

句法分析

孙栩

信息科学技术学院

xusun@pku.edu.cn

□ 上下文无关文法

- 自上而下 与 自下而上
- CKY算法

□ 概率上下文无关文法

□ 依存文法

- 投射 与 非投射
- 图方法
- 局部分类方法

□ 上下文无关文法



- 自上而下 与 自下而上
- CKY算法

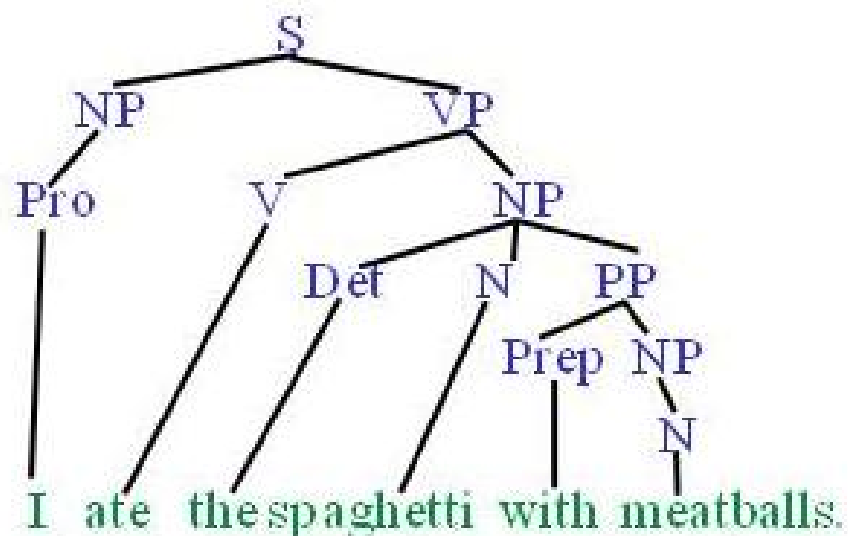
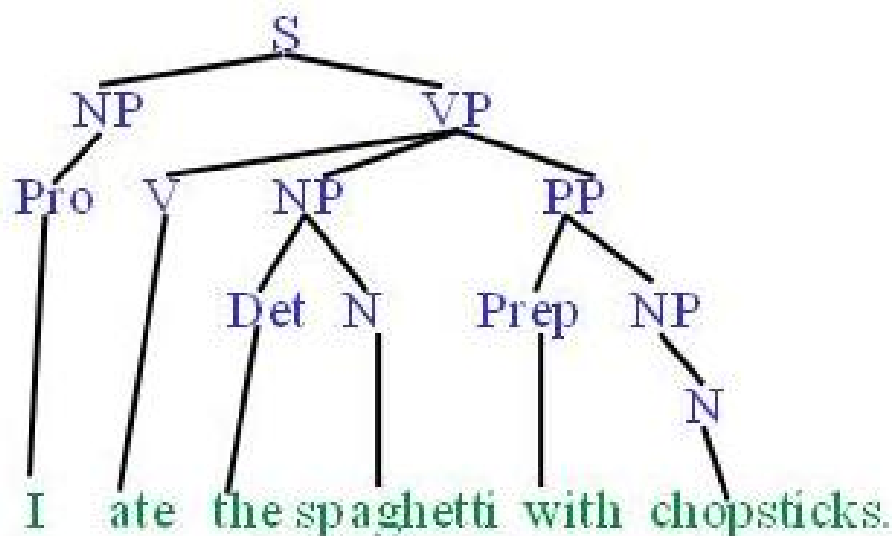
□ 概率上下文无关文法

□ 依存文法

- 投射 与 非投射
- 图方法
- 局部分类方法

句法分析(Syntactic Parsing)

- 给定一个句子，输出其正确的句法结构树



- N 是非终结符集合
- Σ 是终结符集合
- R 是一个规则集合
 - 比如一个规则是 $A \rightarrow \beta$, 其中 A 是非终结符, 而 β 可以是非终结符或终结符
- S 是一个特殊非终结符, 代表树结构的根节点, 称为 *start symbol*

语法规则

$S \rightarrow NP VP$

$S \rightarrow Aux NP VP$

$S \rightarrow VP$

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal Noun$

$Nominal \rightarrow Nominal PP$

$VP \rightarrow Verb$

$VP \rightarrow Verb NP$

$VP \rightarrow VP PP$

$PP \rightarrow Prep NP$

词汇规则

$Det \rightarrow the \mid a \mid that \mid this$

$Noun \rightarrow book \mid flight \mid meal \mid money$

$Verb \rightarrow book \mid include \mid prefer$

$Pronoun \rightarrow I \mid he \mid she \mid me$

$Proper-Noun \rightarrow Houston \mid NWA$

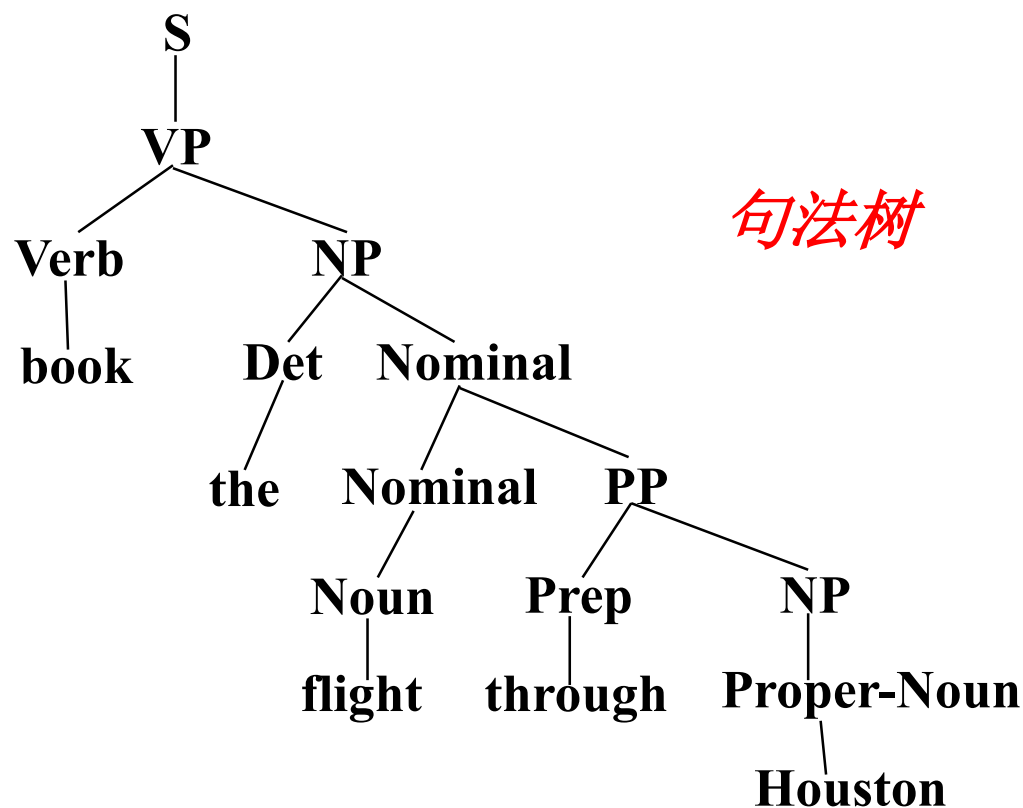
$Aux \rightarrow does$

$Prep \rightarrow from \mid to \mid on \mid near \mid through$

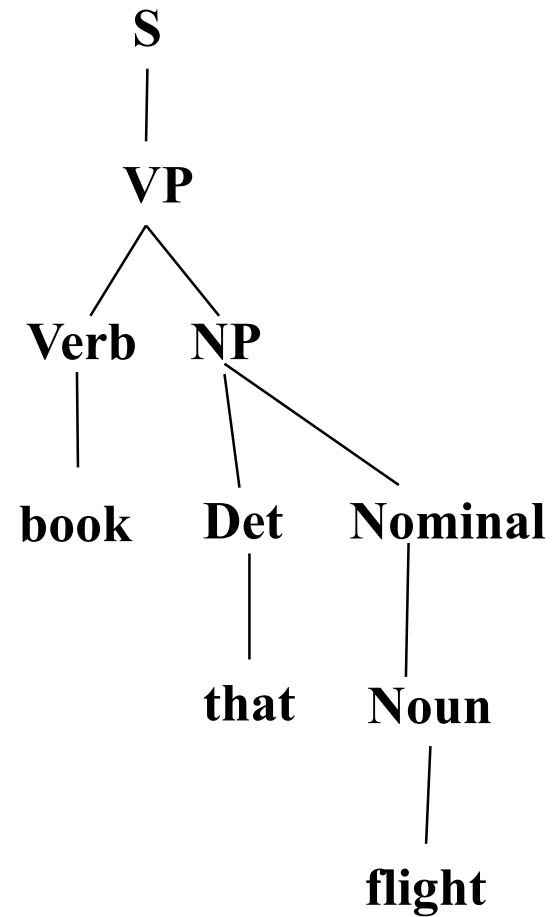
从句法分析看句子是怎么生成的

- 从句法分析的角度来看，一个句子是从根节点开始递归生成句法规则的一个过程，直到最后只有终结符存在。

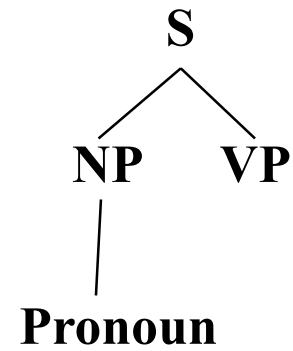
两大类方法：
自上而下句法分析
自下而上句法分析



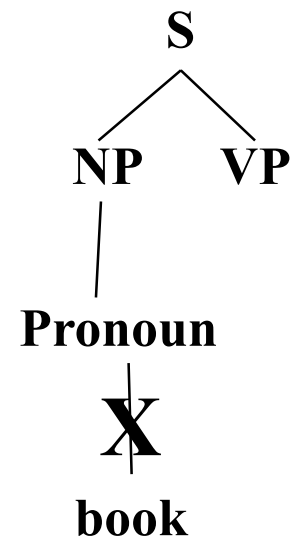
book that flight



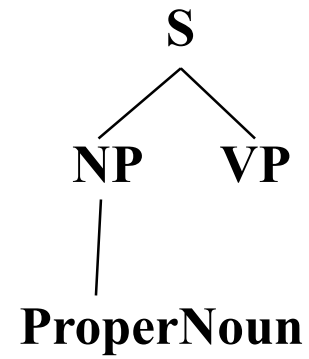
自上而下句法分析(Top Down Parsing)



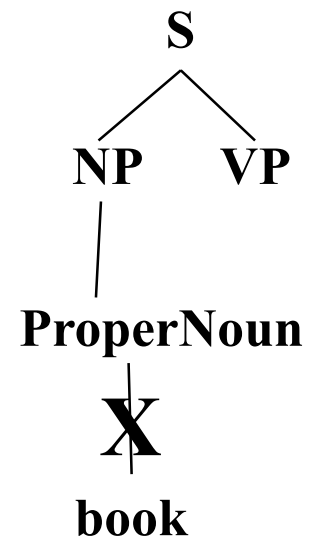
自上而下句法分析(Top Down Parsing)



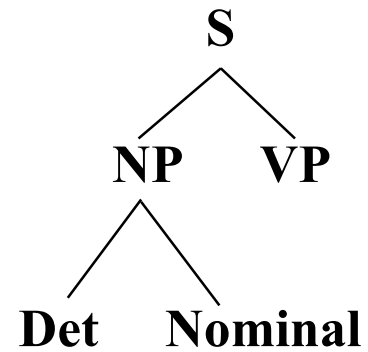
自上而下句法分析(Top Down Parsing)



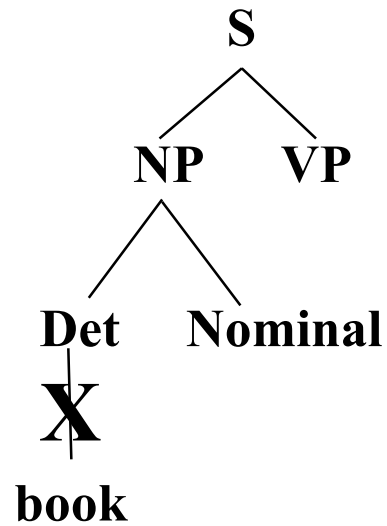
自上而下句法分析(Top Down Parsing)



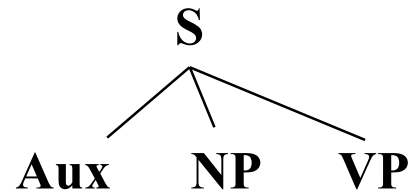
自上而下句法分析(Top Down Parsing)



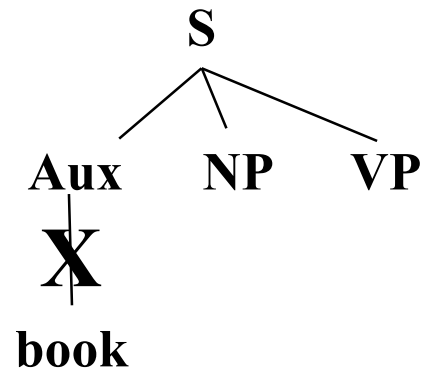
自上而下句法分析(Top Down Parsing)



自上而下句法分析(Top Down Parsing)



自上而下句法分析(Top Down Parsing)



自上而下句法分析(Top Down Parsing)

S
|
VP

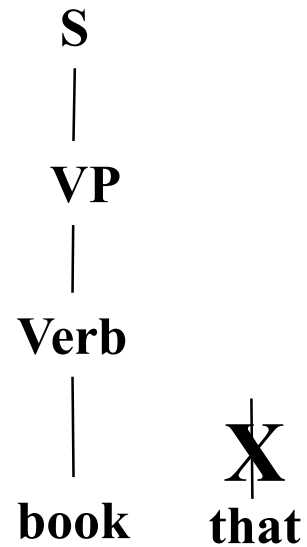
自上而下句法分析(Top Down Parsing)

S
|
VP
|
Verb

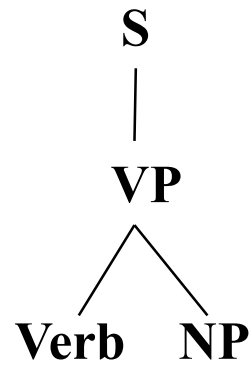
自上而下句法分析(Top Down Parsing)

S
|
VP
|
Verb
|
book

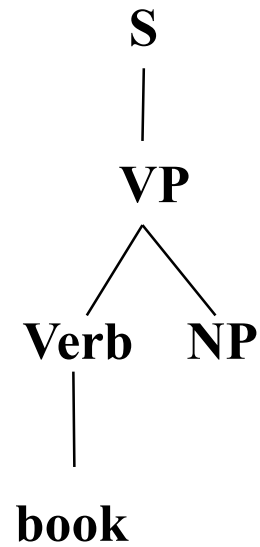
自上而下句法分析(Top Down Parsing)



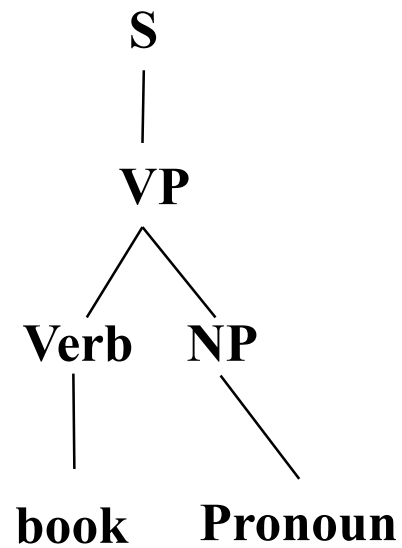
自上而下句法分析(Top Down Parsing)



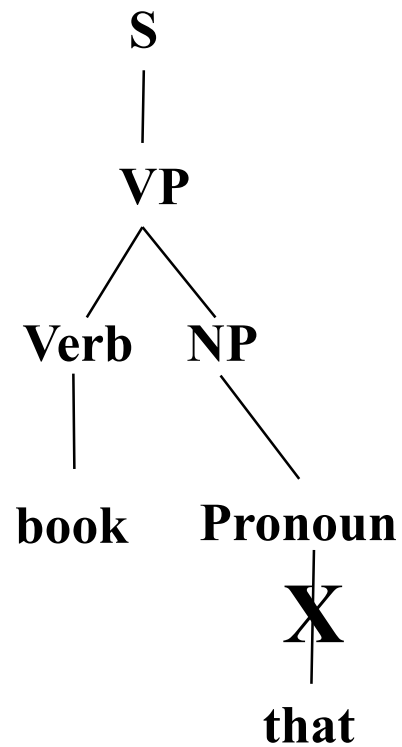
自上而下句法分析(Top Down Parsing)



