

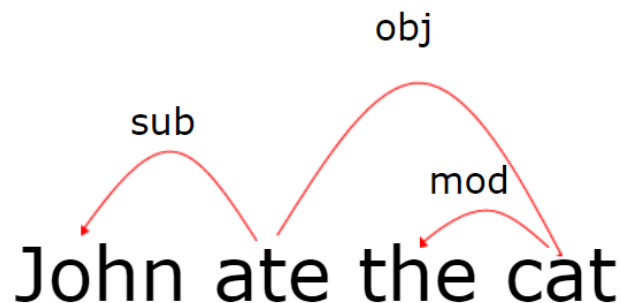
第8章 句法分析

(1/2)

句法分析任务

句法分析(syntactic parsing)任务的目标就是识别句子的**结构关系**。在自然语言处理中，通常有两种句法分析任务：

- 短语结构分析 (constituent phrase parsing)
- 依存句法分析 (dependency parsing)
 - 识别词汇间各种依赖关系





短语结构分析



8.1 概述

8.2 线图分析法

8.3 CYK分析法

8.4 基于PCFG的分析法

8.5 句法分析性能评估

8.6 局部句法分析



8.1 短语结构分析概述

◆回顾例句 (前面第4章)

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



8.1 短语结构分析概述

(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)))

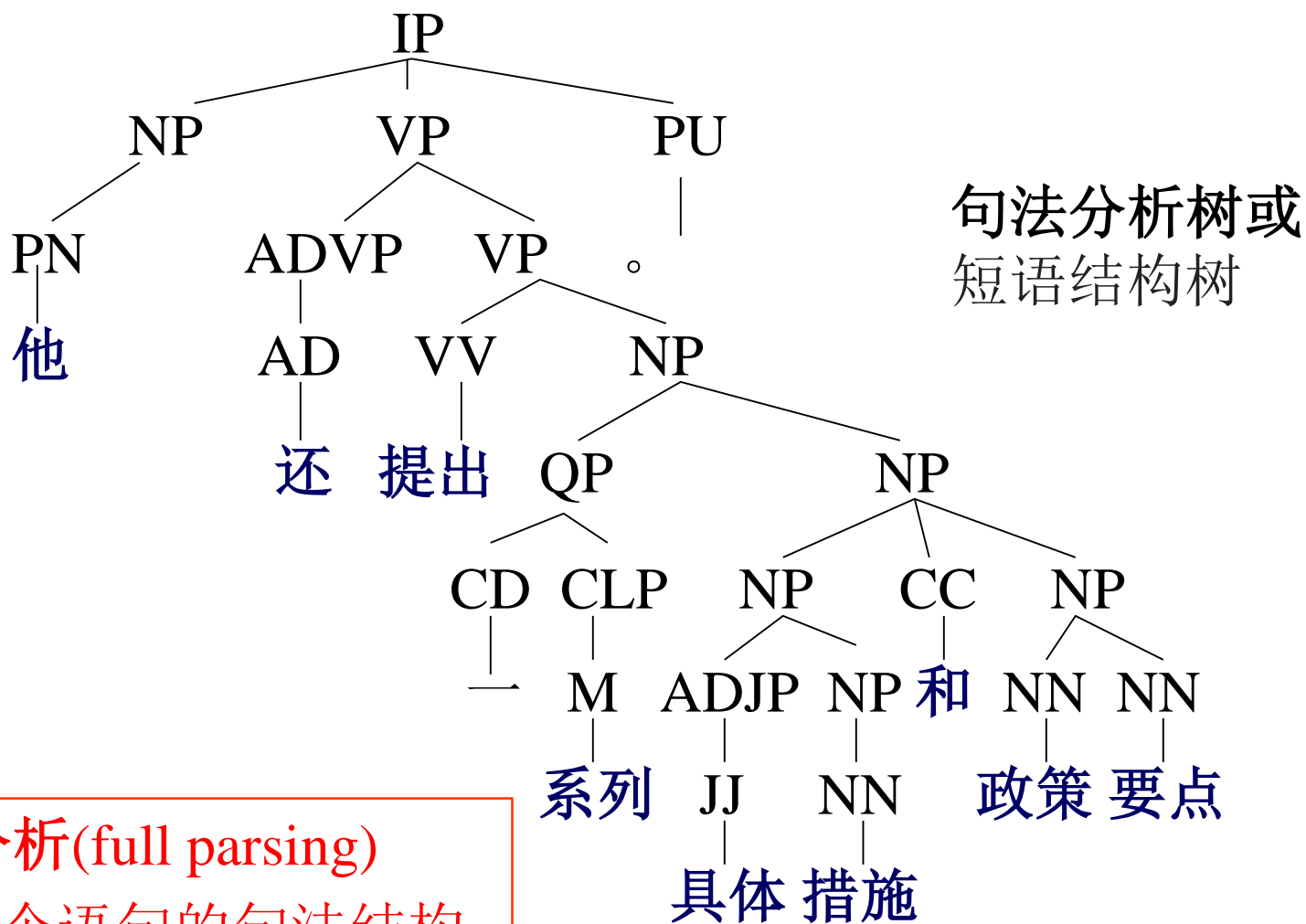
(CC 和)

(NP (NN 政策)

(NN 要点))))))

(PU 。))

8.1 短语结构分析概述



完全句法分析(full parsing)

即:获取整个语句的句法结构



8.1 短语结构分析概述

- ◆ 目标: 实现高准确率、高鲁棒性 (robustness)、快速的句子结构自动分析过程。
- ◆ 困难: 自然语言中存在大量的复杂的结构歧义 (structural ambiguity)。

例如: (1) I saw a boy with a telescope in the park

[I saw a boy] with a telescope

I saw a [boy with a telescope]

(2)关于鲁迅的文章。

(3)把重要的书籍和手稿带走了。



8.1 短语结构分析概述

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 个介词短语， C_N 可由下式获得 [Samuelsson, 2000]:

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



8.1 短语结构分析概述

◆ 基本方法

- 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... ..
 - 基于 PCFG 的分析方法
 - 借助神经网络改进的句法分析方法
-



短语结构分析

8.1 概述

 8.2 线图分析法

8.3 CYK分析法

8.4 基于PCFG的分析法

8.5 句法分析性能评估

8.6 局部句法分析



8.2 线图分析法

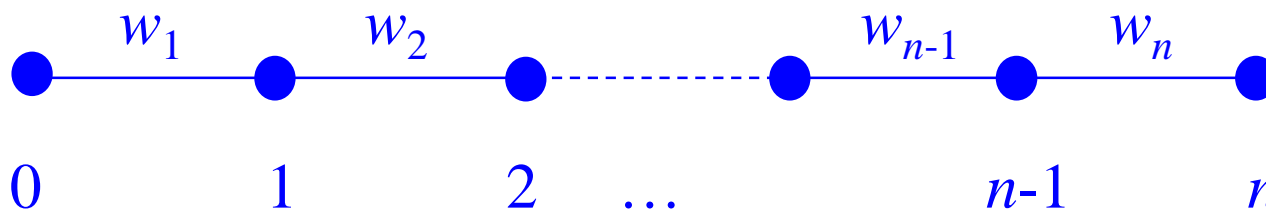
◆ 三种策略

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

8.2 线图分析法

◆ 自底向上的 Chart 分析算法

- 给定一组 CFG(上下文无关文法) 规则: $XP \rightarrow \alpha_1 \dots \alpha_n$ ($n \geq 1$)
- 给定一个句子的词性序列: $S = w_1 w_2 \dots w_n$
- 构造一个线图: 一组结点和边的集合;

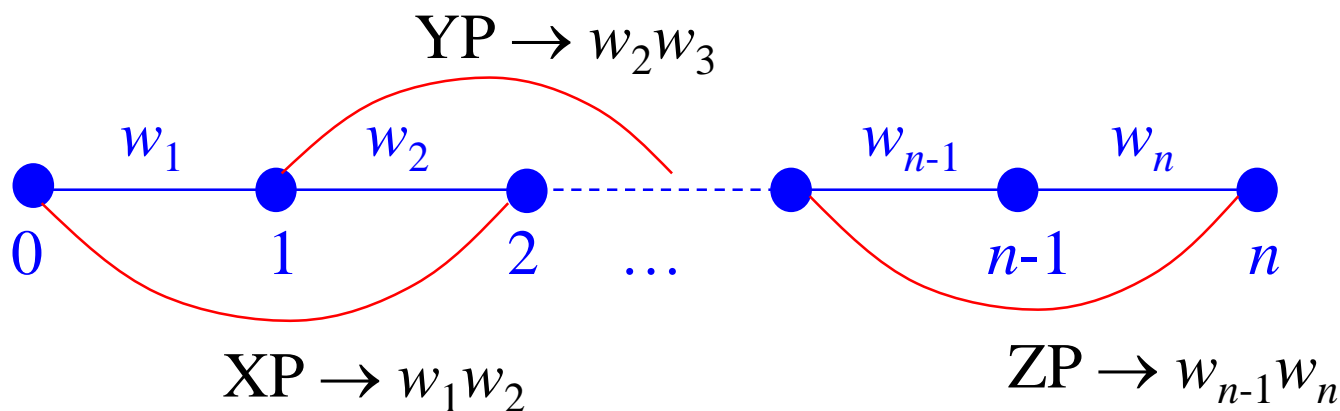


- 建立一个二维表: 记录每条边的起始和终止位置。

8.2 线图分析法

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。



8.2 线图分析法

点规则：用于表示规则右部被归约(reduce)的程度。

设有规则： $NP \rightarrow Det\ A\ N$ $NP \rightarrow Det\ N$ $NP \rightarrow A\ N$

短语： The good book



$NP \rightarrow \underline{Det} \circ A\ N$

$NP \rightarrow Det\ A\ \underline{\circ} N$

$NP \rightarrow Det\ A\ N\ \underline{\circ}$

点的位置左边是匹配到的规则的一部分

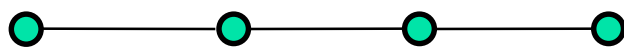
8.2 线图分析法

点规则：用于表示规则右部被归约(reduce)的程度。

设有规则： $NP \rightarrow Det\ A\ N$ $NP \rightarrow Det\ N$ $NP \rightarrow A\ N$

短语： The good book

Det A N



点规则

$NP \rightarrow \underline{Det} \circ A\ N$

$NP \rightarrow Det\ A\ \circ N$

$NP \rightarrow Det\ A\ N\ \circ$

活性边(活动弧):

规则右部未被完全匹配

非活性边(非活动弧, 或完成弧):

规则右部已被完全匹配



8.2 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

8.2 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

①形态分析: the boy hit the dog with a rod

8.2 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

①形态分析: the boy hit the dog with a rod

②词性标注: Det N V Det N Prep Det N

8.2 线图分析法

记录活性表

③ 句法分析

Agenda

ActiveArc

Chart

Acts

① Det (1, 2) ② NP → Det ◦ N (1,2) ③ Det (1, 2) 返回



(1) $S \rightarrow NP \ VP$

(2) $NP \rightarrow Det \ N$

(3) $VP \rightarrow VP \ PP$

(4) $VP \rightarrow V \ NP$

(5) $PP \rightarrow Prep \ NP$

8.2 线图分析法

③ 句法分析

Agenda

ActiveArc

Chart

Acts

- | | | | |
|--------------|----------------------|--------------|--------------------|
| ① Det (1, 2) | ② NP → Det ◦ 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
(将其加入agenda) |



(1) S → NP VP

(2) NP → Det N

(3) VP → VP PP

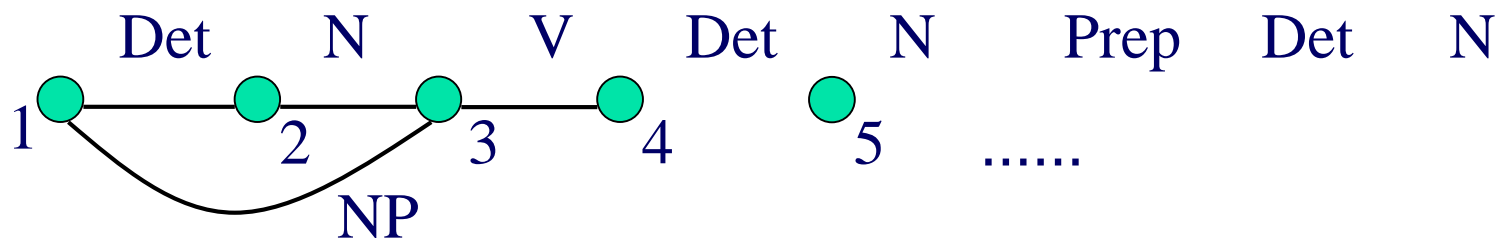
(4) VP → V NP

(5) PP → Prep NP

8.2 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) $VP \rightarrow VP \ PP$

(4) $VP \rightarrow V \ NP$

(5) $PP \rightarrow Prep \ NP$

8.2 线图分析法

② $NP \rightarrow Det \circ N (1,2)$

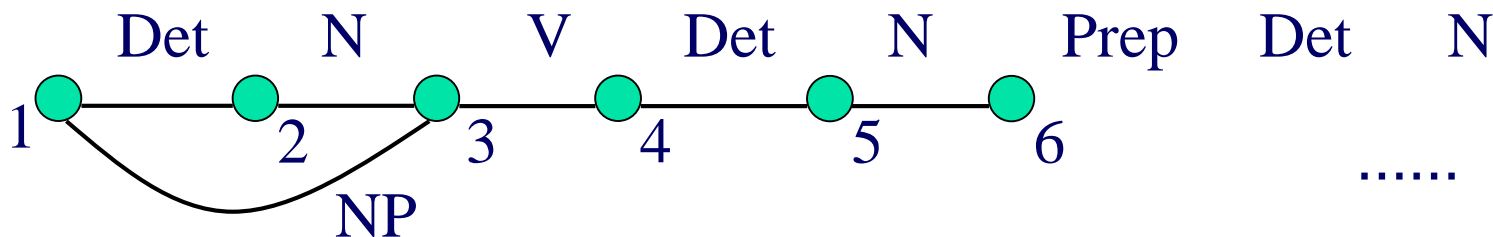
⑥ $NP \rightarrow Det \ N \circ (1,3)$

⑧ $S \rightarrow NP \circ VP (1, 3)$

⑪ $VP \rightarrow V \circ NP (3, 4)$

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) $VP \rightarrow VP \ PP$

(4) $VP \rightarrow V \ NP$

(5) $PP \rightarrow Prep \ NP$

8.2 线图分析法

② $NP \rightarrow Det \circ N (1,2)$

⑥ $NP \rightarrow Det \ N \circ (1,3)$

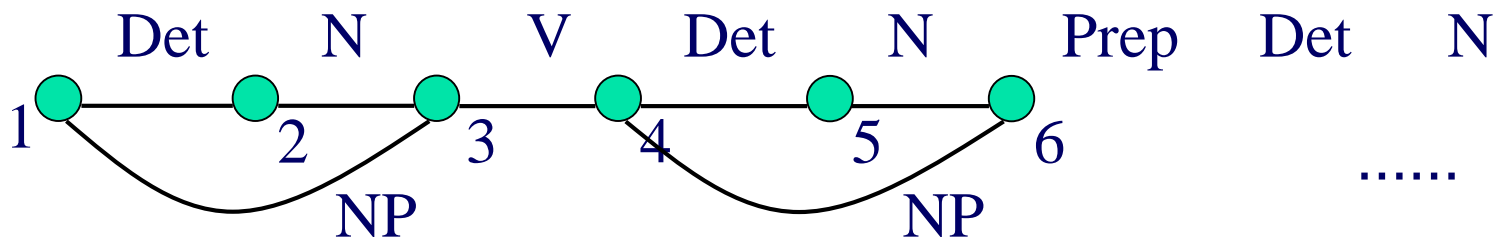
⑧ $S \rightarrow NP \circ VP (1, 3)$

⑪ $VP \rightarrow V \circ NP (3, 4)$

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) $VP \rightarrow VP \ PP$

(4) $VP \rightarrow V \ NP$

(5) $PP \rightarrow Prep \ NP$

8.2 线图分析法

- 14 $NP \rightarrow Det \circ N$ (4,5) 2 $NP \rightarrow Det \circ N$ (1,2)
 18 $NP \rightarrow Det N \circ$ (4,6) 6 $NP \rightarrow Det N \circ$ (1,3)
 20 $S \rightarrow NP \circ VP$ (4, 6) 8 $S \rightarrow NP \circ VP$ (1, 3)
 11 $VP \rightarrow V \circ NP$ (3, 4)

③ 句法分析

Agenda

ActiveArc

Chart

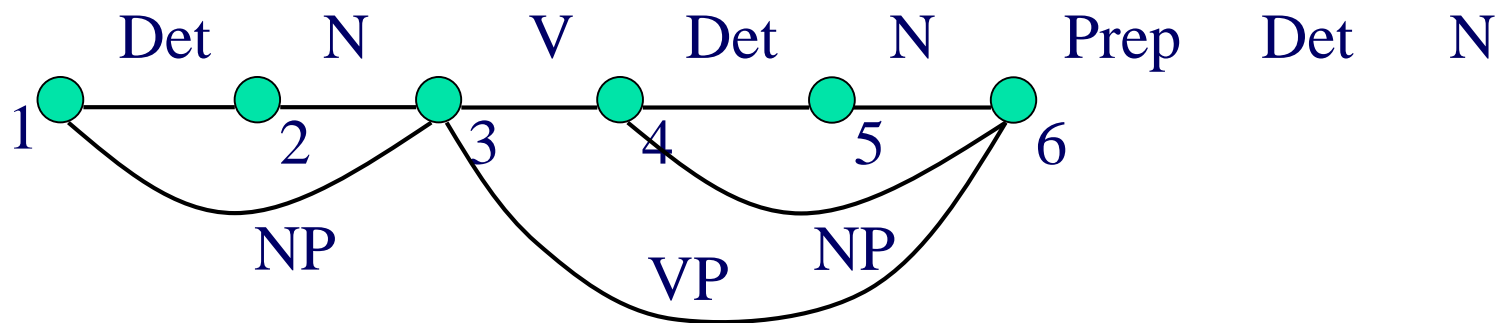
Acts

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?

... ..

注意：这里 **vp(1,3)** 不能通过 **8** 扩展



(1) $S \rightarrow NP VP$

(2) $NP \rightarrow Det N$

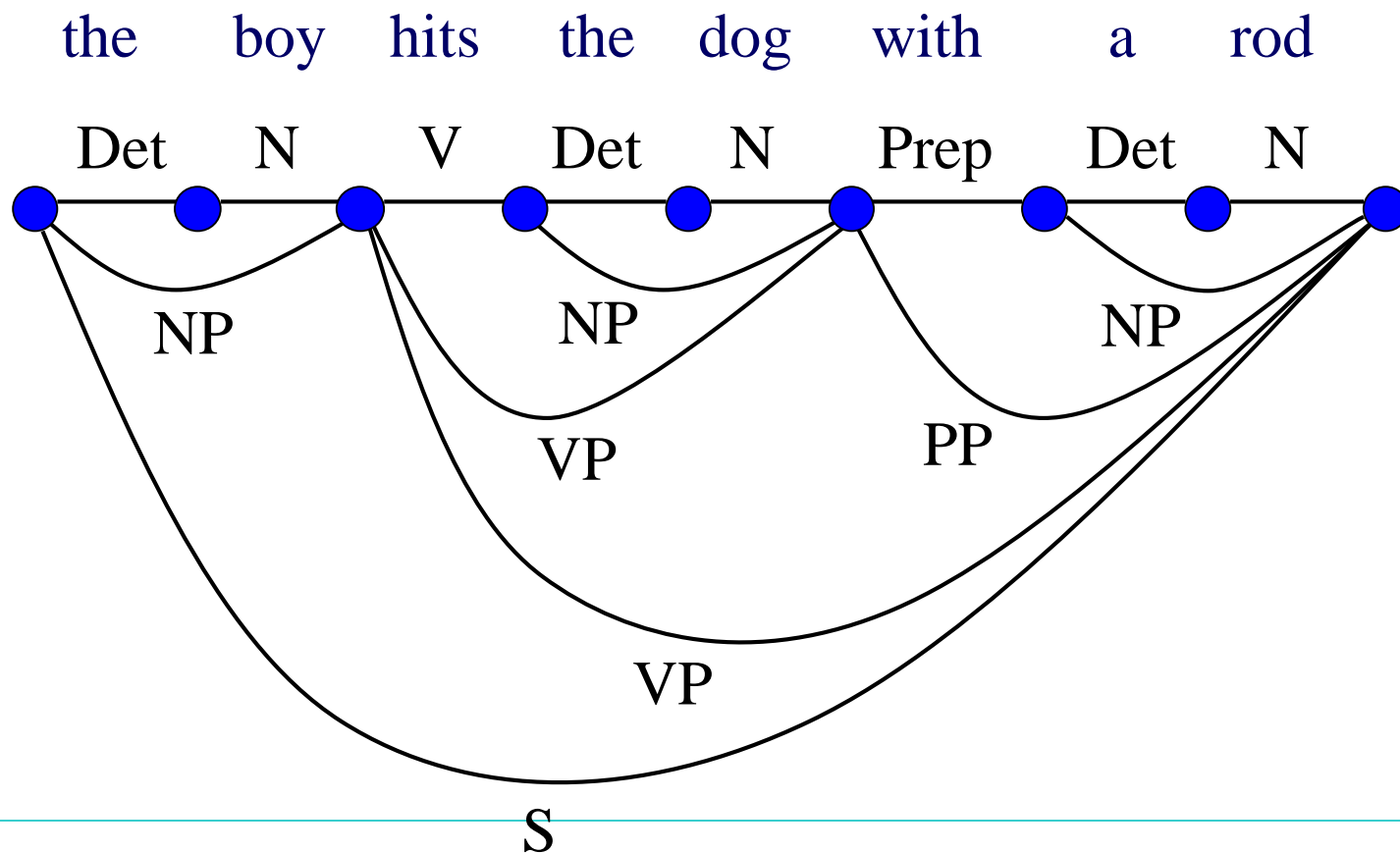
(3) $VP \rightarrow VP PP$

(4) $VP \rightarrow V NP$

(5) $PP \rightarrow Prep NP$

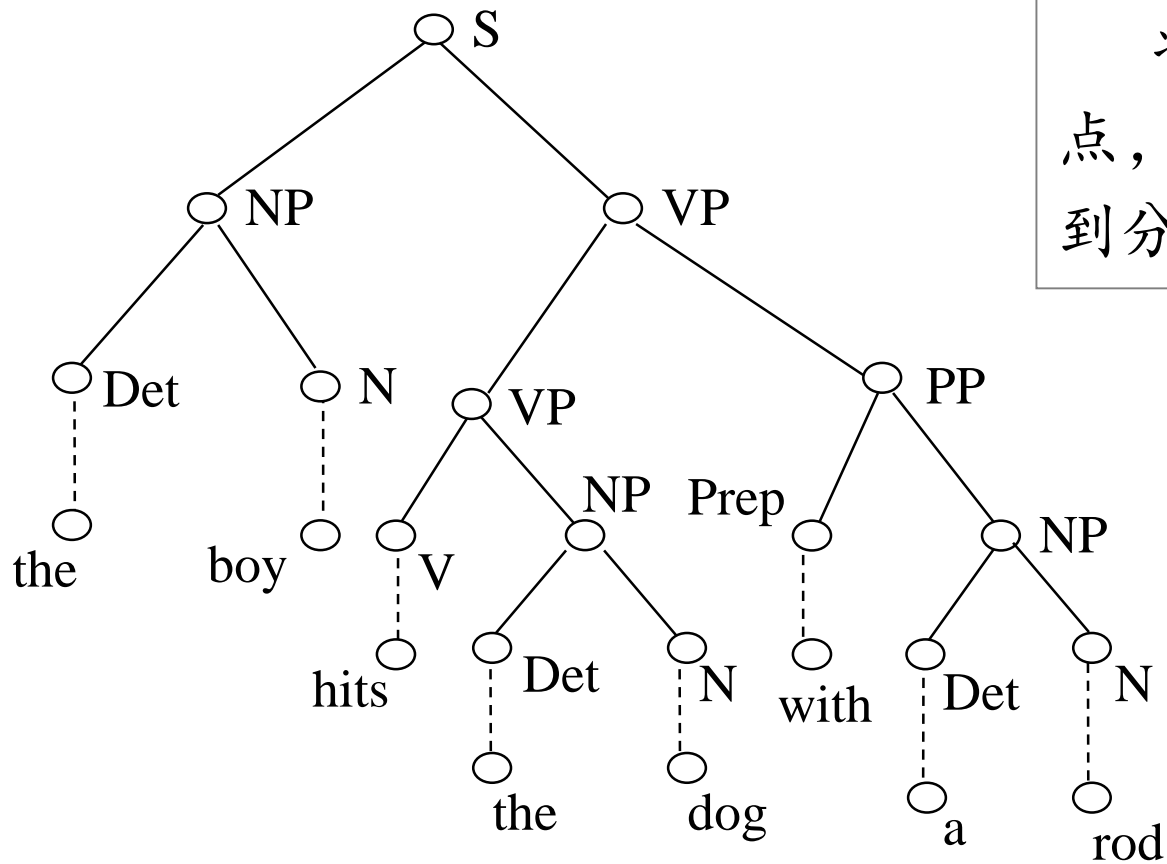
8.2 线图分析法

最后分析结果:



