1

第5章 句法分析(II)

目录

3

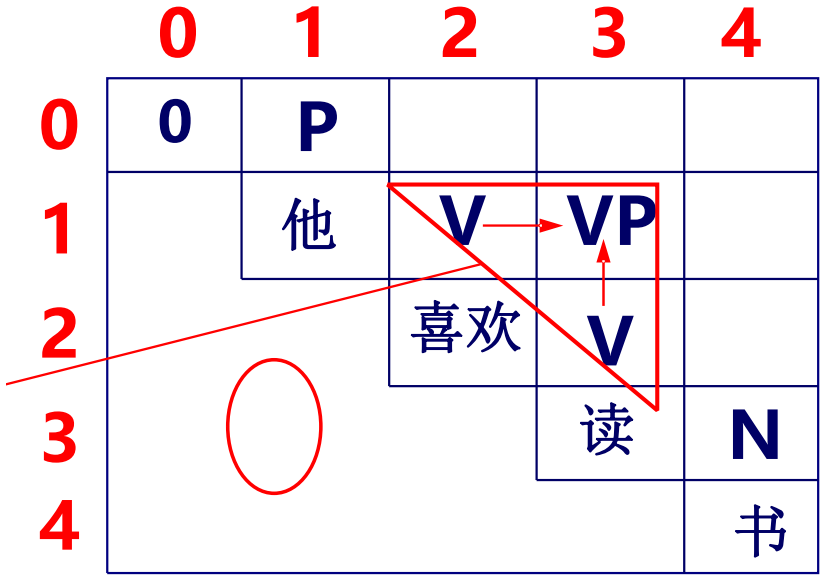
5.1 CYK分析算法

5.1 CYK分析算法

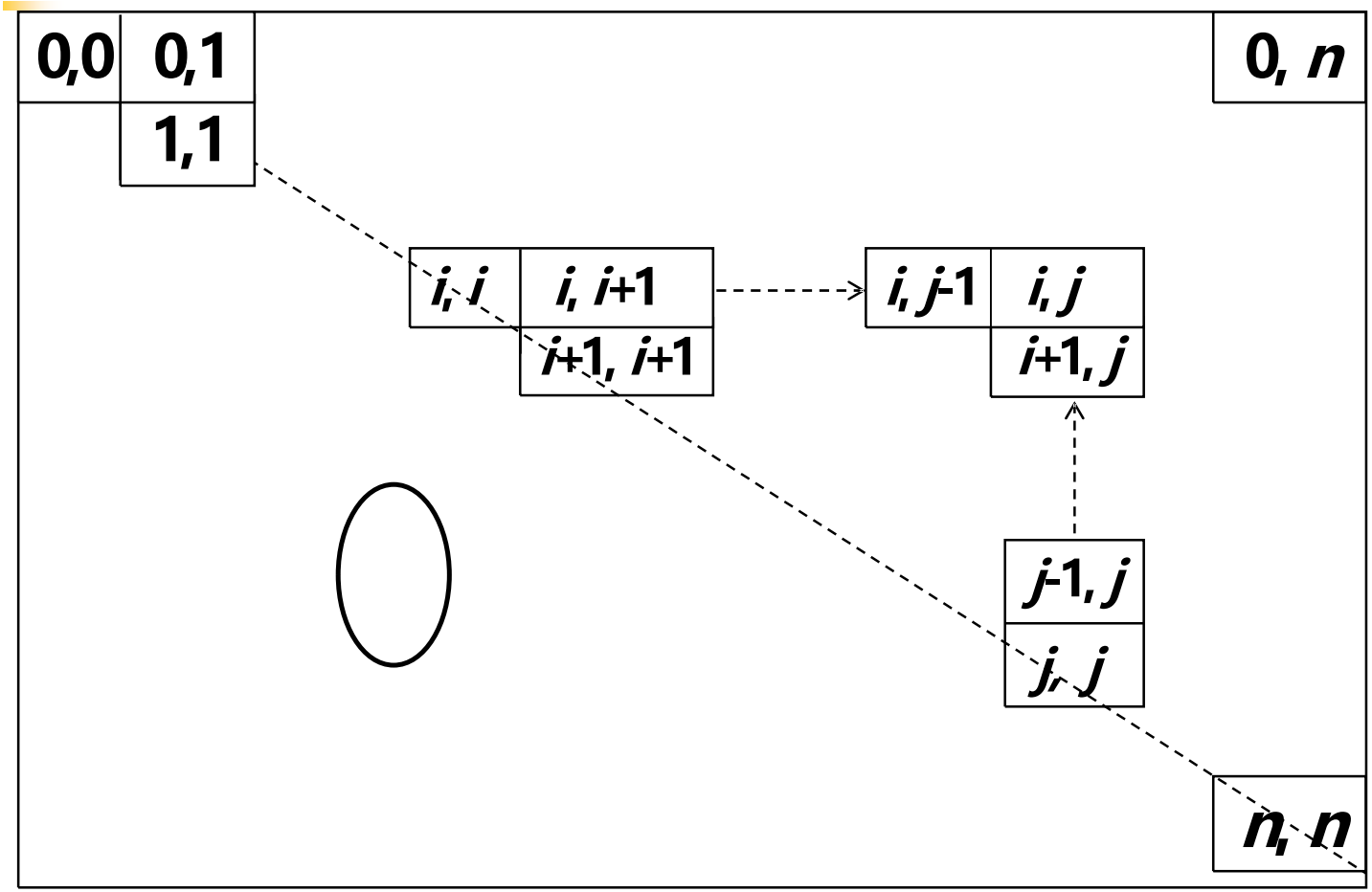
- Coke-Younger-Kasami (CYK)算法
  - 对 Chomsky 文法进行范式化:
    A➔a 或 A➔BC
- 自下而上的分析方法
- 构造$(n+1)\times(n+1)$识别矩阵，n为输入句子长度。
- 假设输入句子$x=a_1a_2...a_n$，a_i为构成句子的单词，$n=|x|$。

5.1 CYK分析算法

- 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



5.1 CYK分析算法



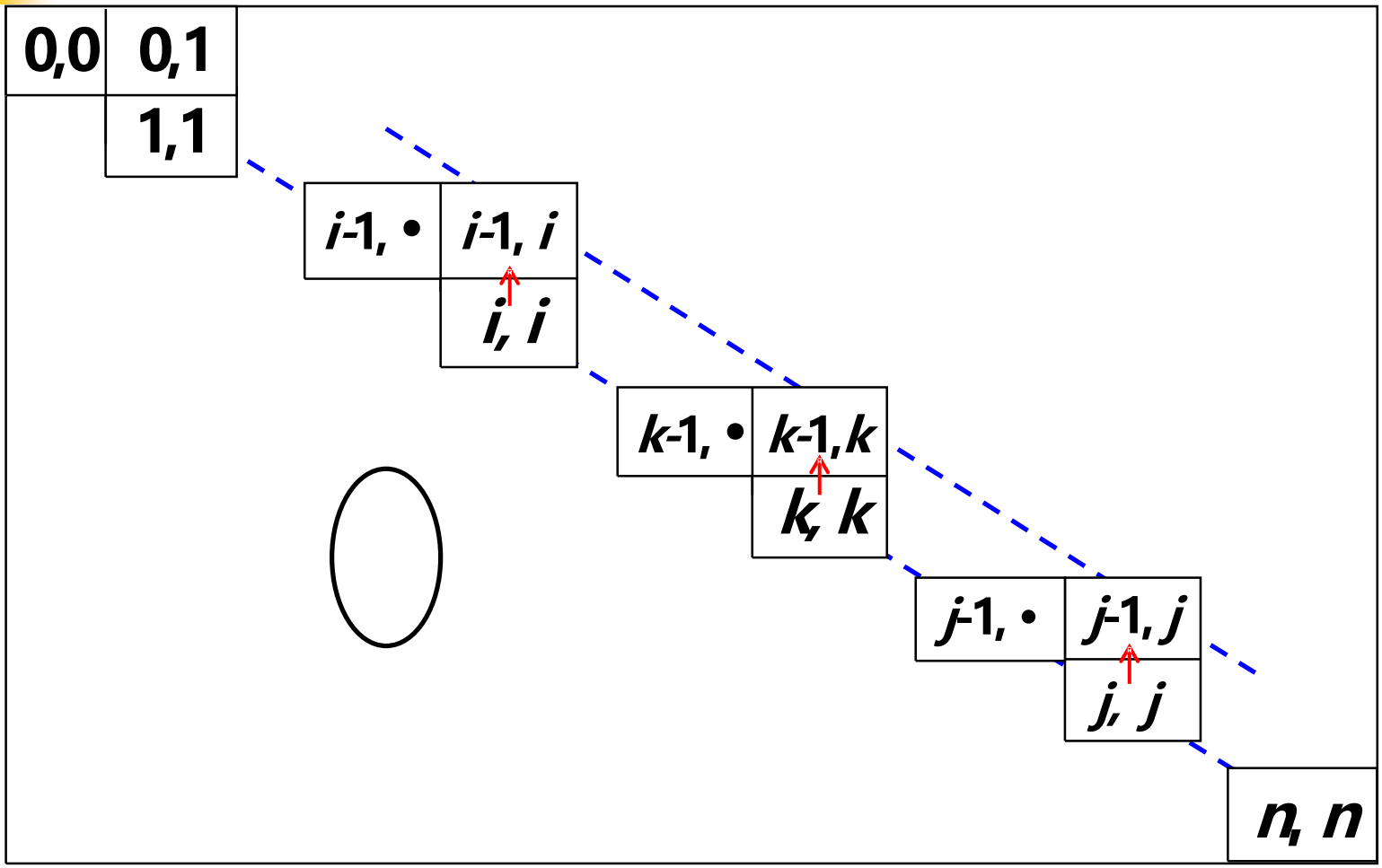
5.1 CYK分析算法

- 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a_i。
  - (2) 构造$t_{i,\ i+1}$主对角线以上紧靠主对角线的元素，其中，$i = 0, 1, 2, \ldots, n-1$。对于输入句子 $x = a_1\, a_2 \ldots a_n$，从 a_1 开始分析。

5.1 CYK分析算法

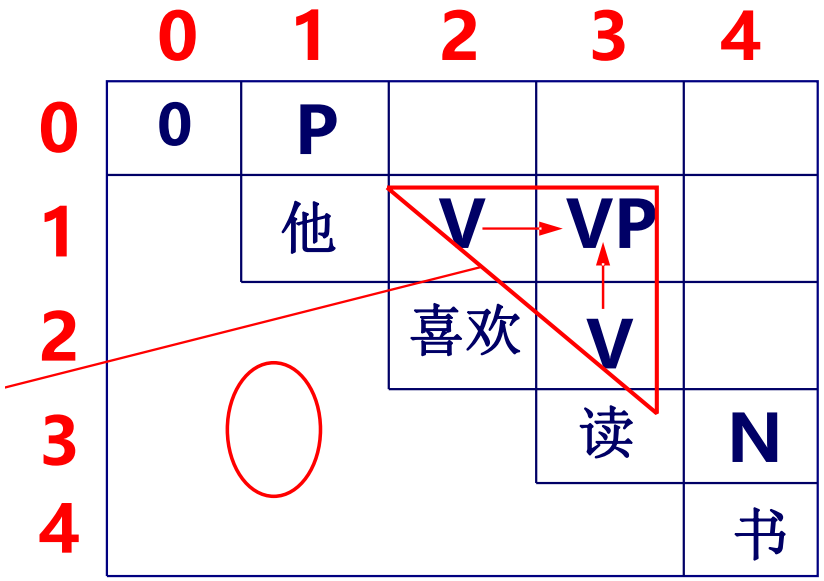
- 如果在文法G的产生式集合中有一条规则：$A \rightarrow a_1$ 则 $t_{0,1}=A$。
- 依此类推，如果有$A \rightarrow a_{i+1}$，则 $t_{i,\ i+1}=A$。
- 即，对于主对角线上的每一个终结符a_i，所有可能推导出它的非终结符写在它的右边主对角线上方的位置上。

5.1 CYK分析算法



5.1 CYK分析算法

(4) P➔他
(5) V ➔ 喜欢
(6) V ➔ 读
(7) N ➔ 书

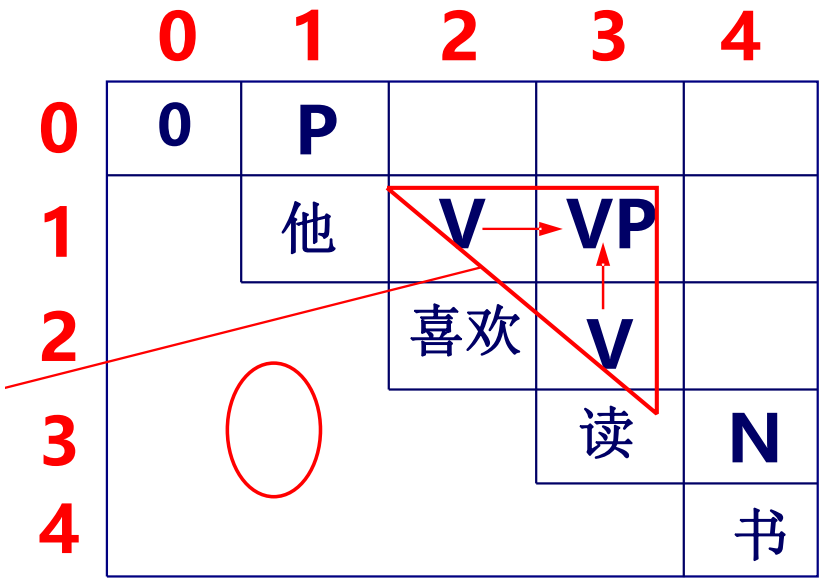


5.1 CYK分析算法

- (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，$i = 0,1, ..., n-d$，$j = d+i$，$d=2, 3, ..., n$。如果存在一个正整数 k，$i+1 \le k \le j-1$，在文法G的规则集中有产生式 A ➔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。