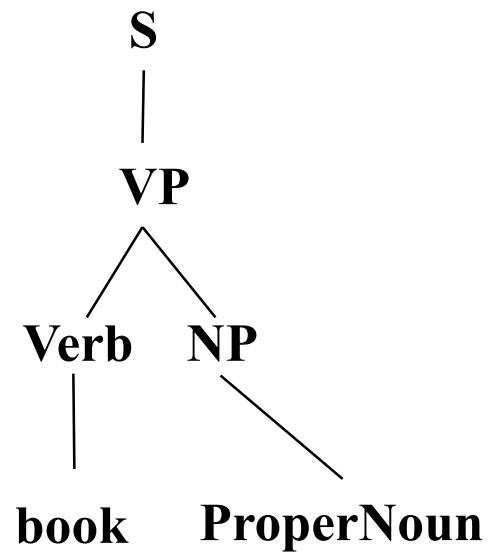
自上而下句法分析(Top Down Parsing)



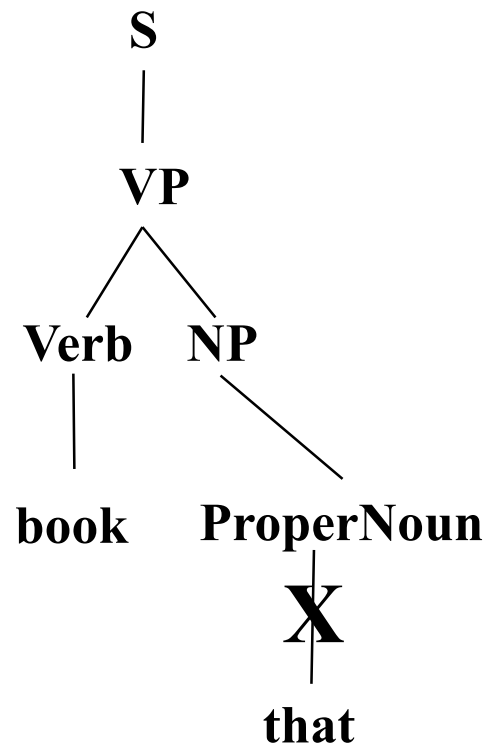
自上而下句法分析(Top Down Parsing)



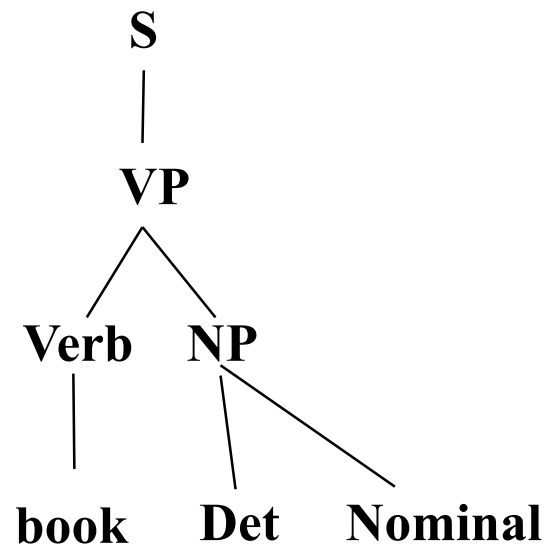
自上而下句法分析(Top Down Parsing)



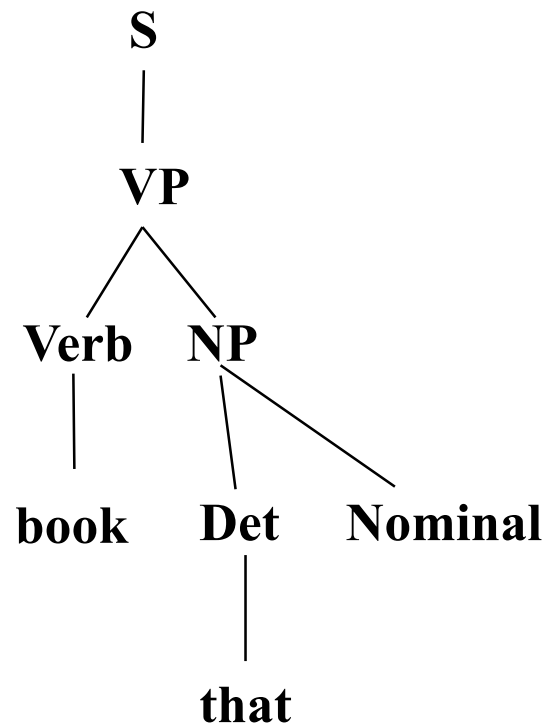
自上而下句法分析(Top Down Parsing)



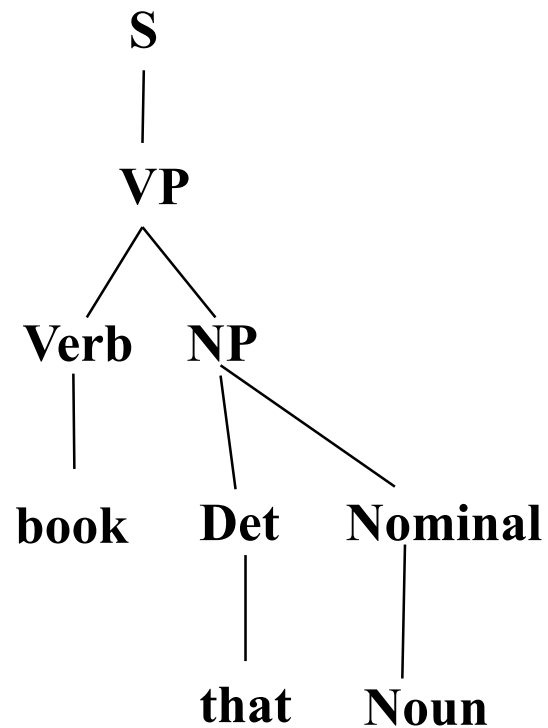
自上而下句法分析(Top Down Parsing)



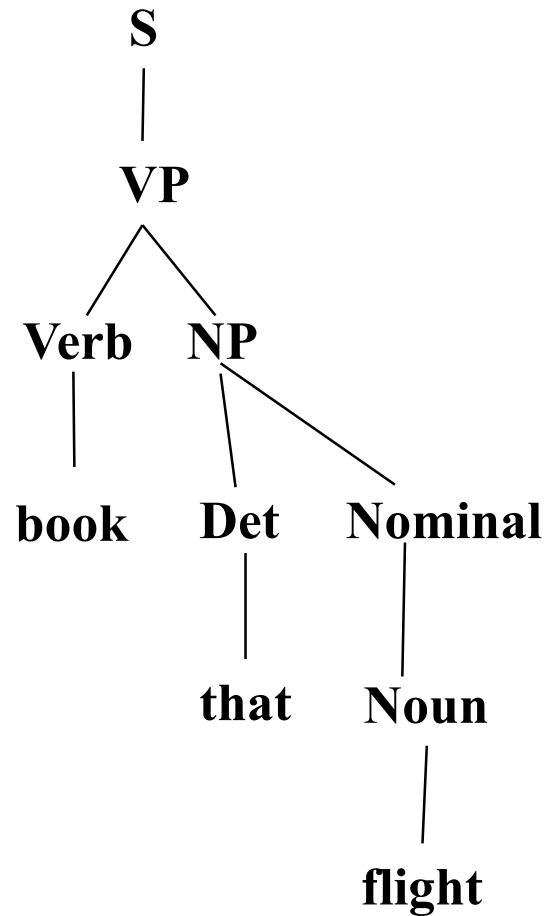
自上而下句法分析(Top Down Parsing)



自上而下句法分析(Top Down Parsing)



自上而下句法分析(Top Down Parsing)



自下而上句法分析(Bottom Up Parsing)

book that flight

自下而上句法分析(Bottom Up Parsing)

Noun

|

book

that

flight

自下而上句法分析(Bottom Up Parsing)

Nominal

|

Noun

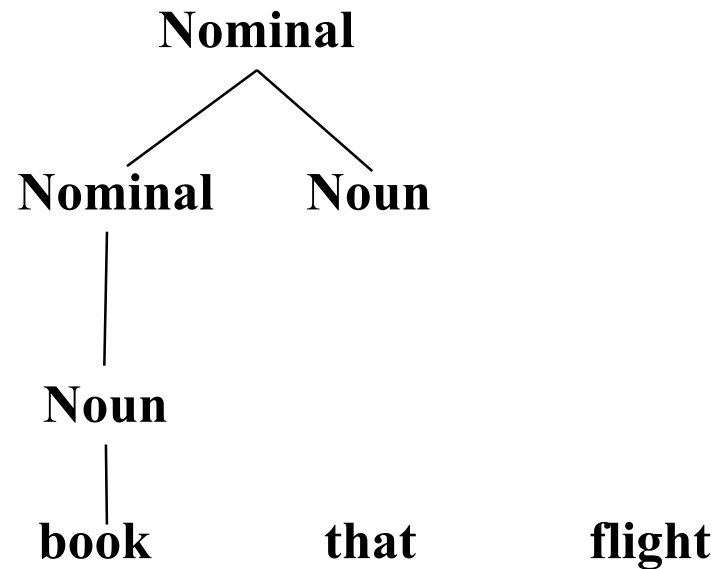
|

book

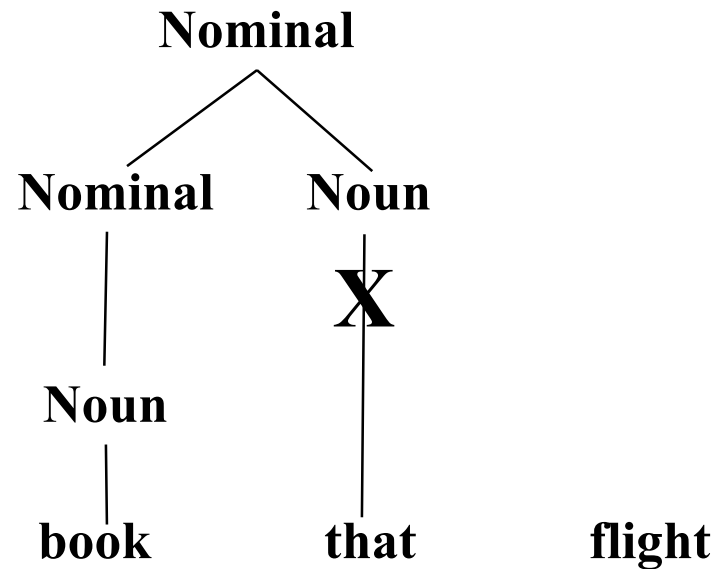
that

flight

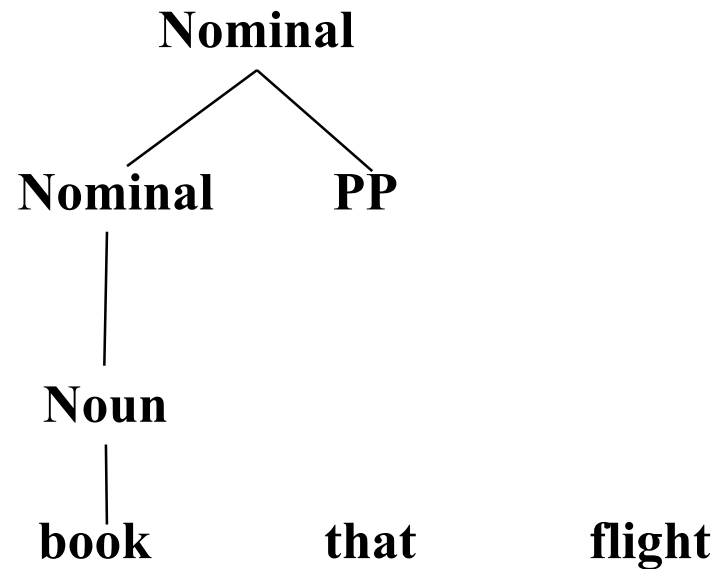
自下而上句法分析(Bottom Up Parsing)



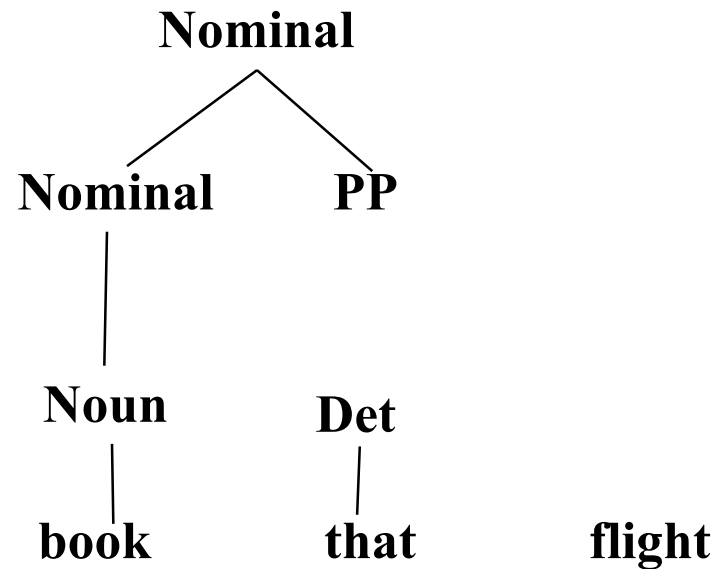
自下而上句法分析(Bottom Up Parsing)



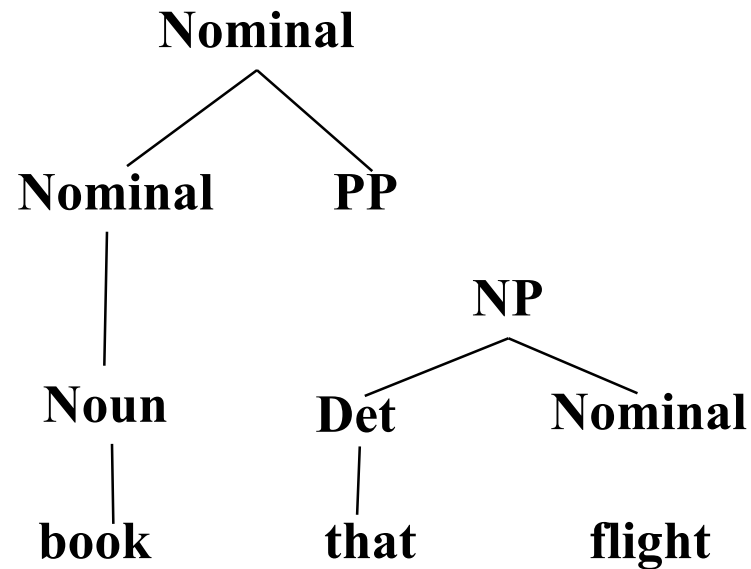
自下而上句法分析(Bottom Up Parsing)



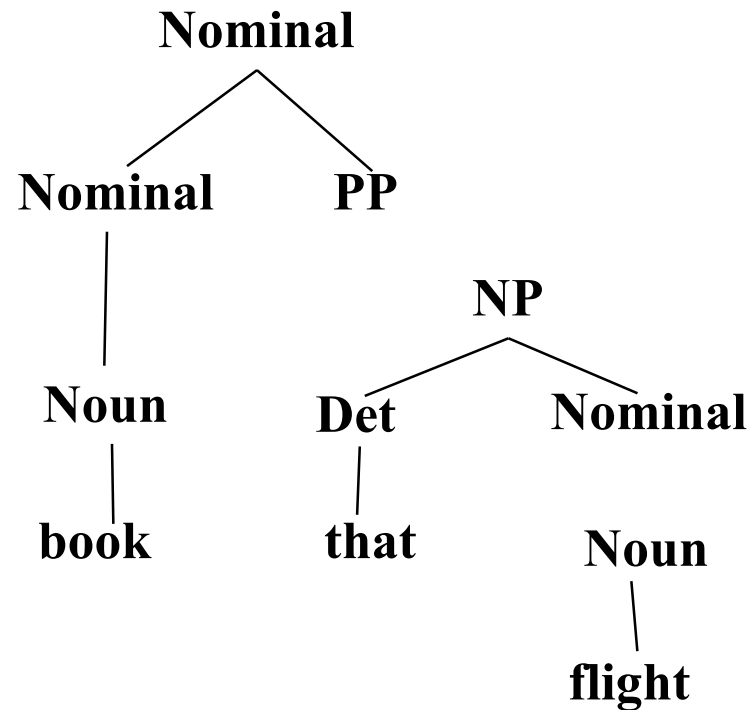
自下而上句法分析(Bottom Up Parsing)



