



# 基于上下文无关语法的句法分析 (Parsing with CFG)

王小捷  
智能科学与技术中心  
北京邮电大学



## ■ POS tagging

## ■ 基于POS确定短语的结构

- 动词短语 V+N、ADV+N、...

- 名词短语 ADJ +N、N+N、...

- .....

## ■ 进一步→句子结构



## ■ 句子结构

■ 1)除了词之外，句子结构是影响句子理解的另一个重要因素：

■ 我打他，他打我

■ 2)复杂句子下结构信息对于句子理解作用更为关键：

■ 我们在所列举的大部分战役里都很容易地找出了不少于三个和陈起威事件中类似的问题。

■ The most commonly used mathematical system for modeling constituent structure in CFG English and other natural languages is the Context-Free Grammar.

# 句子结构信息的应用

- 语法检查
- 机器翻译
- 问答系统
- 信息检索
- .....





## ■ 句子结构：句法

- 句法(syntax)

- 源于希腊语 syntaxis (安排在一起)

- 语言的组合性 (结构)

- vs 语言的聚合性 (词)

- 索绪尔普通语言学

# 句法(Syntax)



## ■两种不同:

### ■规定语法(Prescriptive grammar)

■ how people **ought** to talk

### ■描述语法(Descriptive grammar)

■ how they **do** talk

## ■当前取描述语法的态度:

■句法的目标是建模人们无意识中就具有的关于他们母语的结构知识

■不过: 关于语言净化的争论



# 句法中的几个关键概念

- 单元成分(Constituent)
- 语法关系(Grammatical relations)
- 次范畴(Subcategorization)
- 词汇依赖(Lexical dependencies)
- 其他
- 移动(Movement)
- 长距离依赖(long-distance dependency)
- ...

# 单元(Constituent)



## ■定义

- 一个或一组词，作用如同一个单元。

## ■Noun phrase(NP): 作用如同名词

- They; 他们
- Three persons from Brooklyn; 从布鲁克林来的三个人
- The reason he comes into the house; 他进屋的理由

## ■Verb phrase(VP)

- go; 走
- go to school; 上课
- want to have a look; 想去看一看

## ■PP

## ■ADJP

## ■.....





- 一个或多个词组成了一个单元的判据：
- 1) 共同作为一个整体具有语法性质，而其中的单元单独没有。
  - 以NP为例，整体作为主语，而部分不是
  - Three parties from Brooklyn arrive...
  - 他进屋的理由 是...
  - ...



- 一个或多个词组成了一个单元的判据：
- 2) 出现相似的句法环境中，例如，NP出现在在动词前
  - 以NP为例，他们都出现在动词前面：
  - Three parties from Brooklyn arrive...
  - 他进屋的理由 是...
  - 他们 坐...



- 一个或多个词组成了一个单元的判据:
- 3) 整体移动, 单元的各部分不单独移动
  - 无论出现在哪里, 都是整体移动:
    - On September 17th, I'd like to fly from Atlanta to Denver
    - I'd like to fly on September 17th from Atlanta to Denver
    - I'd like to fly from Atlanta to Denver on September 17th.
  - 一般不会分拆
    - \*On September, I'd like to fly 17th from Atlanta to Denver
    - \*On I'd like to fly September 17th from Atlanta to Denver