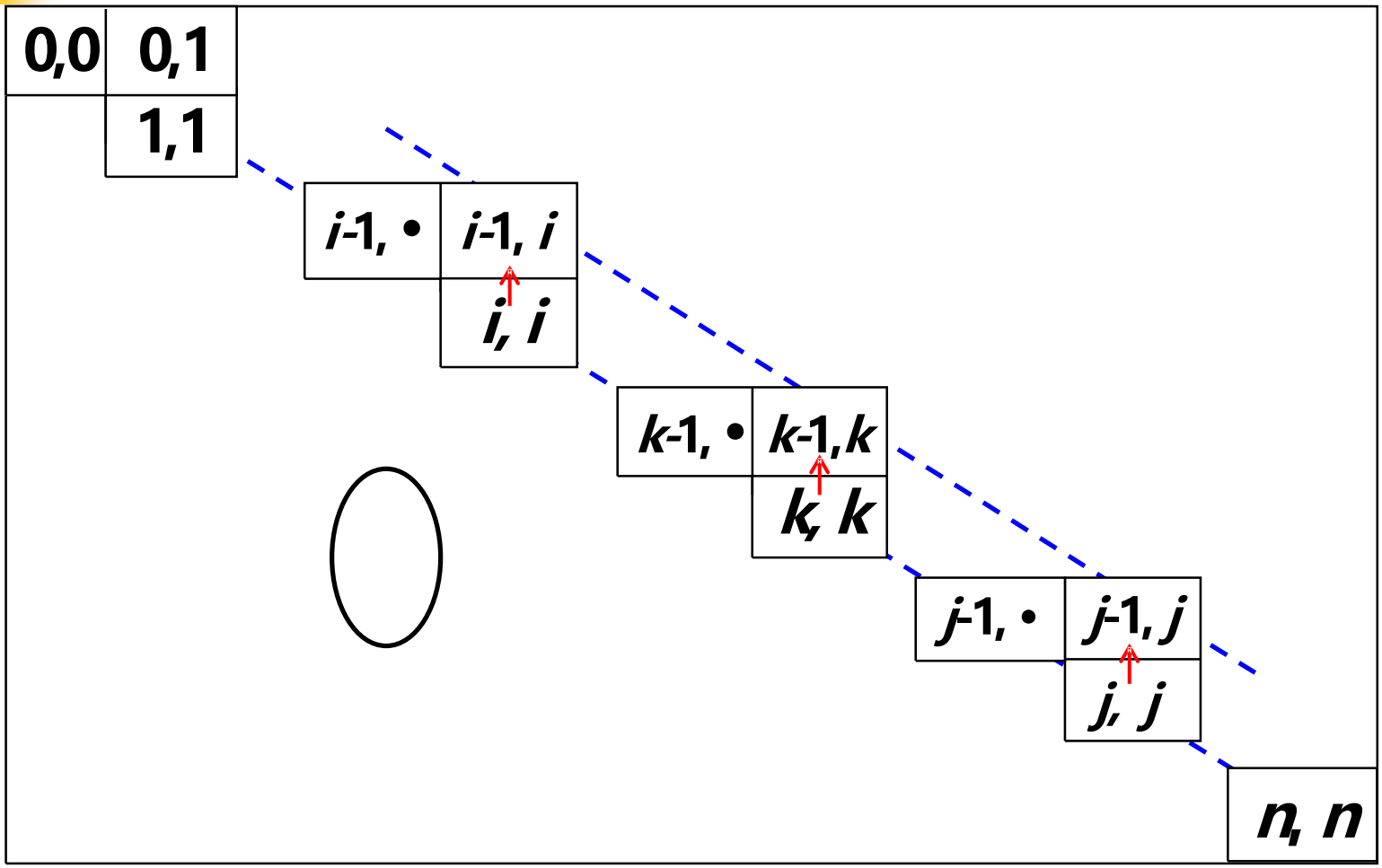
5.1 CYK分析算法

(2) VP➔V V
(4) P➔他
(5) V ➔喜欢
(6) V ➔ 读
(7)N ➔ 书

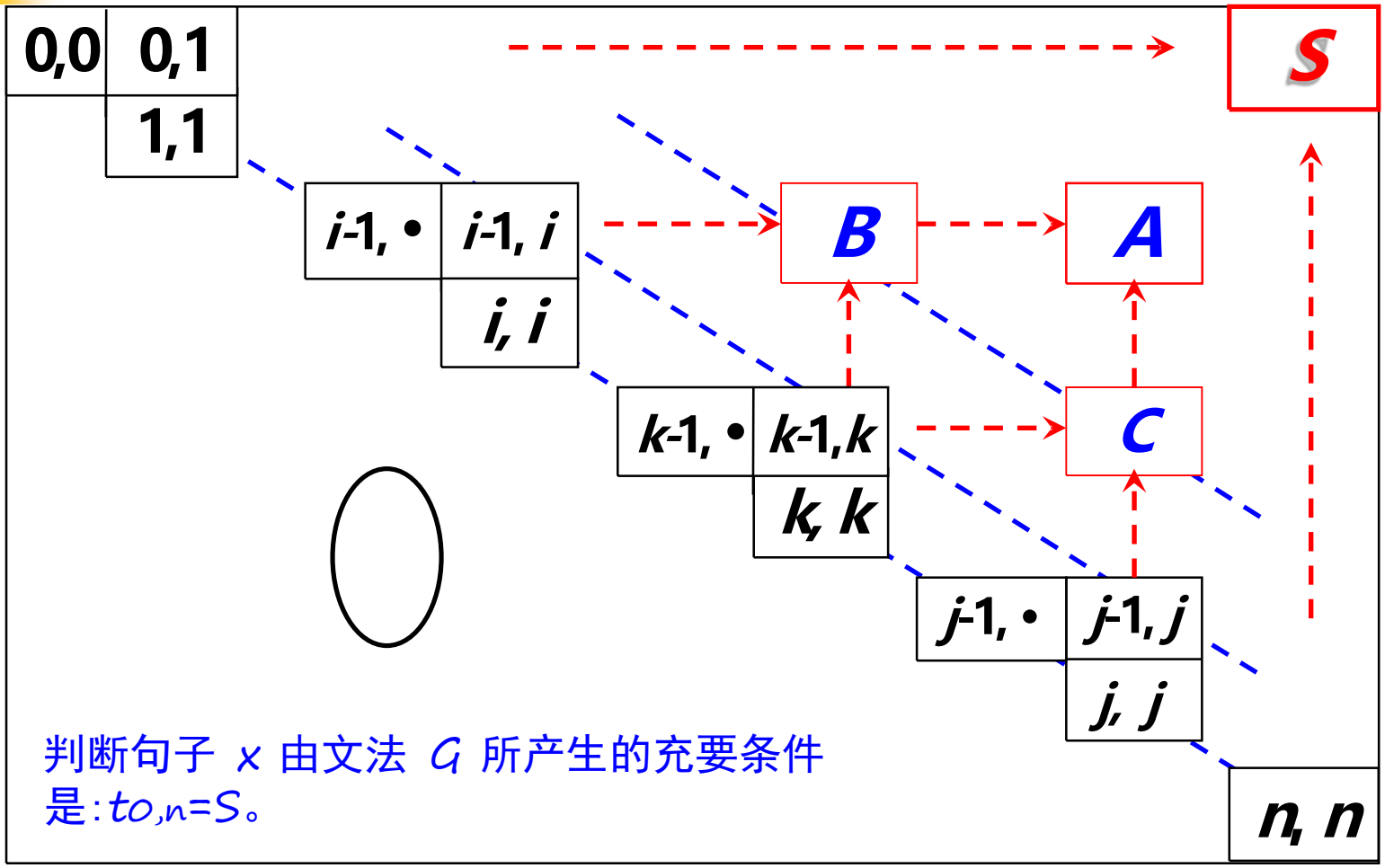


$t_{1,2}$=V, $t_{2,3}$ =V ➔ $t_{1,3}$ =VP

5.1 CYK分析算法



5.1 CYK分析算法


0,0
0,1
1,1
S
i-1, •
i-1, i
i, i
B
A
k-1, •
k-1,k
k, k
C
j-1, •
j-1, j
j, j
n, n
判断句子 x 由文法 G 所产生的充要条件是：$t_{0,n}=S$。

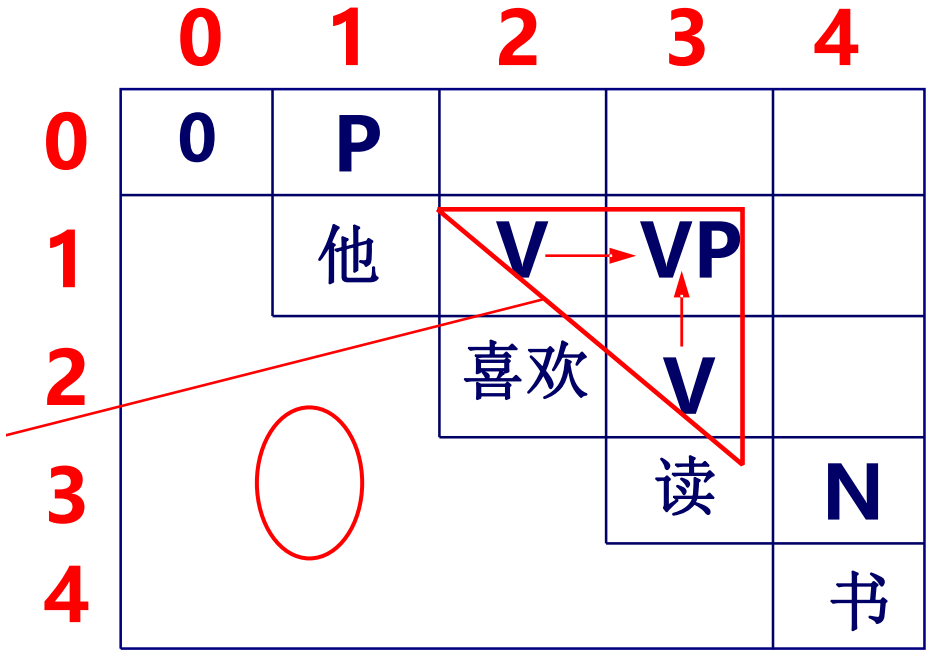
5.1 CYK分析算法

- 例子：给定文法G(S)：
  - (1) S➔P VP
  - (2) VP➔V V
  - (3) VP➔VP N
  - (4) P➔他
  - (5) V ➔喜欢
  - (6) V ➔ 读
  - (7) N ➔ 书
- 请用CYK算法分析句子:他喜欢读书