8.2 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。





8.2 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。
-

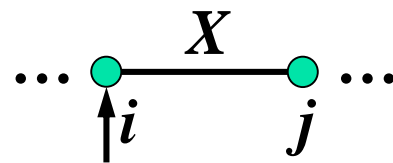
8.2 线图分析法

◆ 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 $X(i, j)$ 。



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

8.2 线图分析法

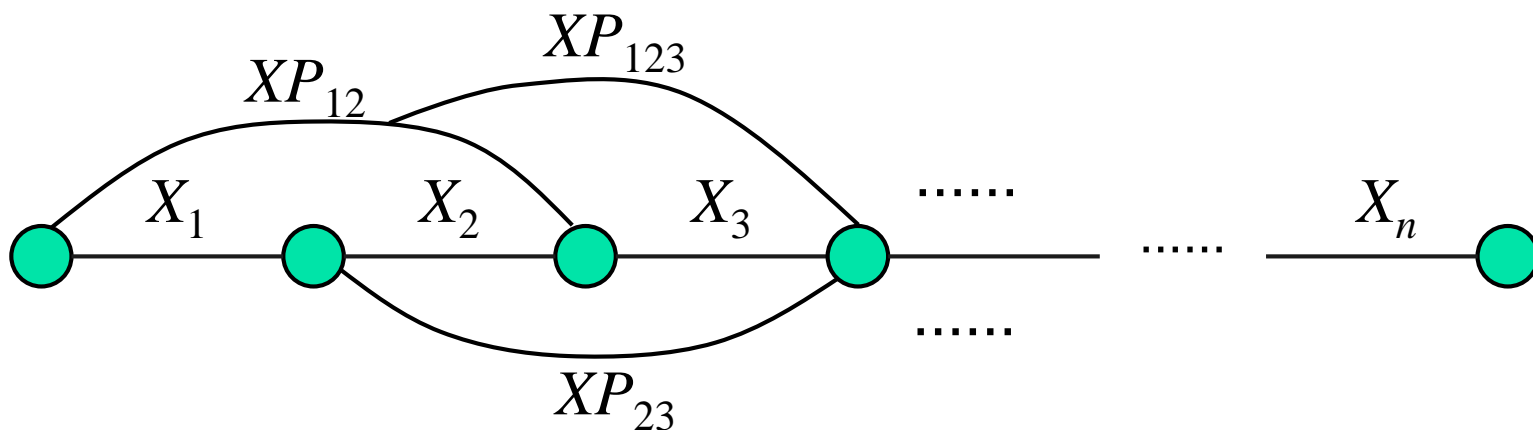
✧ 扩展弧子程序:

- (a) 将 X 插入图表(Char)的 (i, j) 位置中。
- (b) 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: $A \rightarrow \alpha \circ X$, 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, 则将 $A(k, j)$ 加入到Agenda表中。
- (c) 对于每个位置为 (k, i) 的点规则: $A \rightarrow \alpha \circ X\beta$, 则将 $A \rightarrow \alpha X \circ \beta(k, j)$ 加入到活动边集中。

8.2 线图分析法

◆ 算法的时间复杂度分析

设 n 为输入句子的长度, C 为上下文无关文法中的非终结符的数目, S 为点规则的状态数目(大于 CFG 规则的数目), 显然 $S > C$ 。因为 Agenda 表中的元素形式为 $X(i, j)$, 因此, Agenda 表中最大的元素个数为: Cn^2 。





8.2 线图分析法

由于ActiveArc 中的元素形式为： $A \rightarrow \alpha \circ X(i, j)$ ，所以ActiveArc 表中最大的元素数目为： Sn^2 。

{Chart 表中的边的形式为： $A(i, j)$ ，因此，Chart 表中最大的元素数目为： Cn^2 。}

我们来考察算法中每一步执行的最大次数：

8.2 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 $X(i, j)$

最多执行的次数为: C

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

最多执行的次数为: 1

- (3) 对于每条规则 $A \rightarrow X\gamma$, 将 $A \rightarrow X \circ \gamma(i, j)$ 加入活动边集ActiveArc 中, 然后

最多执行的次数为: Sn^2

8.2 线图分析法

✧ 扩展弧子程序:

最多执行的次数为: 1

(a) 将 X 插入图表(Chart)的 (t, j) 位置中。

(b) 对于活动边集(ActiveArc)中每个位置为 (k, i) ($i > k \geq 1$) 的点规则, 如果该规则具有如下形式: $A \rightarrow \alpha \circ X$, 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, 则将

最多执行的次数为: Sn^2

(c) 对于每个位置为 (k, i) 的点规则: $A \rightarrow \alpha \circ X\beta$, 则将 $A \rightarrow \alpha X \circ \beta(k, j)$ 加入到活动边集中

最多执行的次数为: Sn^2



8.2 线图分析法

每处理一个单词需要最多执行的最多操作次数为：

$$C + 1 + Sn^2 + 1 + Sn^2 + Sn^2 = 2 + C + 3Sn^2$$

由于算法对于长度为 n 的输入句子要执行 n 次循环，因此，Chart 算法最大执行的操作次数为：

$$n \times (2 + C + 3Sn^2)$$

所以，Chart算法的时间复杂度为: $O(Kn^3)$, 其中, K 为一常数。



8.2 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 ;
 - 需要高质量的规则，分析结果与规则质量密切相关;
 - 难以区分歧义结构。
-



短语结构分析

8.1 概述

8.2 线图分析法

 8.3 CYK分析法

8.4 基于PCFG的分析法

8.5 句法分析性能评估

8.6 局部句法分析



8.3 CYK分析法

◆ Cocke-Younger-Kasami (CYK) 算法

➤ 对 Chomsky 文法进行范式化:

$A \rightarrow w$ (终结符) 或 $A \rightarrow BC$ (必须是2个非终结符)

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

➤ 自下而上的分析方法

➤ 需构造 $(n+1) \times (n+1)$ 识别矩阵, n 为句子长度。

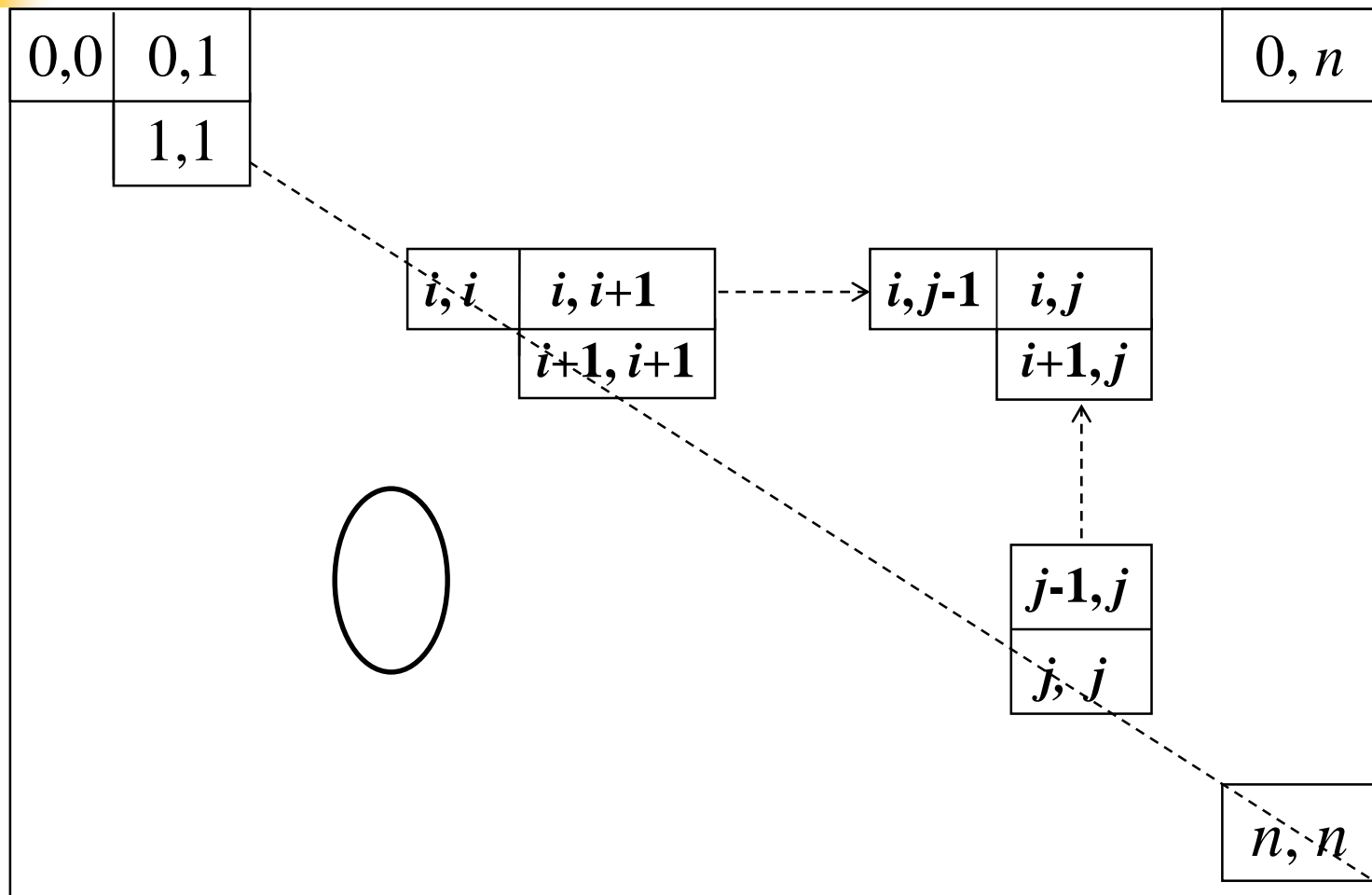


8.3 CYK分析法

◆ 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句子的终结符(单词)构成
-

8.3 CYK分析法





8.3 CYK分析法

◆ 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 $x = w_1 w_2 \dots w_n$ ，从 w_1 开始分析。
-



8.3 CYK分析法

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。

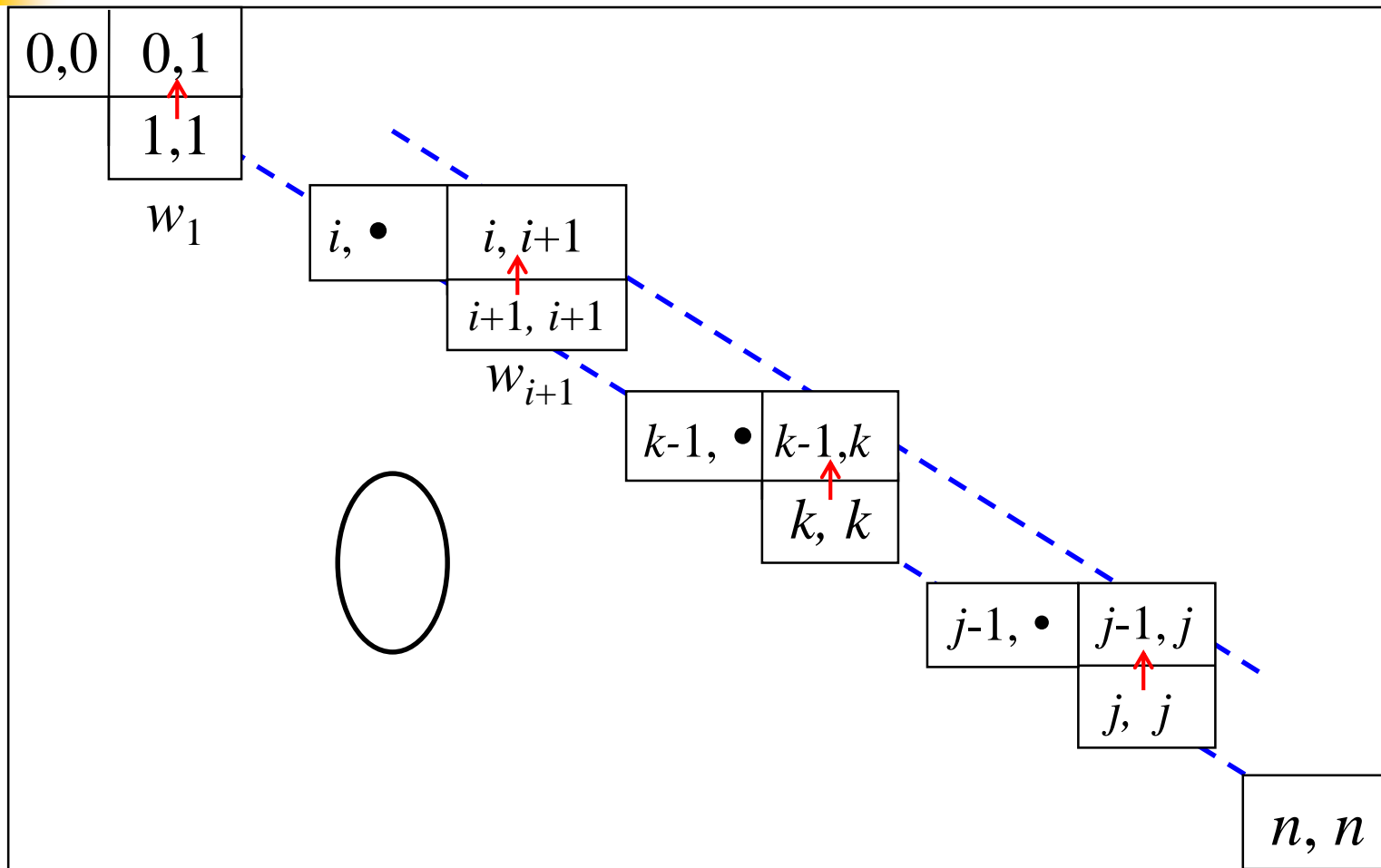


8.3 CYK分析法

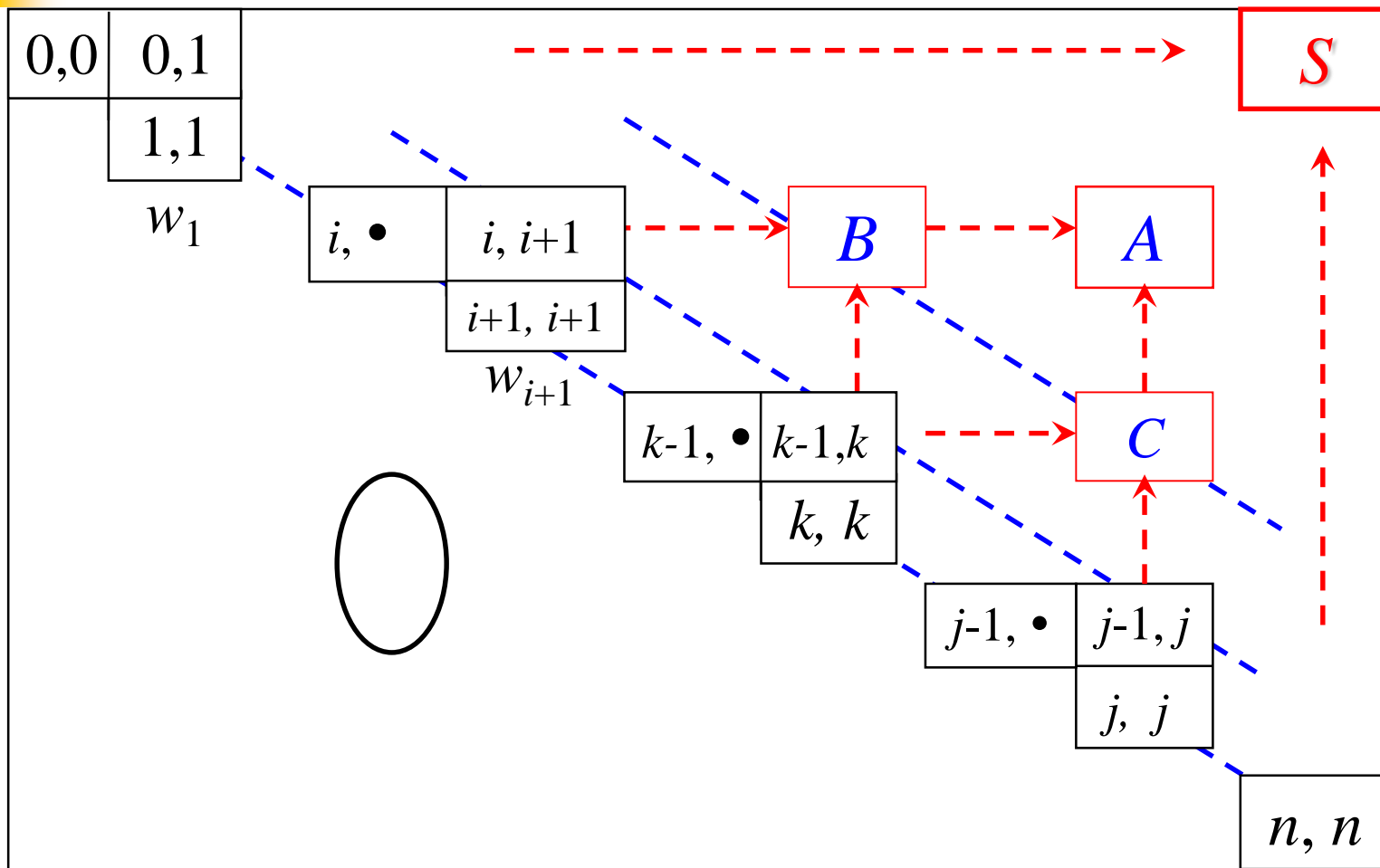
(3) 按平行于主对角线的方向，一层一层地向上填写矩阵的各个元素 $t_{i,j}$ ，其中， $i = 0, 1, \dots, n-d$ ， $j = d+i$ ， $d = 2, 3, \dots, n$ 。如果存在一个正整数 k ， $i+1 \leq k \leq j-1$ ，在文法 G 的规则集中有产生式 $A \rightarrow BC$ ，并且， $B \in t_{i,k}$ ， $C \in t_{k,j}$ ，那么，将 A 写到矩阵 $t_{i,j}$ 位置上。

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

8.3 CYK分析法



8.3 CYK分析法





8.3 CYK分析法

◆ 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

8.3 CYK分析法

(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

8.3 CYK分析法

(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		
2			喜欢	V	N
3				读	N
4					书

8.3 CYK分析法

(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

8.3 CYK分析法

(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

8.3 CYK分析法

(1) 汉语分词和词性标注以后:

他/P 喜欢/V 读/V 书/N $n=4$

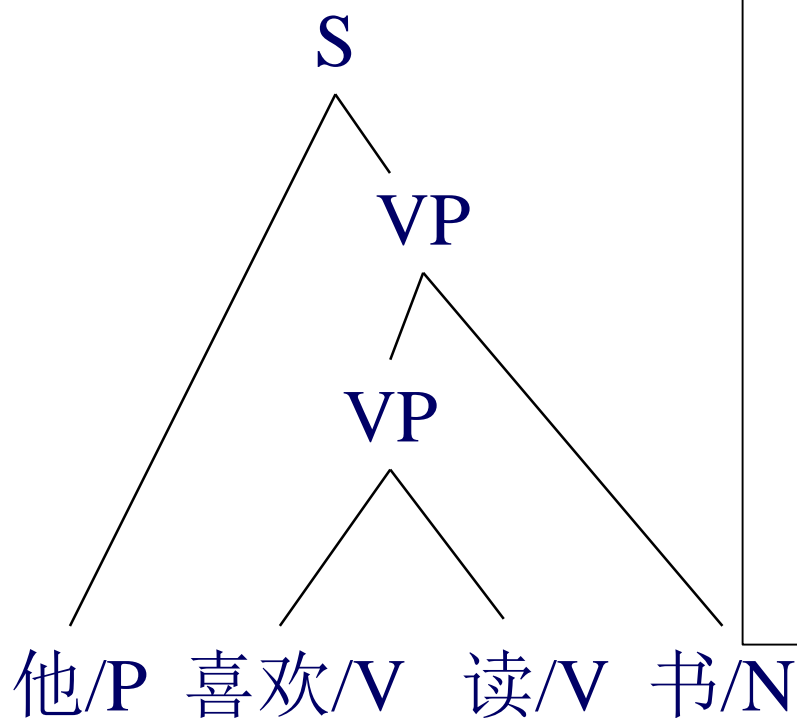
(2) 构造识别矩阵:

(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

8.3 CYK分析法



◆ CYK 算法的评价

● 优点

- 实现简单，执行效率高

● 弱点

- 文法需要范式化处理
(只支持二叉树结构)
- 无法区分歧义



短语结构分析

8.1 概述

8.2 线图分析法

8.3 CYK分析法

 8.4 基于PCFG的分析法

8.5 句法分析性能评估

8.6 局部句法分析

8.4 基于PCFG的分析法

◆ 概率上下文无关文法

(probabilistic/stochastic context-free grammar, PCFG/SCFG)

规则形式: $A \rightarrow \alpha, p$

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

例如:

NP	\rightarrow	NN NN	, 0.60	}	$\Sigma p=1$
NP	\rightarrow	NN CC NN	, 0.40		
CD	\rightarrow	Num Num	, 0.85	}	$\Sigma p=1$
CD	\rightarrow	Num DM	, 0.15		

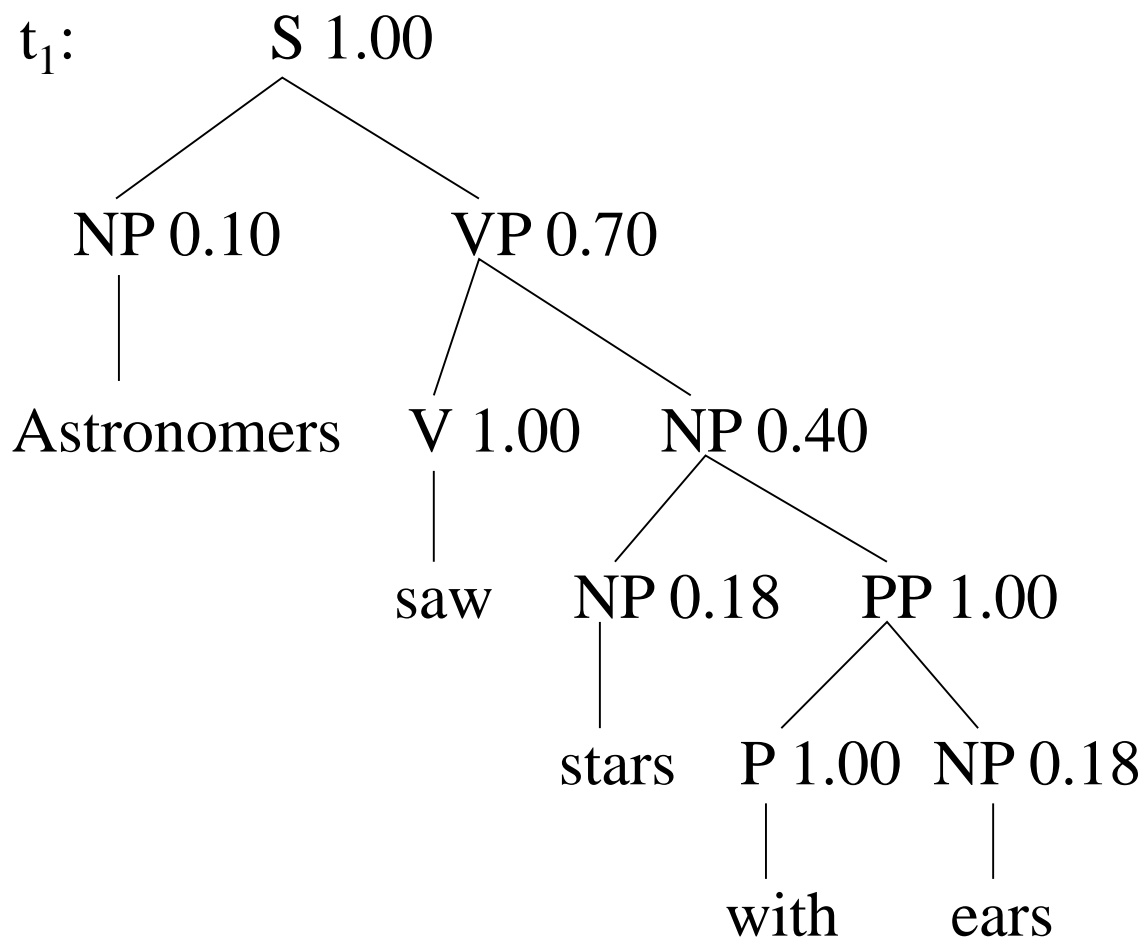


8.4 基于PCFG的分析法

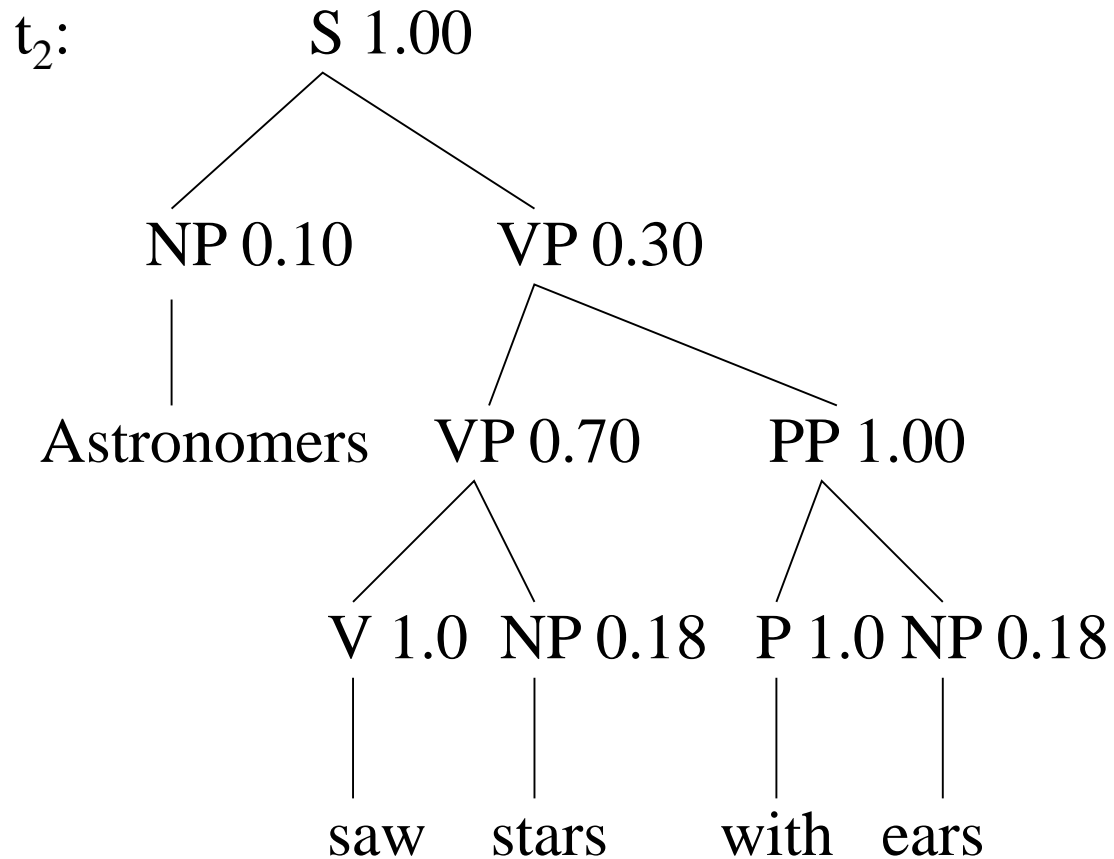
- ◆ 例-1: $S \rightarrow NP \ VP, 1.00$ $NP \rightarrow NP \ PP, 0.40$
 $NP \rightarrow \text{astronomers}, 0.10$
 $NP \rightarrow \text{ears}, 0.18$ $NP \rightarrow \text{saw}, 0.04$
 $NP \rightarrow \text{stars}, 0.18$ $NP \rightarrow \text{telescopes}, 0.1$
 $PP \rightarrow P \ NP, 1.00$ $P \rightarrow \text{with}, 1.00$
 $VP \rightarrow V \ NP, 0.70$ $VP \rightarrow VP \ PP, 0.30$
 $V \rightarrow \text{saw}, 1.00$

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

8.4 基于PCFG的分析法



8.4 基于PCFG的分析法





8.4 基于PCFG的分析法

◆ 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 $p(A_{kl} \rightarrow w / \text{任何除 } A \text{ 以外的祖先结点}) = p(A_{kl} \rightarrow w)。$

8.4 基于PCFG的分析法

t_1 : S 1.00

NP 0.10

Astronomers

VP 0.70

V 1.00

saw

NP 0.40

NP 0.18

stars

PP 1.00

P 1.00

with

NP 0.18

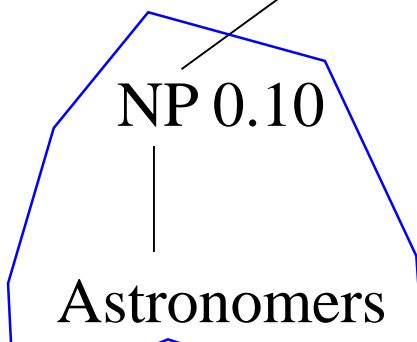
ears

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

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

8.4 基于PCFG的分析法

t_1 : S 1.00



VP 0.70

V 1.00

saw

NP 0.40

NP 0.18

stars

PP 1.00

P 1.00 NP 0.18

with

ears

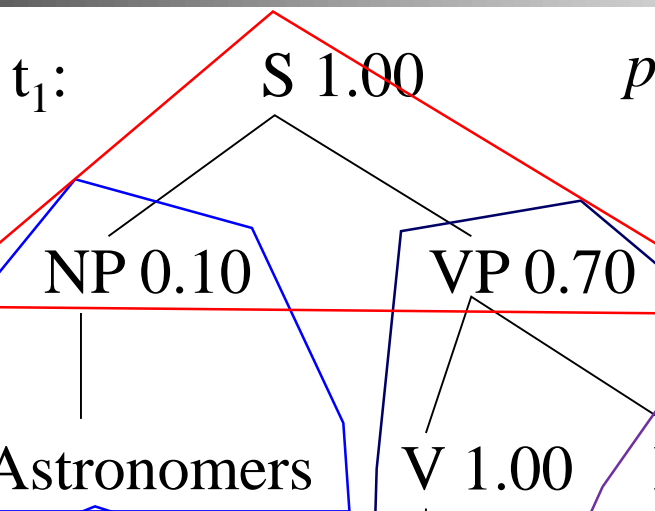
$$\begin{aligned}
 p(\text{tree}_{\text{PP}}) &= p(P \rightarrow \text{with}) \\
 &\quad \times p(\text{NP} \rightarrow \text{ears}) \\
 &\quad \times p(\text{PP} \rightarrow P \text{ NP}) \\
 &= 1.00 \times 0.18 \times 1.00 \\
 &= 0.18
 \end{aligned}$$

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

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

?

8.4 基于PCFG的分析法



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

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

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

$$= 0.009072$$

$$p(\text{tree}_{\text{NP}}) =$$

$$p(\text{NP} \rightarrow \text{astronomers}) = 0.10$$

$$p(\text{tree}_{\text{NP}}) = p(\text{NP} \rightarrow \text{stars}) \times$$

$$= p(\text{tree}_{\text{PP}}) \times p(\text{NP} \rightarrow \text{NP PP})$$

$$= 0.18 \times 0.18 \times 0.4$$

$$= 0.01296$$

$$p(t_1) = p(\text{tree}_{\text{NP}}) \times$$

$$p(\text{VP}) \times$$

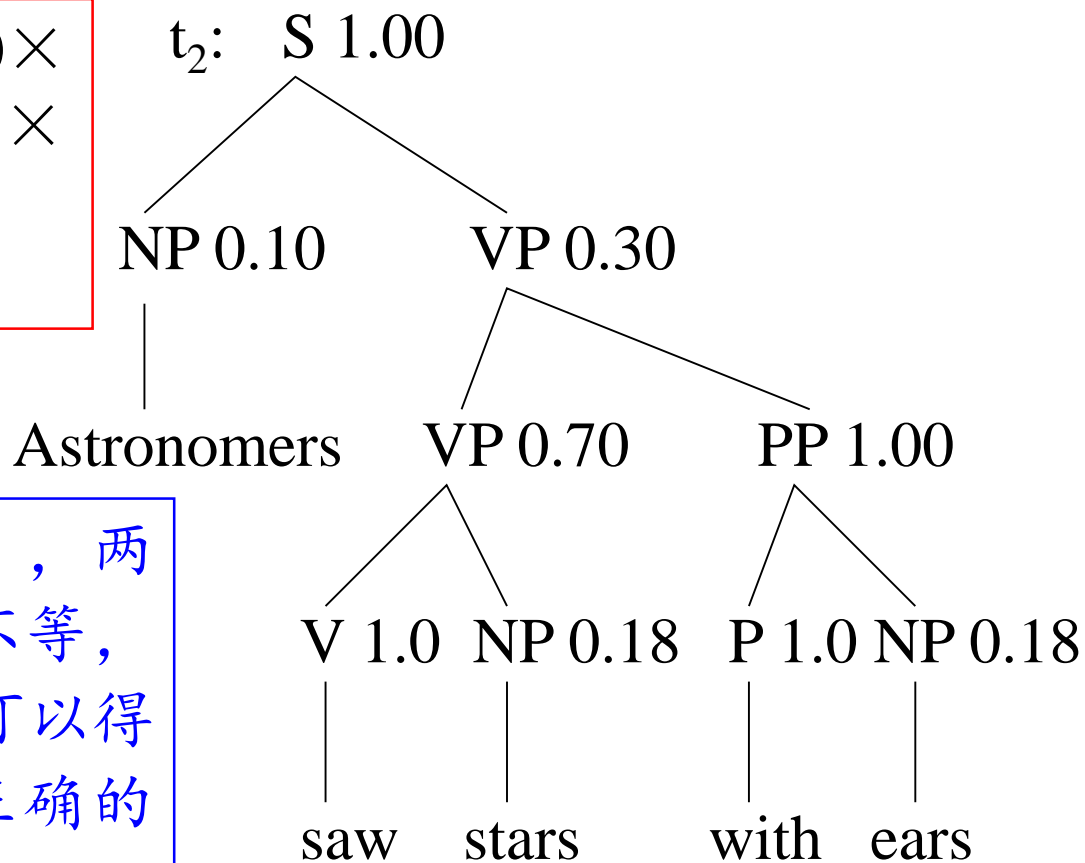
$$p(S \rightarrow \text{NP VP})$$

$$= 0.10 \times 1.0 \times 0.009072$$

$$= 0.0009072$$

8.4 基于PCFG的分析法

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



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 。



8.4 基于PCFG的分析法

◆ PCFG的三个问题

- 给定句子 $W=w_1w_2\dots w_n$ 和 PCFG G , 如何快速地计算 $p(W|G)$?
 - 给定句子 $W=w_1w_2\dots w_n$ 和 PCFG G , 如何快速地选择最佳句法结构树?
 - 给定句子 $W=w_1w_2\dots w_n$ 和 PCFG G , 如何调节 G 的参数, 使得 $p(W|G)$ 最大?
-



8.4 基于PCFG的分析法

假设文法 $G(S)$ 的规则只有两种形式:

$$A \rightarrow w, \quad w \in V_T$$

$$A \rightarrow BC, \quad B, C \in V_N$$

可以通过范式化处理, 使CFG 规则满足上述形式。
这种假设的文法形式称为乔姆斯基范式 (Chomsky normal form, CNF)。



8.4 基于PCFG的分析法

◆ 求解问题1：快速地计算句子的句法树概率- 内向或外向算法

● 内向算法

- 基本思想：利用动态规划算法计算由非终结符 A 推导出的某个字串片段 $w_i w_{i+1} \dots w_j$ 的概率 $\alpha_{ij}(A)$ 。语句 $W = w_1 w_2 \dots w_n$ 的概率即为文法 $G(S)$ 中 S 推导出的字串的概率 $\alpha_{1n}(S)$ 。



8.4 基于PCFG的分析法

- 定义：内向变量 $\alpha_{ij}(A)$ 是由非终结符 A 推导出的语句 W 中子字串 $w_i w_{i+1} \dots w_j$ 的概率：

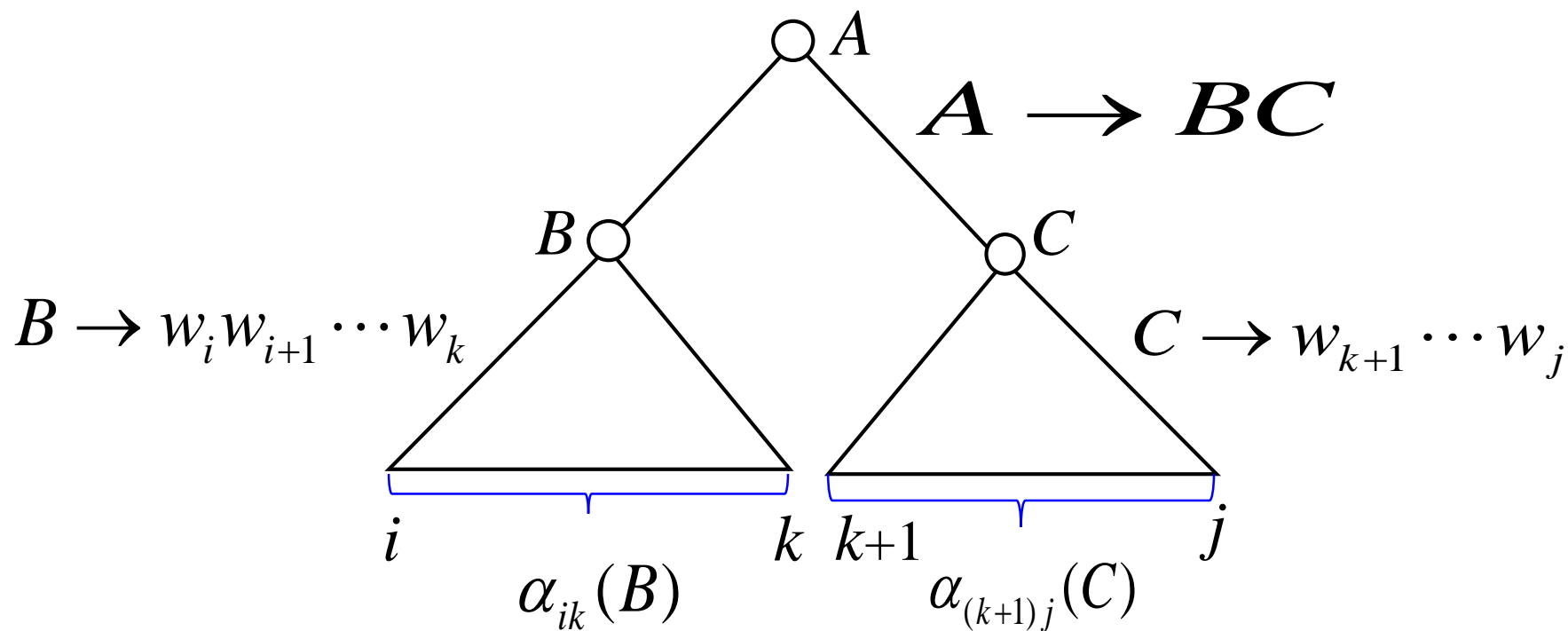
$$\alpha_{ij}(A) = p(A \xRightarrow{*} w_i w_{i+1} \dots w_j)$$

计算 $\alpha_{ij}(A)$ 的递推公式：

$$\alpha_{ii}(A) = p(A \rightarrow w_i)$$

$$\alpha_{ij}(A) = \sum_{B, C \in V_N} \sum_{i \leq k \leq j} p(A \rightarrow BC) \alpha_{ik}(B) \alpha_{(k+1)j}(C)$$

8.4 基于PCFG的分析法





8.4 基于PCFG的分析法

➤解释:

当 $i=j$ 时, 字符串 $w_i w_{i+1} \dots w_j$ 只是一个字 w_{ii} , 可简单记作 w_i , 由 A 推导出 w_i 的概率就是产生式 $A \rightarrow w_i$ 的概率 $p(A \rightarrow w_i)$; 当 $i \neq j$ 时, 也就是说, 字符串 $w_i w_{i+1} \dots w_j$ 至少有两个词, 根据约定, A 要推导出该词串, 必须首先运用产生式 $A \rightarrow BC$, 那么, 可用 B 推导出前半部 $w_i \dots w_k$, 用 C 推导出后半部 $w_{k+1} \dots w_j$ 。由这一推导过程产生 $w_i w_{i+1} \dots w_j$ 的概率为: $p(A \rightarrow BC) \alpha_{ik}(B) \alpha_{(k+1)j}(C)$ 。考虑到 B 、 C 和 k 取值的任意性, 应计算各种情况下概率的总和。

8.4 基于PCFG的分析法

➤ 内向算法描述:

输入: 文法 $G(S)$, 语句 $W = w_1 w_2 \cdots w_n$

输出: $p(S \xRightarrow{*} w_1 w_2 \dots w_n)$

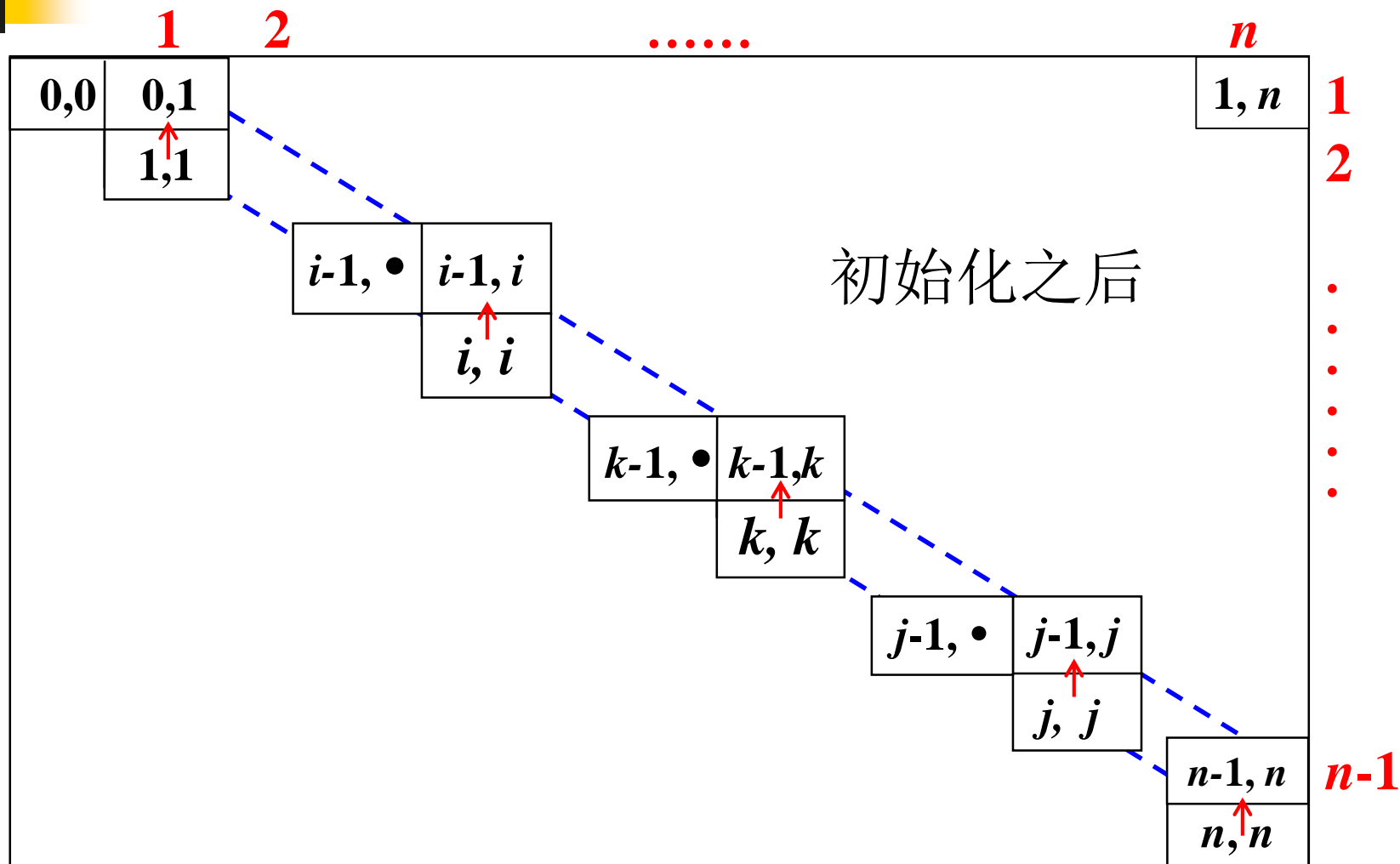
(1) 初始化: $\alpha_{ii}(A) = p(A \rightarrow w_i) \quad A \in V_N, 1 \leq i \leq n$

(2) 归纳计算: $j=1..n, i=1..n-j$, 重复下列计算:

$$\alpha_{i(i+j)}(A) = \sum_{B, C \in V_N} \sum_{i \leq k \leq i+j} p(A \rightarrow BC) \alpha_{ik}(B) \alpha_{(k+1)(i+j)}(C)$$

(3) 终结: $p(S \xRightarrow{*} w_1 w_2 \dots w_n) = \alpha_{1n}(S)$

8.4 基于PCFG的分析法





8.4 基于PCFG的分析法

循环执行：
(假设： $n=20$)

$$\alpha_{i(i+j)}(A) = \sum_{B, C \in V_N} \sum_{i \leq k \leq i+j} p(A \rightarrow BC) \alpha_{ik}(B) \alpha_{(k+1)(i+j)}(C)$$

$j=1..n$, $i=1..n-j$ ($i+j$ 表示列, i 表示行):

$j=1$: $i=1..19$: $\alpha_{12}(A), \alpha_{23}(A), \dots, \alpha_{(19)(20)}(A)$

$j=2$: $i=1..18$: $\alpha_{13}(A), \alpha_{24}(A), \dots, \alpha_{(18)(20)}(A)$

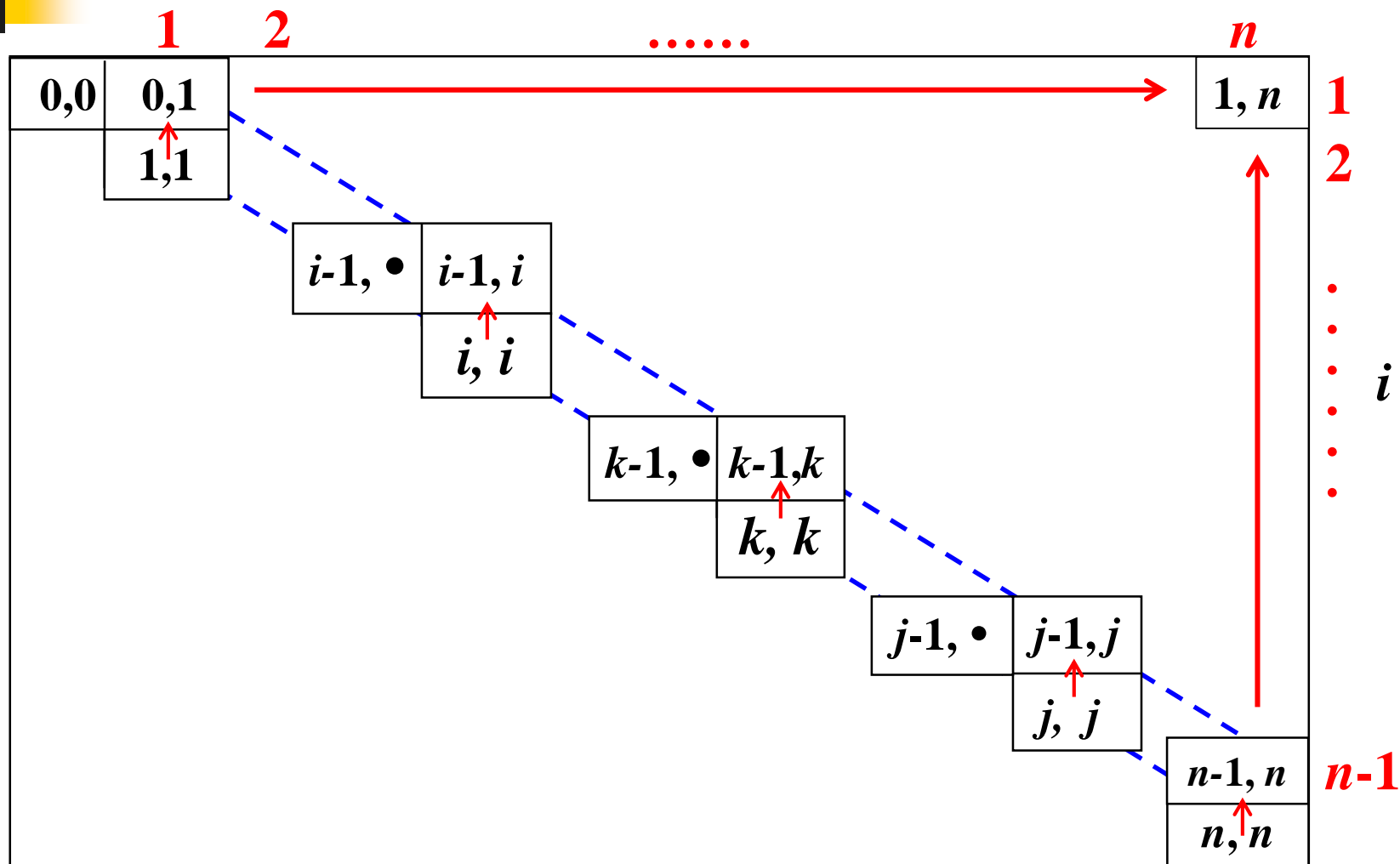
$j=3$: $i=1..17$: $\alpha_{14}(A), \alpha_{25}(A), \dots, \alpha_{(17)(20)}(A)$

.....

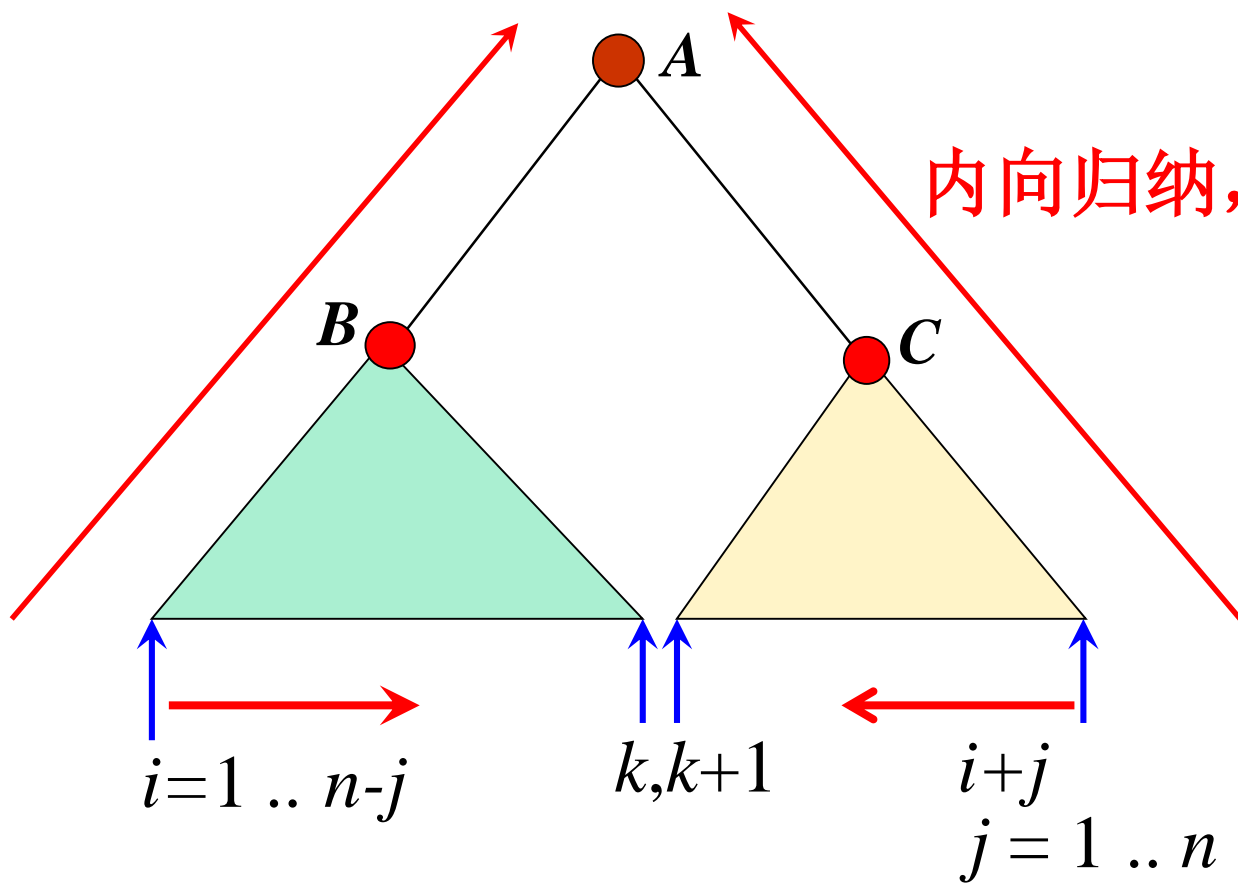
$j=19$: $i=1$: $\alpha_{1(20)}(A)$

$j=20$: 不执行

8.4 基于PCFG的分析法



8.4 基于PCFG的分析法





8.4 基于PCFG的分析法

● 外向算法

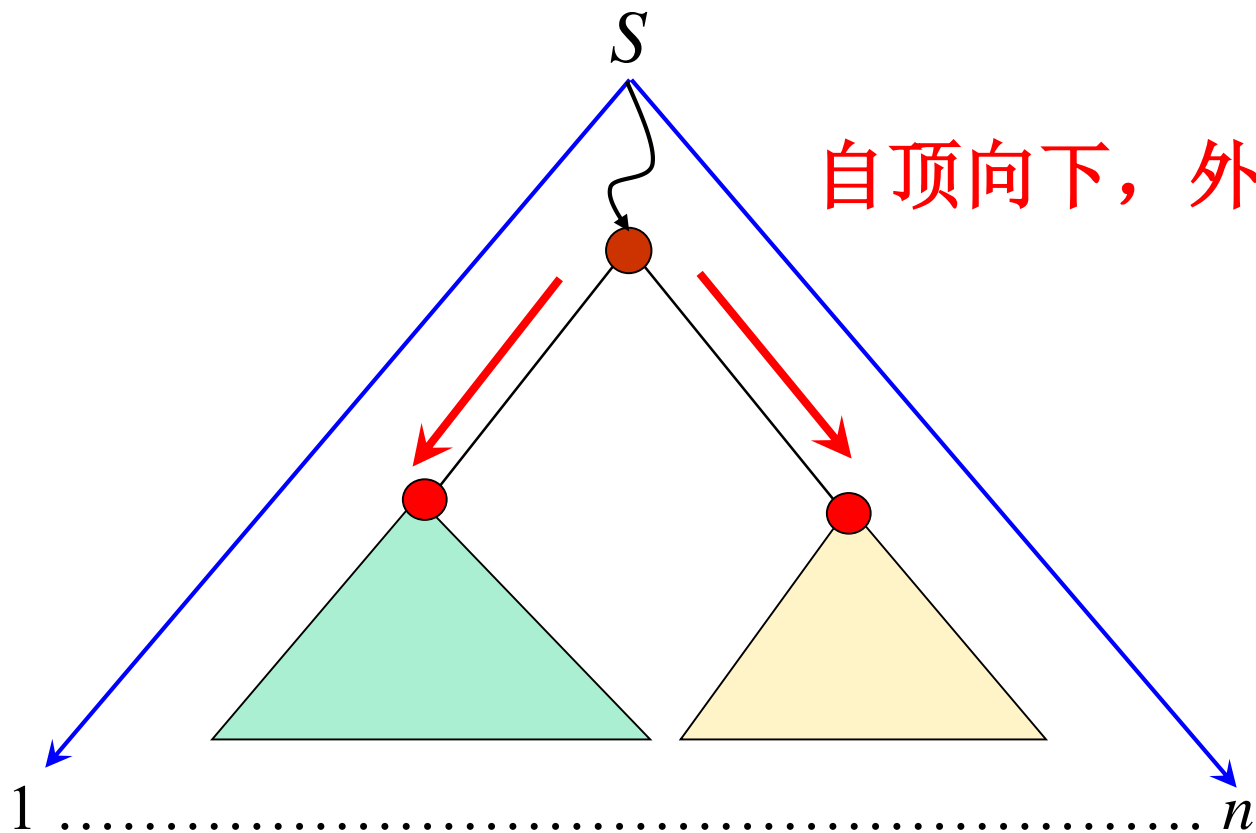
➤ 定义:

外向变量 $\beta_{ij}(A)$ 是由文法初始符号 S 推导出语句 $W=w_1w_2\dots w_n$ 的过程中, 到达扩展符号串 $w_1\dots w_{i-1}\textcircled{A}w_{j+1}\dots w_n$ 的概率($A=w_i\dots w_j$):

$$\beta_{ij}(A) = p(S \xRightarrow{*} w_1 L \ w_{i-1}\textcircled{A}w_{j+1} L \ w_n)$$

$\beta_{ij}(A)$ 表示除了以 A 为根节点的子树以外的概率。

8.4 基于PCFG的分析法



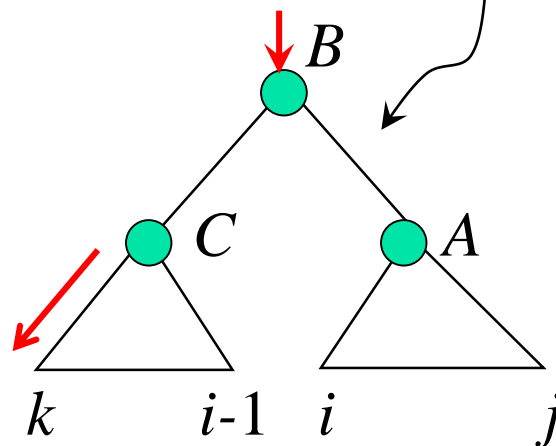
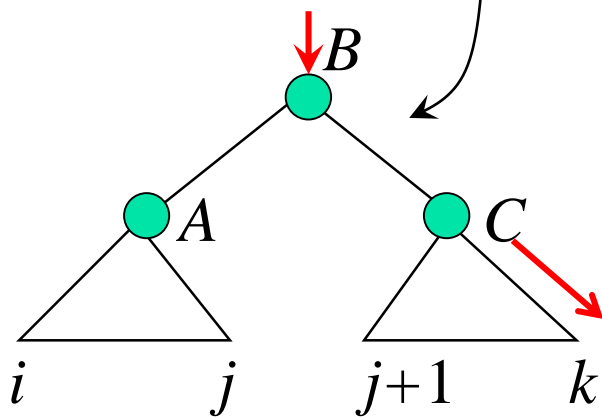
自顶向下，外向扩展。

8.4 基于PCFG的分析法

$\beta_{ij}(A)$ 可由动态规划算法求得, 其递推公式:

$$\beta_{1n}(A) = \delta(A, S) \quad (\text{初始化})$$

$$\beta_{ij}(A) = \sum_{B,C} \sum_{k>j} \beta_{ik}(B) p(B \rightarrow AC) \alpha_{(j+1)k}(C) + \sum_{B,C} \sum_{k<i} \beta_{kj}(B) p(B \rightarrow CA) \alpha_{k(i-1)}(C)$$





8.4 基于PCFG的分析法

➤解释:

(1) 当 $i=1, j=n$ 时, 即 $w_i w_{i+1} \cdots w_j$ 是整个语句 W 时, 根据乔姆斯基语法范式的约定, 不可能有规则 $S \rightarrow A$, 因此, 由 S 推导出 W 的过程中, 如果 $A \neq S$ 的话, A 推导出 W 的概率为 0 (只有 S 才能推导出 W), 即 $\beta_{1n}(A)=0$ 。

如果 $A=S$, $\beta_{1n}(A)$ 为由初始符 S 推导出 W 的概率, 因此, $\beta_{1n}(A)=1$ 。

8.4 基于PCFG的分析法

(2) 当 $i \neq 1$ 或者 $j \neq n$ 时, 如果在 S 推导出 W 的过程中出现了字符串 $w_1 \dots w_k A w_{j+1} \dots w_n$, 则该推导过程必定使用了规则 $B \rightarrow AC$ 或 $B \rightarrow CA$ 。假定运用了规则 $B \rightarrow AC$ 推导出 $w_i \dots w_j w_{j+1} \dots w_k$, 则该推导可以分解为以下三步:

(a) 由 S 推导出 $w_1 \dots w_{i-1} B w_{k+1} \dots w_n$, 其概率为 $\beta_{ik}(B)$;

(b) 运用产生式 $B \rightarrow AC$ 扩展非终结符 B , 其概率为 $p(B \rightarrow AC)$;

(c) 由非终结符 C 推导出 $w_{j+1} \dots w_k$, 其概率为 $\alpha_{(j+1)k}(C)$ 。

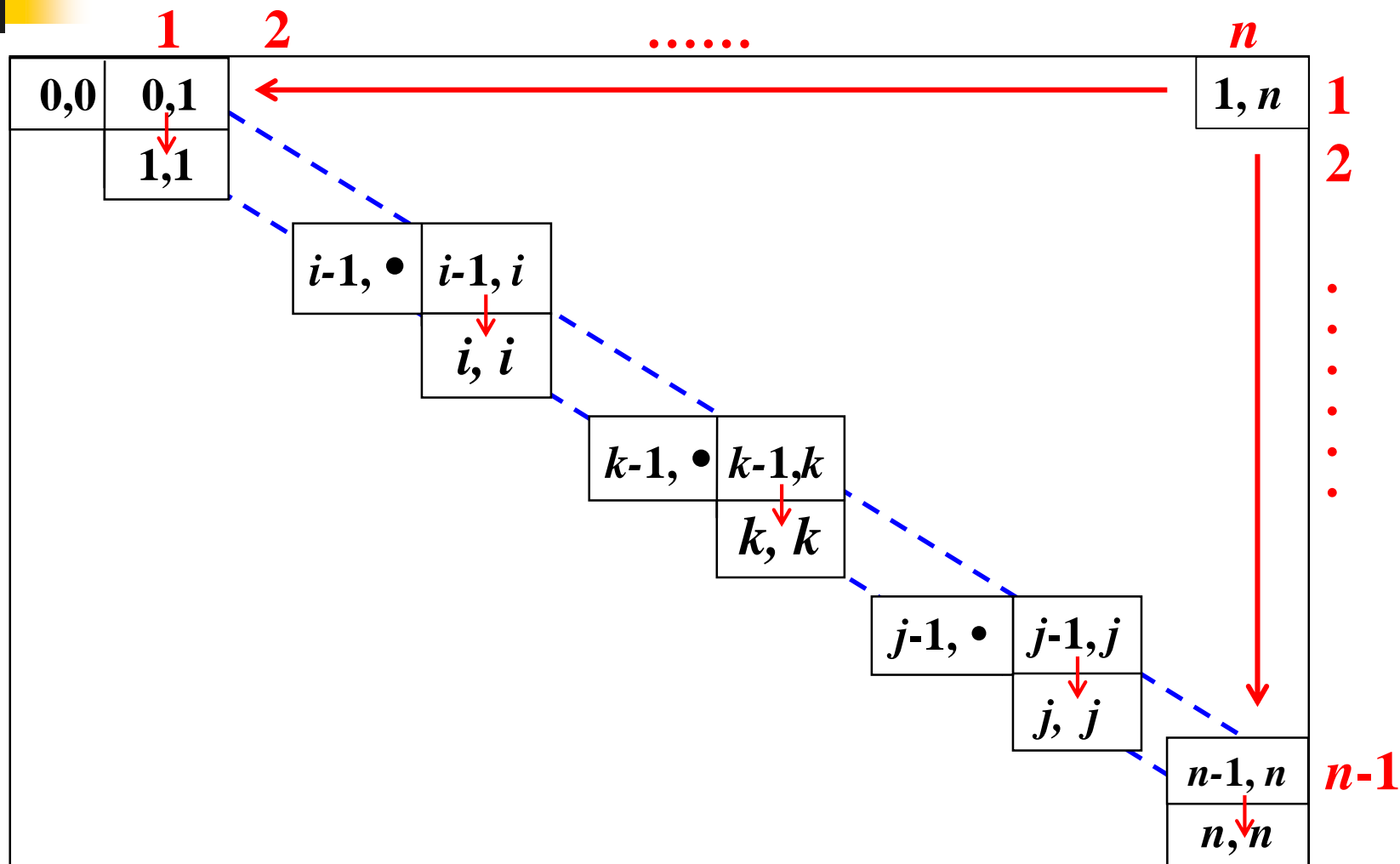


8.4 基于PCFG的分析法

考虑到 B , C 和 k 的任意性, 在计算 $\beta_{ik}(B)$ 时, 必须考虑所有可能的 B , C 和 k , 因此, 计算概率时必须考虑所有情况下的概率之和。