# 其他重要句法概念

## ■ 语法关系(Grammatical Relations)

### ■ 主语(Subject)

■ I booked a flight to New York. 我订了一张机票。

■ The flight was booked by my agent. 飞机晚点了

### ■ 宾语(Object)

■ I booked a flight to New York. 我订了一张机票。

### ■ 补语(Complement)

■ I said that I wanted to leave. 我说不用了



## ■次范畴(Subcategorization)

- 后文会提到：及物动词、不及物动词是动词次范畴

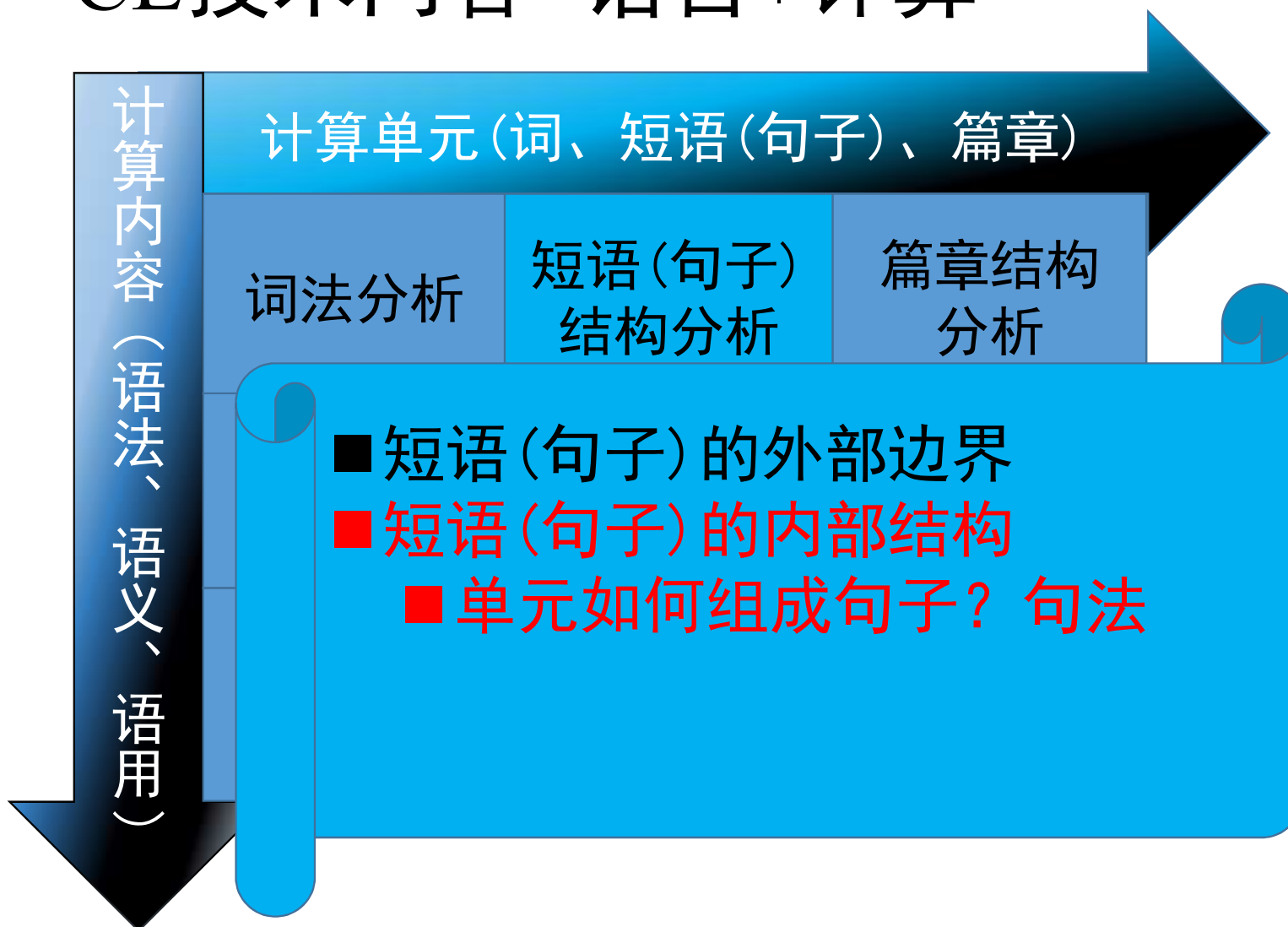
## ■词汇依赖(Lexical dependencies)

- 喝水, 啃玉米...