5.1 CYK分析算法

- 汉语分词和词性标注以后：
  - 他/P 喜欢/V 读/V 书/N n=4
- 构造识别矩阵：
- 执行分析过程。

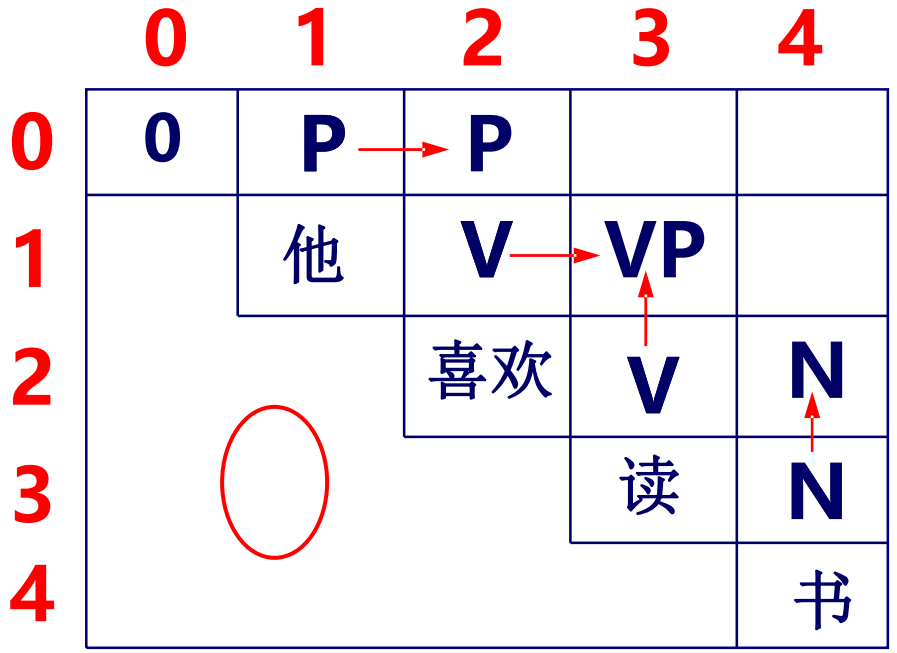
(1) S➔P VP
(2) VP➔V V
(3) VP➔VP N



5.1 CYK分析算法

- 汉语分词和词性标注以后：
  - 他/P 喜欢/V 读/V 书/N n=4
- 构造识别矩阵：
- 执行分析过程。

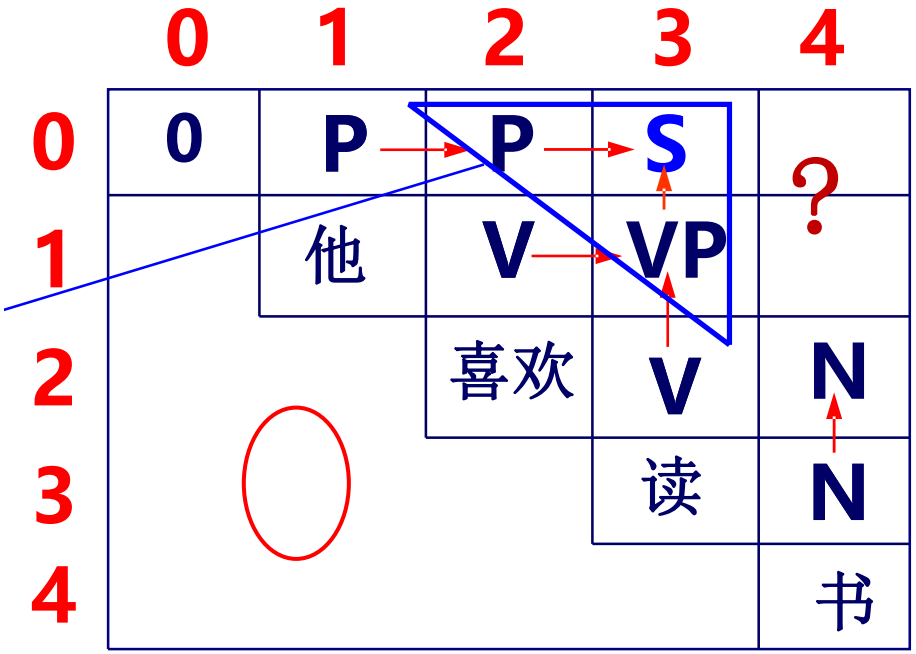
(1) S➔P VP
(2) VP➔V V
(3) VP➔VP N



5.1 CYK分析算法

- 汉语分词和词性标注以后：
  - 他/P 喜欢/V 读/V 书/N n=4
- 构造识别矩阵：
- 执行分析过程。

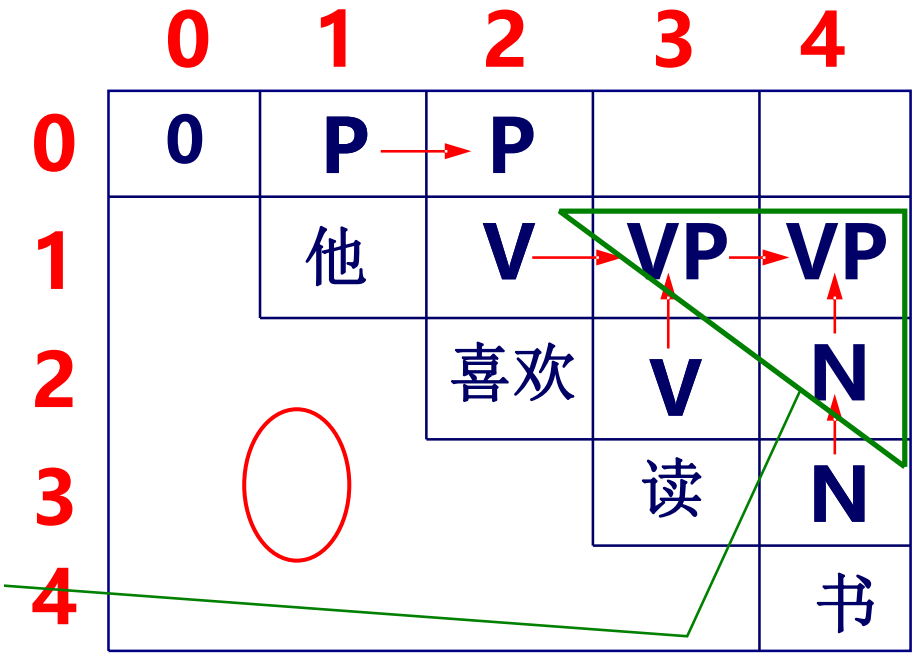
(1) S➔P VP
(2) VP➔V V
(3) VP➔VP N



5.1 CYK分析算法

- 汉语分词和词性标注以后：
  - 他/P 喜欢/V 读/V 书/N n=4
- 构造识别矩阵：
- 执行分析过程。

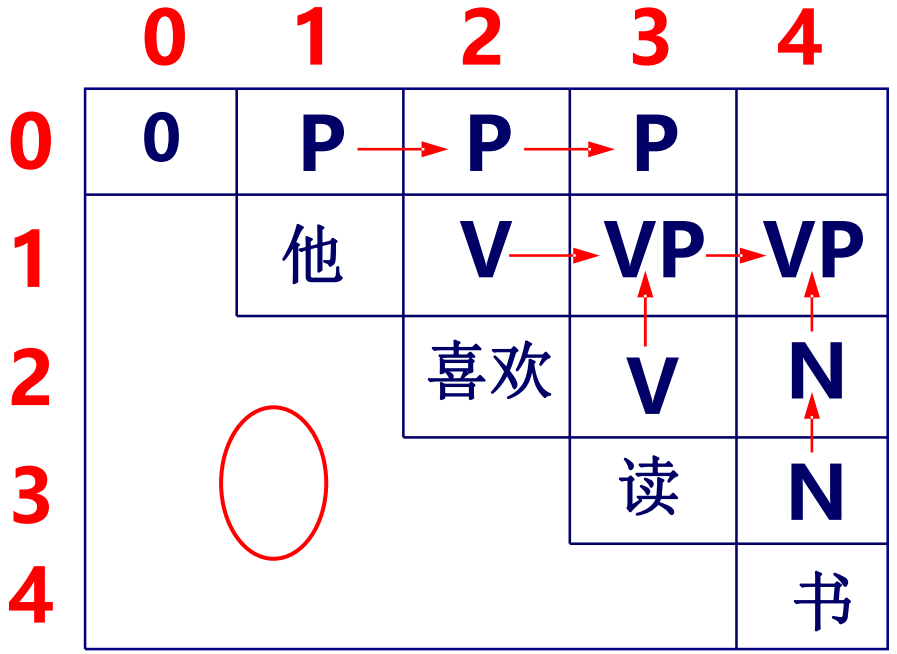
(1) S➔P VP
(2) VP➔V V
(3) VP➔VP N



5.1 CYK分析算法

- 汉语分词和词性标注以后：
  - 他/P 喜欢/V 读/V 书/N n=4
- 构造识别矩阵：
- 执行分析过程。

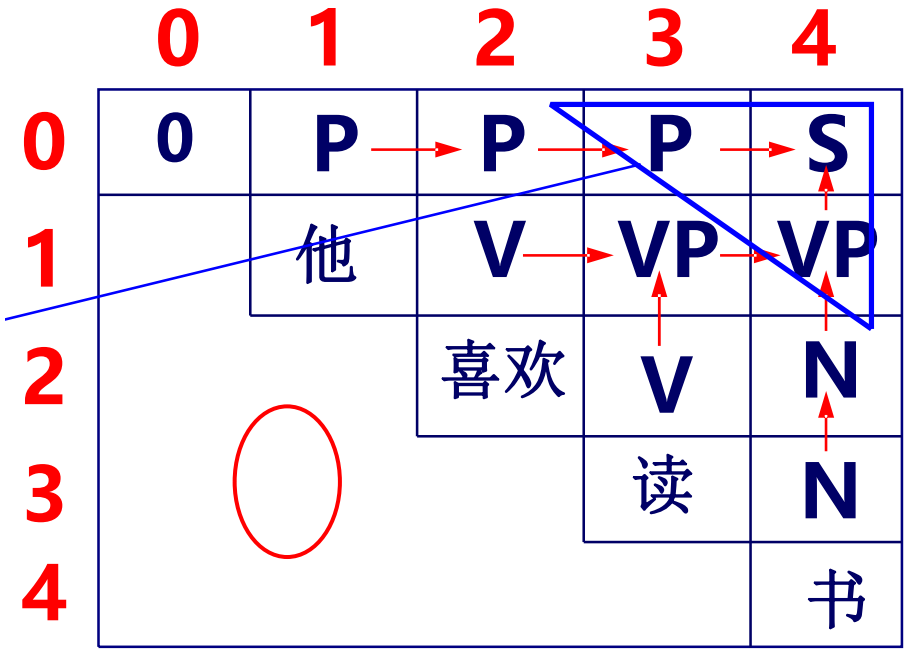
(1) S➔P VP
(2) VP➔V V
(3) VP➔VP N



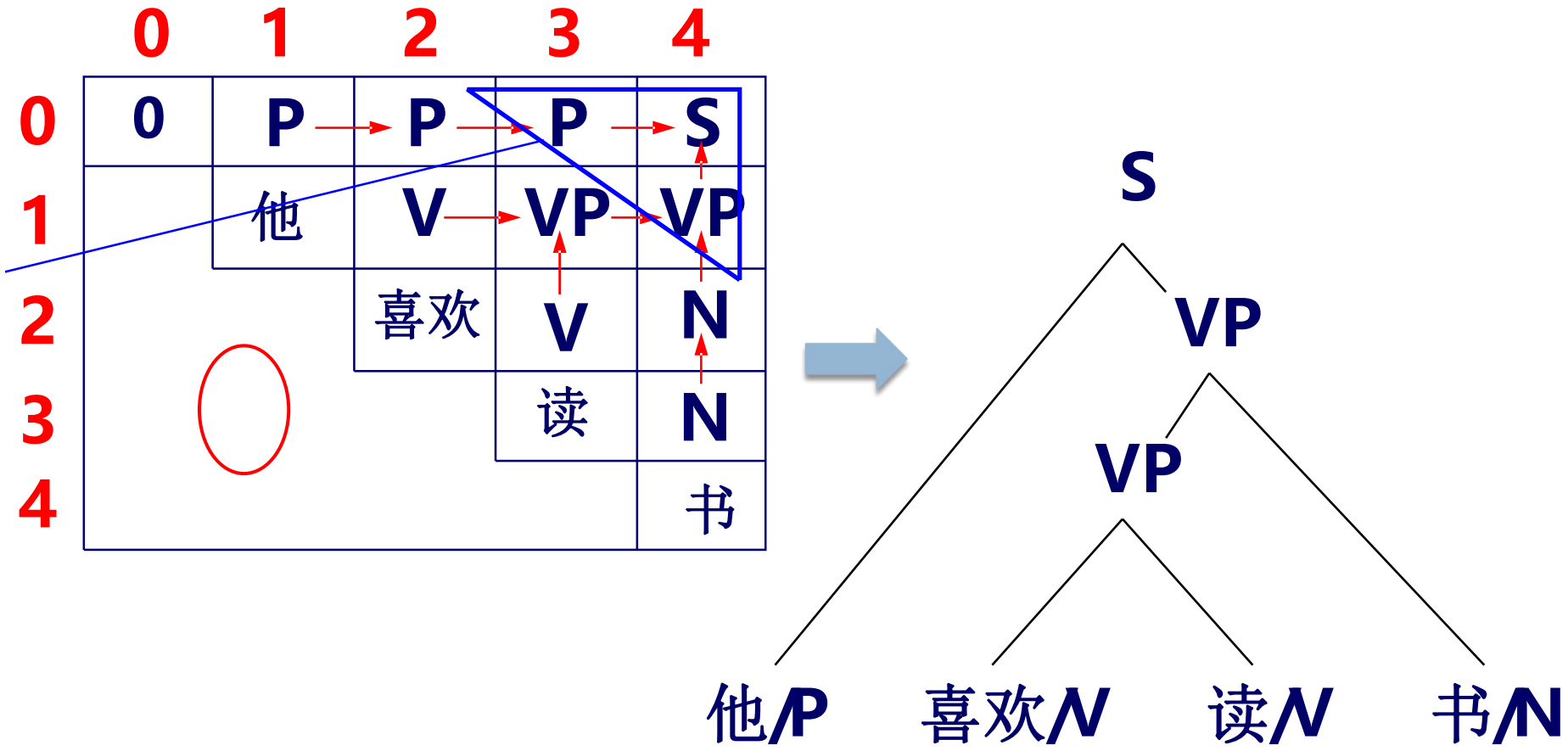
5.1 CYK分析算法

- 汉语分词和词性标注以后：
  - 他/P 喜欢/V 读/V 书/N n=4
- 构造识别矩阵：
- 执行分析过程。

(1) S➔P VP
(2) VP➔V V
(3) VP➔VP N



5.1 CYK分析算法



5.1 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

5.2 概率上下文无关文法PCFG

5.2 PCFG

PCFG规则

形式： $A \to \alpha, \quad p$

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

例如：

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

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