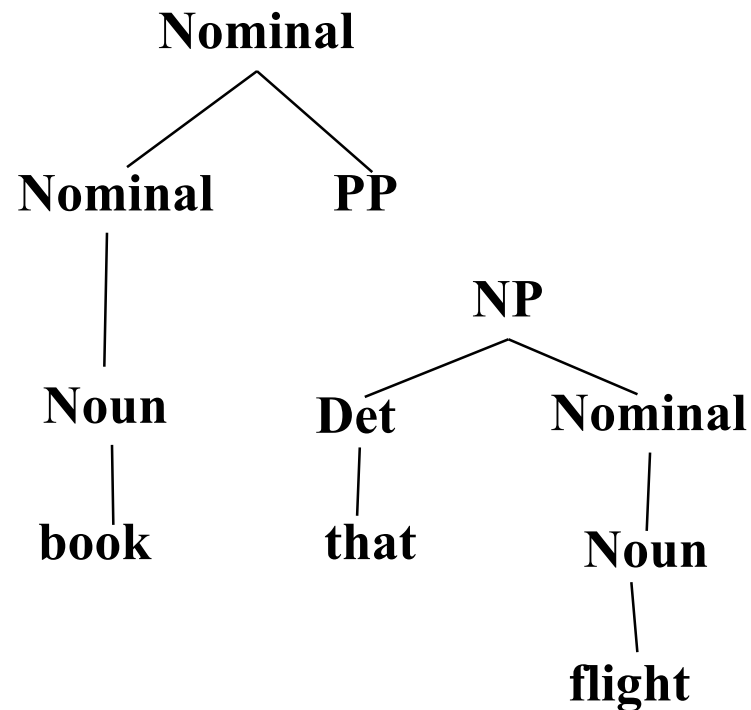
自下而上句法分析(Bottom Up Parsing)



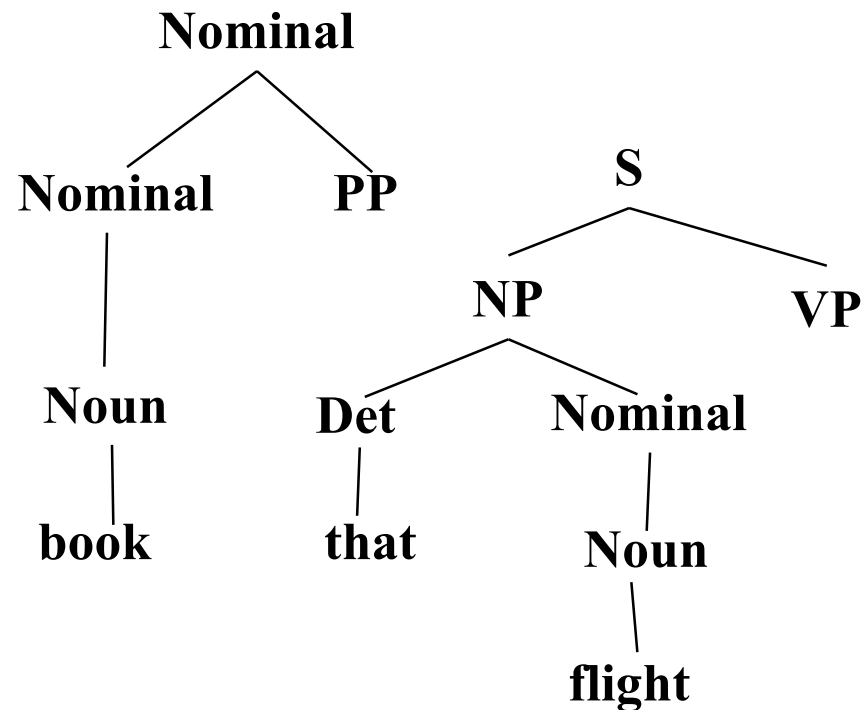
自下而上句法分析(Bottom Up Parsing)



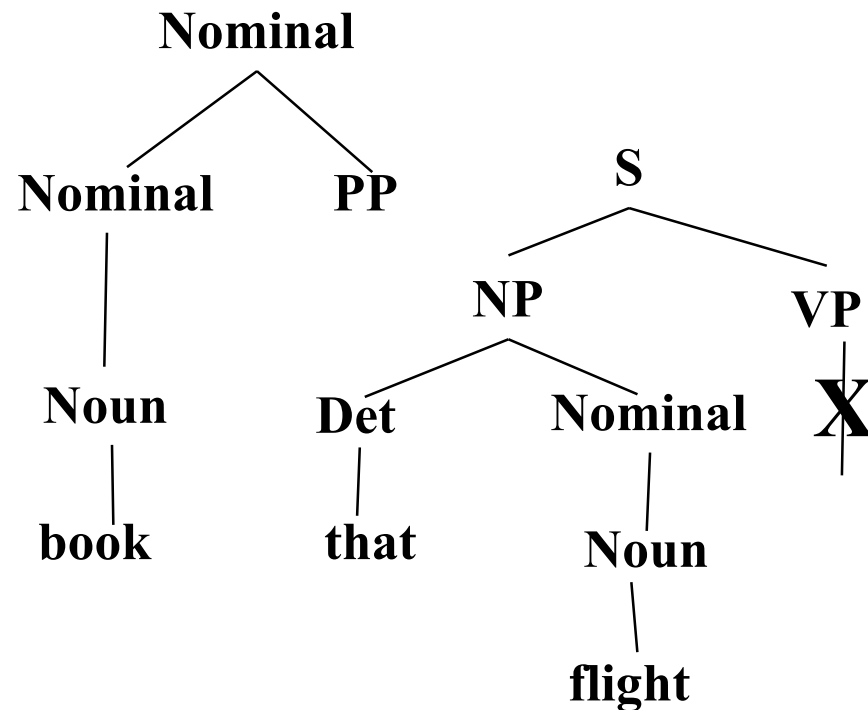
自下而上句法分析(Bottom Up Parsing)



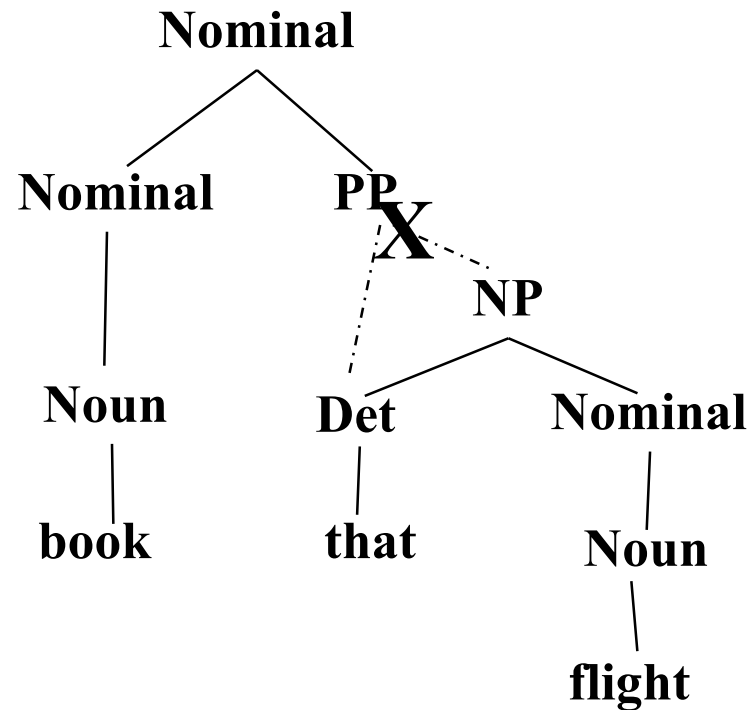
自下而上句法分析(Bottom Up Parsing)



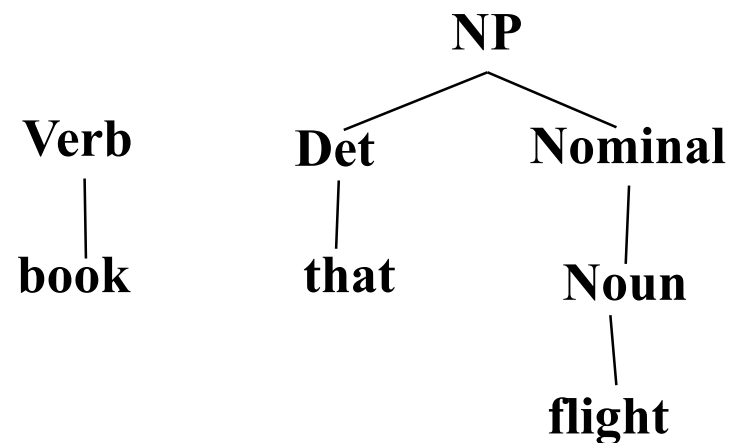
自下而上句法分析(Bottom Up Parsing)



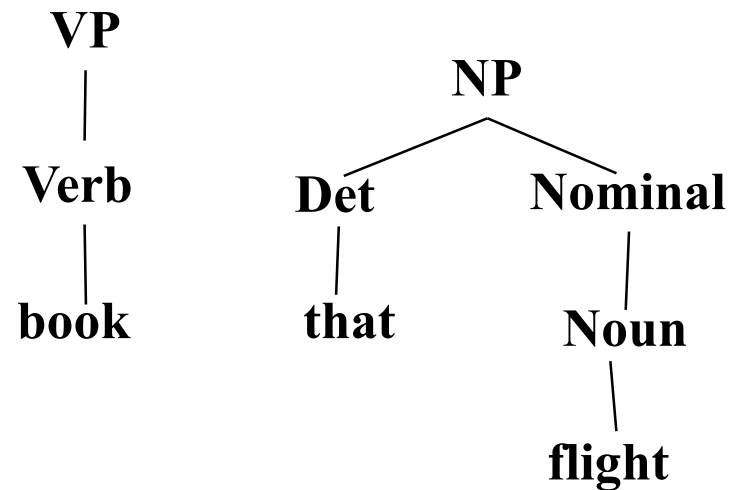
自下而上句法分析(Bottom Up Parsing)



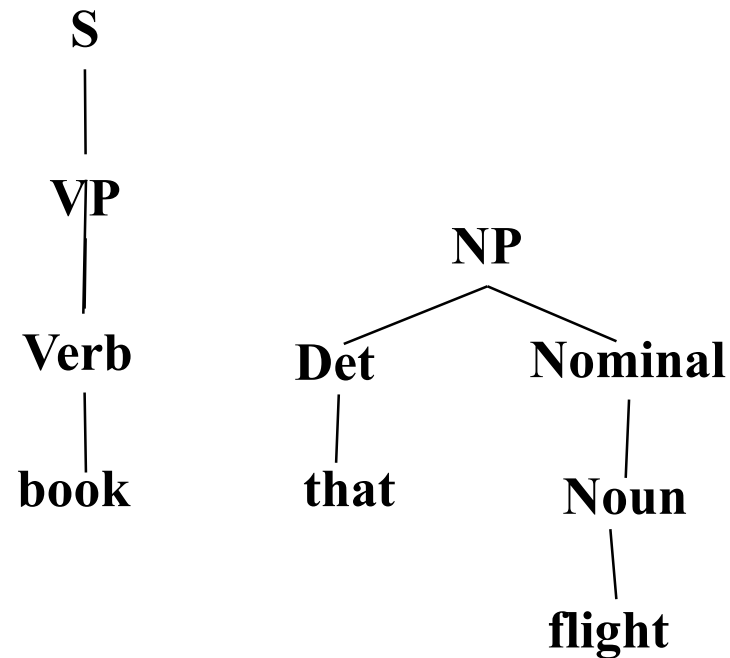
自下而上句法分析(Bottom Up Parsing)



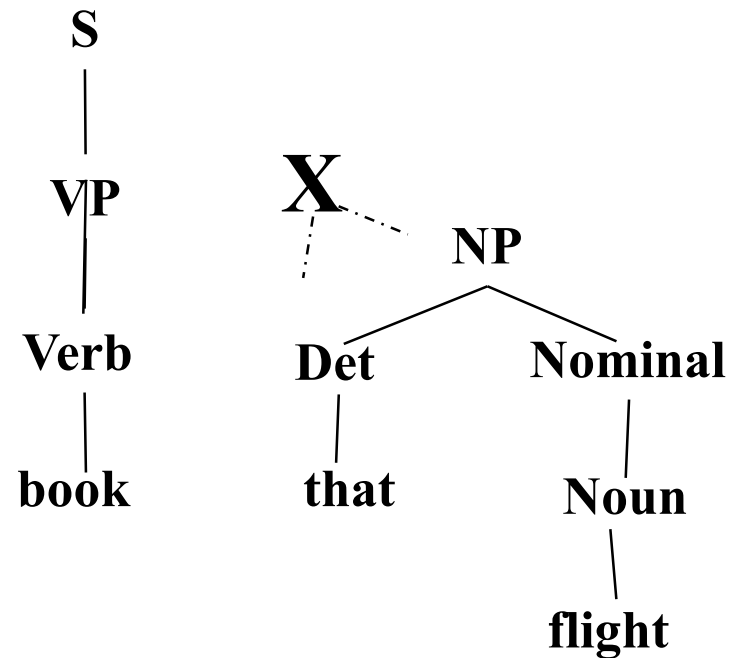
自下而上句法分析(Bottom Up Parsing)



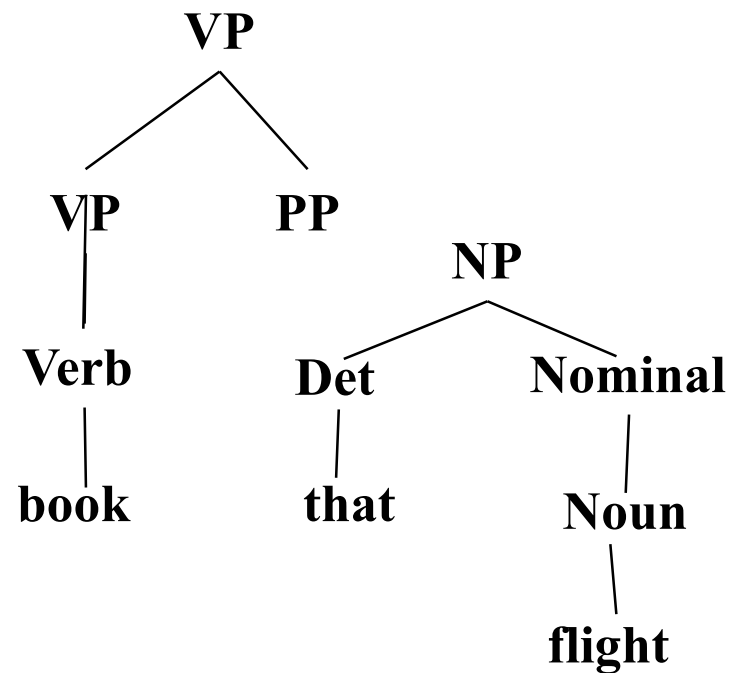
自下而上句法分析(Bottom Up Parsing)



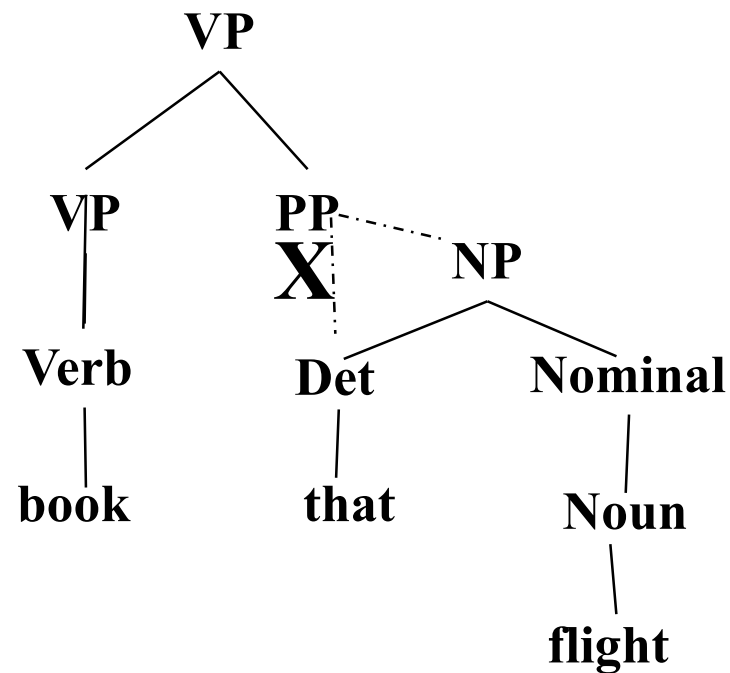
自下而上句法分析(Bottom Up Parsing)