## ■长距离依赖(long-distance dependency)

- 他正在喂那只最近托朋友从德国符腾堡地区重金购买来的良种犬。

# CL技术内容=语言+计算



# 大纲



- 上下文无关语法
- 基于搜索的句法分析方法
- CKY句法分析算法
- Earley句法分析算法
- 句法剖析算法的评估
- 部分句法分析(Partial parsing)



## ■语言(心理)学家提出了很多语法理论:

### ■(转换)生成语法

- 1950s提出以来, 统治地位

### ■依存语法

- 目前得到较为广泛的应用

### ■范畴语法

- 有些人用

- .....





# ■ Chomsky的生成语法体系

## ■ 3-正则语法RG

■ 自然语言 (过弱)

## ■ 2-上下文无关语法CFG

■ 自然语言 (弱)

## ■ 1-上下文相关语法CSG

■ 自然语言 ( ? )

## ■ 0-短语结构语法PSG



# 上下文无关语法 (Context-Free Grammars: CFG)

- 一个CFG是一个 4元组:  $N, \Sigma, P, S$ 
  - 非终止符集合  $N$
  - 终止符集合  $\Sigma$  (和  $N$  的交为空) [词, POS]
  - 产生式集合  $P$ , 集合中每个元素具有:
    - $A \rightarrow \alpha$
    - 其中  $A$  是非终止符,  $\alpha$  属于  $(\Sigma \cup N)^*$
  - 开始符  $S$



# CFG 例

■  $N = \{NP, VP, NOMINAL, Det, Noun, Verb, \dots\}$

■  $\Sigma = \{a, flight, left, \dots\}$

■  $S$

■  $R$ :