5.2 PCFG

◆例-1：S → NP VP, 1.00　　NP → NP PP, 0.40

NP → astronomers, 0.10

NP → ears, 0.18　　NP → saw, 0.04

NP → stars, 0.18　　NP → telescopes, 0.1

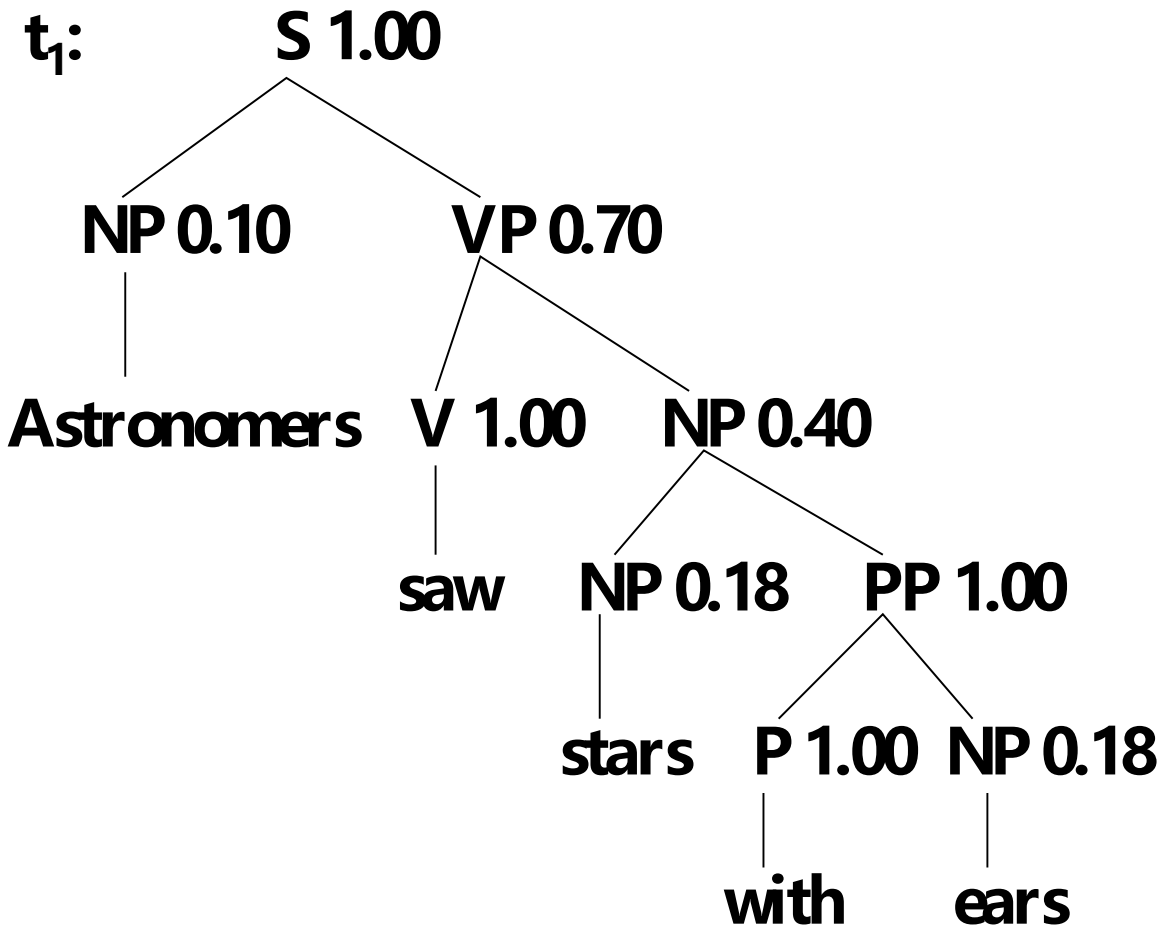
PP → P NP, 1.00　　P → with, 1.00

VP → V NP, 0.70　　VP → VP PP, 0.30

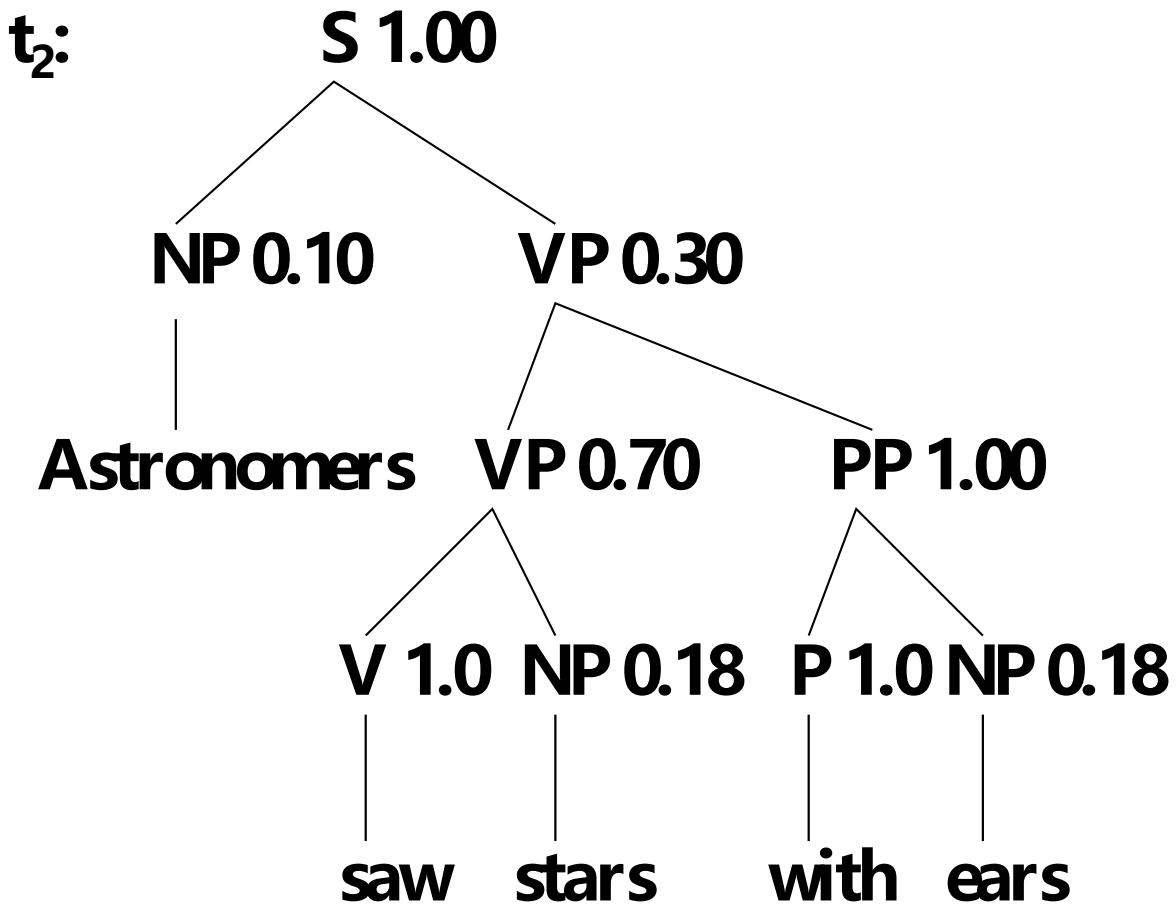
V → saw, 1.00

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

5.2 PCFG


t1:
S 1.00
NP 0.10
VP 0.70
Astronomers
V 1.00
NP 0.40
saw
NP 0.18
PP 1.00
stars
P 1.00
NP 0.18
with
ears

5.2 PCFG

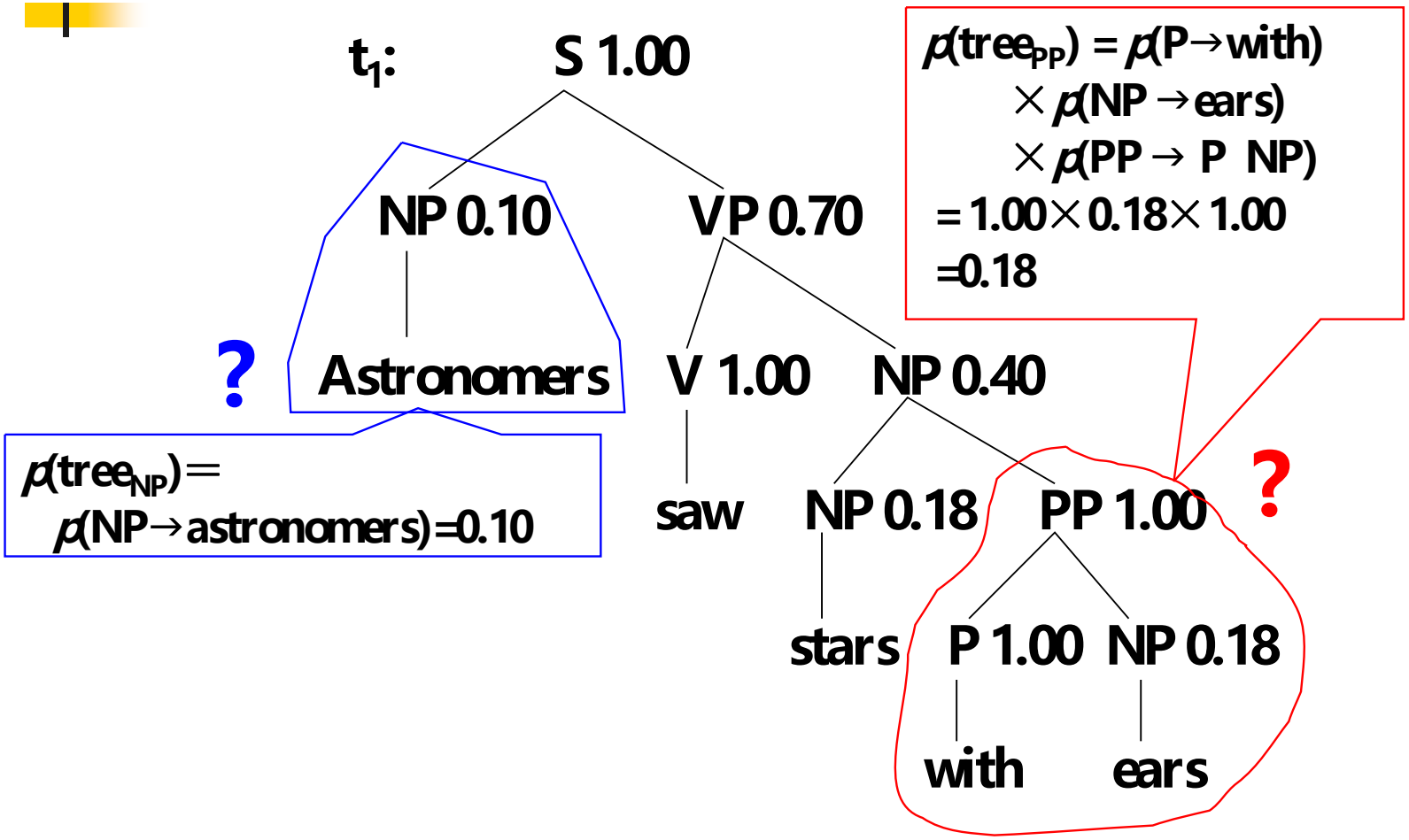

t2:
S 1.00
NP 0.10
VP 0.30
Astronomers
VP 0.70
PP 1.00
V 1.0
NP 0.18
P 1.0
NP 0.18
saw
stars
with
ears

5.2 PCFG

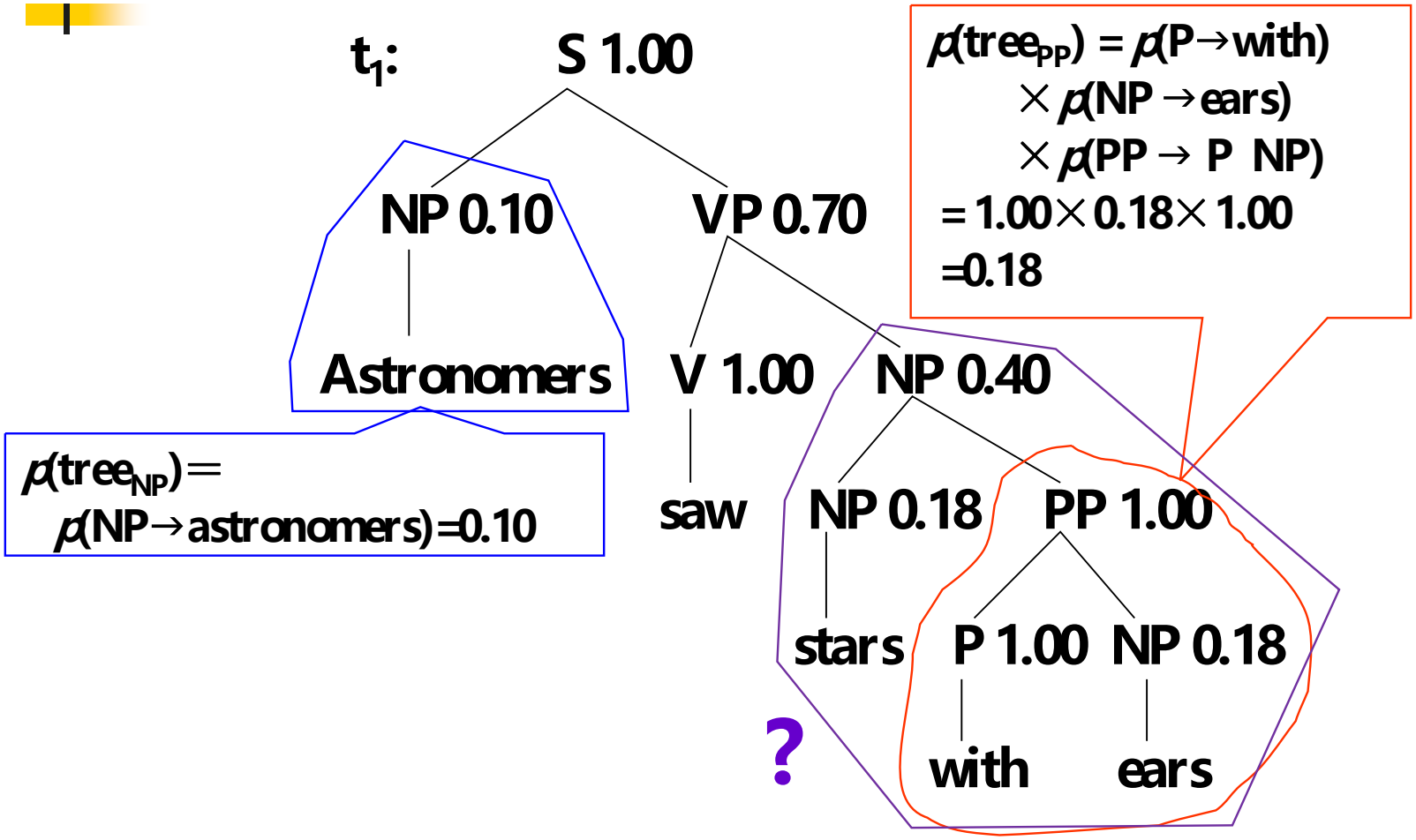
❑ 计算分析树概率的基本假设

- **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 $p(A_{kl} \rightarrow w$|任何超出 k~l 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 $p(A_{kl} \rightarrow w$|任何除 A 以外的祖先结点$) = p(A_{kl} \rightarrow w)$。

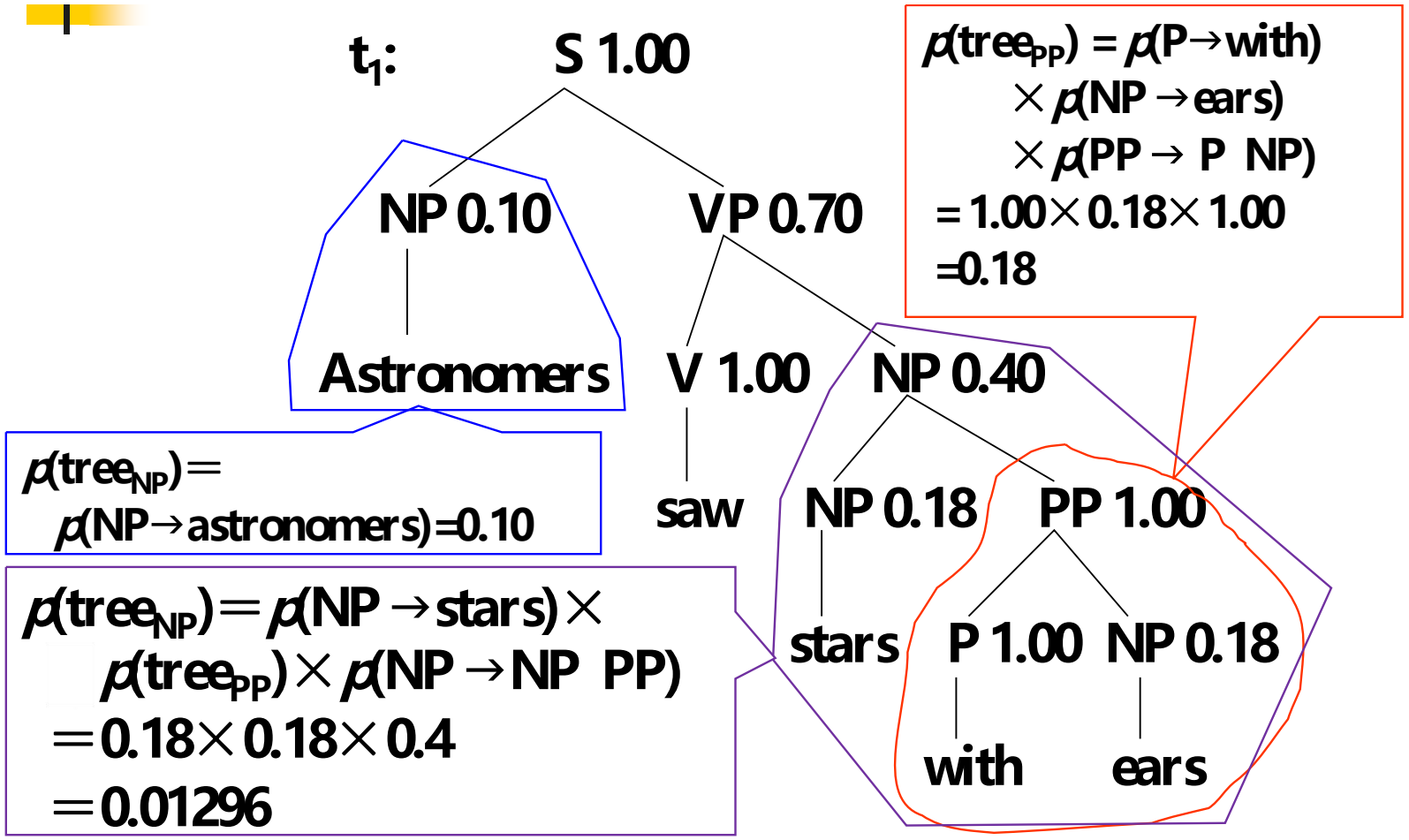
5.2 PCFG


t1:
S 1.00
NP 0.10
VP 0.70
Astronomers
V 1.00
NP 0.40
saw
NP 0.18
PP 1.00
stars
P 1.00
NP 0.18
with
ears
?
?
p(tree_NP)=
p(NP→astronomers)=0.10
p(tree_PP) = p(P→with)
× p(NP→ears)
× p(PP→ P NP)
=1.00×0.18×1.00
=0.18