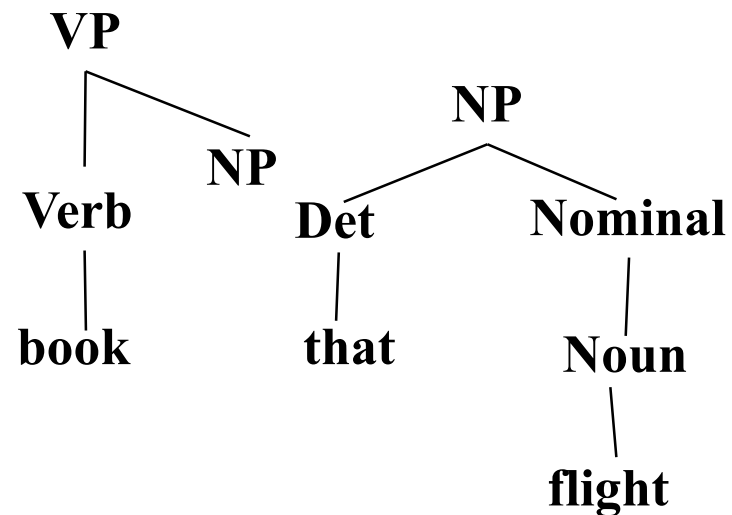
自下而上句法分析(Bottom Up Parsing)



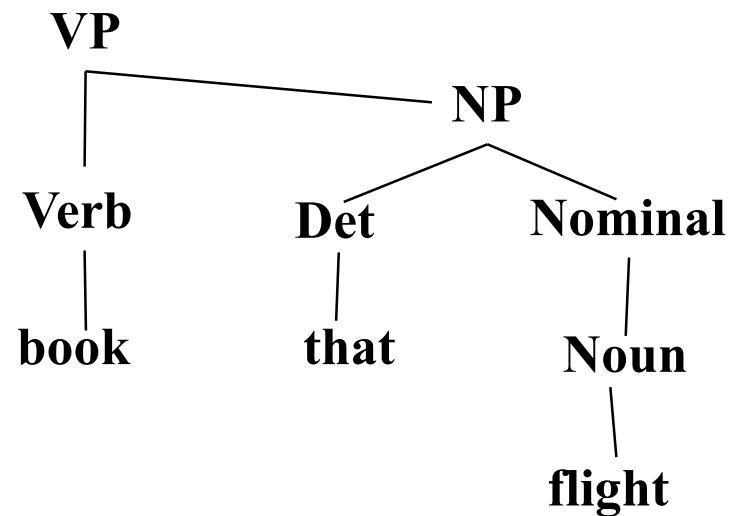
自下而上句法分析(Bottom Up Parsing)



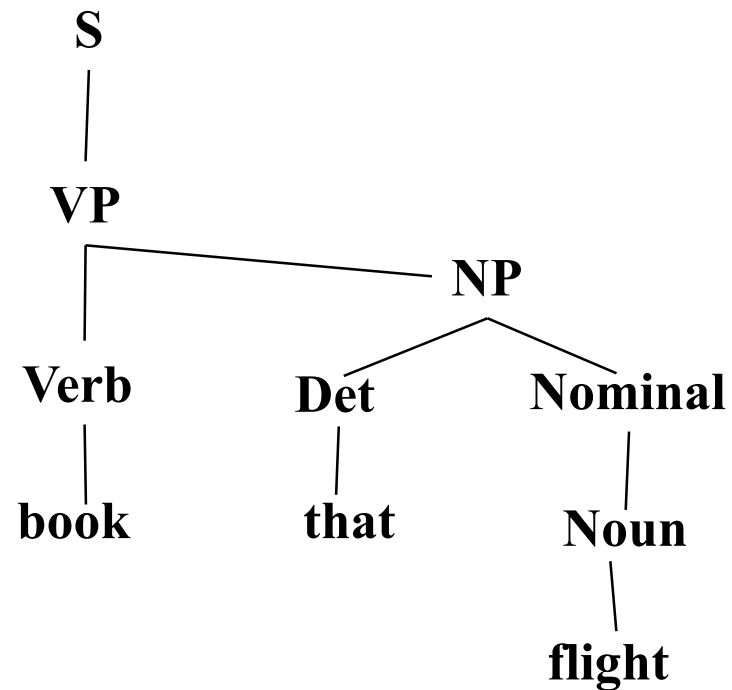
自下而上句法分析(Bottom Up Parsing)



自下而上句法分析(Bottom Up Parsing)



自下而上句法分析(Bottom Up Parsing)



- **自上而下**方法的结果总是一个完整的句法树，但是有可能无法匹配实际的句子
- **自下而上**方法总是能够匹配实际的句子，但是有可能不是一个完整的句法树
- 采用动态规划的方法，这两种方法都可以实现 $O(n^3)$ 的句法分析复杂度，**n**是句子长度

- **CKY (Cocke-Kasami-Younger)** 算法是基于自底向上的动态规划句法分析方法
- **Earley**算法是基于自顶向下的句法分析方法
- **Chart**算法在图表(**chart**)里面保存了完整的短语信息，可以把自顶向下和自底向上的方法结合起来

■ 乔姆斯基范式

- 首先，CKY方法需要把不规则语法转换成乔姆斯基范式 (Chomsky normal form, CNF)
- 在乔姆斯基范式里，一个生成规则必须生成2个非终结符，或者1个终结符

原始语法

$S \rightarrow NP VP$

$S \rightarrow Aux NP VP$

$S \rightarrow VP$

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal Noun$

$Nominal \rightarrow Nominal PP$

$VP \rightarrow Verb$

$VP \rightarrow Verb NP$

$VP \rightarrow VP PP$

$PP \rightarrow Prep NP$

乔姆斯基范式

$S \rightarrow NP VP$

$S \rightarrow X1 VP$

$X1 \rightarrow Aux NP$

$S \rightarrow book \mid include \mid prefer$

$S \rightarrow Verb NP$

$S \rightarrow VP PP$

$NP \rightarrow I \mid he \mid she \mid me$

$NP \rightarrow Houston \mid NWA$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow book \mid flight \mid meal \mid money$

$Nominal \rightarrow Nominal Noun$

$Nominal \rightarrow Nominal PP$

$VP \rightarrow book \mid include \mid prefer$

$VP \rightarrow Verb NP$

$VP \rightarrow VP PP$

$PP \rightarrow Prep NP$

CKY句法分析过程

	Book	the	flight	through	Houston
	j= 1	2	3	4	5
i= 0					
1					
2					
3					
4					

Cell[i,j]包含了
所有的从
 $i+1$ 到 j 的非终
结符信息

Book the flight through Houston

S, VP, Verb, Nominal, Noun	None				
	Det	NP			
		Nominal, Noun			

Book the flight through Houston

S, VP, Verb, Nominal, Noun				
	None	VP		
		NP		
	Det			
		Nominal, Noun		

Book the flight through Houston

S, VP, Verb, Nominal, Noun		S		
	None	VP		
		↓ NP		
	Det			
		Nominal, Noun		

Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP		
	Det	NP		
		Nominal, Noun		

Book the flight through Houston

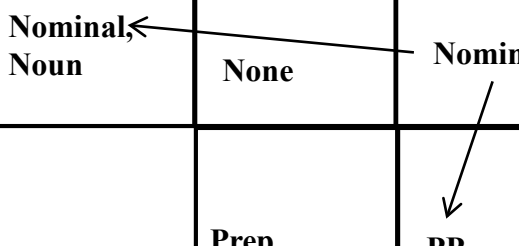
S, VP, Verb, Nominal, Noun	None	S VP	None	
	Det	NP	None	
		Nominal, Noun	None	
			Prep	

Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP	None	
	Det	NP	None	
		Nominal, Noun	None	
			Prep ← PP	
				NP ProperNoun

Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP	None	
	Det	NP	None	
		Nominal, Noun	None	Nominal
			Prep	PP
				NP ProperNoun



Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP	None	
	Det	NP	None	NP
		Nominal, Noun	None	Nominal
			Prep	PP
				NP ProperNoun

The diagram illustrates the CKY parsing process for the sentence "Book the flight through Houston". The table shows the partial parse tree structure. Arrows indicate the derivation of specific words from non-terminals:

- An arrow points from the **NP** non-terminal in the second row, second column to the word **the** (labeled as **Det**).
- An arrow points from the **NP** non-terminal in the second row, fifth column to the word **Houston** (labeled as **Nominal**).

Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP	None	VP
	Det	NP	None	NP
		Nominal, Noun	None	Nominal
			Prep	PP
				NP ProperNoun

Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP	None	S VP
	Det	NP	None	NP
		Nominal, Noun	None	Nominal
			Prep	PP
				NP ProperNoun

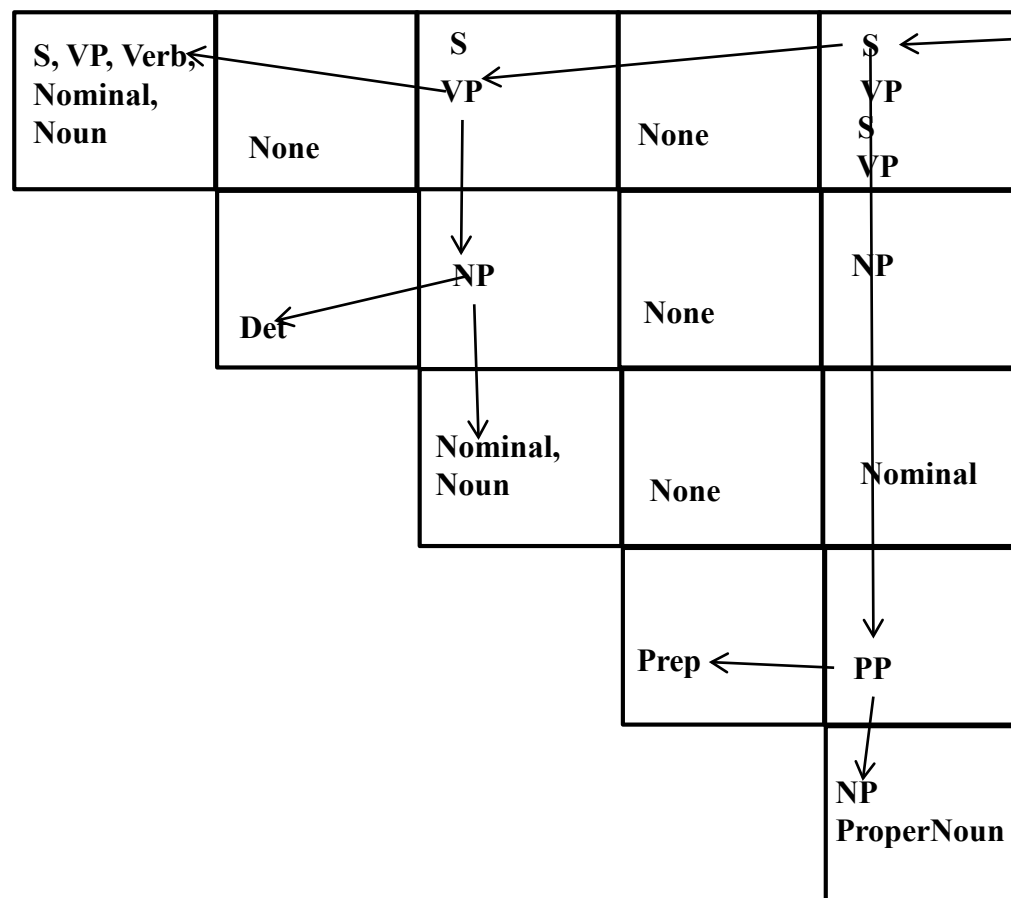
Book the flight through Houston

S, VP, Verb, Nominal, Noun	None	S VP←	None	VP \$ VP
	Det	NP	None	NP
		Nominal, Noun	None	Nominal
			Prep	PP
				NP ProperNoun

Book the flight through Houston

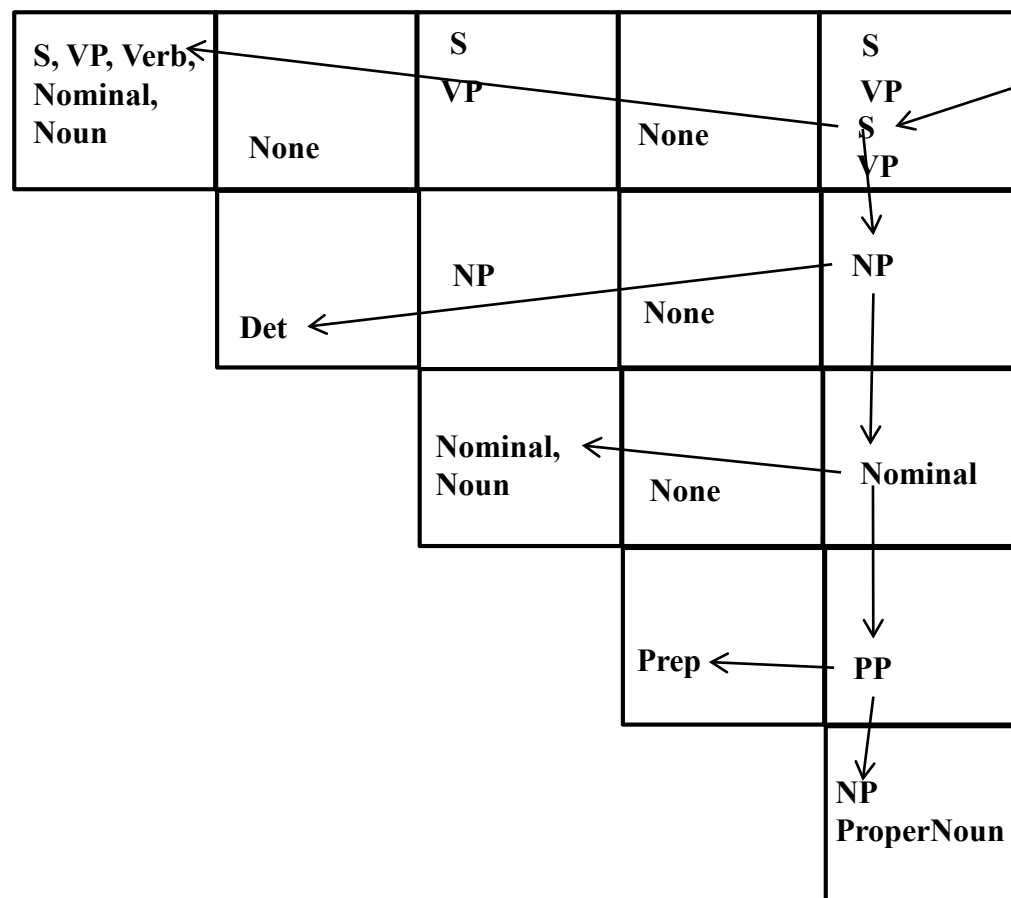
S, VP, Verb, Nominal, Noun	None	S VP ←	None	S VP S VP
	Det	NP	None	NP
		Nominal, Noun	None	Nominal
			Prep	↓ PP
				NP ProperNoun

Book the flight through Houston



句法树#1

Book the flight through Houston



句法树#2

CKY (recognition)的复杂度

- 总共有 $(n(n+1)/2) = O(n^2)$ 个单元格
- 总体的时间复杂度是 $O(n^3)$
- 分析得到的句法树 (**parse tree**) 是基于乔姆斯基范式的
- 有必要的話，可以用一个后处理过程把乔姆斯基范式转换回原来的语法规则结构

- 自动句法分析有助于计算机理解句子的意思
 - John ate the spaghetti with meatballs with chopsticks.
 - How did John eat the spaghetti?
 - What did John eat?
- 动态规划算法可以在**3**次方时间内计算一棵句法树，或指数时间内计算所有的句法树
- 问题：只是单纯的输出符合规则的句法树，无法确定哪个句法树的概率最大
 - 解决方法：基于概率的句法分析

□ 上下文无关文法

- 自上而下 与 自下而上
- CKY算法

□ 概率上下文无关文法



□ 依存文法

- 投射 与 非投射
- 图方法
- 局部分类方法

- 使用概率模型对每个句法树赋予一个概率信息
 - 通过概率信息消解句法分析中的歧义现象
 - 在标注好的树库的基础上，实现有监督学习
 - 也可以实现无监督学习，但是目前的无监督学习效果比较有限
- 基于概率的上下文无关文法 (Probabilistic Context Free Grammar, PCFG)
 - 基于概率的上下文无关文法(PCFG)是上下文无关文法(CFG)的概率版本
 - 每个生成规则都带有概率信息