■  $S \rightarrow NP VP$

■  $NP \rightarrow Det NOMINAL$

■  $NOMINAL \rightarrow Noun$

■  $VP \rightarrow Verb$

■  $Det \rightarrow a$

■  $Noun \rightarrow flight$

■  $Verb \rightarrow left$

# CFG例

■  $N = \{NP, VP, NOMINAL, \dots\}$

■  $\Sigma = \{Det, Noun, Verb, \dots\}$

■  $S$

■  $R:$

■  $S \rightarrow NP VP$

■  $NP \rightarrow Det NOMINAL$

■  $NOMINAL \rightarrow Noun$

■  $VP \rightarrow Verb$





# 对CFG的说明

## ■ $S \rightarrow NP VP$

- S由一个NP紧接着一个VP组成
- 但并不是说这是组成S的唯一方式
- 也不是说NP和VP只能出现在这里

## ■ 上下文无关：规则的使用不受上下文影响

- $A \rightarrow B C$
- 只要看到A，就可以重写为B 和C



# 乔姆斯基范式

## ■ Chomsky Normal Formulas

- $A \rightarrow a$

- $A \rightarrow B C$

## ■ From CFG to CNF: **Equivalence**

- $A \rightarrow a B \rightarrow ?$

- $A \rightarrow A B C \rightarrow ?$

## ■ Why CNF: CKY

# 派生(Derivations)



## ■连续使用规则而产生的句子,例如

■  $S \rightarrow NP VP$

■  $NP \rightarrow Pro$

■  $Pro \rightarrow I$

■  $VP \rightarrow Verb NP$

■  $Verb \rightarrow prefer$

■  $NP \rightarrow Det NOMINAL$

■  $Det \rightarrow a$

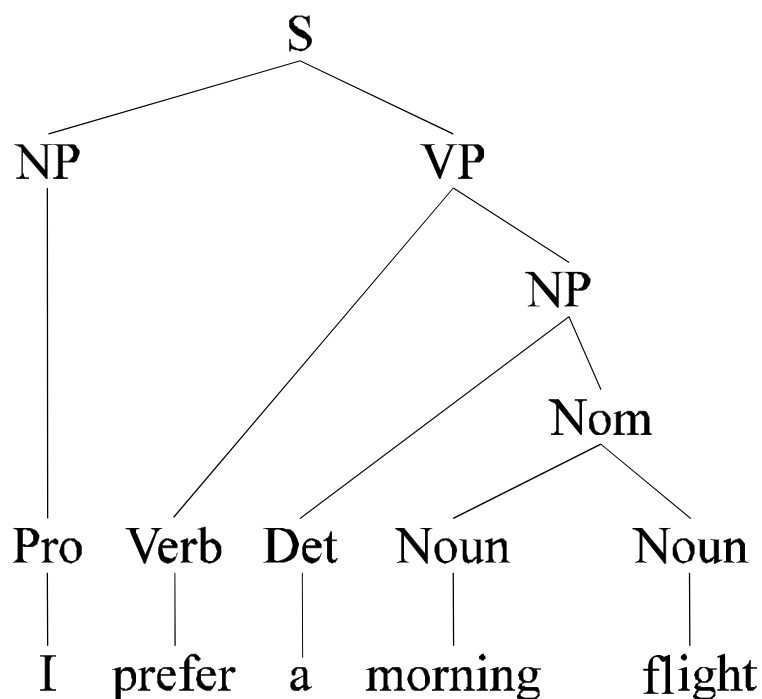
■  $NOMINAL \rightarrow Noun Noun$

■  $Noun \rightarrow morning$

■  $Noun \rightarrow flight$

■  $\rightarrow I prefer a morning flight.$

# 派生的树表示：句法树



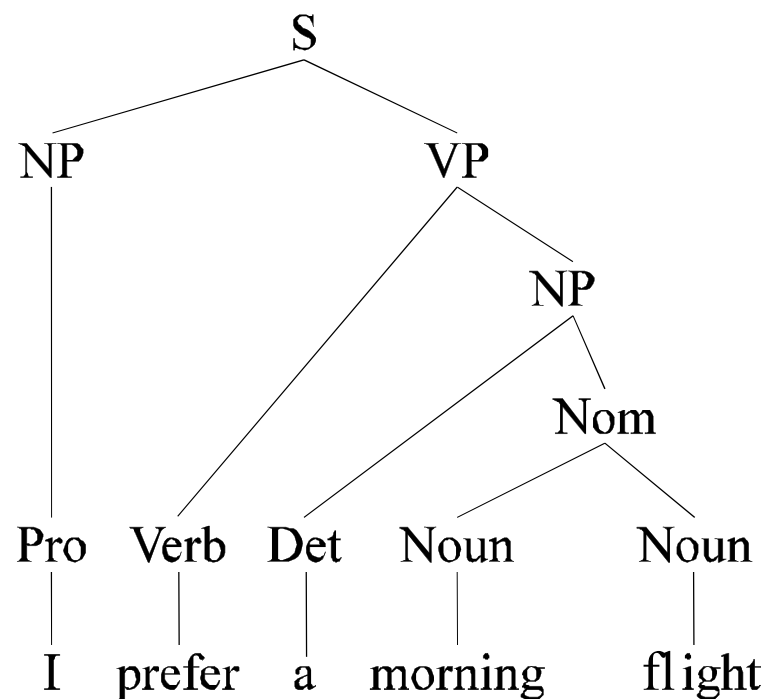
- 所用规则：
- $S \rightarrow NP VP$
  - $NP \rightarrow Pro$
  - $Pro \rightarrow I$
  - $VP \rightarrow Verb NP$
  - $Verb \rightarrow prefer$
  - $NP \rightarrow Det NOMINAL$
  - $Det \rightarrow a$
  - $NOMINAL \rightarrow Noun$
  - $Noun \rightarrow morning$
  - $Noun \rightarrow flight$





# 括号表示(Bracketed Notation)

■[<sub>S</sub> [<sub>NP</sub> [<sub>PRO</sub> I] [<sub>VP</sub> [<sub>V</sub> prefer [<sub>NP</sub> [<sub>NP</sub> [<sub>Det</sub> a] [<sub>Nom</sub> [<sub>N</sub> morning] [<sub>N</sub> flight]]]]]]]