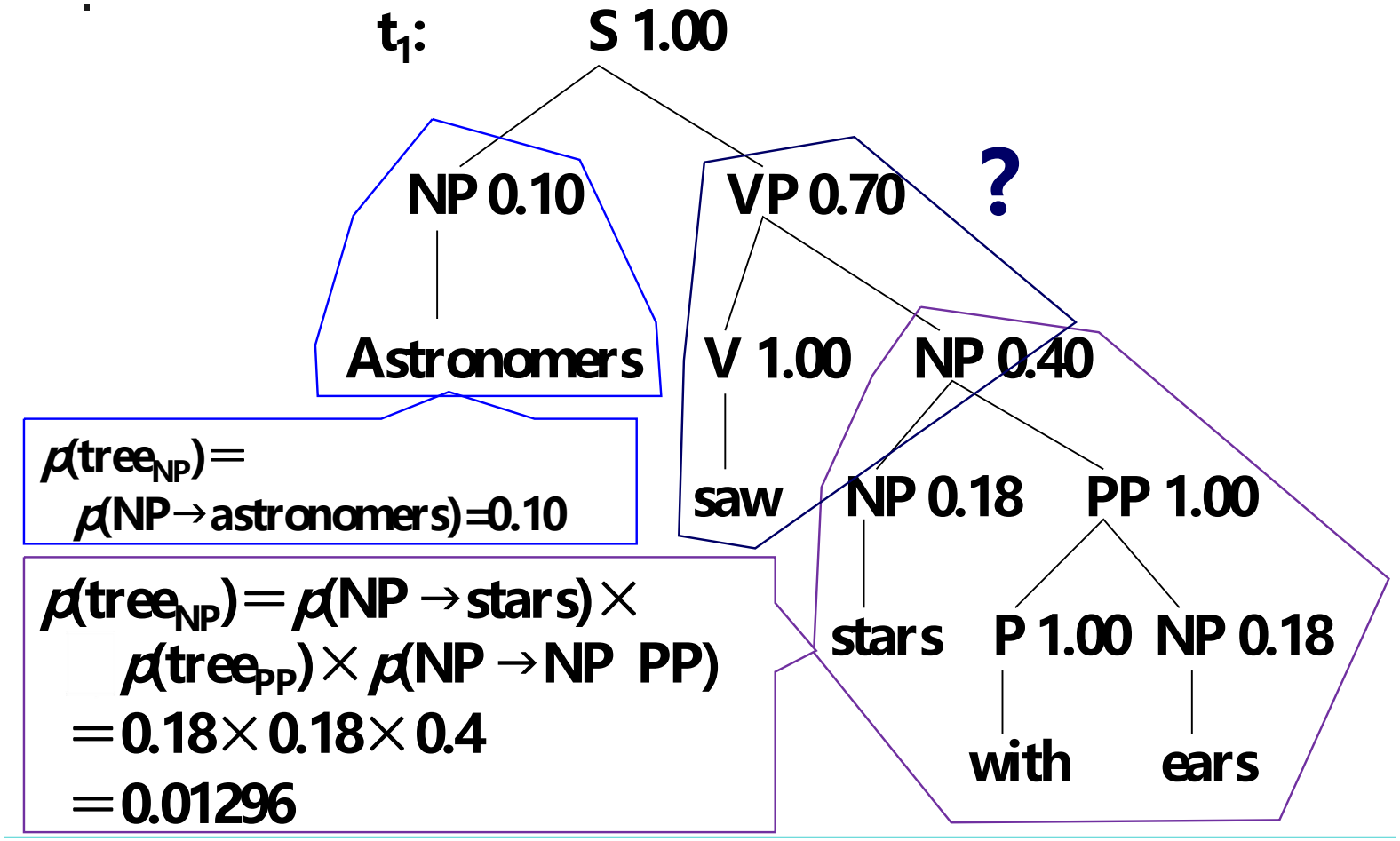
5.2 PCFG


t1: S 1.00
NP 0.10
VP 0.70
Astronomers
V 1.00
NP 0.40
saw
NP 0.18
PP 1.00
stars
P 1.00
NP 0.18
with
ears
?
p(tree_NP) = p(NP→astronomers) = 0.10
p(tree_PP) = p(P→with) × p(NP→ears) × p(PP→P NP)
= 1.00 × 0.18 × 1.00
= 0.18

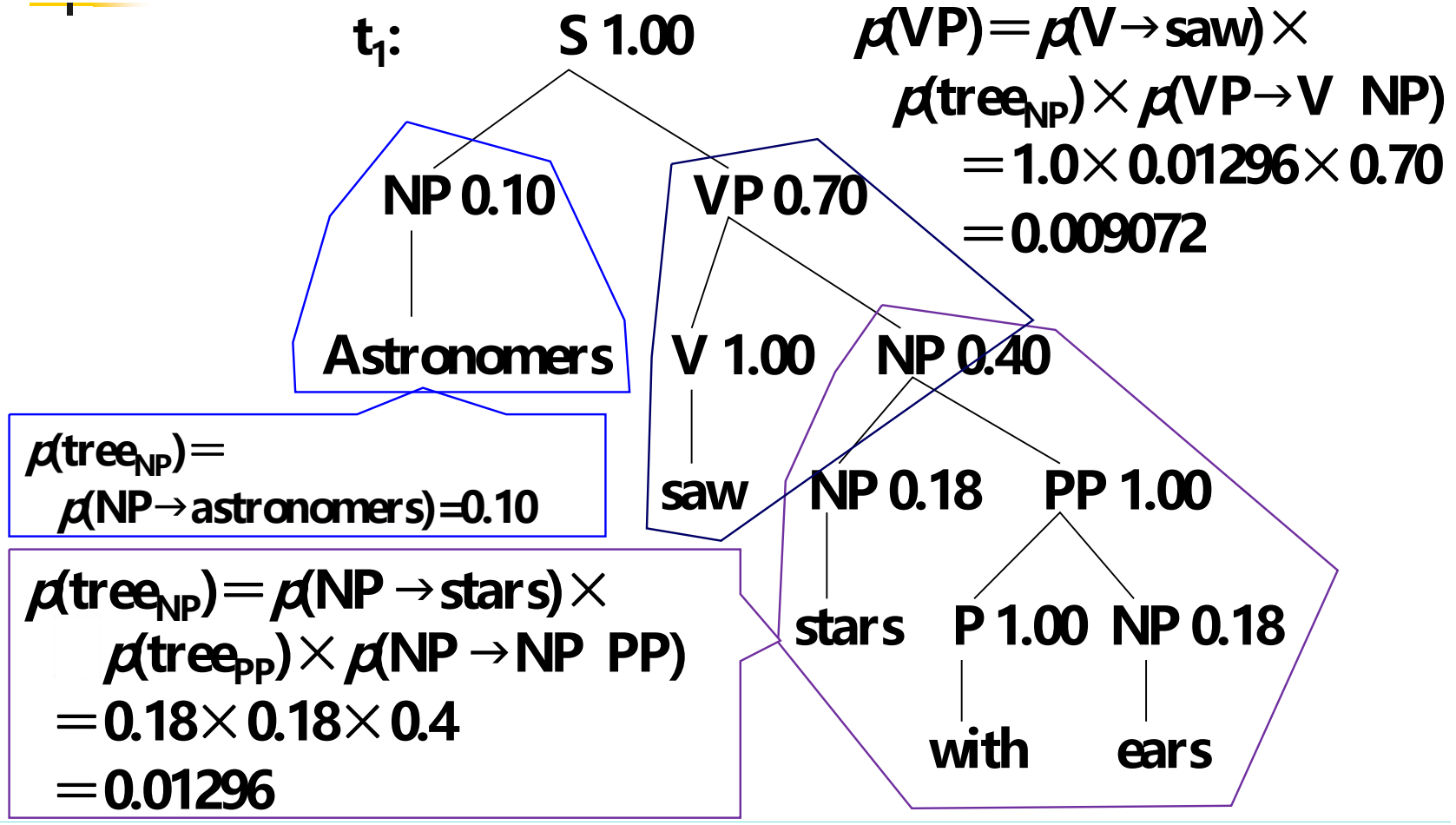
5.2 PCFG


t1: S 1.00
NP 0.10
VP 0.70
Astronomers
V 1.00
NP 0.40
saw
NP 0.18
PP 1.00
stars
P 1.00
NP 0.18
with
ears
p(tree_PP) = p(P→with) × p(NP→ears) × p(PP→ P NP) = 1.00×0.18×1.00 =0.18
p(tree_NP)= p(NP→astronomers)=0.10
p(tree_NP)=p(NP→stars)× p(tree_PP)×p(NP→NP PP) =0.18×0.18×0.4 =0.01296

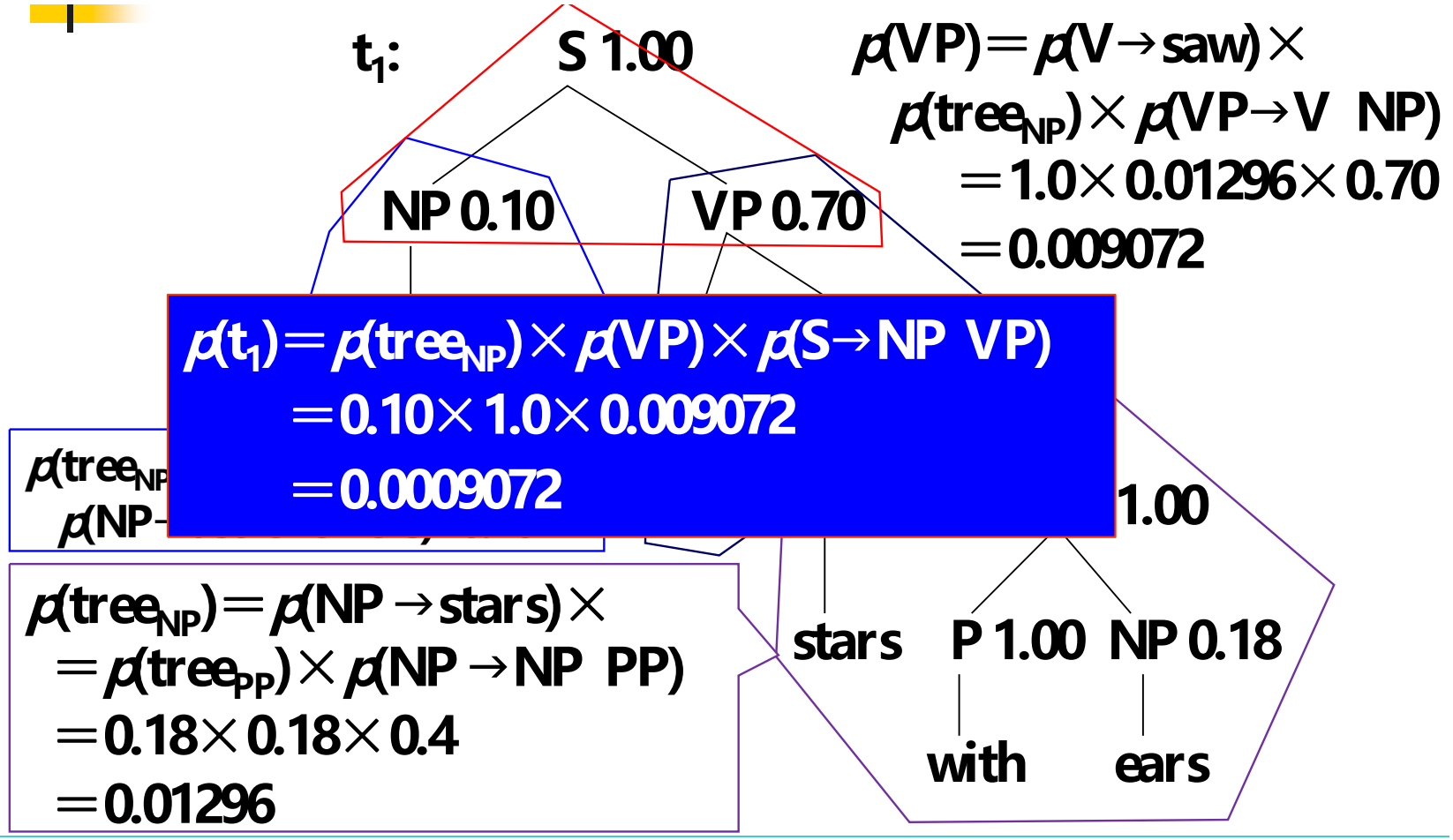
5.2 PCFG


t1:
S 1.00
NP 0.10
VP 0.70
?
Astronomers
V 1.00
NP 0.40
saw
NP 0.18
PP 1.00
stars
P 1.00
NP 0.18
with
ears
p(tree_NP)=
p(NP→astronomers)=0.10
p(tree_NP)=p(NP→stars)×
p(tree_PP)×p(NP→NP PP)
=0.18×0.18×0.4
=0.01296