语法规则

$S \rightarrow NP VP$	0.8	+ 1.0
$S \rightarrow Aux NP VP$	0.1	
$S \rightarrow VP$	0.1	
$NP \rightarrow Pronoun$	0.2	+ 1.0
$NP \rightarrow Proper-Noun$	0.2	
$NP \rightarrow Det Nominal$	0.6	
$Nominal \rightarrow Noun$	0.3	+ 1.0
$Nominal \rightarrow Nominal Noun$	0.2	
$Nominal \rightarrow Nominal PP$	0.5	
$VP \rightarrow Verb$	0.2	+ 1.0
$VP \rightarrow Verb NP$	0.5	
$VP \rightarrow VP PP$	0.3	
$PP \rightarrow Prep NP$	1.0	

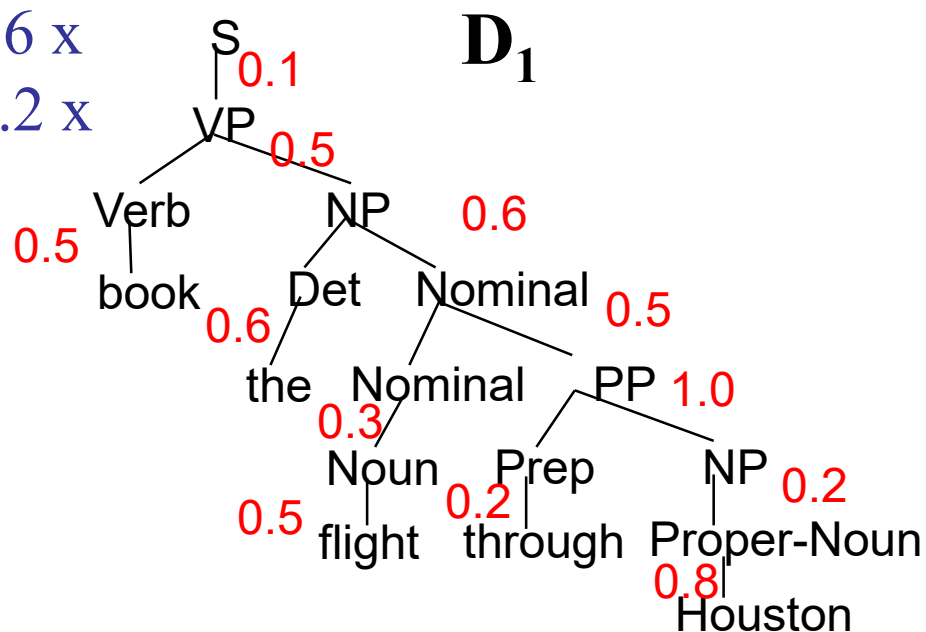
概率

词汇规则

$Det \rightarrow the \mid a \mid that \mid this$	0.6	0.2	0.1	0.1
$Noun \rightarrow book \mid flight \mid meal \mid money$	0.1	0.5	0.2	0.2
$Verb \rightarrow book \mid include \mid prefer$	0.5	0.2	0.3	
$Pronoun \rightarrow I \mid he \mid she \mid me$	0.5	0.1	0.1	0.3
$Proper-Noun \rightarrow Houston \mid NWA$	0.8	0.2		
$Aux \rightarrow does$	1.0			
$Prep \rightarrow from \mid to \mid on \mid near \mid through$	0.25	0.25	0.1	0.2
	0.2			0.2

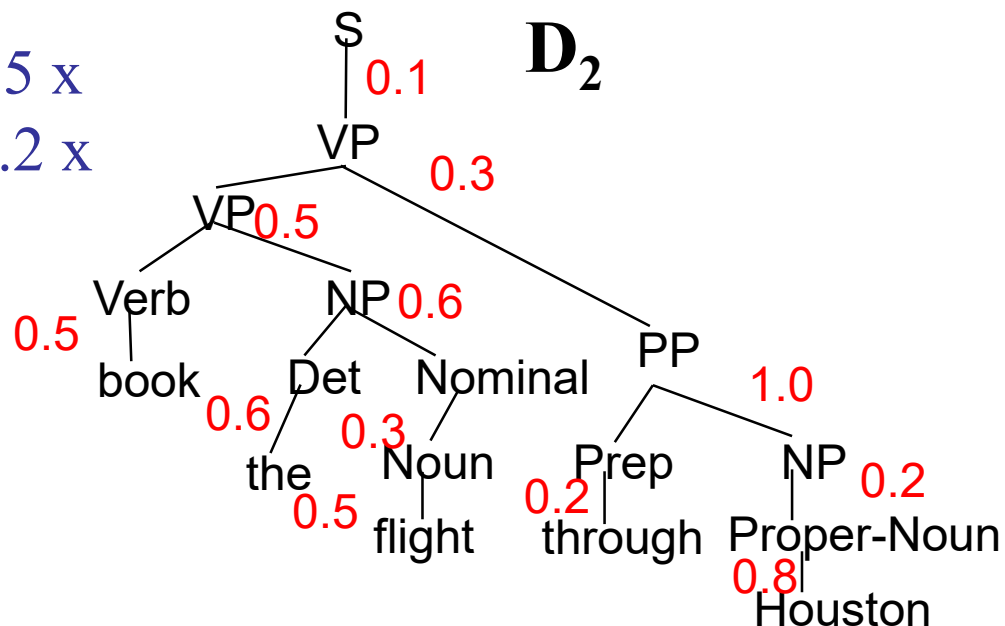
- 独立性假设
- 概率相乘

$$\begin{aligned} P(D_1) &= 0.1 \times 0.5 \times 0.5 \times 0.6 \times 0.6 \times \\ &\quad 0.5 \times 0.3 \times 1.0 \times 0.2 \times 0.2 \times \\ &\quad 0.5 \times 0.8 \\ &= 0.0000216 \end{aligned}$$



- 挑选概率最大的句法树作为句法分析的结果

$$\begin{aligned} P(D_2) &= 0.1 \times 0.3 \times 0.5 \times 0.6 \times 0.5 \times \\ &\quad 0.6 \times 0.3 \times 1.0 \times 0.5 \times 0.2 \times \\ &\quad 0.2 \times 0.8 \\ &= 0.00001296 \end{aligned}$$



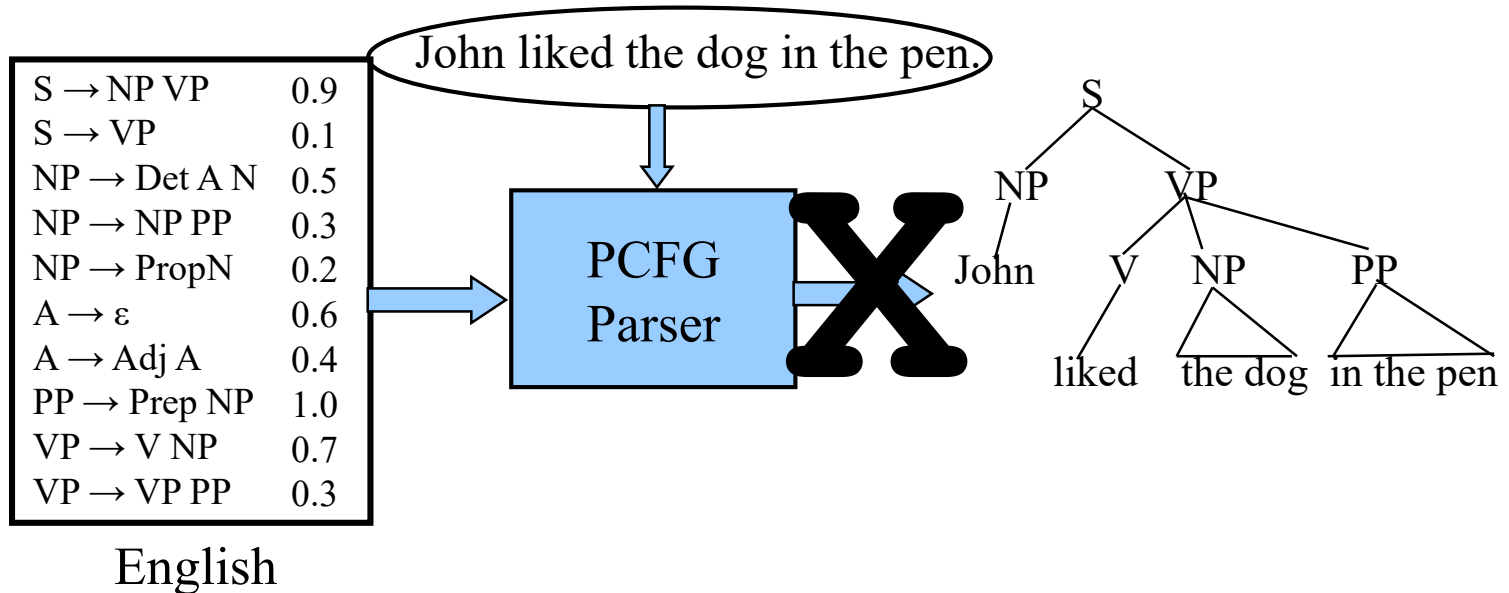
- 也可以计算句子本身的概率，句子本身的概率是其所有句法树概率之和

$$\begin{aligned} P(\text{"book the flight through Houston"}) &= \\ P(D_1) + P(D_2) &= 0.0000216 + 0.00001296 \\ &= 0.00003456 \end{aligned}$$

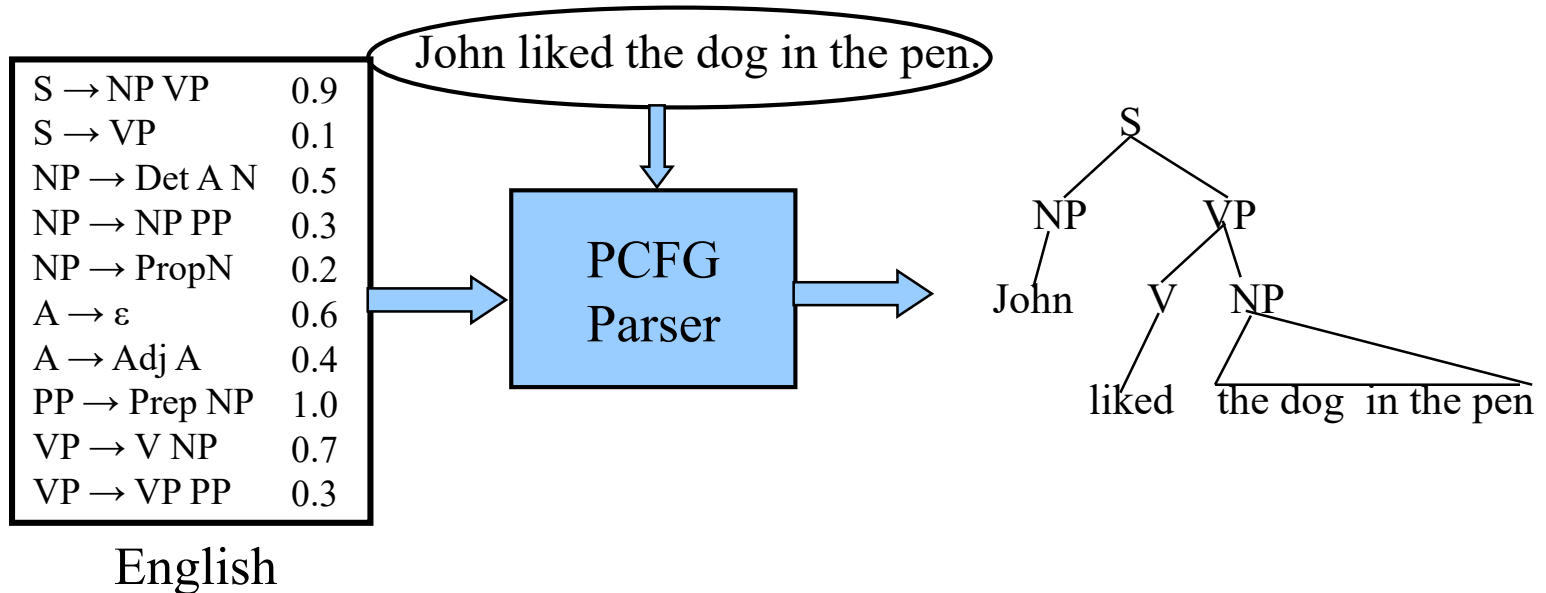
三个主要的PCFG 任务

- **观测概率** (Observation likelihood)
 - 用于对句子排序等
- **最大概率句法树** (Most likely derivation)
 - 找出最大概率句法树
- **最大似然训练** (Maximum likelihood training)
 - 基于训练数据训练一个句法分析器

- 可以使用维特比（**Viterbi**）算法确定最大可能的句法树



- 可以使用维特比（**Viterbi**）算法确定最大可能的句法树



基于概率的CKY算法 (Probabilistic CKY)

- 在原来的**CKY**算法的基础上加入概率信息
- 当转换为乔姆斯基范式的时候，需要重新设置概率信息从而保证原有的概率分布

基于概率的乔姆斯基范式转换

原始语法规则

$S \rightarrow NP VP$	0.8
$S \rightarrow Aux NP VP$	0.1
$S \rightarrow VP$	0.1
$NP \rightarrow Pronoun$	0.2
$NP \rightarrow Proper-Noun$	0.2
$NP \rightarrow Det Nominal$	0.6
$Nominal \rightarrow Noun$	0.3
$Nominal \rightarrow Nominal Noun$	0.2
$Nominal \rightarrow Nominal PP$	0.5
$VP \rightarrow Verb$	0.2
$VP \rightarrow Verb NP$	0.5
$VP \rightarrow VP PP$	0.3
$PP \rightarrow Prep NP$	1.0

乔姆斯基范式

$S \rightarrow NP VP$	0.8
$S \rightarrow X1 VP$	0.1
$X1 \rightarrow Aux NP$	1.0
$S \rightarrow book \mid include \mid prefer$ 0.01 0.004 0.006	
$S \rightarrow Verb NP$	0.05
$S \rightarrow VP PP$	0.03
$NP \rightarrow I \mid he \mid she \mid me$ 0.1 0.02 0.02 0.06	
$NP \rightarrow Houston \mid NWA$ 0.16 .04	
$NP \rightarrow Det Nominal$	0.6
$Nominal \rightarrow book \mid flight \mid meal \mid money$ 0.03 0.15 0.06 0.06	
$Nominal \rightarrow Nominal Noun$	0.2
$Nominal \rightarrow Nominal PP$	0.5
$VP \rightarrow book \mid include \mid prefer$ 0.1 0.04 0.06	
$VP \rightarrow Verb NP$	0.5
$VP \rightarrow VP PP$	0.3
$PP \rightarrow Prep NP$	1.0

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None			
	Det:.6	NP:.6*.6*.15 =.054		
		Nominal:.15 Noun:.5		

基于概率的CKY算法

Book the flight through Houston

S :.01, VP:.1, Verb:.5 ← Nominal:.03 Noun:.1	None	VP:.5*.5*.054 =.0135		
	Det:.6	NP:.6*.6*.15 =.054		
		Nominal:.15 Noun:.5		

Book the flight through Houston

S :.01, VP:.1, Verb:.5 ← Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135		
	Det:.6	NP:.6*.6*.15 =.054		
		Nominal:.15 Noun:.5		

基于概率的CKY算法

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	
	Det:.6	NP:.6*.6*.15 =.054	None	
		Nominal:.15 Noun:.5	None	
			Prep:.2	

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	
	Det:.6	NP:.6*.6*.15 =.054	None	
		Nominal:.15 Noun:.5	None	
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

基于概率的CKY算法

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	
	Det:.6	NP:.6*.6*.15 =.054	None	
		Nominal:.15 Noun:.5	None	Nominal: .5*.15*.032 =.0024
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	
	Det:.6 ←	NP:.6*.6*.15 =.054	None	NP:.6*.6* .0024 =.000864
		Nominal:.15 Noun:.5	None	Nominal: .5*.15*.032 =.0024
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

Book the flight through Houston

S :.01, VP:.1, Verb:. 5 Nominal:.03 Noun:.1		S:.05*.5*.054 =.00135		S:.05*.5* .000864 =.0000216
	None	VP:.5*.5*.054 =.0135	None	
				NP:.6*.6* .0024 =.000864
	Det:.6	NP:.6*.6*.15 =.054	None	
		Nominal:.15 Noun:.5	None	Nominal: .5*.15*.032 =.0024
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

基于概率的CKY算法

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	S:.03*.0135* .032 =.00001296 S:.0000216
	Det:.6	NP:.6*.6*.15 =.054	None	NP:.6*.6* .0024 =.000864
		Nominal:.15 Noun:.5	None	Nominal: .5*.15*.032 =.0024
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

