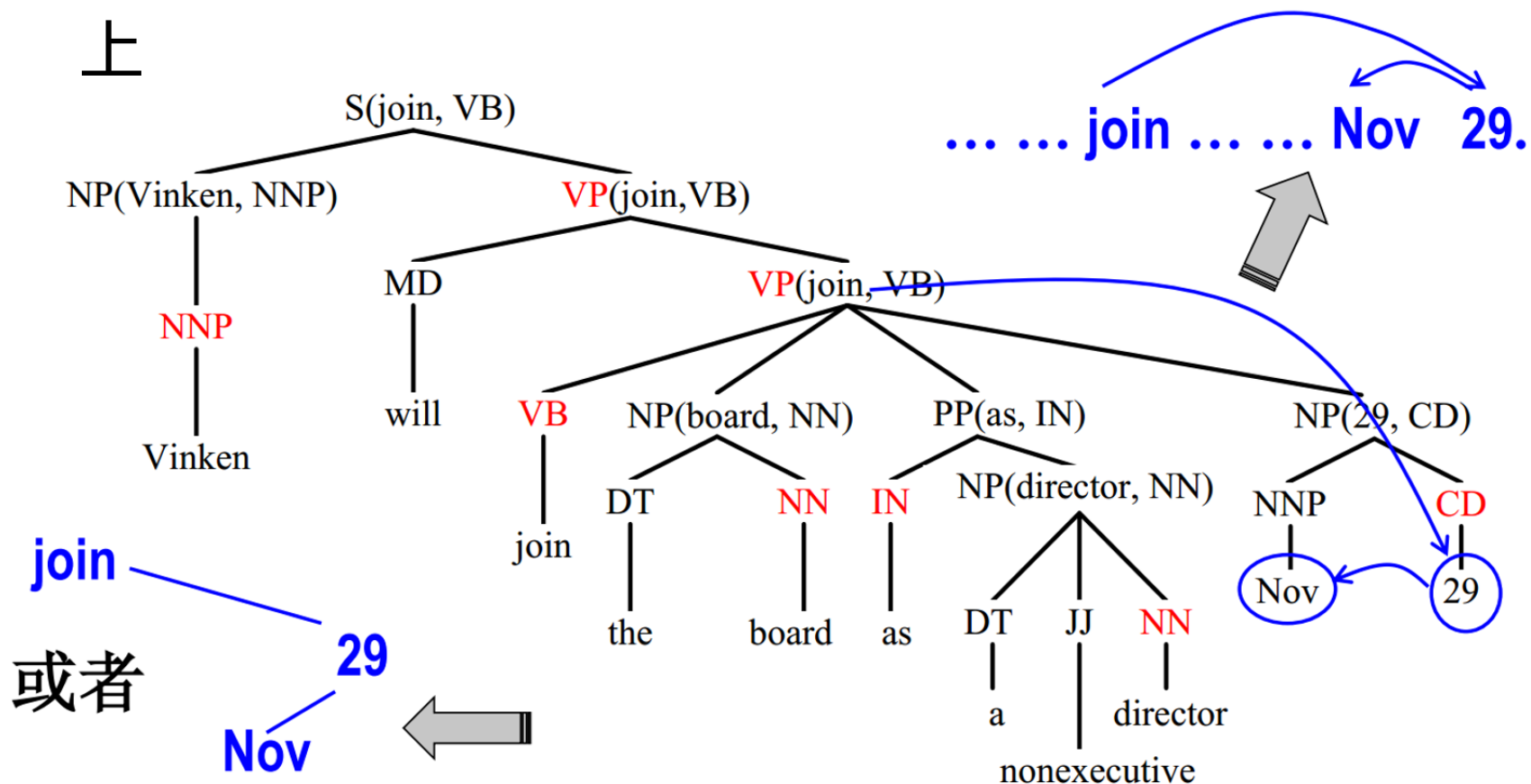
- 将非中心子节点的中心词依存到中心子节点的中心词上



9.8 短语结构与依存结构的关系

◆ 短语结构可转换为依存结构

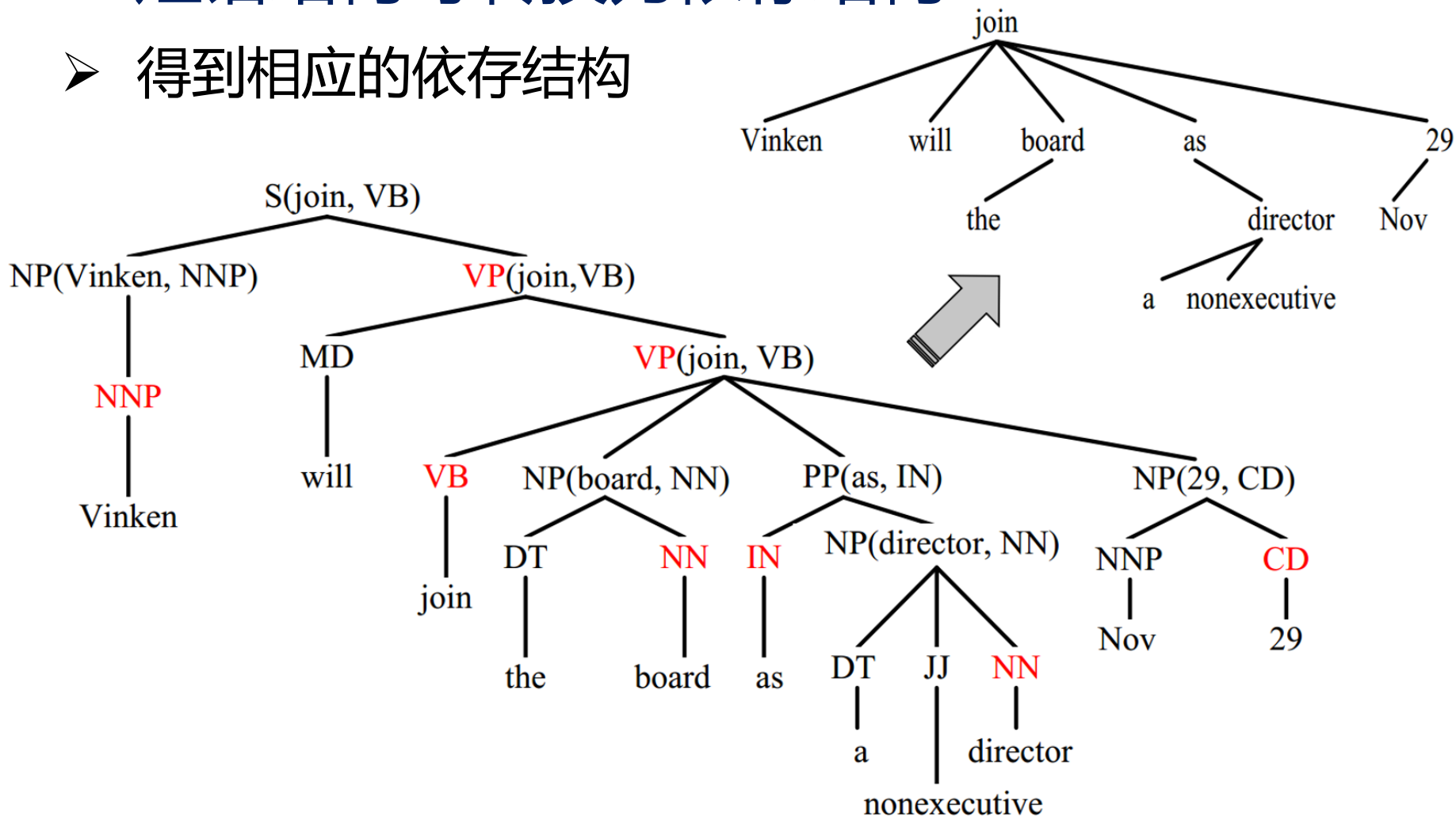
- 将非中心子节点的中心词依存到中心子节点的中心词上



9.8 短语结构与依存结构的关系

◆ 短语结构可转换为依存结构

➤ 得到相应的依存结构



本章小结

◆ 句法分析的任务，面临的困难

◆ 短语结构分析方法

➤ 基于规则的方法：

- Chart Parsing
- CYK 方法

➤ 基于概率上下文无关文法 PCFG

◆ 依存句法分析

➤ 基本方法

➤ 短语结构与依存结构

习题

- 9-1. 编写程序实现自顶向下(top-down)的 Chart 分析器，体会自顶向下和自底向上(bottom-up)分析算法的不同。
- 9-2. 自学Left Corner 分析算法和 Tomita GLR 句法分析算法。
- 9-3. 如有条件，利用树库语料抽取 PCFG 规则，结合Chart 分析算法实现一个基于 PCFG 的句法分析器。

谢谢!