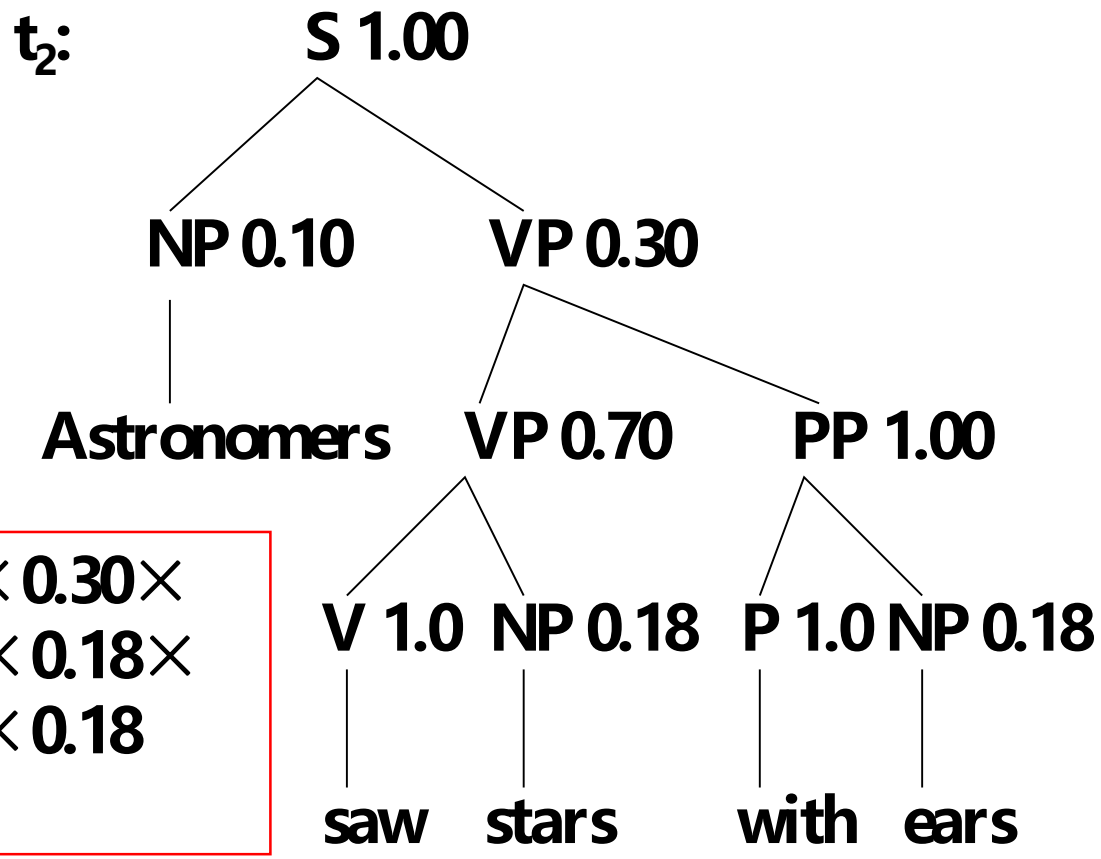
5.2 PCFG


t1: S 1.00
NP 0.10
VP 0.70
Astronomers
V 1.00
NP 0.40
saw
NP 0.18
PP 1.00
stars
P 1.00
NP 0.18
with
ears
p(VP)=p(V→saw)×
p(treeNP)×p(VP→V NP)
=1.0×0.01296×0.70
=0.009072
p(treeNP)=
p(NP→astronomers)=0.10
p(treeNP)=p(NP→stars)×
p(treePP)×p(NP→NP PP)
=0.18×0.18×0.4
=0.01296

5.2 PCFG


t_1:
S 1.00
NP 0.10
VP 0.70
$p(VP) = p(V \rightarrow saw) \times$
$p(tree_{NP}) \times p(VP \rightarrow V\ NP)$
$= 1.0 \times 0.01296 \times 0.70$
$= 0.009072$
$p(t_1) = p(tree_{NP}) \times p(VP) \times p(S \rightarrow NP\ VP)$
$= 0.10 \times 1.0 \times 0.009072$
$= 0.0009072$
$p(tree_{NP}) = p(NP \rightarrow stars) \times$
$= p(tree_{PP}) \times p(NP \rightarrow NP\ PP)$
$= 0.18 \times 0.18 \times 0.4$
$= 0.01296$
1.00
stars
P 1.00
NP 0.18
with
ears

5.2 PCFG



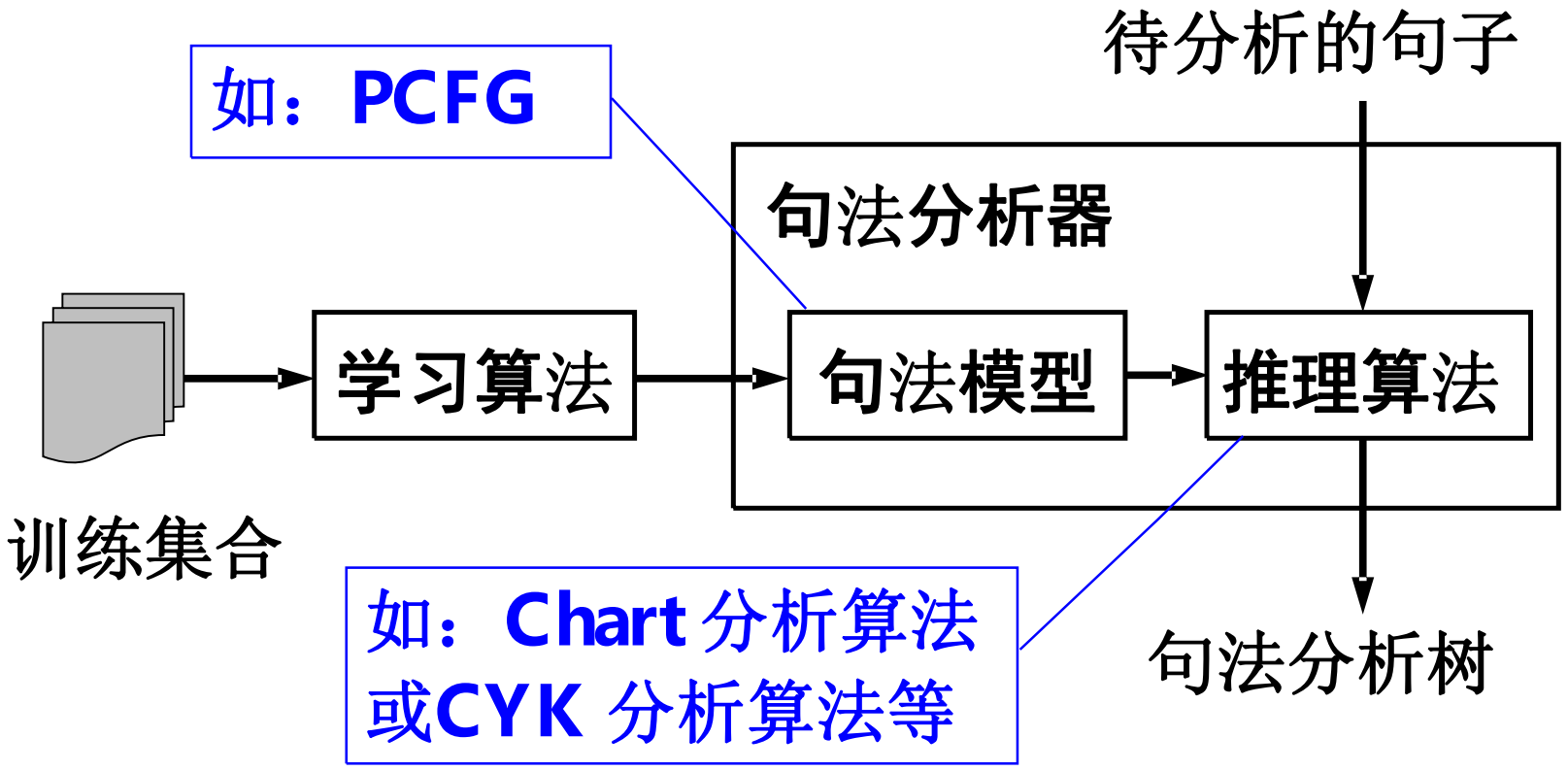
$$p(t_2) = 1.00 \times 0.10 \times 0.30 \times 0.70 \times 1.00 \times 0.18 \times 1.00 \times 1.00 \times 0.18 = 0.0006804$$

5.2 PCFG

- 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。

5.2 PCFG的分析实例

统计句法分析器实现的一般方法：



5.2 PCFG的分析实例

给定如下 PCFG G(S)：

非终结符集合：N={S, NP, VP, PP, DT, Vi, Vt, NN, IN}

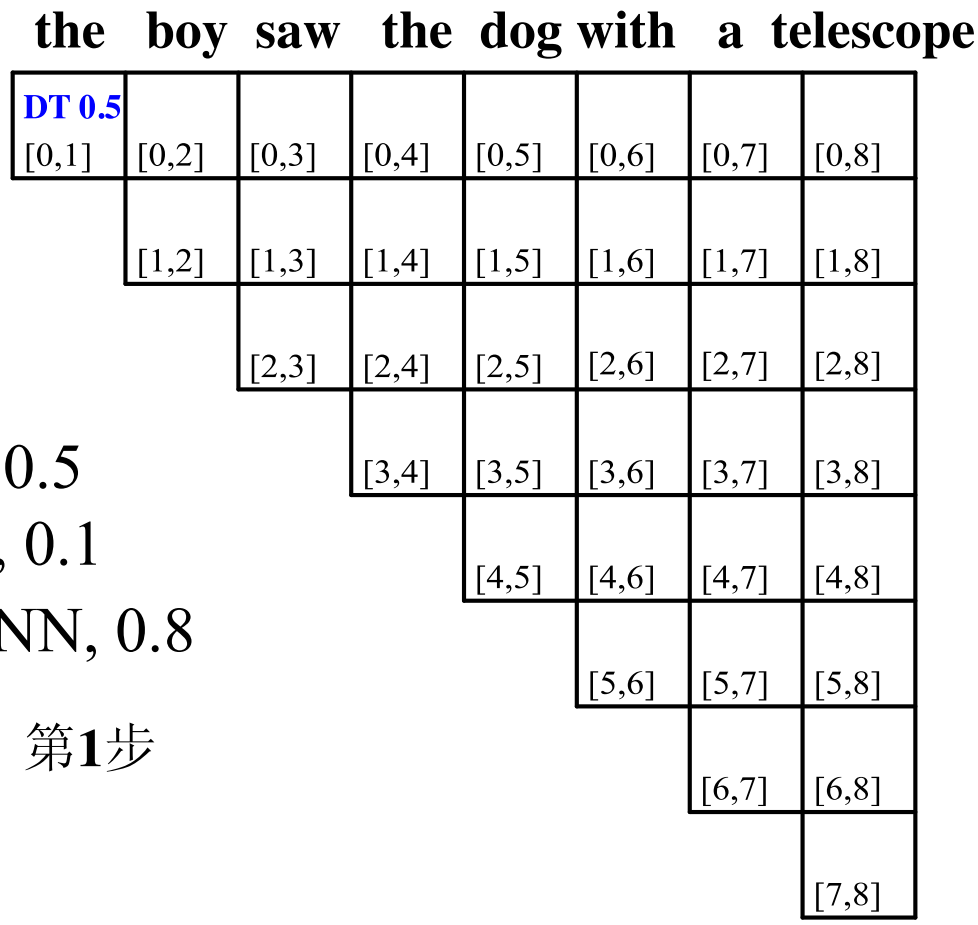
终结符集合：Σ={sleeps, saw, man, woman, dog, telescope, the, with, in}