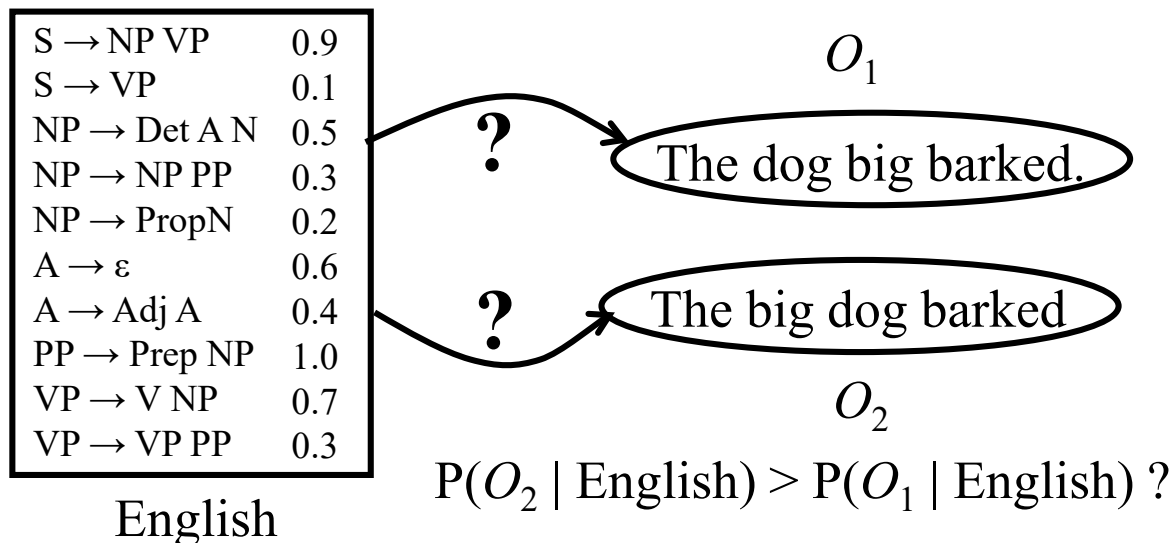
Book the flight through Houston

S :.01, VP:.1, Verb:. 5 Nominal:.03 Noun:.1		S:. $.05*.5*.054$ =.00135		S:. $.0000216$
	None	VP:. $.5*.5*.054$ =.0135	None	
				NP:. $.6*.6*$.0024 =.000864
	Det:. 6	NP:. $.6*.6*.15$ =.054	None	
				Nominal: .5*.15*.032 =.0024
		Nominal:.15 Noun:.5	None	
			Prep:. 2	PP:. $1.0*.2*.16$ =.032
				NP:.16 PropNoun:. 8

计算最大概率的句法树

PCFG: 观测概率的计算(Observation Likelihood)

- 类似于序列标注问题中的观测概率的计算方法——前向算法(**Forward algorithm**), 可以使用 **Inside algorithm** 这个算法进行动态规划计算观测概率
 - 把Viterbi算法的max计算改为sum计算
- 可以使用**PCFG**作为一个语言模型, 从而对句子的概率进行计算, 用于语音识别、机器翻译等



PCFG: 观测概率的计算(Observation Likelihood)

Probabilistic CKY Parser for Inside Computation

Book the flight through Houston

S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	S:...00001296 S:.0000216
	Det:.6	NP:.6*.6*.15 =.054	None	NP:.6*.6* .0024 =.000864
		Nominal:.15 Noun:.5	None	Nominal: .5*.15*.032 =.0024
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

PCFG: 观测概率的计算(Observation Likelihood)

Probabilistic CKY Parser for Inside Computation

Book the flight through Houston

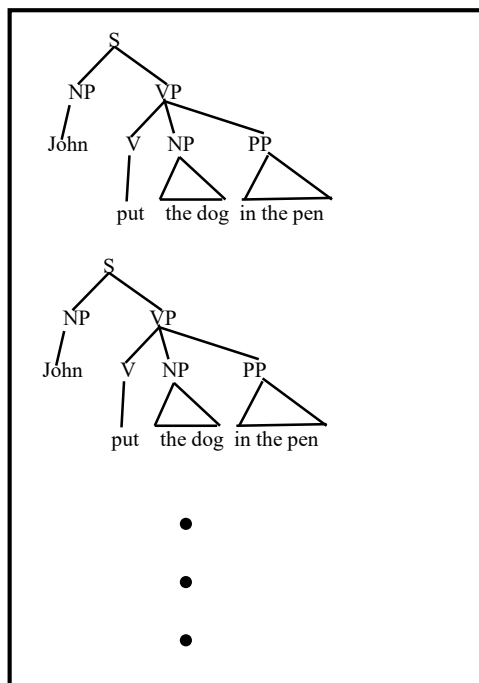
S :.01, VP:.1, Verb:.5 Nominal:.03 Noun:.1	None	S:.05*.5*.054 =.00135 VP:.5*.5*.054 =.0135	None	S: .00001296 +.0000216 =.00003456
	Det:.6	NP:.6*.6*.15 =.054	None	NP:.6*.6* .0024 =.000864
		Nominal:.15 Noun:.5	None	Nominal: .5*.15*.032 =.0024
			Prep:.2	PP:1.0*.2*.16 =.032
				NP:.16 PropNoun:. 8

计算观测概率

PCFG: 有监督学习(Supervised Training)

- 给定了训练数据的话（一般是标注好的树库**tree bank**），**PCFG**的有监督学习相对比较简单
 - 可以通过相对频率来计算

Tree Bank



Supervised
PCFG
Training

$S \rightarrow NP VP$	0.9
$S \rightarrow VP$	0.1
$NP \rightarrow Det A N$	0.5
$NP \rightarrow NP PP$	0.3
$NP \rightarrow PropN$	0.2
$A \rightarrow \epsilon$	0.6
$A \rightarrow Adj A$	0.4
$PP \rightarrow Prep NP$	1.0
$VP \rightarrow V NP$	0.7
$VP \rightarrow VP PP$	0.3

English

- 可以从树库收集语法规则
- 语法规则对应的概率可以通过相对频率来计算

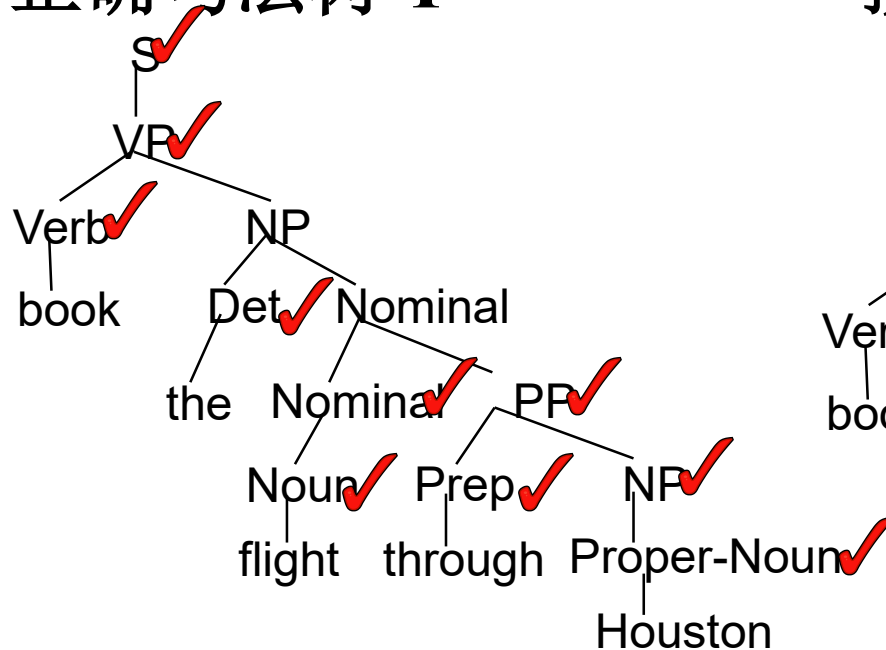
$$P(\alpha \rightarrow \beta \mid \alpha) = \frac{\text{count}(\alpha \rightarrow \beta)}{\sum_{\gamma} \text{count}(\alpha \rightarrow \gamma)} = \frac{\text{count}(\alpha \rightarrow \beta)}{\text{count}(\alpha)}$$

句法分析的效果打分

- 句法分析器给出了预测的句法树之后，可以计算句法树和已经标注好的句法树的相似度
- 假设 **P** 是系统输出的句法树，假设 **T** 是标注好的句法树：
 - **召回率(Recall)** = $(\#P \text{中正确的元素}) / (\#T \text{中正确的元素})$
 - **准确率(Precision)** = $(\#P \text{中正确的元素}) / (\#P \text{中总的元素})$
- **F值(F-score, F1)**是召回率和准确率之间的调和平均数 (harmonic mean)

目前好的句法分析系统在标准数据集上可以达到**90%**以上的Precision, Recall, F-score

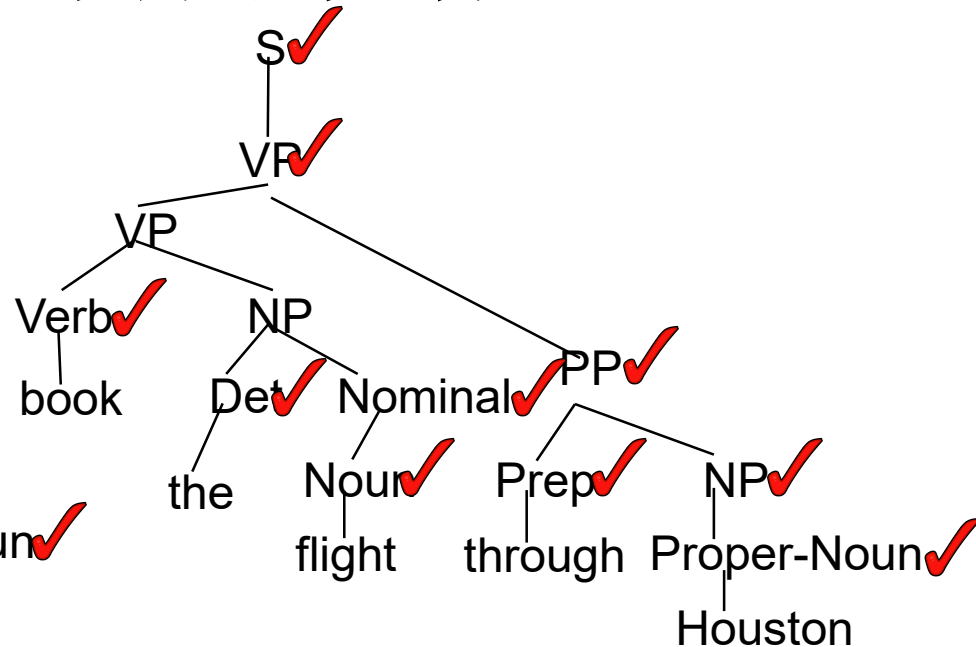
正确句法树 T



元素: 12

正确的元素: 10

预测的句法树P

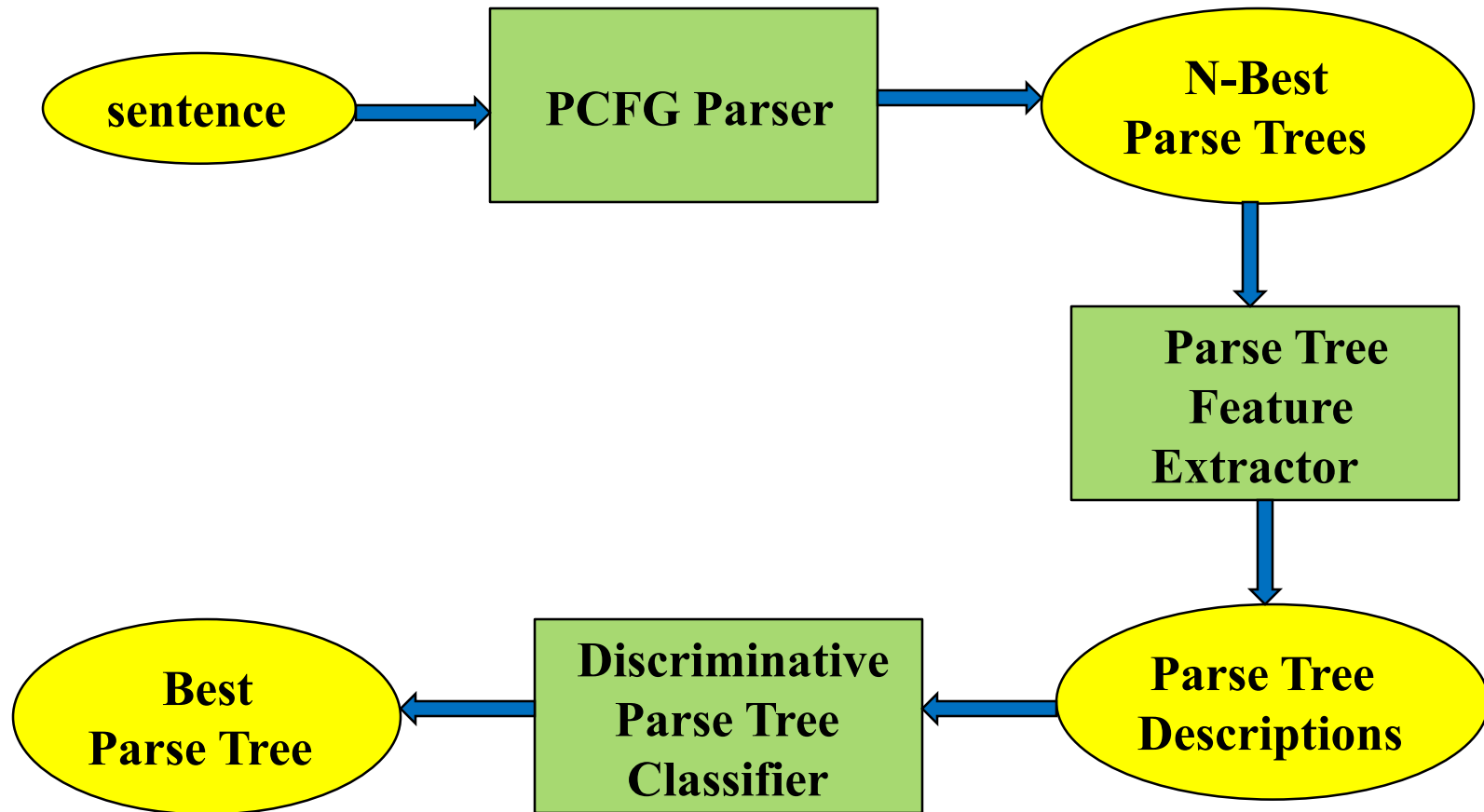


元素: 12

Recall = $10/12 = 83.3\%$ Precision = $10/12 = 83.3\%$

$F_1 = 83.3\%$

基于重排序的句法分析(re-ranking parser)



- 通过概率信息，消解句法分析中的歧义，得到最大概率的句法分析树等
- 通过标注好的树库，可以学习到概率句法分析器
- 现有的概率句法分析技术已经有很高的准确度

□ 参考书

- 《统计自然语言处理》 第8章：句法分析
 - 8.1 句法结构分析概述
 - Page 179 – 184
 - 8.2 基于PCFG的基本分析方法
 - Page 184 - 192

□ 上下文无关文法

- 自上而下 与 自下而上
- CKY算法

□ 概率上下文无关文法

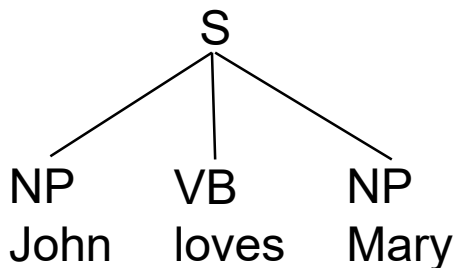
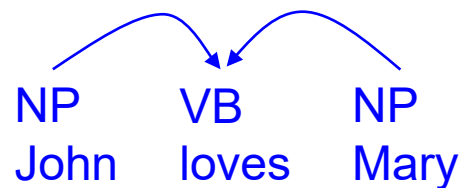
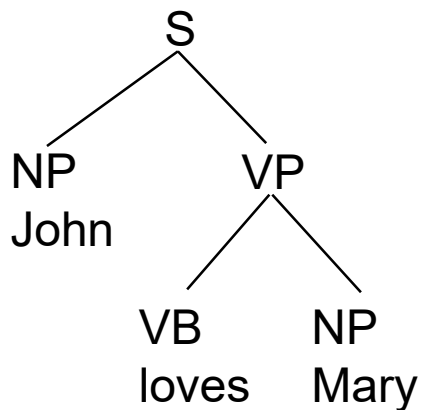
□ 依存文法