同样方法, 可以计算出运用产生式 $B \rightarrow CA$ 推导出 $w_k \cdots w_{i-1} w_i \cdots w_j$ 的概率。

8.4 基于PCFG的分析法



8.4 基于PCFG的分析法

➤ 外向算法描述:

输入: PCFG $G=(S, N, T, P)$, 语句 $W = w_1 w_2 \cdots w_n$;

输出: $p(W)$, $A \in N$, $1 \leq i \leq j \leq n$ 。

(1) 初始化: $\beta_{1n}(A) = \delta(A, S)$, $A \in N$;

(2) 归纳: $j = n-2 \dots 0$, $i = 1 \dots n-j$, 重复计算:

$$\begin{aligned} \beta_{i(i+j)}(A) = & \sum_{B,C} \sum_{i+j < k \leq n} p(B \rightarrow AC) \alpha_{(i+j+1)k}(C) \beta_{ik}(B) \\ & + \sum_{B,C} \sum_{1 \leq k < i} p(B \rightarrow CA) \alpha_{k(i-1)}(C) \beta_{k(i+j)}(B) \end{aligned}$$

(3) 终结: $p(S \xRightarrow{*} w_1 w_2 \dots w_n) = \sum_A \beta_{1n}(A) \times p(A \rightarrow w_i)$

8.4 基于PCFG的分析法

假设 $W=w_1w_2...w_{10}$, 即 $n=10$ 。

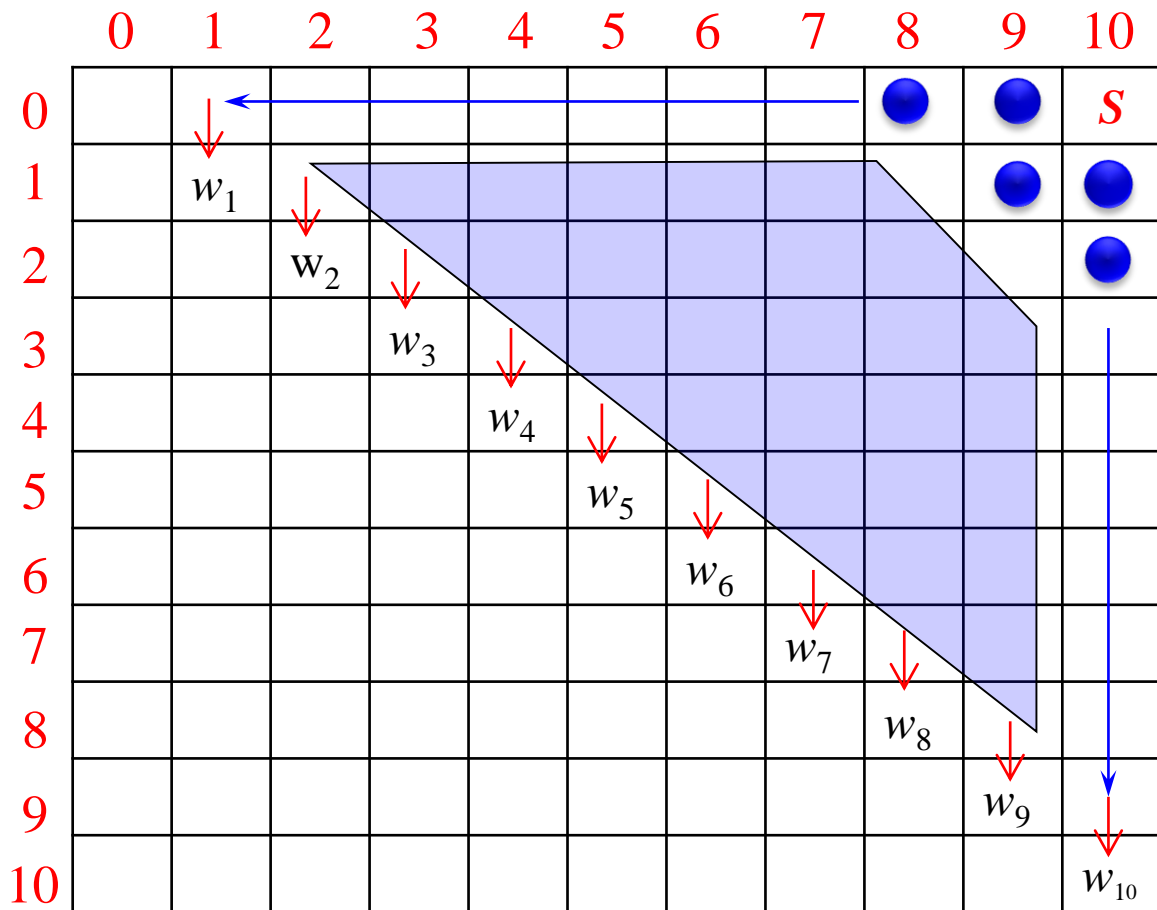
那么, $j = n-2 \dots 0$

$$i = 1 \dots n - j$$

• • • • •

$$j = 8$$
$$i = 1, \quad \beta_{i(i+j)} = \beta_{1,9}$$
$$i = 2, \quad \beta_{i(i+j)} = \beta_{2,10}$$
 $j = 7$
$$i = 1, \quad \beta_{i(i+j)} = \beta_{1,8}$$
$$i = 2, \quad \beta_{i(i+j)} = \beta_{2,9}$$
$$i = 3, \quad \beta_{i(i+j)} = \beta_{3,10}$$

• • • • •





8.4 基于PCFG的分析法

◆ 求解问题2: 最佳搜索- Viterbi 算法

● 定义:

Viterbi 变量 $\gamma_{ij}(A)$ 是由非终结符 A 推导出语句 W 中子字符串 $w_i w_{i+1} \dots w_j$ 的最大概率。

变量 $\psi_{i,j}$ 用于记忆字符串 $w_1 w_2 \dots w_n$ 的 Viterbi 句法分析结果。

注意: γ 与内向算法中的 α 的区别。

8.4 基于PCFG的分析法

●Viterbi 算法描述:

输入: 文法 $G(S)$, 语句 $W = w_1 w_2 \dots w_n$;

输出: $\gamma_{1n}(S)$

(1) 初始化: $\gamma_{ii}(A) = p(A \rightarrow w_i) \quad A \in V_N, 1 \leq i \leq n$

(2) 归纳计算: $j=1..n, i=1..n-j$, 重复下列计算:

$$\gamma_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

$$\psi_{i(i+j)}(A) = \max_{B, C \in V_N; i \leq k \leq i+j} p(A \rightarrow BC) \gamma_{ik}(B) \gamma_{(k+1)(i+j)}(C)$$

(3) 终结: $p(S \xRightarrow{*} w_1 w_2 \dots w_n) = \gamma_{1n}(S)$



8.4 基于PCFG的分析法

◆ 求解问题3: 参数估计- 内外向算法

● 基本思路

如果有大量已标注句法结构的训练语料，则可直接通过计算每个句法规则的使用次数，用最大似然估计方法计算 PCFG 规则的概率参数，即：

$$\hat{p}(N^j \rightarrow \zeta) = \frac{C(N^j \rightarrow \zeta)}{\sum_{\gamma} C(N^j \rightarrow \gamma)}$$



8.4 基于PCFG的分析法

多数情况下，没有可利用的标注语料，只好借助EM (Expectation Maximization) 迭代算法估计PCFG的概率参数。

初始时随机地给参数赋值，得到语法 G_0 ，依据 G_0 和训练语料，得到语法规则使用次数的期望值，以期望次数运用于最大似然估计，得到语法参数新的估计值，由此得到新的语法 G_1 ，由 G_1 再次得到语法规则的使用次数的期望值，然后又重新估计语法参数。循环这个过程，语法参数将收敛于最大似然估计值。

8.4 基于PCFG的分析法

● 算法设计

给定 CFG G 和训练数据 $W = w_1 w_2 \dots w_n$, 句法规则 $A \rightarrow BC$ 使用次数的期望值为:

$$\begin{aligned} \text{Count}(A \rightarrow BC) &= \sum_{1 \leq i \leq k \leq j \leq n} p(A_{ij}, B_{ik}, C_{(k+1)j} \mid w_1 L w_n, G) \\ &= \frac{1}{p(w_1 L w_n \mid G)} \sum_{1 \leq i \leq k \leq j \leq n} p(A_{ij}, B_{ik}, C_{(k+1)j}, w_1 L w_n \mid G) \\ &= \frac{1}{p(w_1 L w_n \mid G)} \sum_{1 \leq i \leq k \leq j \leq n} \beta_{ij}(A) p(A \rightarrow BC) \alpha_{ik}(B) \alpha_{(k+1)j}(C) \\ &\quad \dots \text{(E1)} \end{aligned}$$

8.4 基于PCFG的分析法

● 解释：给定语句 $W=w_1w_2 \dots w_n$,

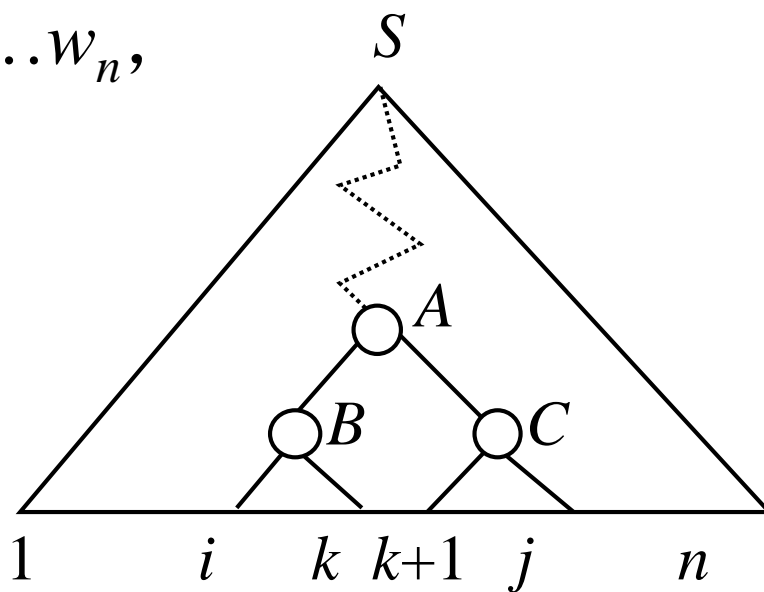
PCFG G 中产生式 $A \rightarrow BC$

在产生 W 的过程中被使用

次数的期望值为：在所有可能的情况下，即在条件：

$1 \leq i \leq k \leq j \leq n$ 下， W 的句

法分析结构中 $w_i \dots w_k$ 由 B 导出， $w_{k+1} \dots w_j$ 由 C 导出， $w_i \dots w_j$ 由 A 导出的概率总和。





8.4 基于PCFG的分析法

类似地，语法规则 $A \rightarrow a$ 的使用次数的期望值为：

$$\begin{aligned} \text{Count}(A \rightarrow a) &= \sum_{1 \leq i \leq n} p(A_{ii} \mid w_1 L w_n, G) \\ &= \frac{1}{p(w_1 L w_n \mid G)} \sum_{1 \leq i \leq n} p(A_{ii}, w_1 L w_n \mid G) \\ &= \frac{1}{p(w_1 L w_n \mid G)} \sum_{1 \leq i \leq n} \beta_{ii}(A) p(A \rightarrow a) \delta(a, w_i) \end{aligned}$$

... (E2)



8.4 基于PCFG的分析法

G 的参数可由下面的公式重新估计：

$$\hat{p}(A \rightarrow \mu) = \frac{Count(A \rightarrow \mu)}{\sum_{\mu} Count(A \rightarrow \mu)} \quad \dots (E3)$$

其中， μ 要么为终结符号，要么为两个非终结符号串，即 $A \rightarrow \mu$ 为乔姆斯基语法范式要求的两种形式。



8.4 基于PCFG的分析法

- 内外向算法描述:

(1) 初始化: 随机地给 $p(A \rightarrow \mu)$ 赋值, 使得

$$\sum_{\mu} p(A \rightarrow \mu) = 1, \text{ 由此得到语法 } G_0。 \text{ 令 } i=0;$$

(2) EM迭代:

E-步: 由 G_i 根据公式(E1)和(E2), 计算

期望值 $Count(A \rightarrow BC)$ 和 $Count(A \rightarrow a)$;

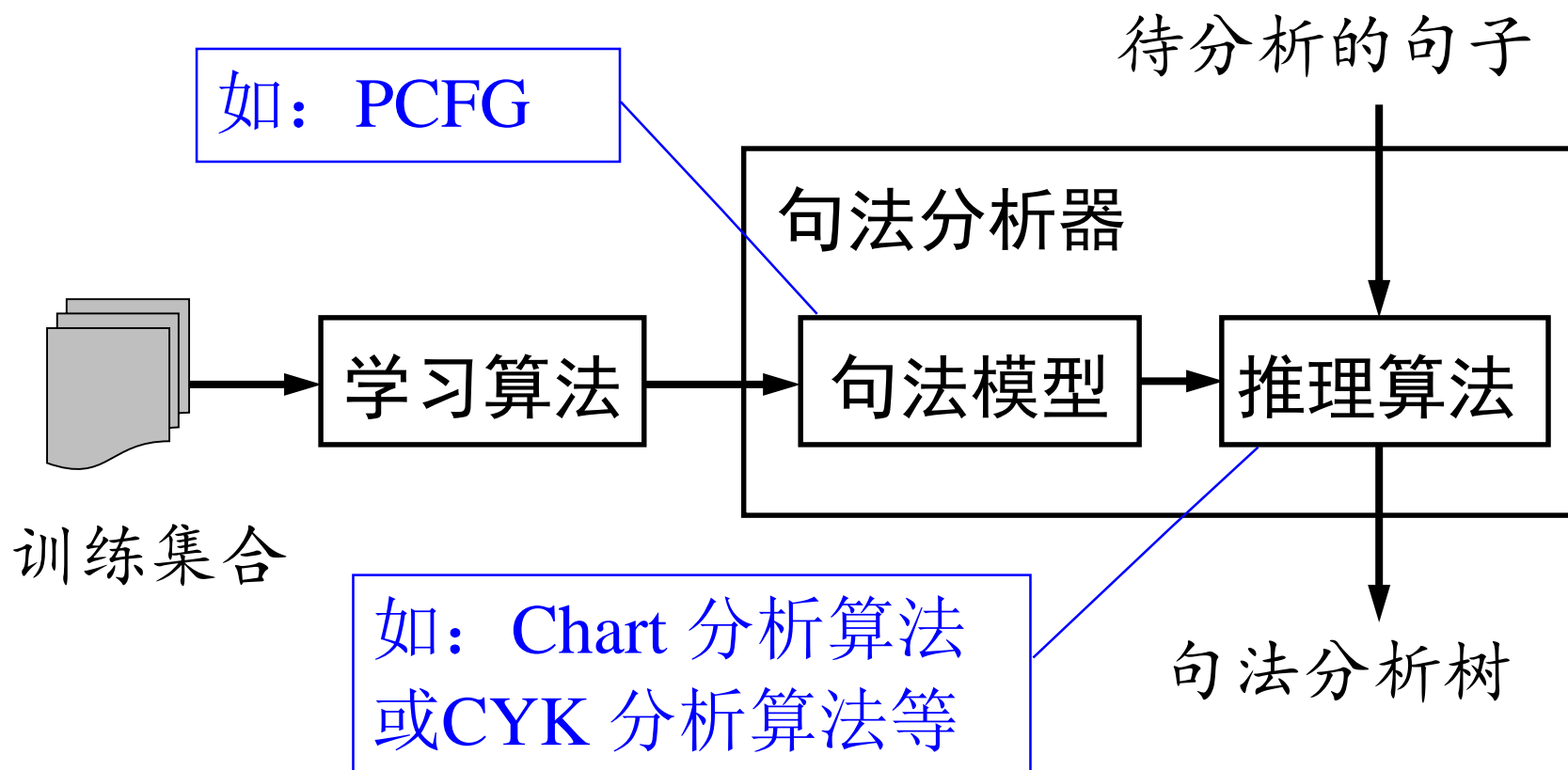
M-步: 用 E-步所得的期望值, 根据

公式(E3)重新估计 $p(A \rightarrow \mu)$, 得到 G_{i+1} 。

(3) 循环: $i=i+1$, 重复EM步骤, 直至 $p(A \rightarrow \mu)$ 值收敛。

8.4 基于PCFG的分析法

◆统计句法分析器实现的一般方法



8.4 基于PCFG的分析法

◆PCFG Parser 执行过程示例

给定如下 PCFG $G(S)$: $V_N = \{S, NP, VP, PP, DT, Vi, Vt, NN, IN\}$;
 $V_T = \{\text{sleeps, saw, man, woman, dog, telescope, the, with, in}\}$;
规则集 P :

(1) $S \rightarrow NP VP$	1.0	(7) $PP \rightarrow IN NP$	1.0	(13) $NN \rightarrow \text{dog}$	0.5
(2) $VP \rightarrow Vi$	0.3	(8) $Vi \rightarrow \text{sleeps}$	1.0	(14) $DT \rightarrow \text{the}$	0.5
(3) $VP \rightarrow Vt NP$	0.4	(9) $Vt \rightarrow \text{saw}$	1.0	(15) $DT \rightarrow \text{a}$	0.5
(4) $VP \rightarrow VP PP$	0.3	(10) $NN \rightarrow \text{man}$	0.1	(16) $IN \rightarrow \text{with}$	0.6
(5) $NP \rightarrow DT NN$	0.8	(11) $NN \rightarrow \text{woman}$	0.1	(17) $IN \rightarrow \text{in}$	0.4
(6) $NP \rightarrow NP PP$	0.2	(12) $NN \rightarrow \text{telescope}$	0.3		

输入句子: the man saw the dog with a telescope

DT 0.5							
[0,1]	[0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]
the	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]
	man	[2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]
		saw	[3,4]	[3,5]	[3,6]	[3,7]	[3,8]
			the	[4,5]	[4,6]	[4,7]	[4,8]
				dog	[5,6]	[5,7]	[5,8]
					with	[6,7]	[6,8]
						a	[7,8]
							telescope

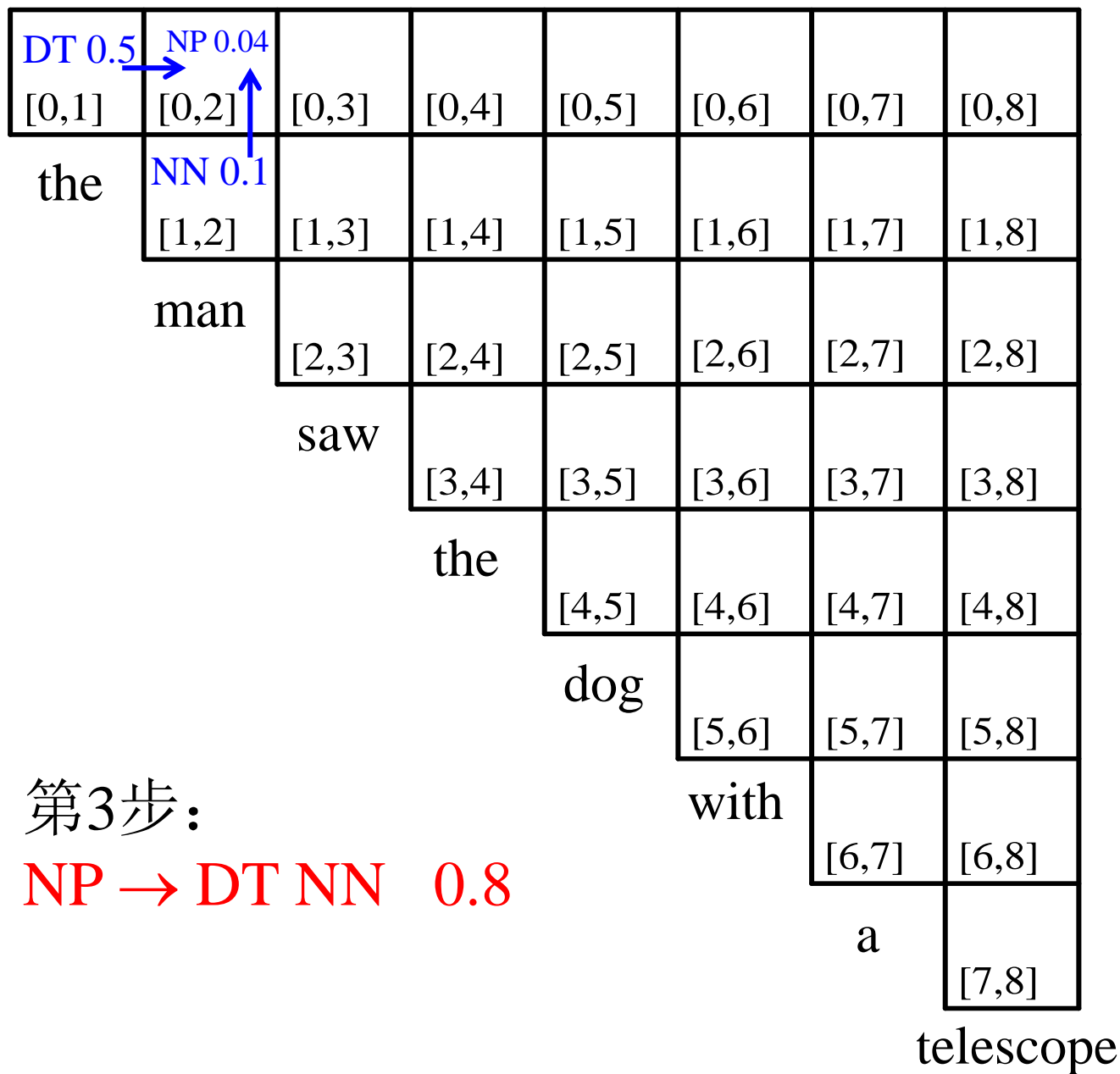
第1步:

DT → the 0.5

DT 0.5							
[0,1]	[0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]
the	NN 0.1						
	[1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]
	man						
		[2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]
		saw					
			[3,4]	[3,5]	[3,6]	[3,7]	[3,8]
			the				
				[4,5]	[4,6]	[4,7]	[4,8]
				dog			
					[5,6]	[5,7]	[5,8]
					with		
						[6,7]	[6,8]
						a	
							[7,8]
							telescope

第2步:

NN → man 0.1



DT 0.5 [0,1]	NP 0.04 [0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]
the	NN 0.1 [1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]
man	Vt 1.0 [2,3]	[2,4]	[2,5]	[2,6]	[2,7]	[2,8]	
saw	DT 0.5 [3,4]	[3,5]	[3,6]	[3,7]	[3,8]		
the	NN 0.5 [4,5]	[4,6]	[4,7]	[4,8]			
dog	[5,6]	[5,7]	[5,8]				
with	[6,7]	[6,8]					
a	[7,8]						

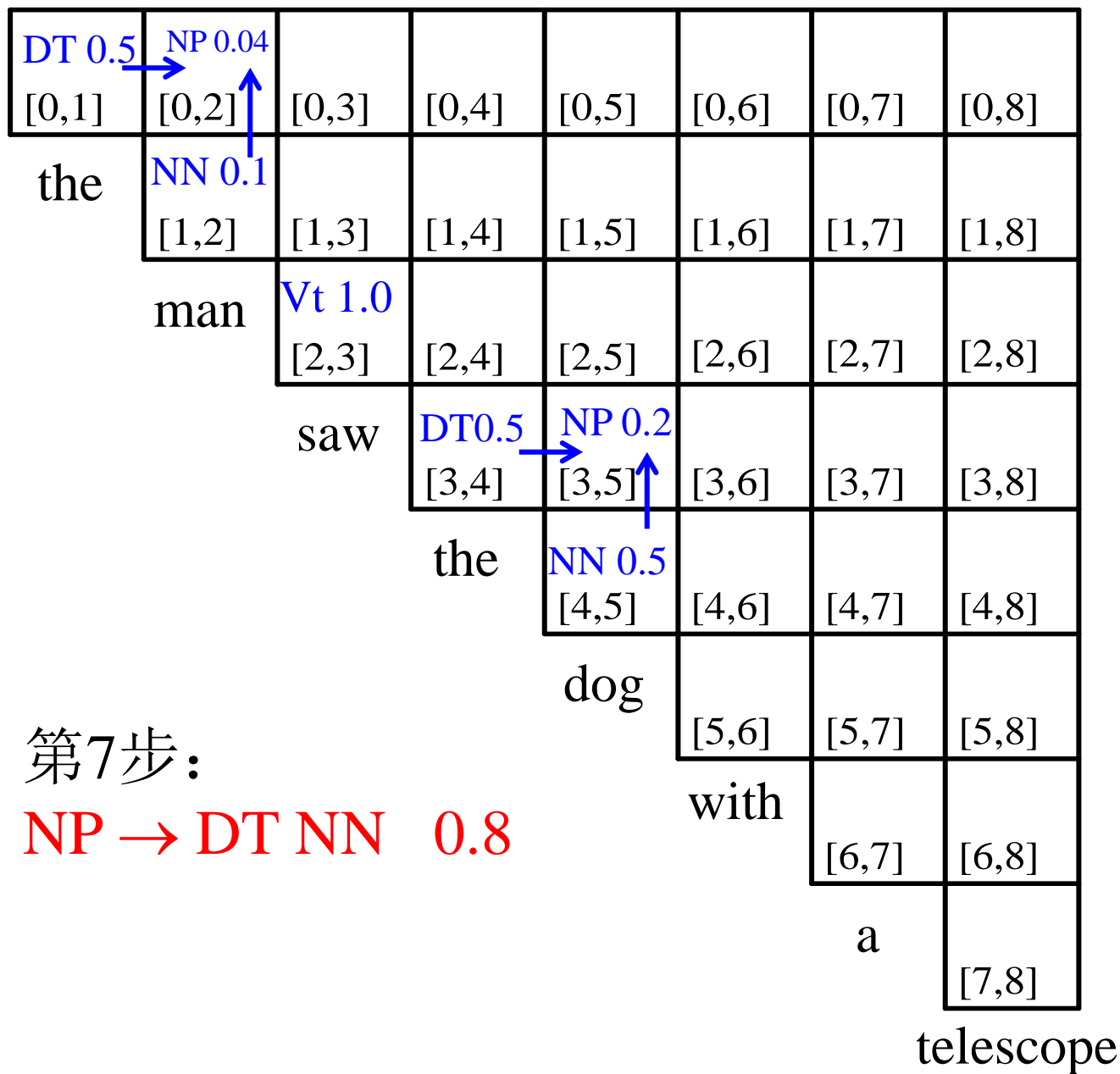
第4~6步:

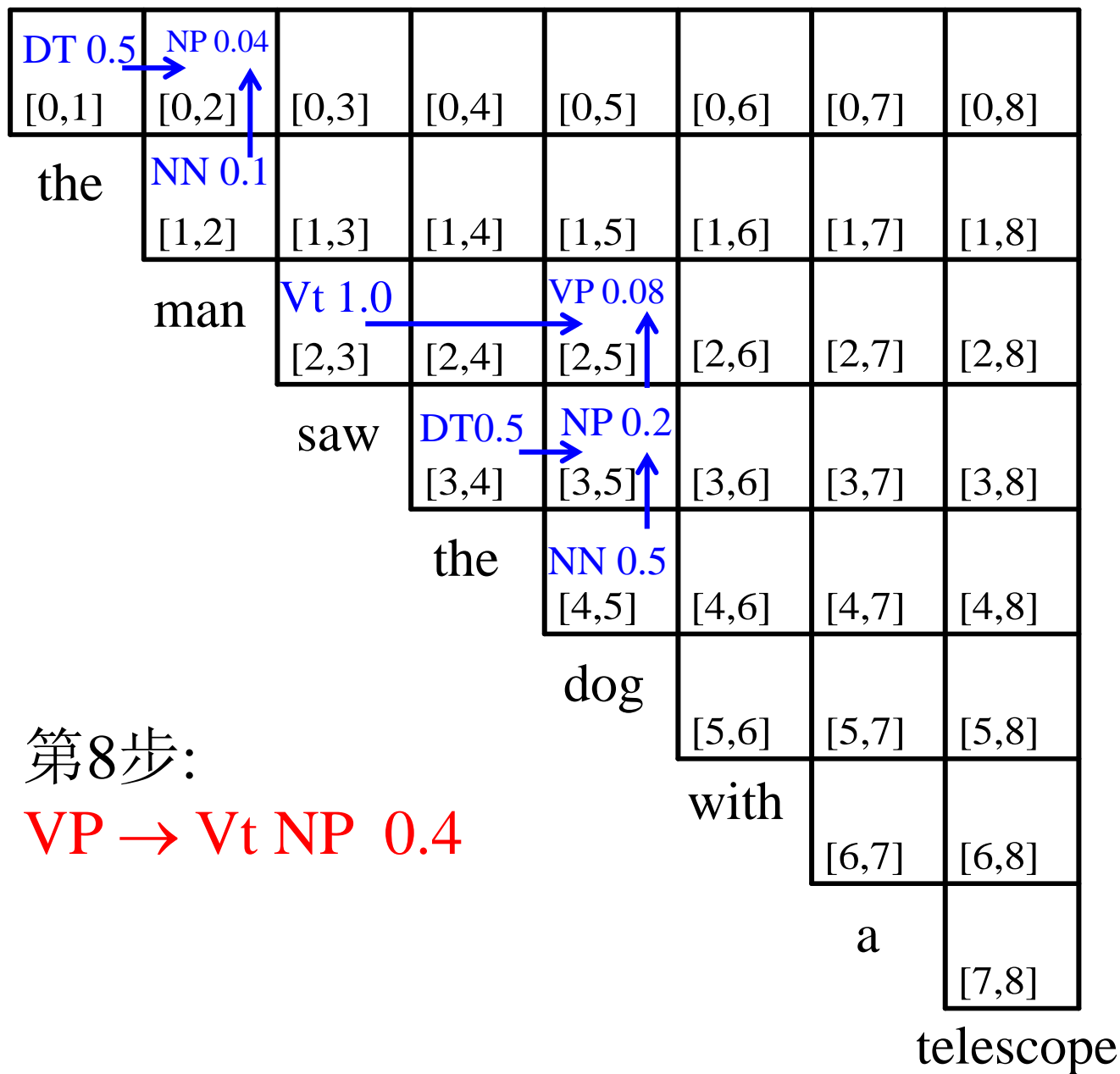
Vt → saw 1.0

DT → the 0.5

NN → dog 0.5

telescope





DT 0.5 [0,1]	NP 0.04 [0,2]	[0,3]	[0,4]	[0,5]	[0,6]	[0,7]	[0,8]
the	NN 0.1 [1,2]	[1,3]	[1,4]	[1,5]	[1,6]	[1,7]	[1,8]
man	Vt 1.0 [2,3]	[2,4]	VP 0.08 [2,5]	[2,6]	[2,7]	[2,8]	
saw	DT 0.5 [3,4]	NP 0.2 [3,5]	[3,6]	[3,7]	[3,8]		
the	NN 0.5 [4,5]	[4,6]	[4,7]	[4,8]			
dog	IN 0.6 [5,6]	[5,7]	[5,8]				
with	DT 0.5 [6,7]	[6,8]					
a	NN 0.3 [7,8]						

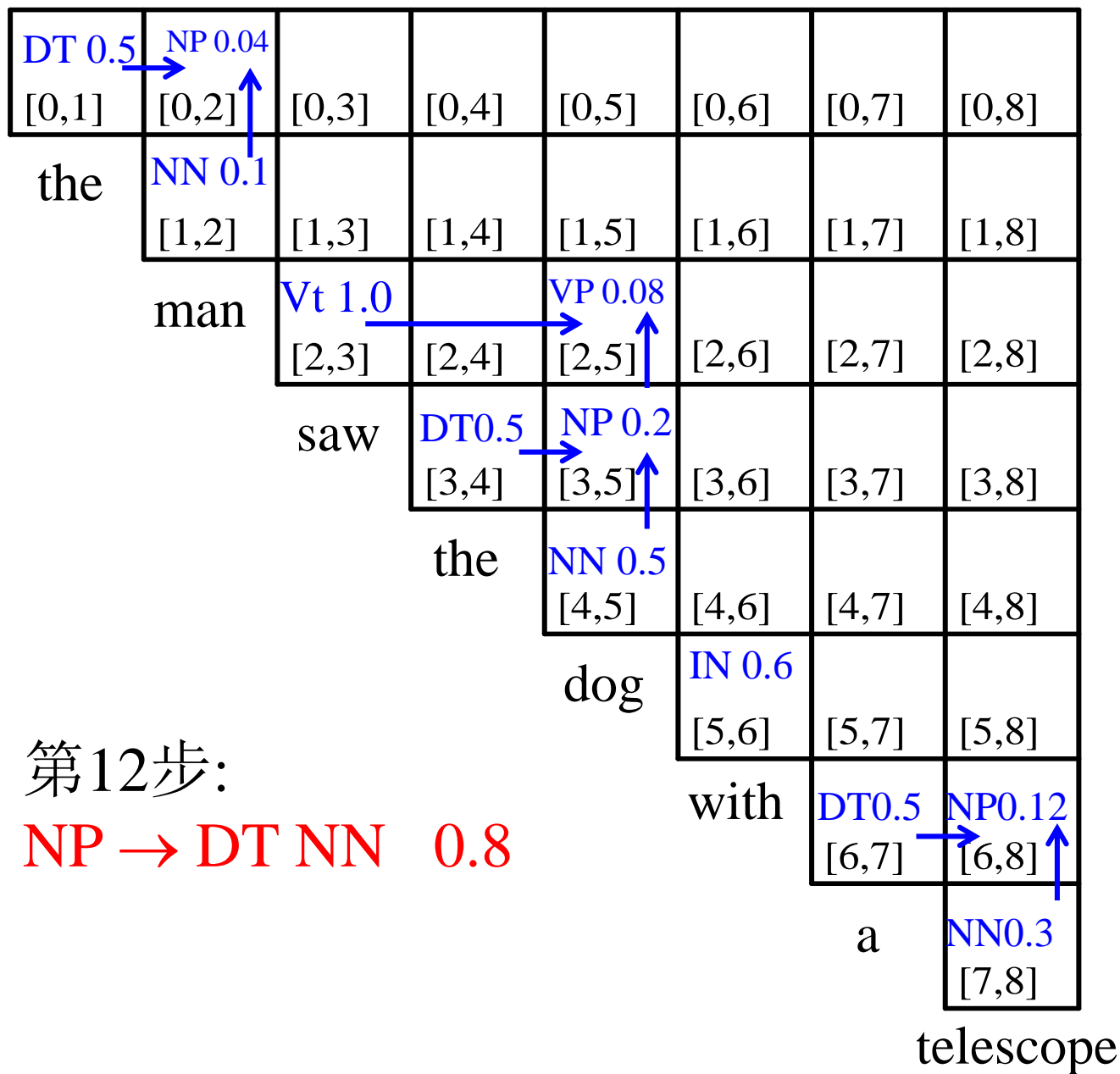
第9~11步:

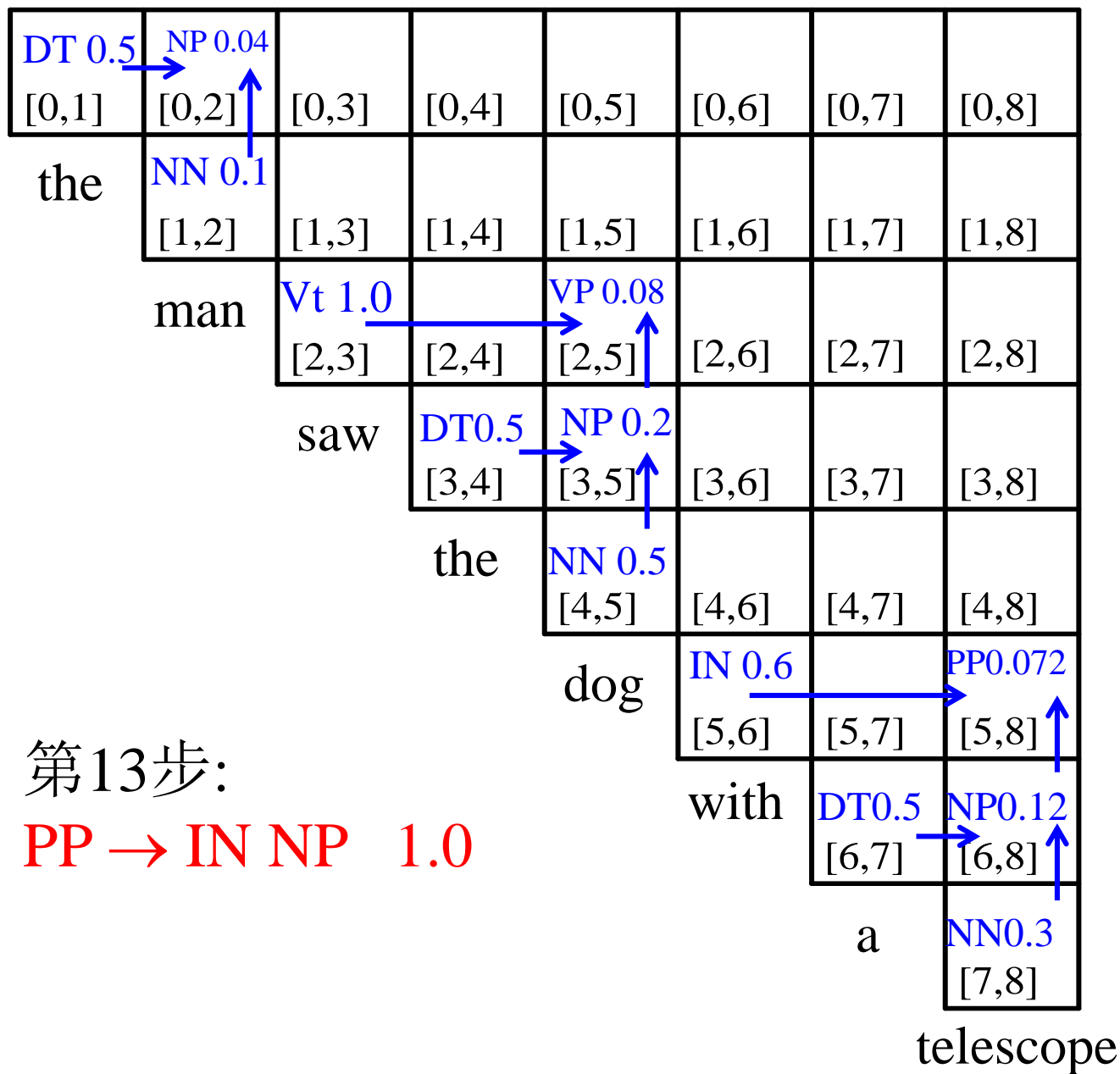
IN → with 0.6

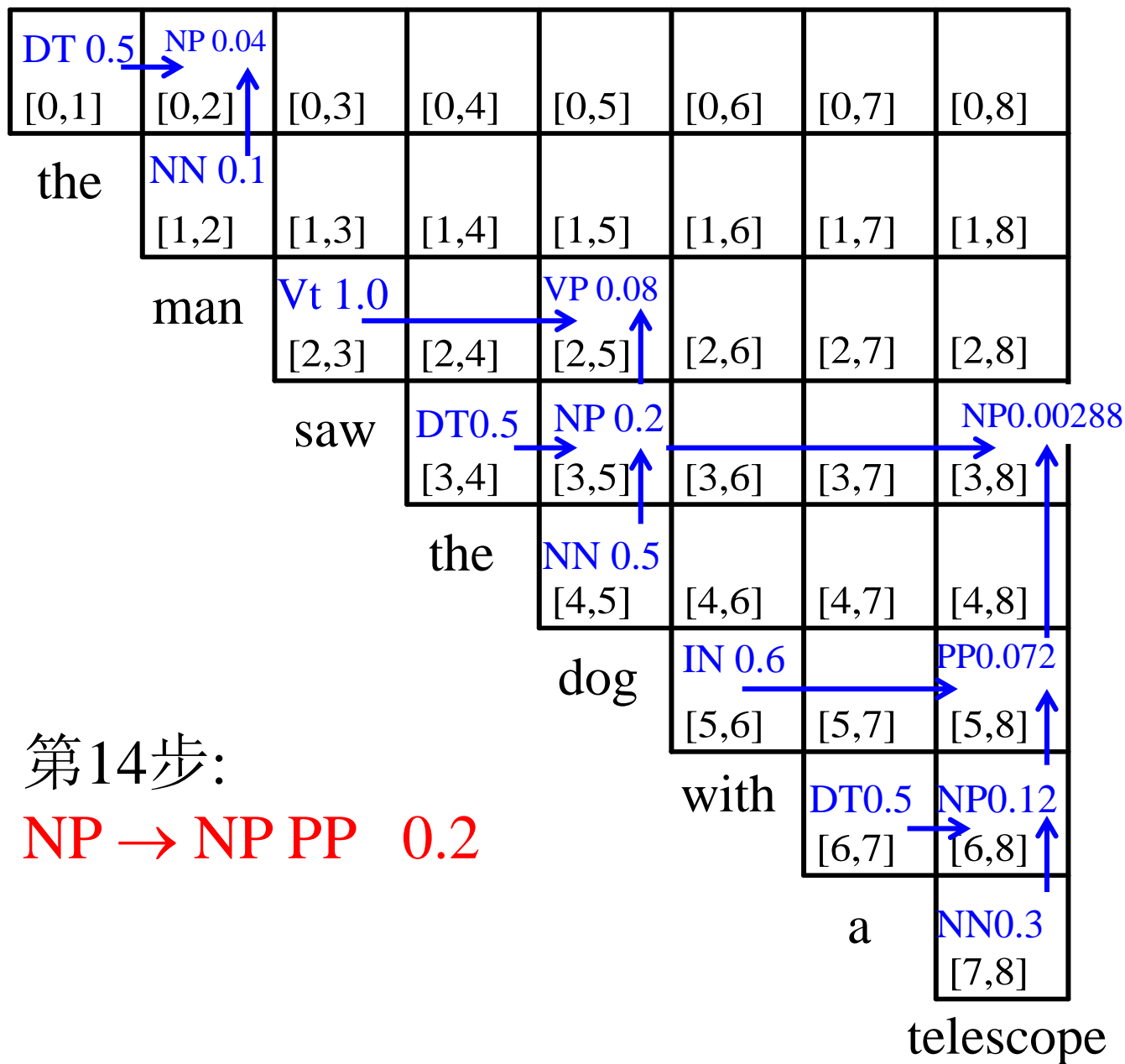
DT → a 0.5

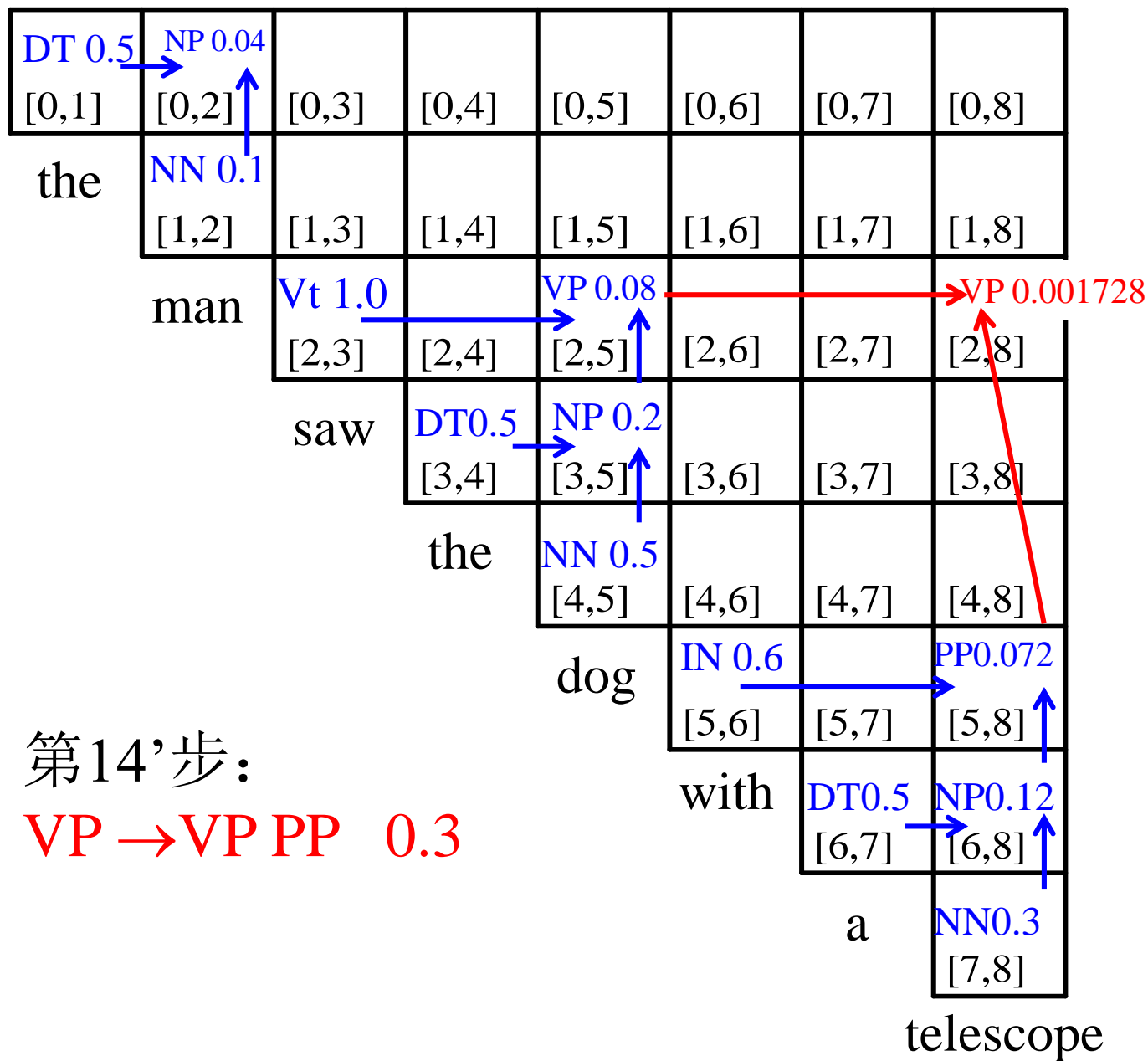
NN → telescope 0.3

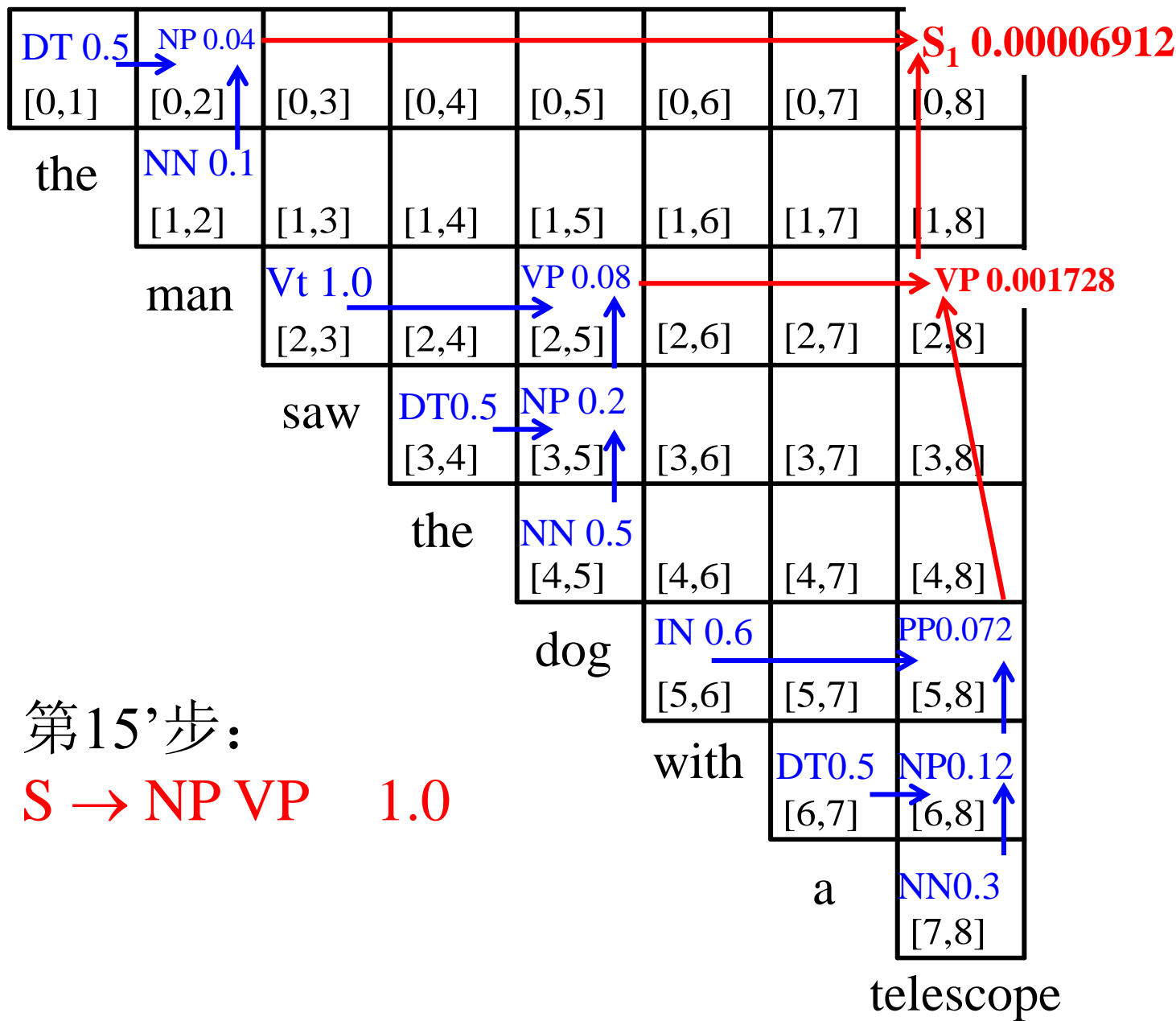
telescope

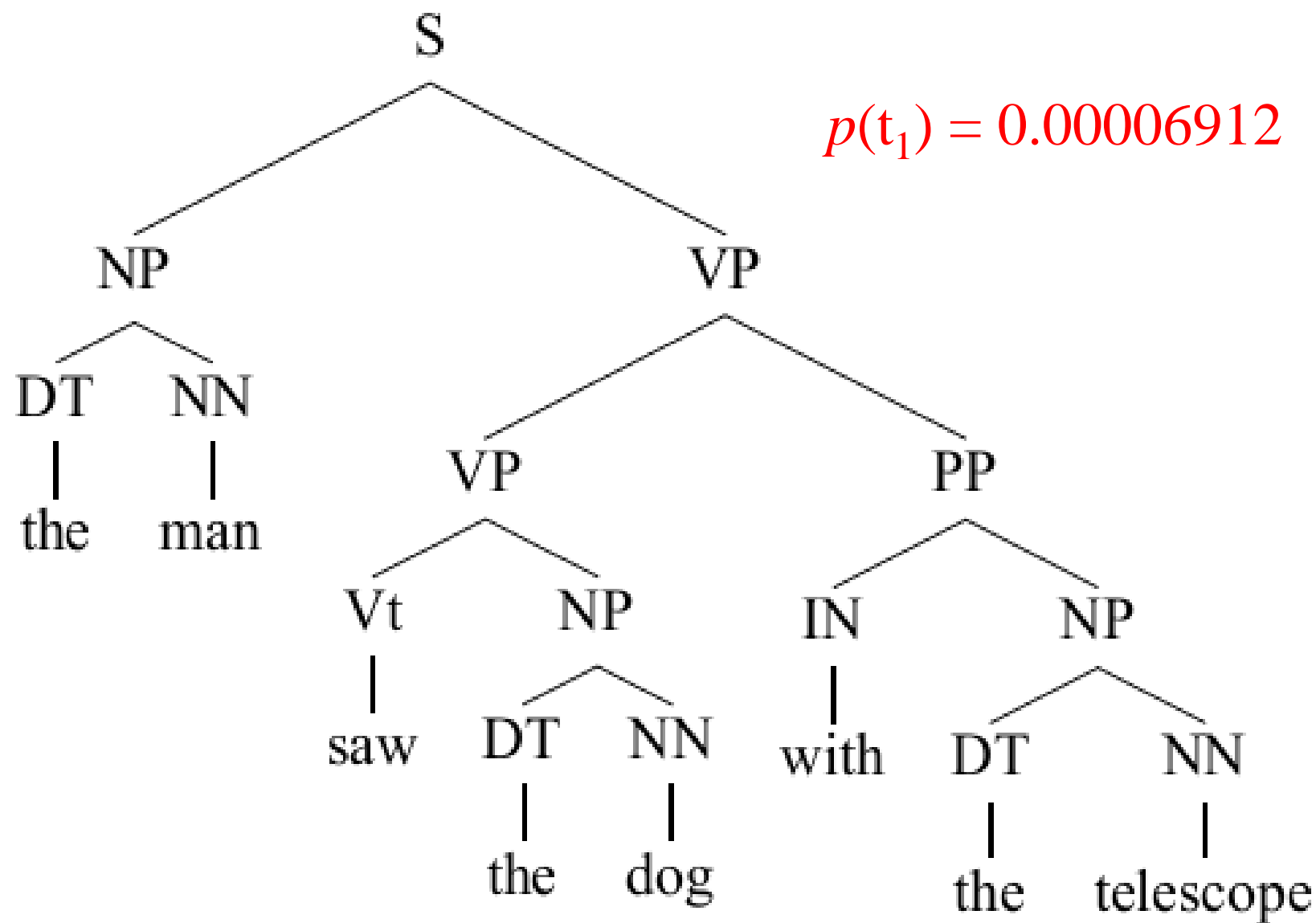




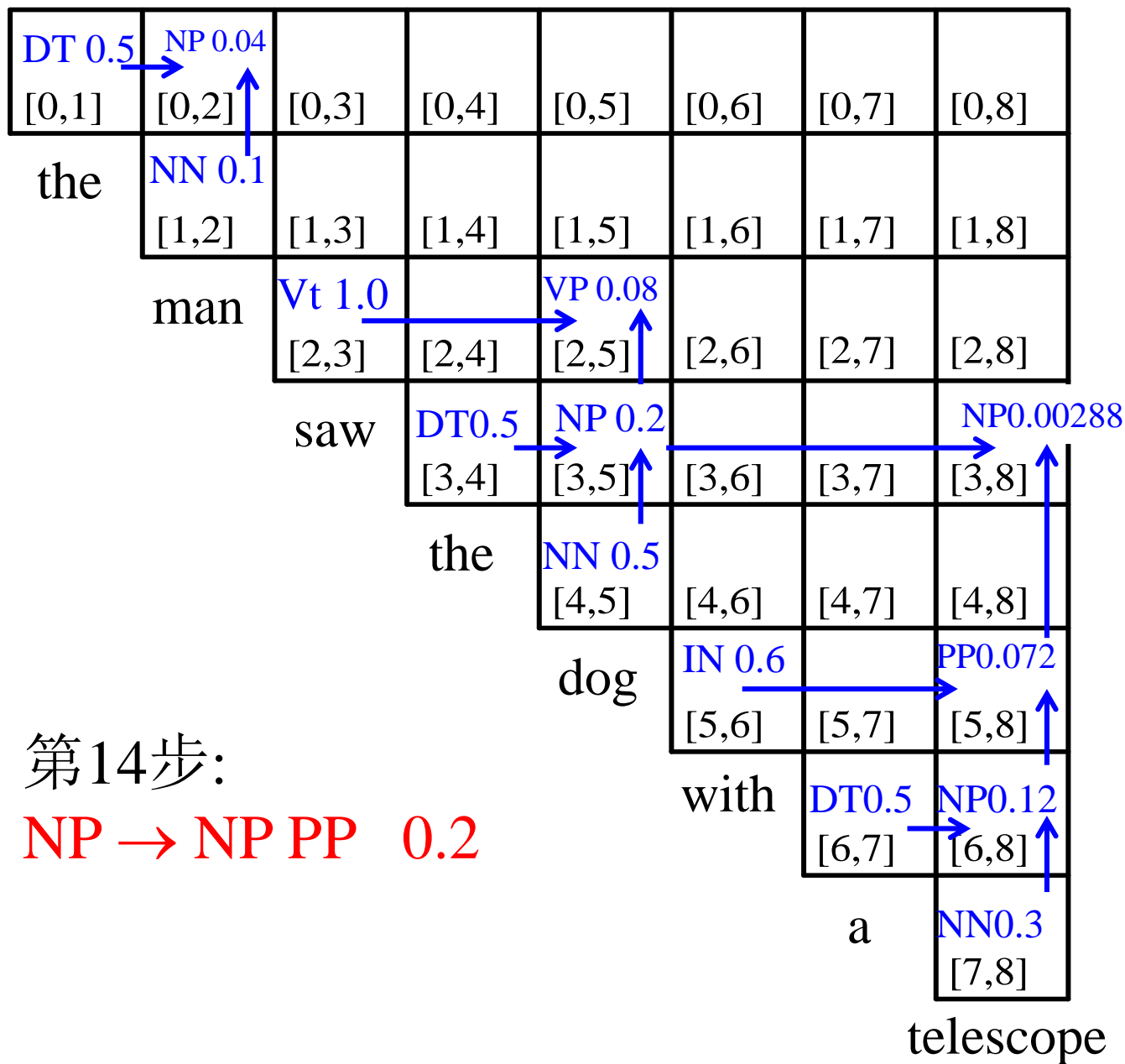


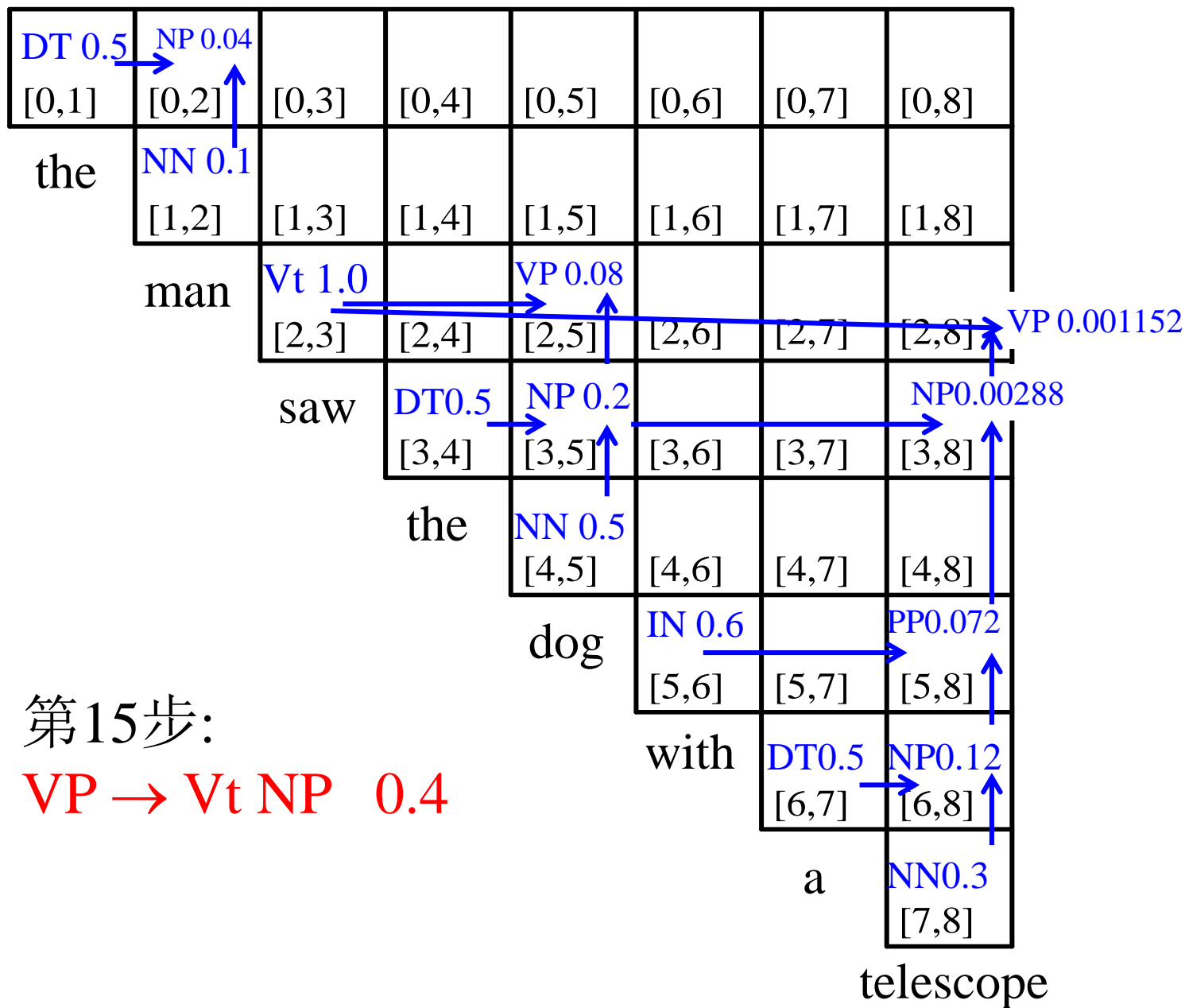


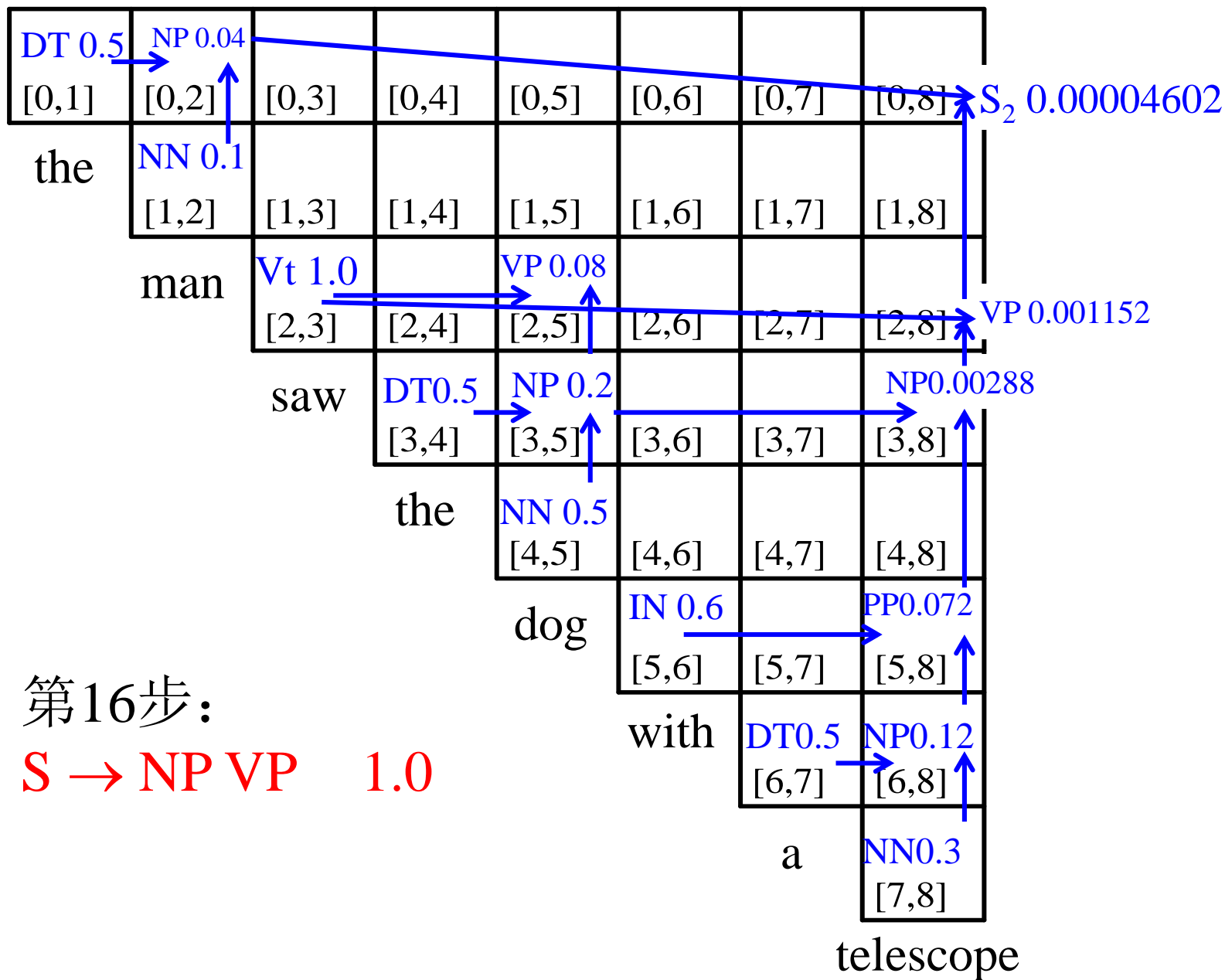


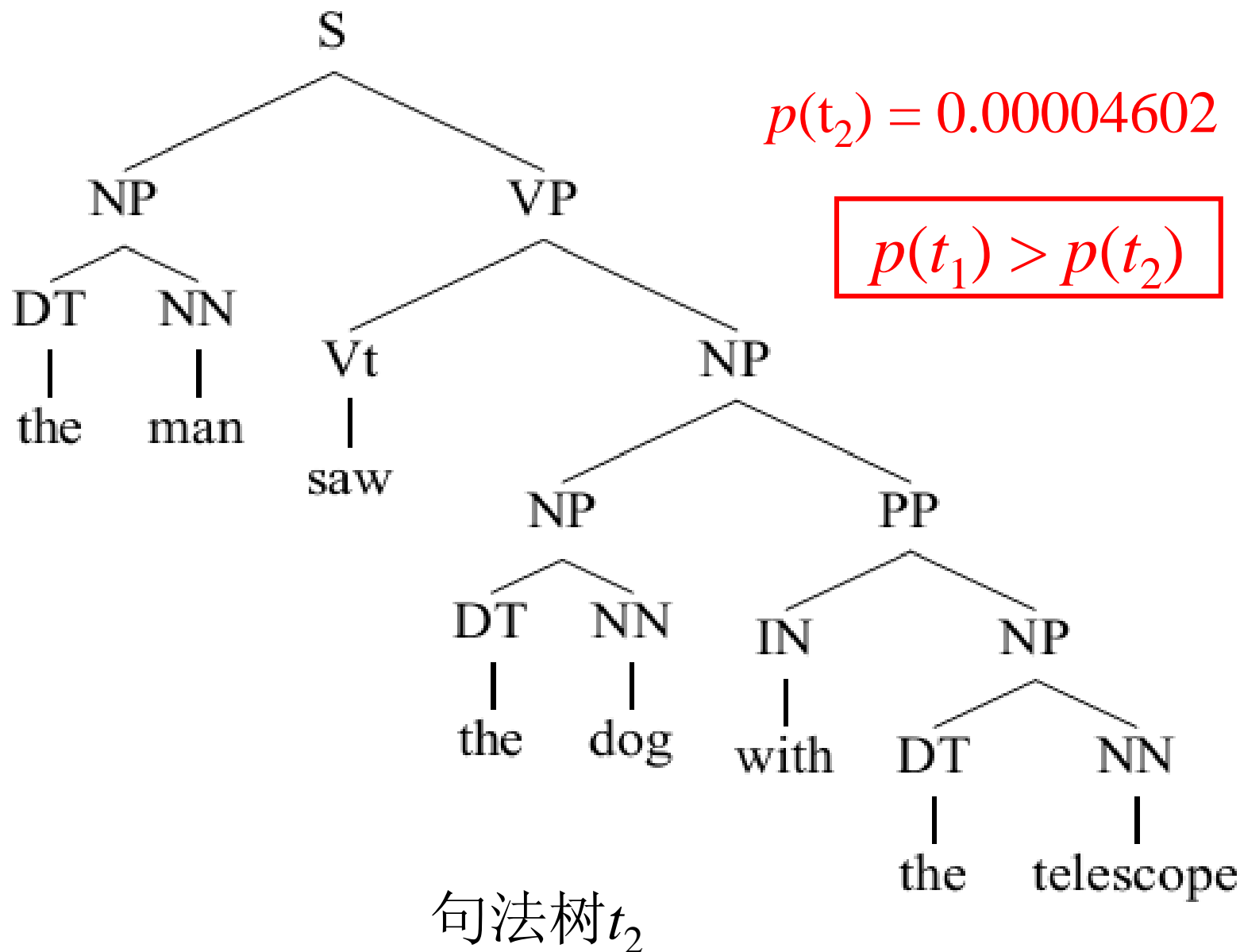


句法树 t_1











8.4 基于PCFG的分析法

◆ PCFG 的评价

● 优点:

- 可利用概率进行子树剪枝，减少分析过程的搜索空间，加快分析效率；
- 可以定量地比较两个句法分析器的性能。

● 弱点:

- 分析树的概率计算条件比较苛刻，甚至不够合理。
-



短语结构分析

8.1 概述

8.2 线图分析法

8.3 CYK分析法

8.4 基于PCFG的分析法

 8.5 句法分析性能评估

8.6 局部句法分析



8.5 句法分析性能评估

◆ 句法分析器性能评价指标

目前广泛使用的句法分析器性能评价指标是PARSEVAL评测提出的，主要包括如下几个：

- **精度(precision)**：句法分析结果中正确的短语个数所占的比例，即分析结果中与标准分析树（答案）中的短语相匹配的个数占分析结果中所有短语个数的比例，即：

$$P = \frac{\text{分析得到的正确的短语个数}}{\text{分析得到的所有的短语个数}} \times 100\%$$



8.5 句法分析性能评估

- 召回率(recall): 句法分析结果中正确的短语个数占标准分析树中全部短语个数的比例, 即:

$$R = \frac{\text{分析得到的正确的短语个数}}{\text{标准树库中(答案)的短语个数}} \times 100\%$$

- F-measure:
$$F = \frac{(\beta^2 + 1) \times P \times R}{\beta^2 \times P + R} \times 100\%$$

一般地, $\beta=1$, 称作 $F1$ 测度。



8.5 句法分析性能评估

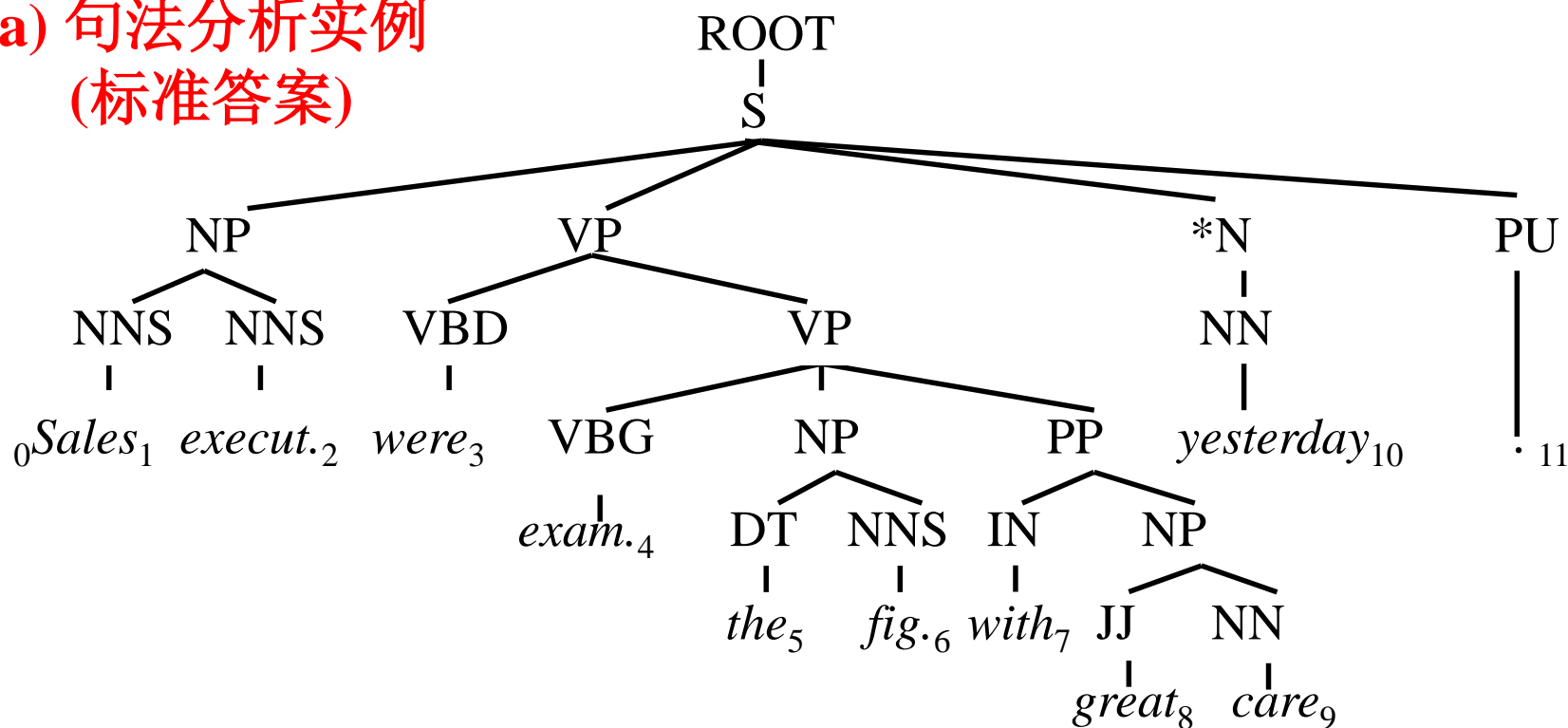
- 交叉括号数 (crossing brackets): 一棵分析树中与其他分析树中边界相交叉的成分个数的平均值。

分析树中除了词性标注符号以外的其他非终结符节点采用如下标记格式: **XP-(起始位置: 终止位置)**。其中, **XP**为短语名称; **(起始位置: 终止位置)**为该节点的跨越范围, **起始位置**指该节点所包含的子节点的起始位置, **终止位置**为该节点所包含的子节点的终止位置。在计算PARSEVAL指标时, 通常需要计算分析结果与标准分析树之间括号匹配的数目或括号交叉的数目。

例如, 下面的图(a)为句子 “*Sales executives were examining the figures with great care yesterday.*” 的正确分析树(答案标准)。

8.5 句法分析性能评估

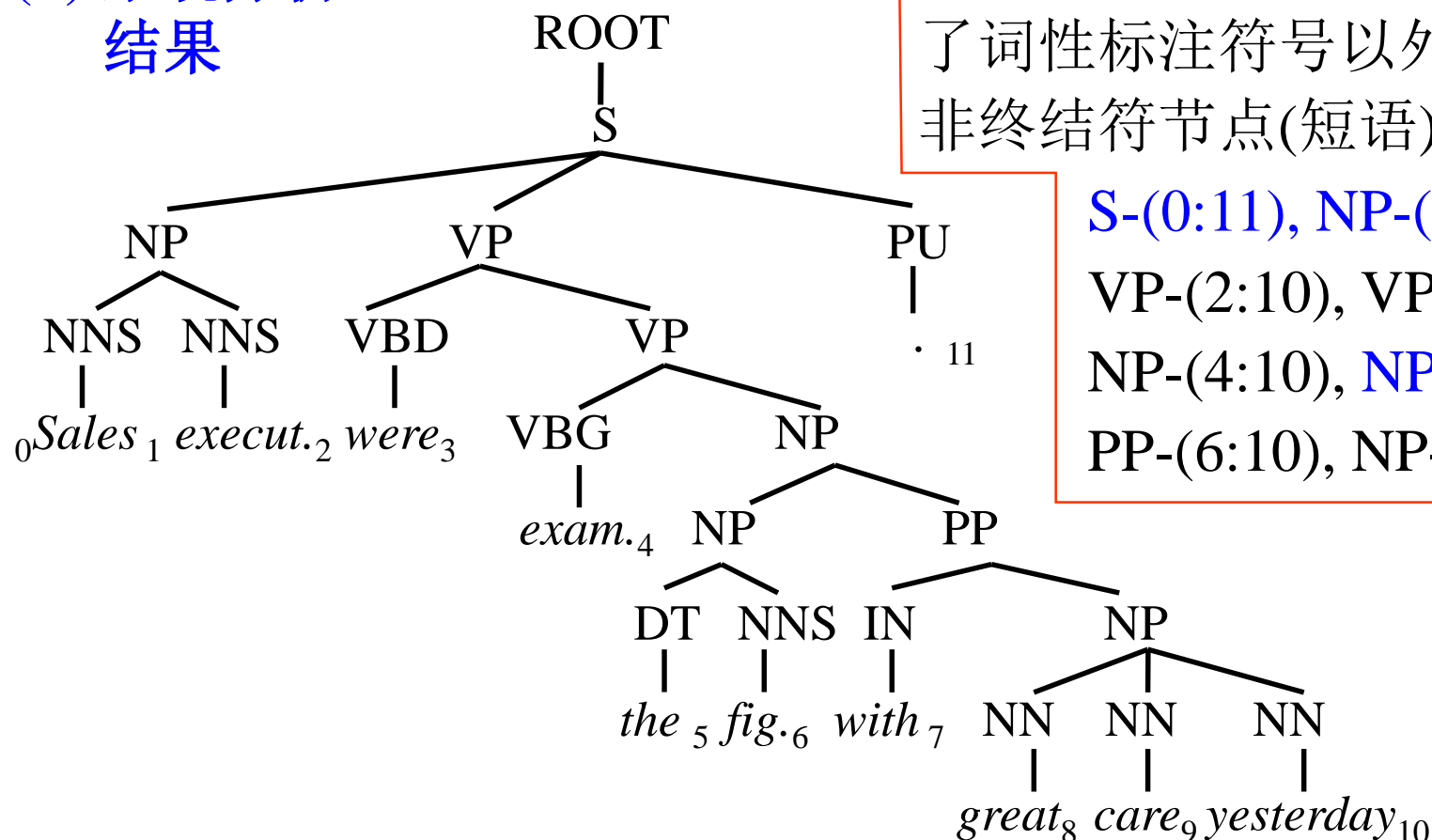
(a) 句法分析实例 (标准答案)



在标准答案树中，除了词性标注符号以外(即除了叶子节点和其直接父节点以外)的其他非终结符节点(短语)有：S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), NP-(4:6), PP-(6:9), NP-(7:9), *NP-(9:10)。

8.5 句法分析性能评估

(b) 系统分析结果



在系统输出的分析树中，除了词性标注符号以外的其他非终结符节点(短语)有：

S-(0:11), NP-(0:2),
VP-(2:10), VP-(3:10),
NP-(4:10), NP-(4:6),
PP-(6:10), NP-(7:10)。

8.5 句法分析性能评估

标准答案: S-(0:11), NP-(0:2), VP-(2:9), VP-(3:9), NP-(4:6), PP-(6:9), NP-(7:9), *NP-(9:10)

系统结果: S-(0:11), NP-(0:2), VP-(2:10), VP-(3:10), NP-(4:10), NP-(4:6), PP-(6:10), NP-(7:10)

只有这3个短语与标准答案完全一样，因此，

$$\text{Precision} = \frac{3}{8} \times 100\% = 37.5\%$$

$$\text{Recall} = \frac{3}{8} \times 100\% = 37.5\%$$

注：图(a)中加*号的一元的节点(*NP)在计算时应该被排除在外，但在这里也被包括进来了。



8.5 句法分析性能评估

- 另外两个指标:

- ① 词性标注的准确率(tagging accuracy)

在本例的句子分析树中，11个词中除了 *great* 被错误地标注以外，其他的词性标注均为正确的，因此，词性标注准确率为 $10/11 = 90.9\%$

- ② 交叉括号数(crossing brackets)和交叉准确率(crossing accuracy)

在本例中不存在交叉括号，因此，交叉括号数为0，交叉准确率为100%。

(说明：通常是对应位置的短语比较。该例中存在范围覆盖，但不存在交叉括号情况。)



8.5 句法分析性能评估

PARSEVAL评测方法的区分能力不是很强，而且在某些特殊情况下计算出的正确率和召回率存在较大的偏差。因此，有关专家提出了一些其他的评测方法，如基于依存结构的评测方法[D. Lin, 1995]、基于语法关系的评测方法[J. Carroll, 1998]等。

Lin, Dekang. 1995. A dependency-based method for evaluating broad-coverage parsers. In *Proceedings of the IJCAI-95*, Montreal, Canada, pp 1420-1425

Carroll, John, Ted Briscoe, and Antonio Sanfilippo. 1998. Parser Evaluation: a Survey and a New Proposal. In *Proceedings of the First Conference on Linguistic Resources*, pp 447-455



部分公开的短语结构分析器

- ✧ Berkeley Parser: <http://nlp.cs.berkeley.edu/Main.html#Parsing>
 - ✧ Stanford Parser:
<http://nlp.stanford.edu/downloads/lex-parser.shtml>
 - ✧ Collins Parser: <http://people.csail.mit.edu/mcollins/code.html>
 - ✧ Charniak Parser: <http://www.cs.brown.edu/people/ec/#software>
 - ✧ Bikel Parser:
<http://www.cis.upenn.edu/~dbikel/software.html#stat-parser>
 - ☀ Oboe Parser (可执行程序):
<http://www.openpr.org.cn/index.php/NLP-Toolkit-for-Natural-Language-Processing/>
-

8.5 句法分析性能评估

◆性能现状

- 2012年前后我们利用宾州树库对Berkeley Parser 进行了测试:

语种	训练集	开发集	测试集
汉语(句子数)	18089	350	348
英文(句子数)	39832	1700	2416

- 性能表现: Charniak Parser (汉语) F1=85.97% (≤ 40), 82.41%。

语种	句子类型	正确率(%)	召回率(%)	F1
英语	长度小于等于40个词	90.7	90.5	90.6
	所有的句子	90.2	8	90.5
汉语	长度小于等于40个字	86.9	85.7	86.3
	所有的句子	84.8	81.9	83.32



8.5 句法分析性能评估

- 汉语句子的分析性能低于英文
- 在特定领域的分析性能急剧下降
 - 生物医药等专业领域 (67%左右)
 - 微博等非规范语句 (更差)
- 深度学习方法改善了句法分析的性能 (后面介绍)



短语结构分析

8.1 概述

8.2 线图分析法

8.3 CYK分析法

8.4 基于PCFG的分析法

8.5 句法分析性能评估

 8.6 局部句法分析



8.6 局部句法分析

S. Abney (1991) 提出了浅层句法分析(shallow parsing)的概念，也被称为局部（部分）句法分析(partial parsing)，或语块划分(chunking)，其目的是识别句子中某些结构相对简单的独立成分，如：非递归的名词短语、动词短语等。

浅层句法分析通常包括：①语块识别；②语块之间的依存关系分析。



8.6 局部句法分析

根据 S. Abney 对语块的解释，语块是介于词和句子之间的具有**非递归特征**的核心成分。S. Abney(1995) 对英语语块的定义包含三个层次：

- 词 (words)
- 非递归的名词短语 (NP)、动词词组 (VG)、副词短语 (DP) 和介词短语 (PP)
- 子句 (clause)

由于NP、VG 和 DP、PP属于不同的类别，因此，又将第二类进一步划分成“非递归的名词短语和动词词组”和“非递归的介词短语和副词短语”两类。通常非递归的名词短语和动词短语分别称为基本名词短语(Base NP)和基本动词短语(Base VP)。



8.6 局部句法分析

◆ Base NP 定义

基本名词短语指的是简单的、非嵌套的名词短语，不含有其他的子短语。它的主要特点有两个：短语的中心语为名词；短语中不含有其他的子项短语，并且base NP之间结构上是独立的。

Base NP 的形式化定义：

$\text{base NP} \rightarrow \text{base NP} + \text{base NP}$

$\text{base NP} \rightarrow \text{base NP} + \text{名词} \mid \text{名动词}$

$\text{base NP} \rightarrow \text{限定性定词} + \text{base NP} \mid \text{名词}$

$\text{base NP} \rightarrow \text{限定性定词} + \text{名词} \mid \text{名动词}$

$\text{限定性定词} \rightarrow \text{形容词} \mid \text{区别词} \mid \text{动词} \mid \text{名词} \mid \text{处所词} \mid \text{数量词} \mid \text{外文字串} \mid \text{数词和量词}$



8.6 局部句法分析

例1: [Pierre Vinken], [61 years] old, will join [the board] as [a non-executive director] on [Nov. 29].

例2: When [it] is [time] for [their biannual powwow], [the nation]'s [manufacturing titans] typically jet off to [the sunny confines] of [resort towns] like [Boca Raton and Hot Springs].

例3: 一个于 [半个 世纪] 之后 重新 聚集 在 “[西南 联大]” [旗帜] 下 的 [奉献 活动] 开始了!



8.6 局部句法分析

◆Base NP 识别方法

Base NP 识别可以简单地看作分类问题：判断一个短语的边界，识别该短语是 base NP 或非 base NP 两类。

●常用的方法：序列标注方法

●数据标注

两种标记方法：

- 括号分隔法 (the open/close bracketing)
 - IOB 标注方法 (IOB tagging)
-



8.6 局部句法分析

例如：在IOB标注方法中，字母 ‘B’ (Begin)表示当前词语位于base NP的开端，字母 ‘I’ (In) 表示当前词语在base NP内 (非短语首词语)，字母 ‘O’ (Out) 表示词语位于base NP 之外。例如：

外商/B 投资/I 成为/O 中国/B 外贸/I 重要/B 增长/I 。/O

与IOB方法类似的标注方法还有：IOE (In, Out, End) 表示方法，或者采用5个标志符号：O, B, E, I, S 等。

●分类器

➤SVM

➤CRFs

➤WINNOWN



8.6 局部句法分析

➤ 基于SVM 的识别方法

T. Kudo等(2003) 在利用 SVM 识别 base NP的系统(YamCha¹) 中, 主要使用了三类特征:

- 词: $w_{i-2} w_{i-1} w_i w_{i+1} w_{i+2}$
- 词性: $t_{i-2} t_{i-1} t_i t_{i+1} t_{i+2}$
- base NP 标志: $c_{i-2} c_{i-1}$

其中, w_i 为句子中位置 i 处的词, t_i 为词的词性, c_i 为待识别的第 i 个词的base NP标记。

¹<http://chasen.org/~taku/software/yamcha>

8.6 局部句法分析

YamCha 系统识别 base NP 过程示意图:

	COL: 0	COL: 1	TAG	
POS:-4	He	PRP	B-NP	
POS:-3	reckons	VBZ	B-VP	
POS:-2	the	DT	B-NP	Feature Sets
POS:-1	current	JJ	I-NP	
POS: 0	deficit	NN	I-NP	Eestimated TAG
POS:+1	will	MD	B-VP	
POS:+2	narrow	VB	I-NP	
POS:+3	to	TO	B-PP	

其中, POS列表示当前词(POS: 0)的前后词的位置; COL: 0 列表示给定句子; COL: 1列为给定句子中各个词对应的词类标记; TAG列为给定句子中的各个词被标记为base NP的标记。当要估计位置POS: 0处词的base NP标记时, 该词的前后各两个位置上的词和它们的词性标记, 以及前面两个词的base NP 标记共同作为被选取的特征。



8.6 局部句法分析

- 用于**base NP** 识别语料资源

- ✧ 英文：CoNLL-2000 (Conference on Computational Natural Language Learning)提供的《华尔街日报》语料

- 训练语料：15—18章，211,727个词

- <http://www.cnts.ua.ac.be/conll2000/chunking/train.txt.gz>

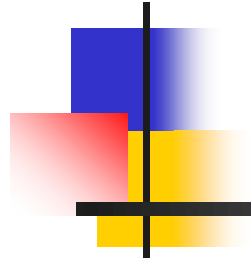
- 测试语料：第20章，47,377个词

- <http://www.cnts.ua.ac.be/conll2000/chunking/test.txt.gz>

- 工具Chunkkink 用于语料格式转换。

- <http://ilk.kub.nl/~sabine/chunklink/>

- ✧ 汉语：宾州 LDC 中文树库。



Thanks

ကျေးဇူးတင်ပါတယ်။