规则集：

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

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

5.2 PCFG的分析实例



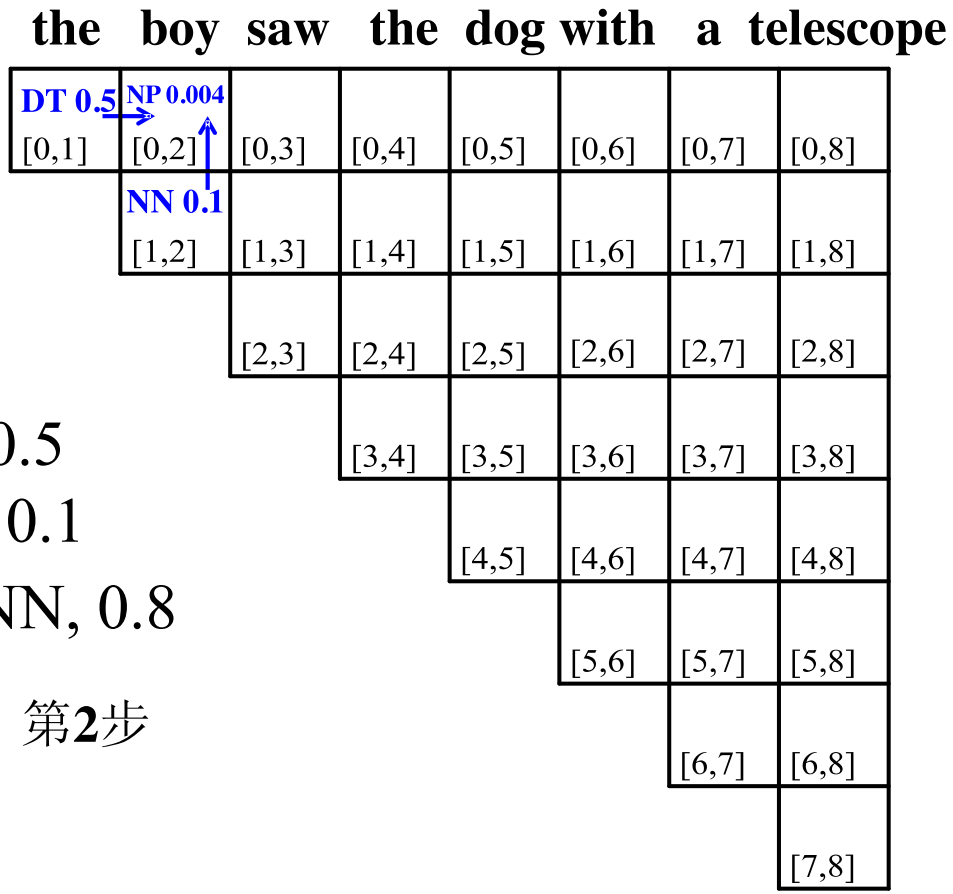
DT➔the, 0.5

NN➔boy, 0.1

NP➔DT NN, 0.8

第1步

5.2 PCFG的分析实例



DT➔the, 0.5

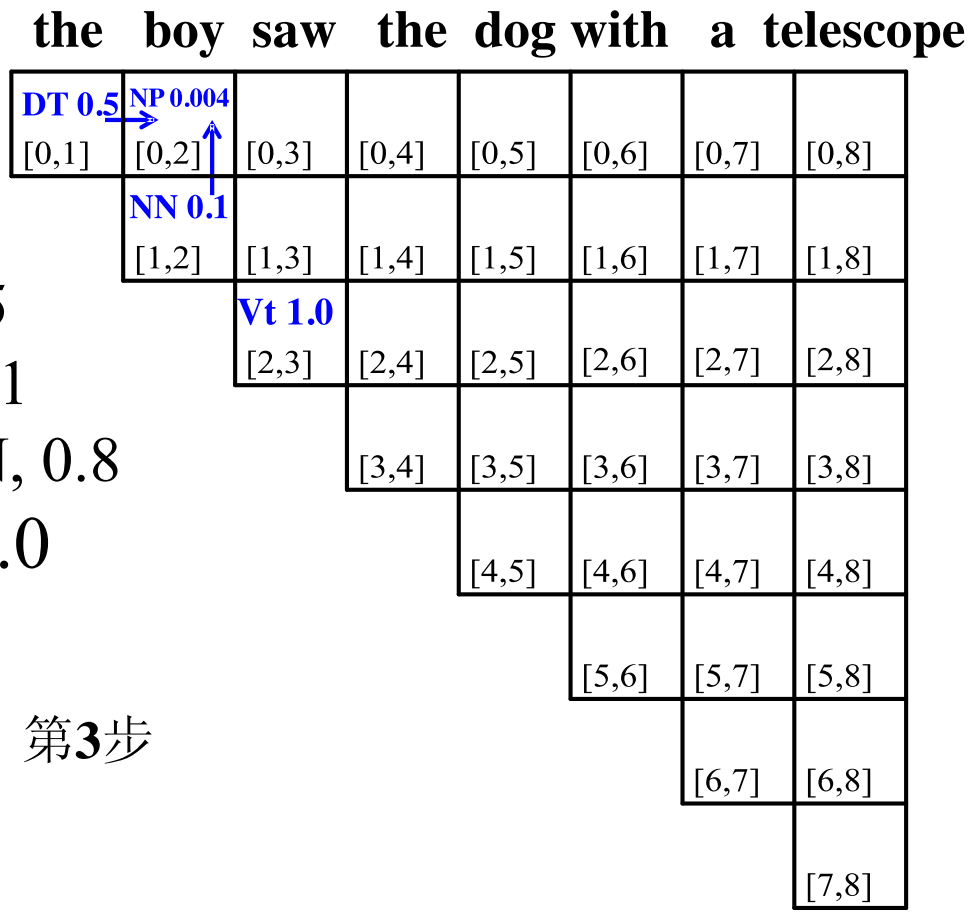
NN➔boy, 0.1

NP➔DT NN, 0.8

第**2**步

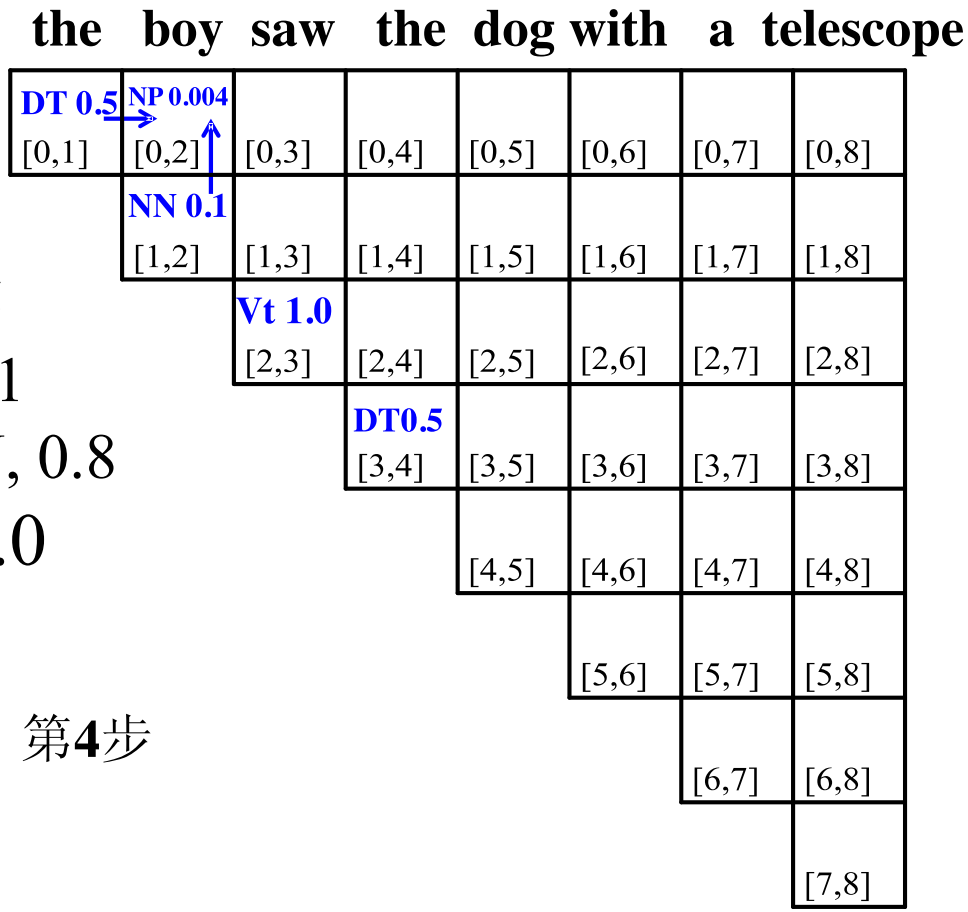
5.2 PCFG的分析实例

DT➔the, 0.5
NN➔boy, 0.1
NP➔DT NN, 0.8
Vt➔saw, 1.0



第3步

5.2 PCFG的分析实例

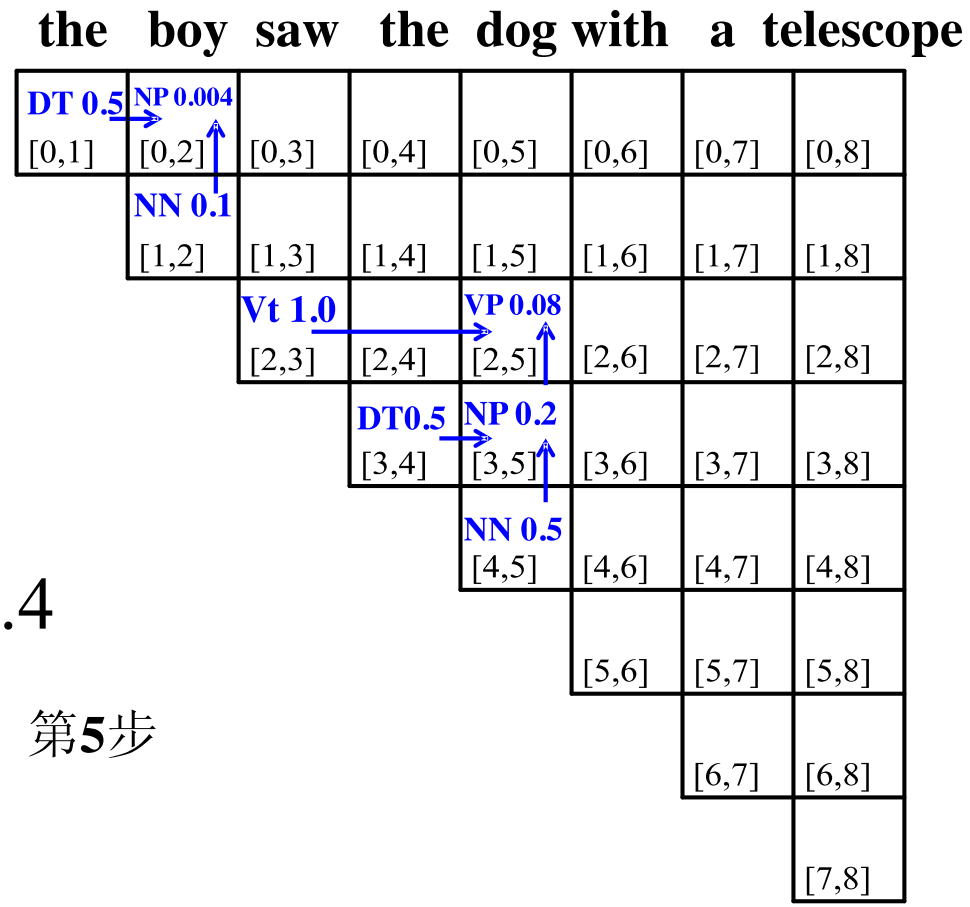


DT➔the, 0.5
NN➔boy, 0.1
NP➔DT NN, 0.8
Vt➔saw, 1.0

第**4**步

5.2 PCFG的分析实例

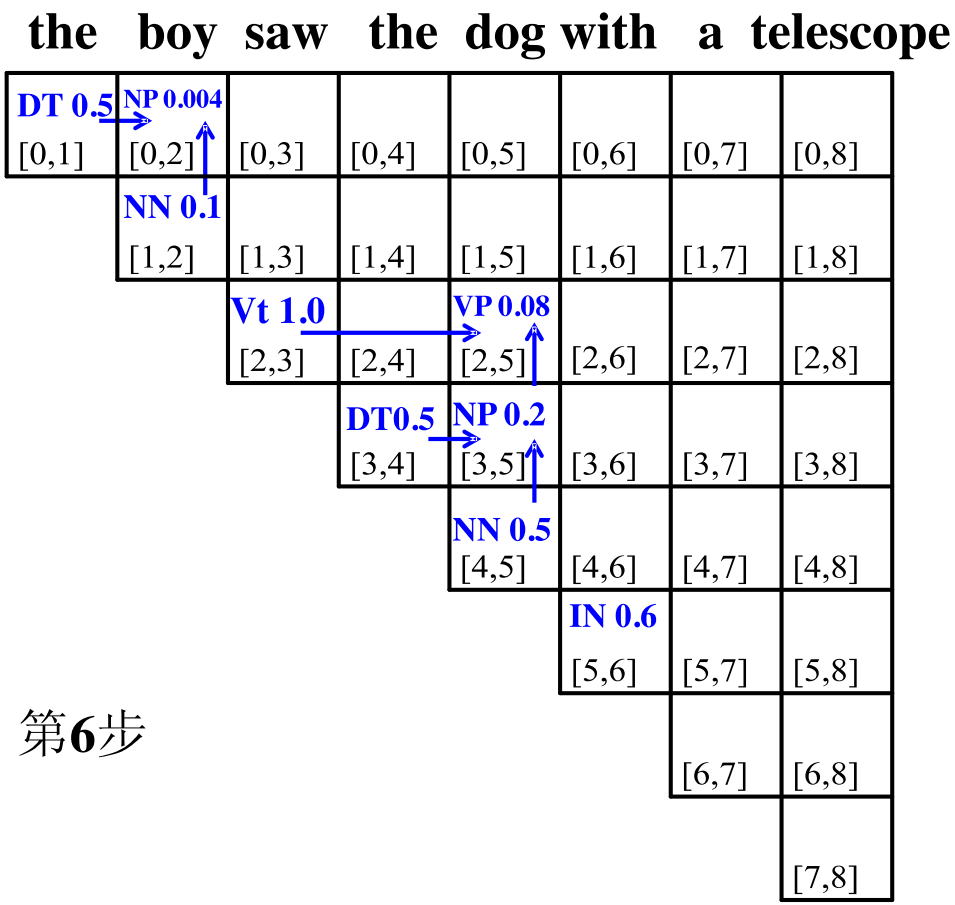
DT➔the, 0.5
NN➔boy, 0.1
NP➔DT NN, 0.8
Vt➔saw, 1.0
NN➔dog, 0.5
VP ➔Vt Np, 0.4



第5步

5.2 PCFG的分析实例

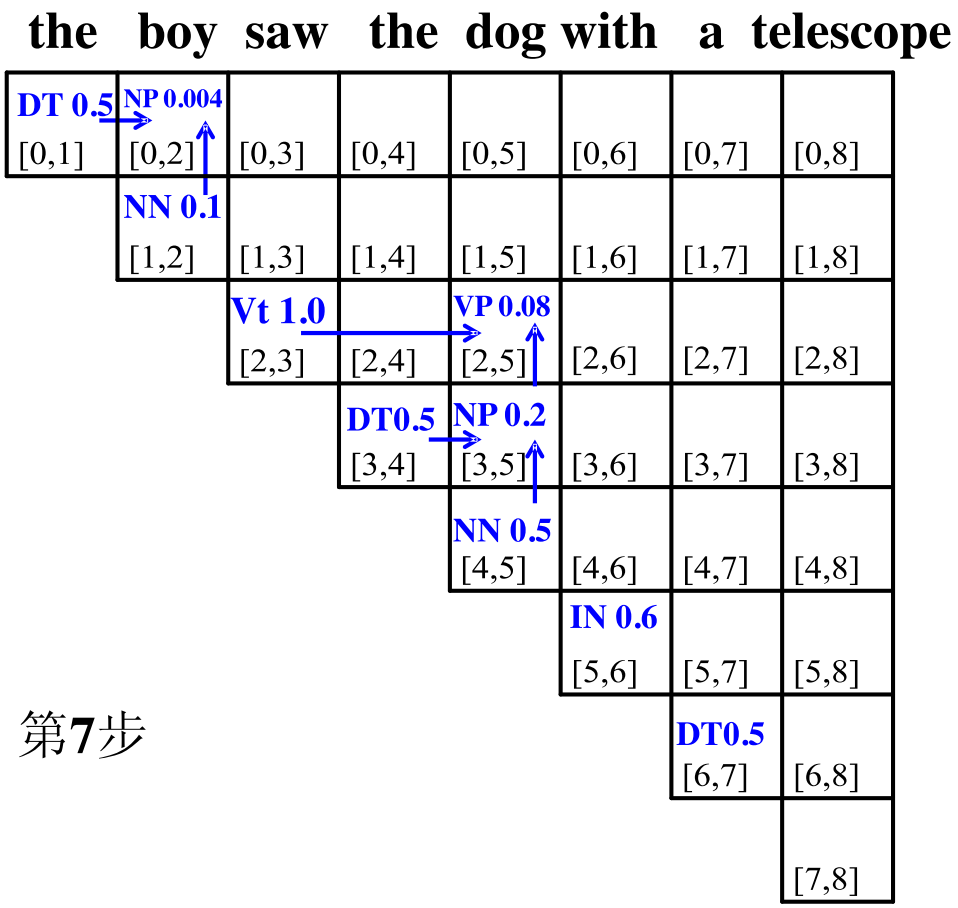
DT➔the, 0.5
NN➔boy, 0.1
NP➔DT NN, 0.8
Vt➔saw, 1.0
NN➔dog, 0.5
VP ➔Vt Np, 0.4
IN➔with, 0.6