- 投射 与 非投射
- 图方法
- 局部分类方法

□ 输出的类别

- ▣ 传统句法分析模式（如之前讲的上下文无关文法CFG、PCFG），依存分析模式



□ 依存句法分析(Dependency-based grammar parsing)

- 根据依存语法规则 G , 给定一个输入字符串 $x \in \Sigma^*$, 计算 x 的依存句法树 y

□ 也有依存篇章分析(Dependency-based text parsing)

依存语法 (Dependency Grammar)

□ 依存语法

- 依存语法 (Dependency grammar, DG) 是相对比较晚出现的语法分析理论, 核心是基于依存句法关系, 可以追溯到Lucien Tesnière提出的相关依存句法理论
- 基本思想: 用若干基于词对的非对称依存箭头来刻画句法结构

□ 依存关系

- 依存句法关系是以动词为依存结构的核心, 一般把动词作为依存关系的头节点 (head)
- 依存结构取决于头节点 (head) 和附属节点 (dependents)之间的依存关系、类别

□ 动词为中心的思想

- 给定一个句子, 认为所有的句法单元(e.g. 词) 要么直接或者间接地依赖于某个动词

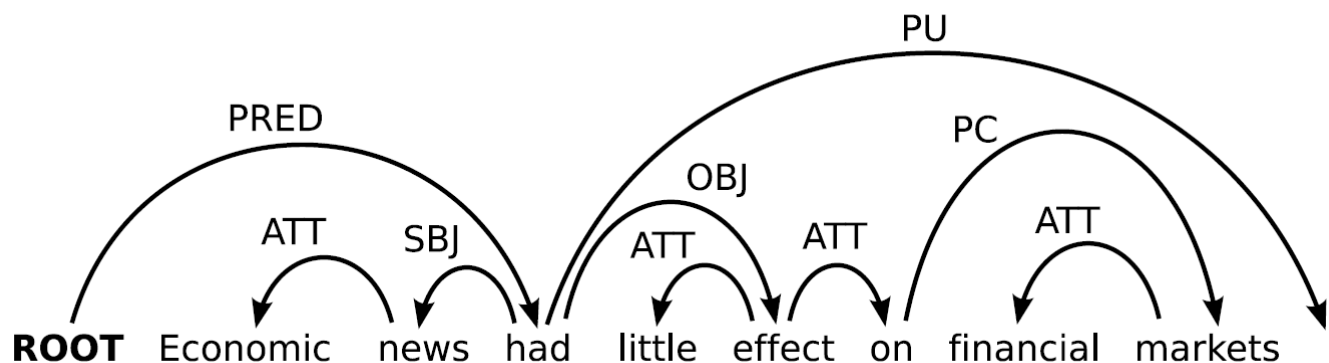
□ 依存语法的优点

- 很适合对词序比较自由、灵活的语言进行句法结构分析
- 适合捕捉长距离的句法结构依存关系

□ 有向图(Directed graphs)

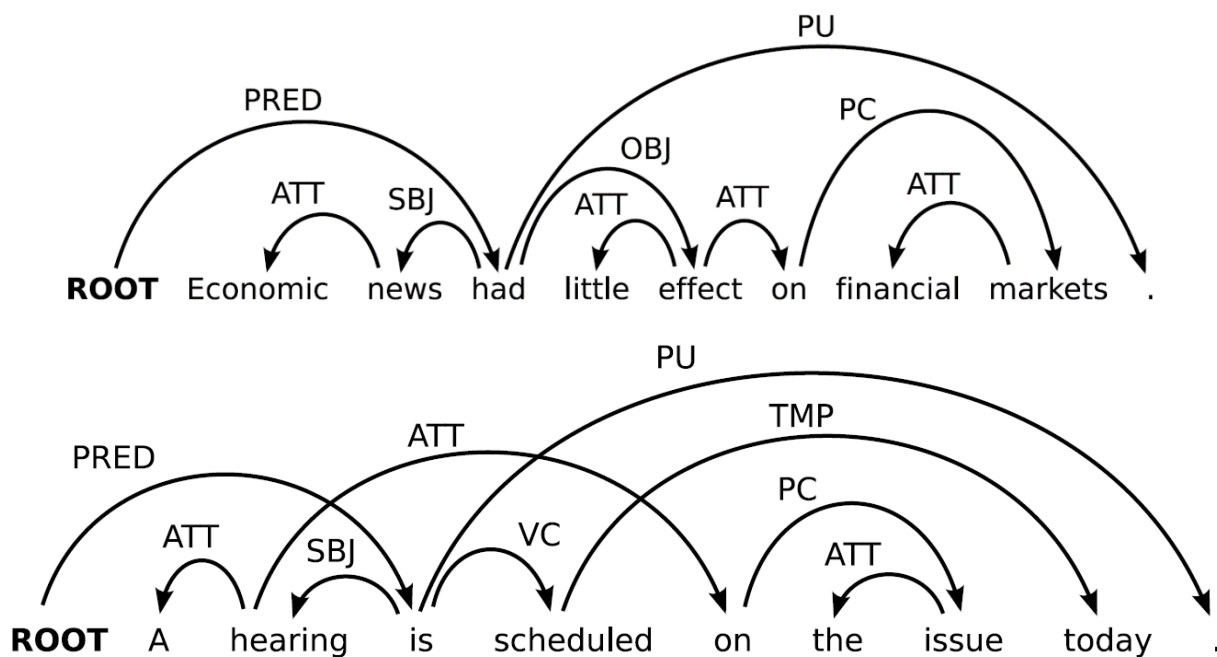
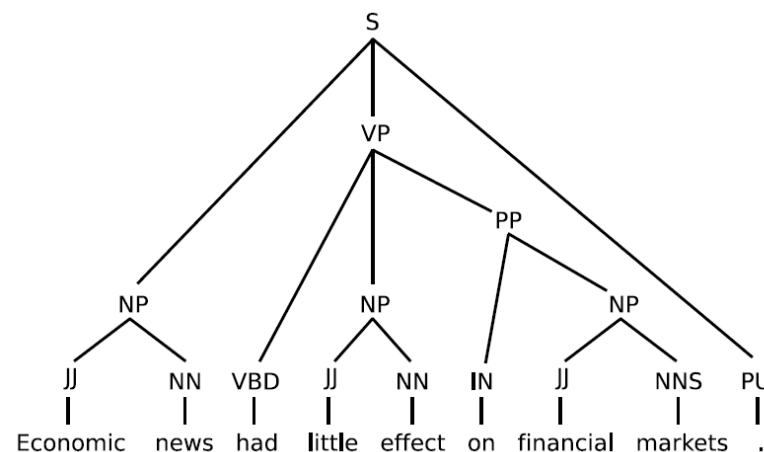
- V 代表节点的集合(词)
- E 代表箭头的集合(依存关系)
- L 代表 E 的类别集合(依存关系的类别)

□ 举例:



依存表示(Dependency Representations)

依存结构表达的信息和
传统的短语结构句法树
表达的信息是不一样，
可以表达更长距离的信
息依存关系



依存表示(Dependency Representations)

怎么确定头结点(head, H)和其依赖节点(dependent, D)

□ 一些比较通用的判断原则:

1. H 决定了依存关系的类别
2. H 决定了依存关系的语义的类别; D 只是对语义信息进行了一定的补充
3. H 在结构上是必须的; D 在结构上不是必须的

□ 依存结构的一些限制条件

□ 单头结点(single-head)

- 对于任何一个节点/词，其只能有最多一个头节点

□ 联通图(connectedness)

- 没有完全独立的节点/词

□ 无环图(acyclicity)

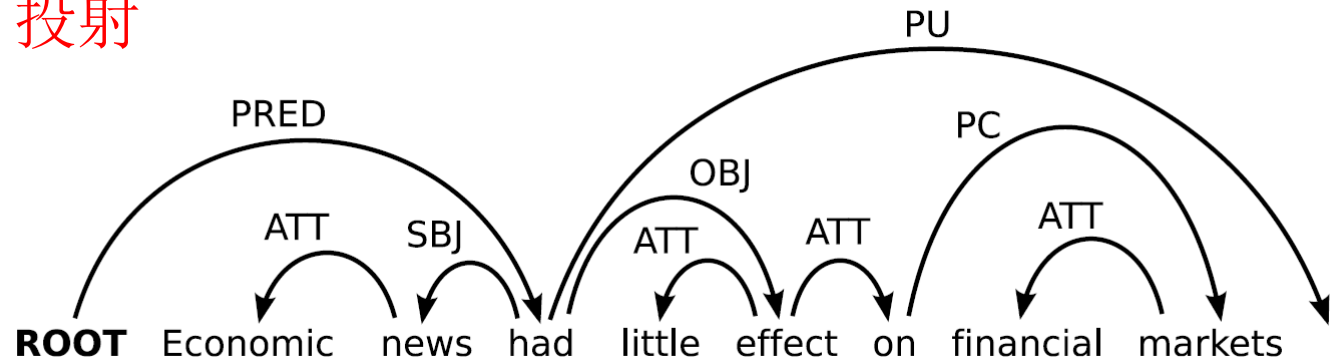
- 依存结构不会存在环形

□ 可选：投射性(projectivity)

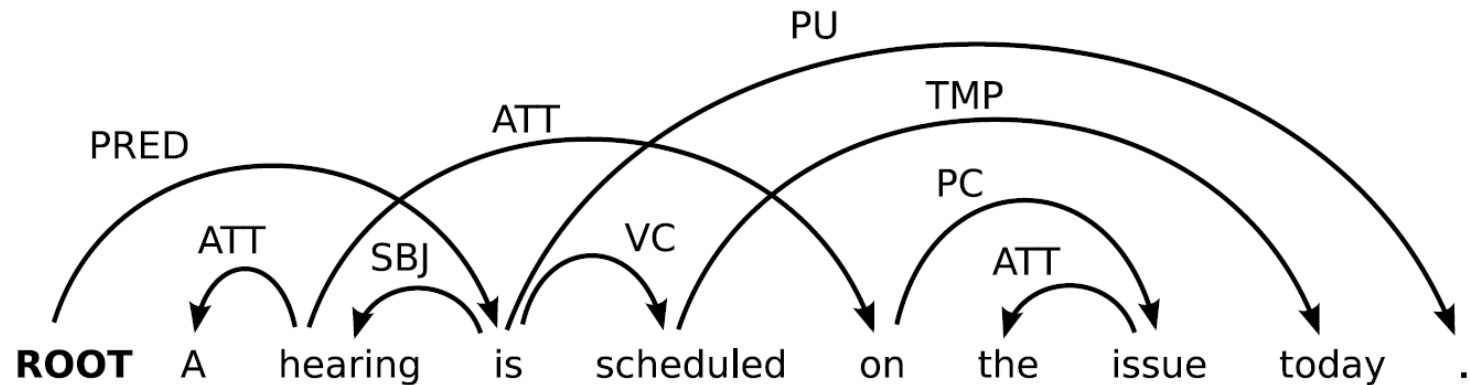
- 没有交叉的依存箭头
- 也就是依存结构是嵌套的、局部的
- 这主要是为了高效率的依存句法分析算法，为了实现动态规划
- 这是可选的，不是必须的。也有非投射性的依存语法、对应的算法

依存表示(Dependency Representations)

投射

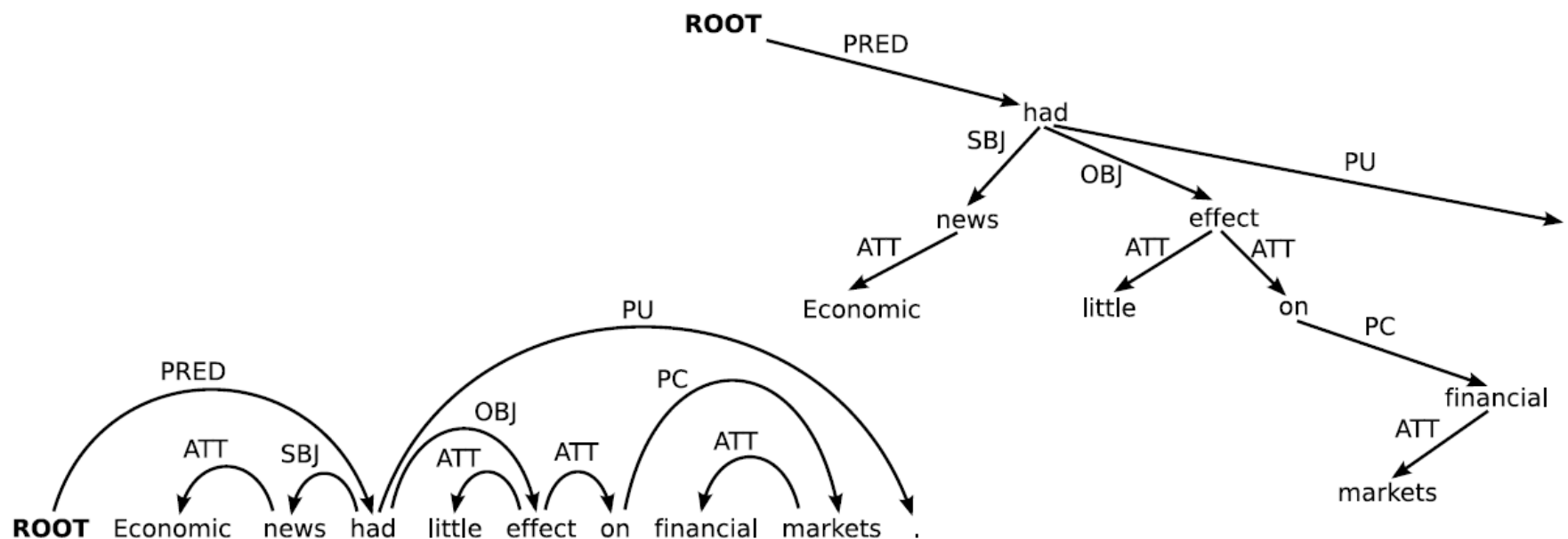


非投射



依存表示(Dependency Representations)

□ 依存句法分析其实也是树状结构!



□ 主要的依存句法分析算法

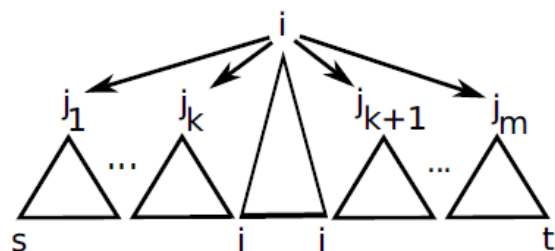
□ 图方法(Graph-Based Parsing)

- 用动态规划算法寻找全局最优的句法树
- 3次方复杂度

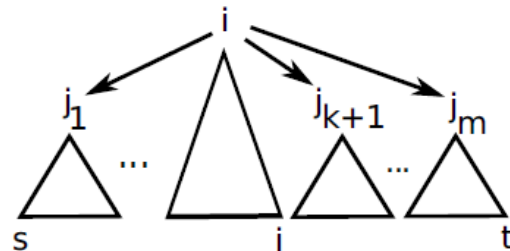
□ 局部分类方法(Deterministic parsing)