第6步

5.2 PCFG的分析实例

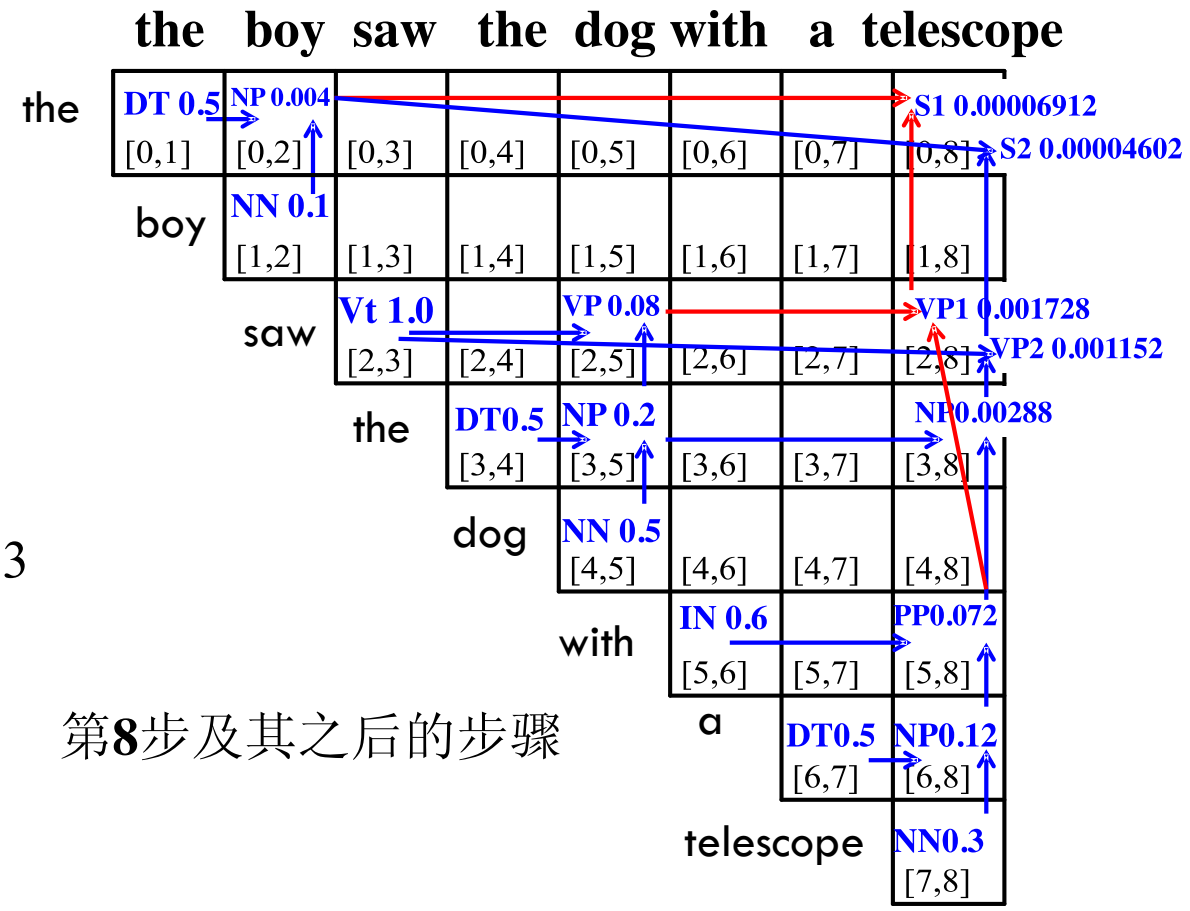
DT➔the, 0.5
NN➔boy, 0.1
NP➔DT NN, 0.8
Vt➔saw, 1.0
NN➔dog, 0.5
VP ➔Vt Np, 0.4
IN➔with, 0.6
DT➔a, 0.5



第7步

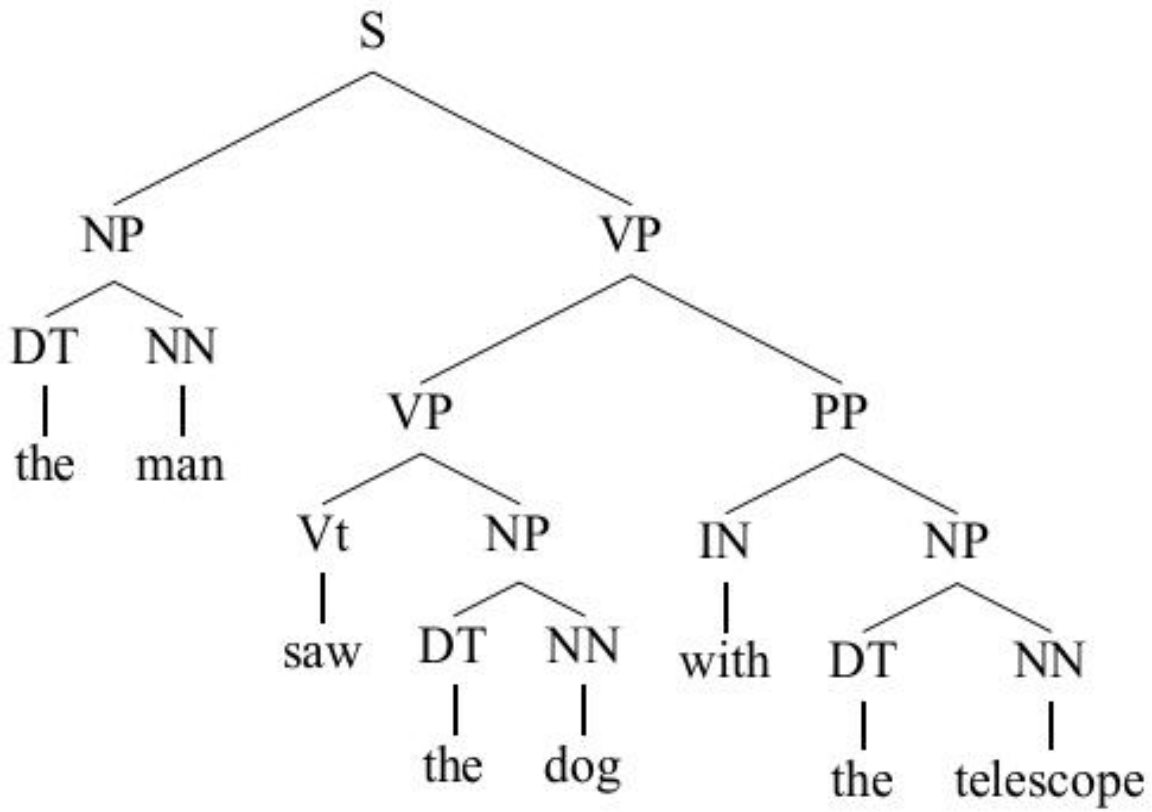
5.2 PCFG的分析实例

DT➔the, 0.5
NN➔boy, 0.1
NP➔DT NN, 0.8
Vt➔saw, 1.0
NN➔dog, 0.5
VP ➔Vt Np, 0.4
IN➔with, 0.6
DT➔a, 0.5
NN➔telescope, 0.3
PP➔IN NP, 1.0
NP➔NN PP, 0.2
VP➔VP PP, 0.3
……



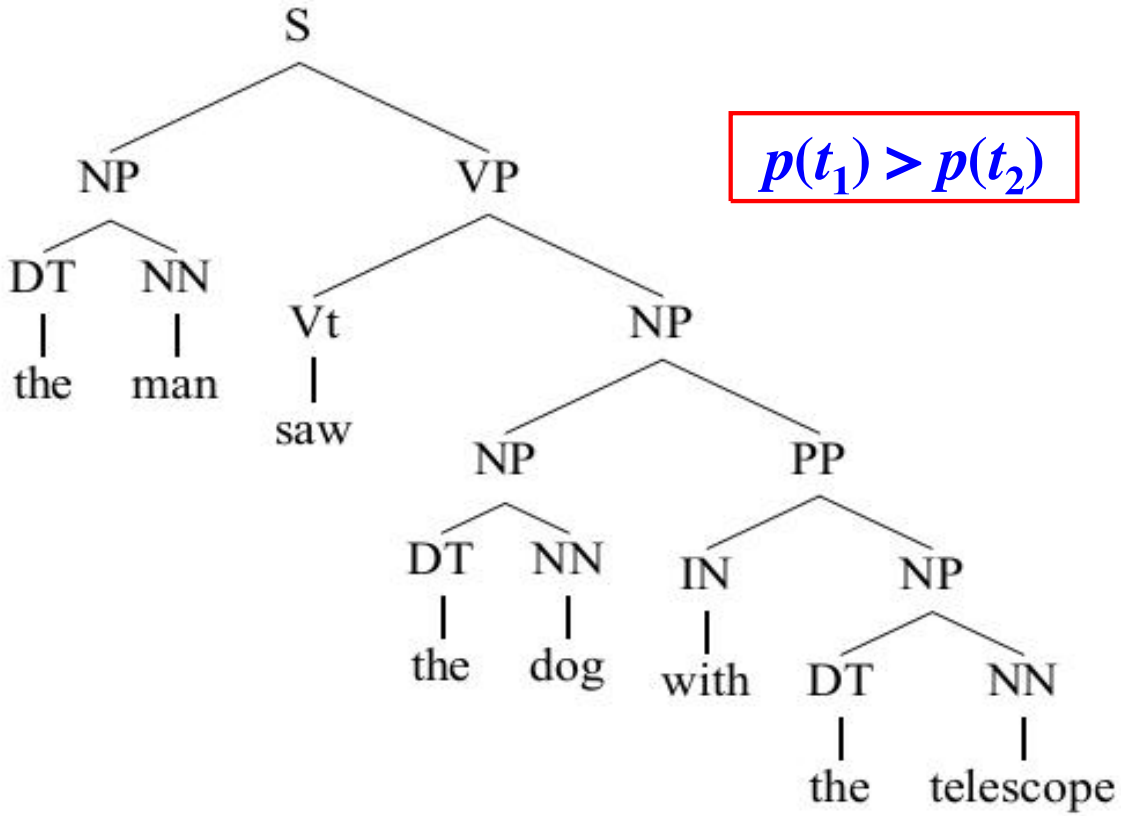
第8步及其之后的步骤

5.2 PCFG的分析实例



句法树t_1

5.2 PCFG的分析实例



$p(t_1) > p(t_2)$

句法树t_2

5.2 PCFG的分析实例

- PCFG的评价
  - 优点:
    - 可利用概率减少分析过程的搜索空间;
    - 可利用概率对概率较小的子树剪枝，加快分析效率;
    - 可以定量地比较两个语法的性能。
  - 弱点:
    - 分析树的概率计算条件非常苛刻，甚至不够合理。

5.2 基于PCFG分析方法的改进

- 代表性论文
- **Collins'Parser**
  - Michael Collins. 2003. Head-driven statistical models for natural language parsing. Computational Linguistics, 29(4): 589-637
- **Charniark Parser**
  - Eugene Charniak. 2000. A maximum-entropy-inspired parser. In Proceedings of NAACL, pp.132-139
  - Eugene Charniak and Mark Johnson. 2005. Coarse-to-fine n-best parsing and MaxEnt discriminative reranking. In Proceedings of ACL, pp.173-180
- **Bikel Parser**
  - Daniel M. Bikel. 2004. Intricacies of Collins' parsing model. Computational Linguistics, 30(4): 479-511

5.2 基于PCFG分析方法的改进

- 代表性论文
- **Stanford Parser**
  - Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In Proceedings of ACL, pp. 423-430
- **Berkeley Parser**
  - Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. Proc. NAACL-HLT, pp. 404–411
  - Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. Proc. the 21st COLING and 44th ACL, pp.433 - 440

5.2 基于PCFG分析方法的改进

- 开源句法分析器
- **Collins Parser**
  - http://people.csail.mit.edu/mcollins/code.html
- **Bikel Parser**
  - http://www.cis.upenn.edu/~dbikel/software.html#stat-parser
- **Charniak Parser**
  - http://www.cs.brown.edu/people/ec/#software
- **Oboe Parser (可执行程序)**
  - http://www.openpr.org.cn/index.php/NLP-Toolkit-for-Natural-Language-Processing/

5.2 基于PCFG分析方法的改进

- 开源句法分析器
- Berkeley Parser
  - http://nlp.cs.berkeley.edu/Main.html#Parsing
- Stanford Parser
  - http://nlp.stanford.edu/downloads/lex-parser.shtml

5.2 PCFG的三个问题

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

5.2 PCFG的三个问题

- 内向算法或外向算法解决第一个问题
- 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$w_i w_{i+1} \ldots w_j$的概率$a_{ij}(A)$。语句$W=w_1 w_2 \ldots w_n$的概率即为文法G(S)中S推导出的字串的概率$a_{1n}(S)$。

5.2 PCFG的三个问题

- Viterbi算法解决第二个问题
- Viterbi变量是由非终结符A推导出语句W中子字串$w_i w_{i+1} \ldots w_j$的最大概率。
- 变量ψ用于记忆字串$w_1 w_2 \ldots w_n$的Viterbi语法分析结果。

5.2 PCFG的三个问题

- 内外向算法解决第三个问题
- 基本思路：
  - 如果有大量已标注语法结构的训练语料，则可直接通过计算每个语法规则的使用次数，用最大似然估计方法计算 PCFG 规则的概率参数，即：

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

5.2 PCFG的三个问题

- 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，然后又可以重新估计语法参数。循环这个过程，语法参数将收敛于最大似然估计值。