- 用局部分类方法寻找局部最优的句法树
- 线性复杂度

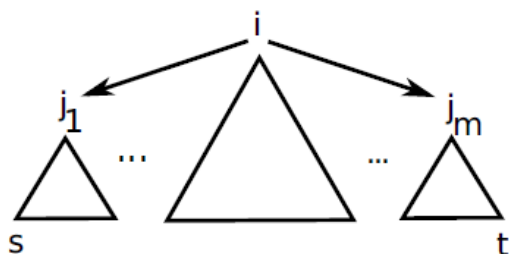
图方法1: Cocke-Kasami-Younger (CKY)算法



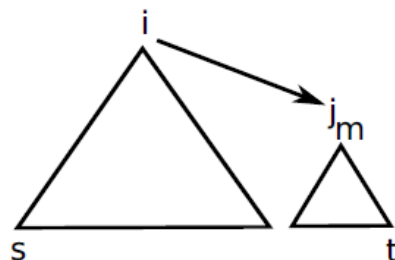
a



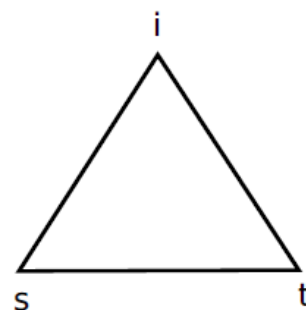
b



c



d



e

动态规划+分治法

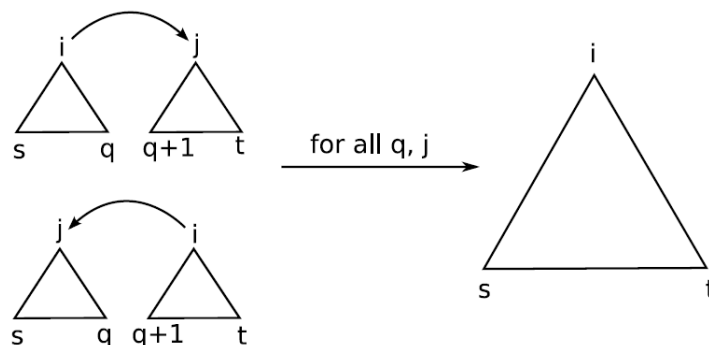
□ 图方法1: Cocke-Kasami-Younger (CKY)算法

初始化

$$C[i][i][i] = 0.0, \text{ for all } 0 \leq i \leq n$$

递归计算

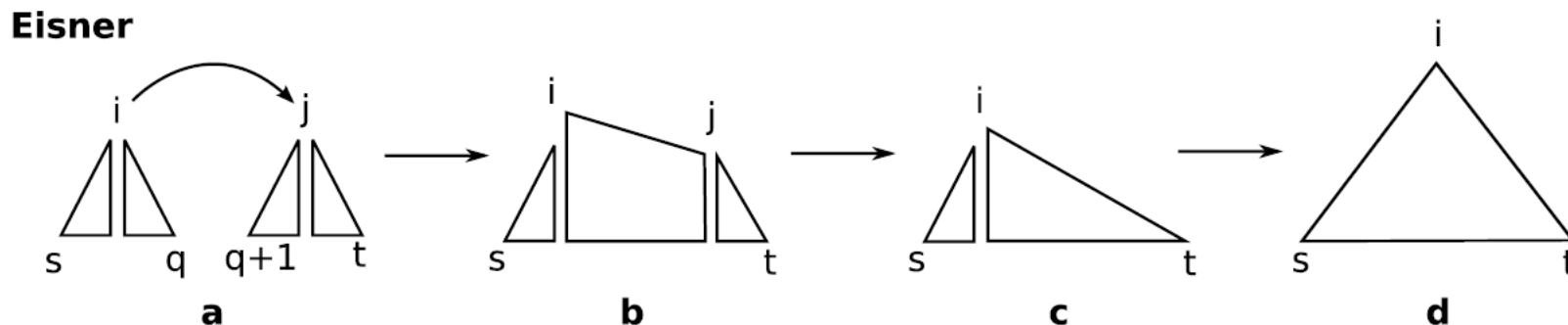
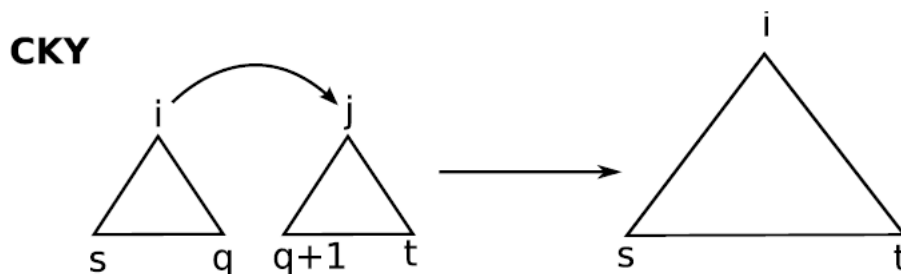
$$C[s][t][i] = \max_{s \leq q < t, s \leq j \leq t} \begin{cases} C[s][q][i] + C[q+1][t][j] + \lambda(w_i, w_j) & \text{if } j > i \\ C[s][q][j] + C[q+1][t][i] + \lambda(w_i, w_j) & \text{if } j < i \end{cases}$$



□ 图方法1: Cocke-Kasami-Younger (CKY)算法

□ 时间复杂度 $O(n^5)$

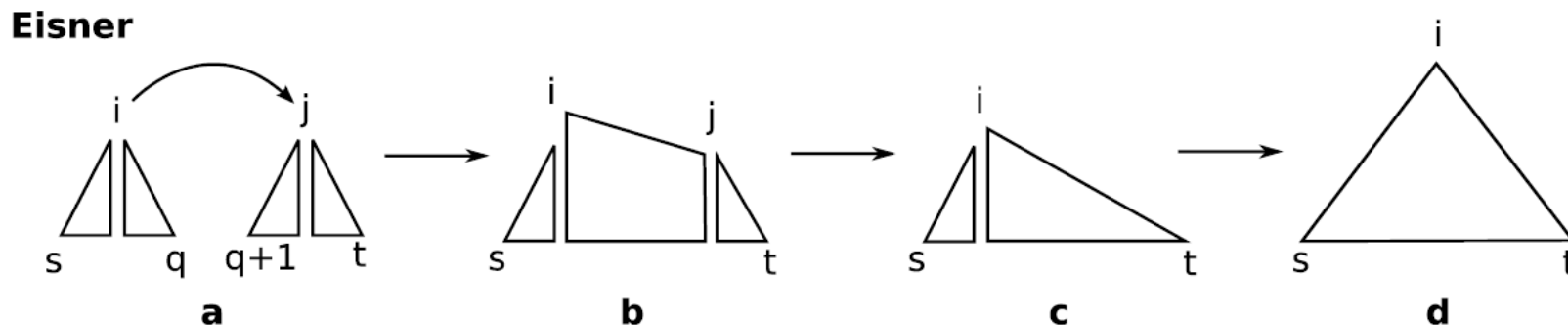
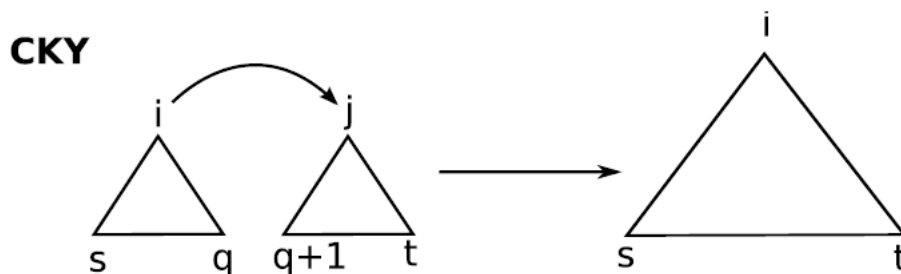
□ 改进: Eisner动态规划算法



□ 图方法1: Cocke-Kasami-Younger (CKY)算法

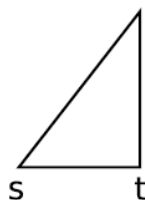
□ 时间复杂度 $O(n^5)$

□ 改进: Eisner动态规划算法

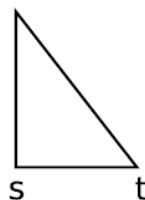


图方法(Graph-Based Parsing)

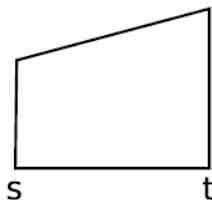
$E[s][t][0][0]$



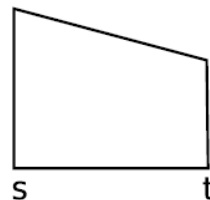
$E[s][t][1][0]$



$E[s][t][0][1]$



$E[s][t][1][1]$



Eisner(S, Γ, λ)

Sentence $S = w_0 w_1 \dots w_n$

Arc weight parameters $\lambda_{(w_i, w_j)} \in \lambda$

1 Instantiate $E[n][n][2][2] \in \mathbb{R}$

2 Initialization: $E[s][s][d][c] = 0.0$ for all s, d, c

3 for $m : 1..n$

4 for $s : 1..n$

5 $t = s + m$

6 if $t > n$ then break

 % Create subgraphs with $c = 1$ by adding arcs (step a-b in figure 4.5)

7 $E[s][t][0][1] = \max_{s \leq q < t} (E[s][q][1][0] + E[q+1][t][0][0] + \lambda_{(w_t, w_s)})$

8 $E[s][t][1][1] = \max_{s \leq q < t} (E[s][q][1][0] + E[q+1][t][0][0] + \lambda_{(w_s, w_t)})$

 % Add corresponding left/right subgraphs (step b-c in figure 4.5)

9 $E[s][t][0][0] = \max_{s \leq q < t} (E[s][q][0][0] + E[q][t][0][1])$

10 $E[s][t][1][0] = \max_{s < q \leq t} (E[s][q][1][1] + E[q][t][1][0])$

图方法2: Eisner动态规划算法

时间复杂度 $O(n^3)$

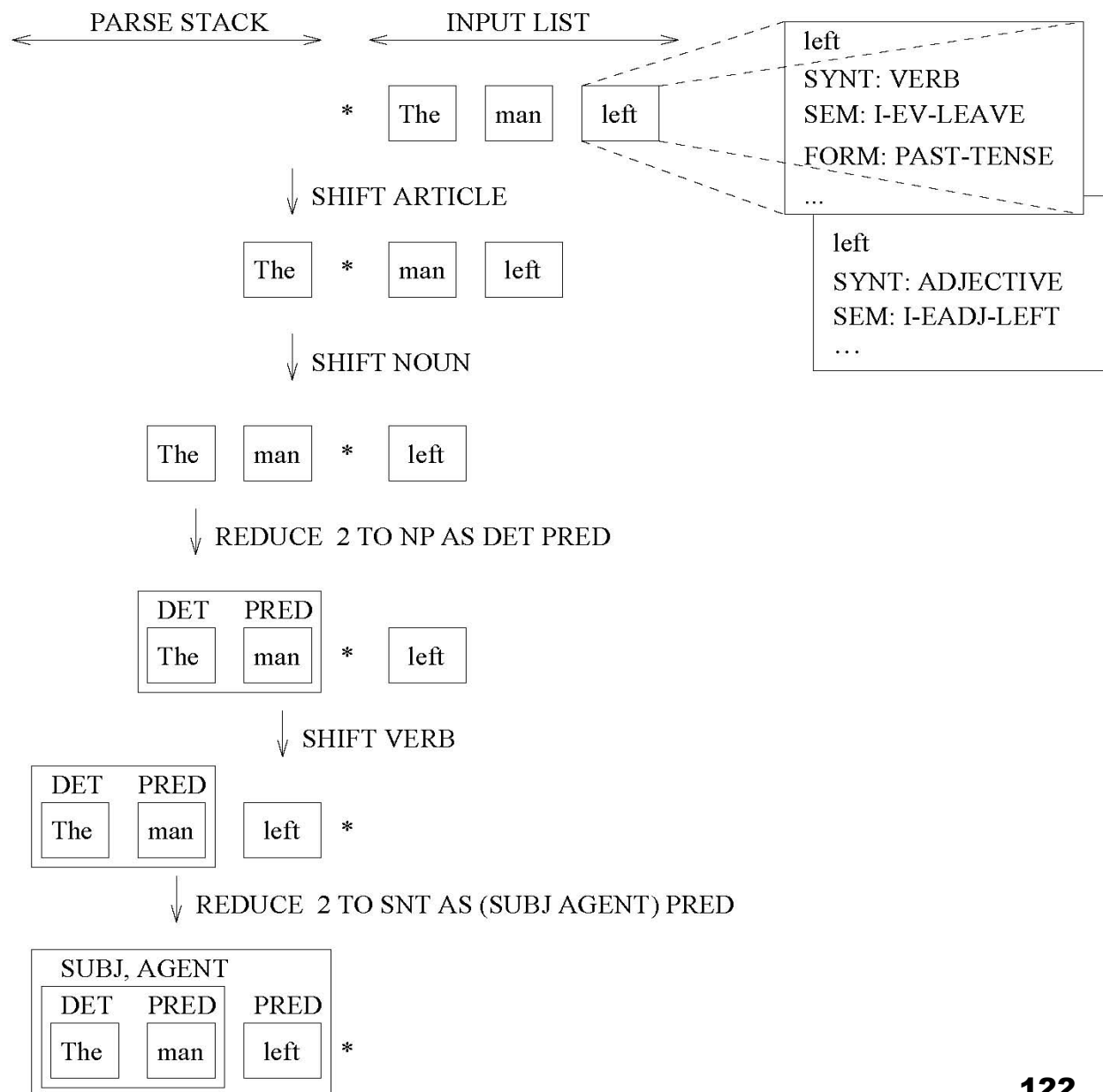
□ 主要思路

- 类似一个滑动窗口的方法，一步一步进行局部分类，最后获得句法树
- 无法获得全局最优的句法树，但是速度快，线性复杂度

□ 代表方法

- Shift-reduce句法分析算法
 - Standard (Kudo, Matsumoto, Yamada)
 - Arc-eager (Nivre)

举例

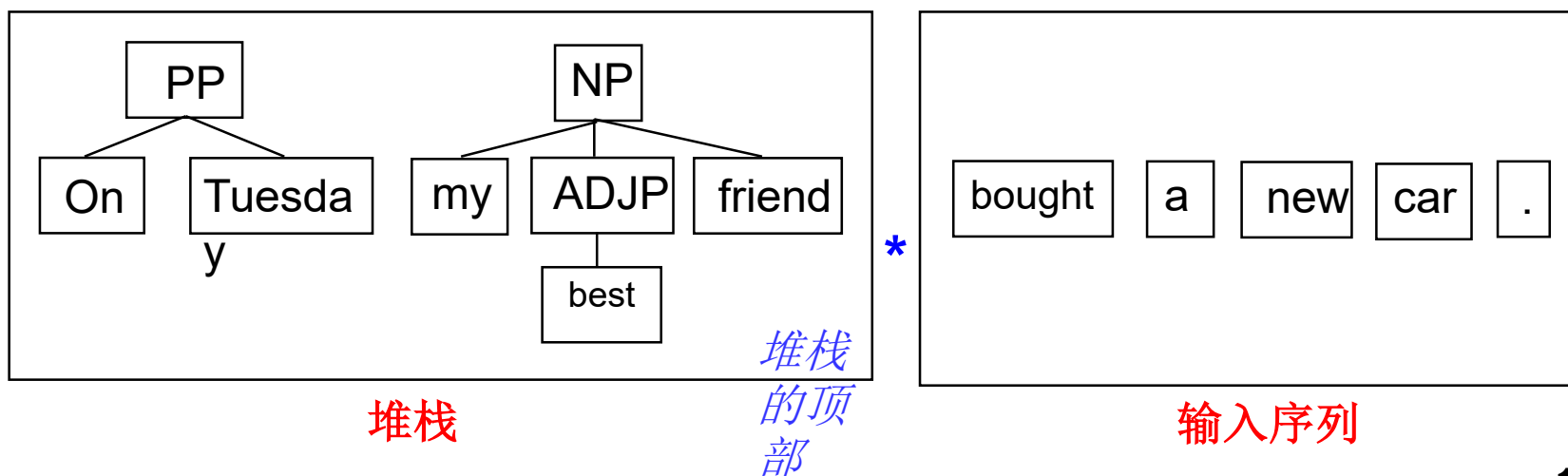


1. 输入序列

- 输入的词序列，也就是待分析的句子
- 逐步压入分析堆栈，直到词序列为空
- 输入序列为空的时候，分析完成，获得依存句法树

2. 分析堆栈

- 分析堆栈里面代表的是部分分析好的句法结构
- 堆栈的顶部是局部分类的“活跃部分”
- 输入序列为空的时候，堆栈中获得最后的完整句法分析树



Shift-Reduce算法的步骤

□ 2类主要的步骤:

□ Shift步骤

- 把元素从输入队列shift到堆栈

□ Reduce步骤

- 在堆栈内分析依存结构

怎么定义特征(feature)

■ 为了在堆栈进行准确的分类

- 句法或者语义类别
- 时态、数字、大小写信息
- 词性信息
- ...

Feature stem	Value
syntactic class of item at position 1	noun
semantic class of item at position 1	relative-temporal-interval
semantic class of object of item at position -1	monetary-quantity
tense of item at position -1	past tense
np-vp agreement of items at position -2 and -1	true
subcat affinity of 1 to -1 relative to -2	positive

□ 图方法

- 目前能达到93%以上

□ 局部分类方法

- 目前能达到91%以上

□ 优点:

- 依存关系和实际的语义关系比较接近，有助于对句子的语义方面的理解
- 依存关系的定义相对比较简单，有助于高效率的句法分析
- 因为能够有效建模长距离依赖关系，依存句法更适合词序比较自由、灵活的语言

□ 参考书

- 《统计自然语言处理》第8章：句法分析
 - 8.9 依存语法理论简介
 - 8.10 依存句法分析
 - 8.11 依存分析器性能评价
 - 8.12 短语结构与依存结构之间的关系
 - Page 220 - 240

THANKS!