- 按不同的次序使用不同的规则可以派生出各种终止符序列
- 由一个CFG按各种可能的方式派生的所有非终止符的集合是一个CF语言。
- CF语言有限？无限？
- $N, \Sigma, R$ 均有限：非常巨大，可认为是无限
- 而一些特殊的 $R \rightarrow$



# 递归结构使得句子更复杂和丰富

■ **Flights** from Denver:

NP → NP PP

■ [[flights] [from Denver]]

■ **Flights from Denver** to Miami:

■ [[[Flights] [from Denver]] [to Miami]]

■ **Flights from Denver to Miami** in February:

■ [[[[Flights] [from Denver]] [to Miami]] [in February]]

■ **Flights from Denver to Miami in February** on a Friday:

■ [[[[[Flights] [from Denver]] [to Miami]] [in February]] [on a Friday]] ...

■ **Flights from Denver to Miami in February on a Friday** under \$300

■ ...理论上的无限可能



# 并列结构

■  $S \rightarrow S \text{ and } S$

■ John went to NY and Mary followed him and...

■  $NP \rightarrow NP \text{ and } NP$

■ 左边的椅子和右边的凳子...

■  $VP \rightarrow VP \text{ and } VP$

■ 他经常看书与写字...

■ ...理论上的无限可能



## CFG的问题(局限性)

- 一致性(Agreement)
- 次范畴(Subcategorization)
- 移动(Movement)



# 一致性(Agreement)

■ This dog

■ Those dogs

■ This dog eats

■ Those dogs eat

■ \*This dogs

■ \*Those dog

■ \*This dog eat

■ \*Those dogs eats

■ NP → Det N

■ S → NP VP

■ ...



# 可能的解决方案

NP → Det Nominal	➔	SgNP → SgDet SgNom
		PlNP → PlDet PlNom
S → NP VP	➔	SgS → SgNP SgVP
		PlS → PlNp PlVP
...		

似乎可以接受：代价是增加了几个规则  
继续...



# 次范畴(Subcategorization)

## ■次范畴表达了谓词对其论元的进一步约束

■Sneeze: John sneezed

■Find: Please find [a flight to NY]<sub>NP</sub>

■Give: Give [me]<sub>NP</sub> [a cheaper fare]<sub>NP</sub>

■Help: Can you help [me]<sub>NP</sub> [with a flight]<sub>PP</sub>

■Prefer: I prefer [to leave earlier]<sub>TO-VP</sub>

■Said: You said [United has a flight]<sub>S</sub>

■...

■\*John sneezed the book

■\*I prefer United Air has a flight

■\*Give with a flight

■  $VP \rightarrow V$

■  $VP \rightarrow V NP$

■  $VP \rightarrow V NP PP$

■ ...





## 可能的解决方案

$VP \rightarrow V$	$\rightarrow$	$VP \rightarrow \text{Intrans}V$
$VP \rightarrow V \text{ NP}$	$\rightarrow$	$VP \rightarrow \text{Trans}V \text{ NP}$
$VP \rightarrow V \text{ NP PP}$	$\rightarrow$	$VP \rightarrow \text{Trans}V \text{ NP PP}$

似乎可以接受：代价是增加了几个规则  
继续...

# ■现在，同时处理一致性和次范畴问题时？

NP → Det Nominal	➔	SgNP → SgDet SgNom
		PlNP → PlDet PlNom
S → NP VP	➔	SgS → SgNP <b>SgVP</b>
		PlS → PlNp <b>PlVP</b>
VP → V	➔	VP → IntransV
VP → V NP	➔	VP → TransV NP
VP → V NP PP	➔	VP → TransV NP PP

SgVP、PlVP每一个都要与不同的次范畴组合：2\*3

代价：规则组合增长的速度难以接受了！



# 移动(Movement): wh-movement

## ■例子:

■He **booked** the **flight**.

## ■正常

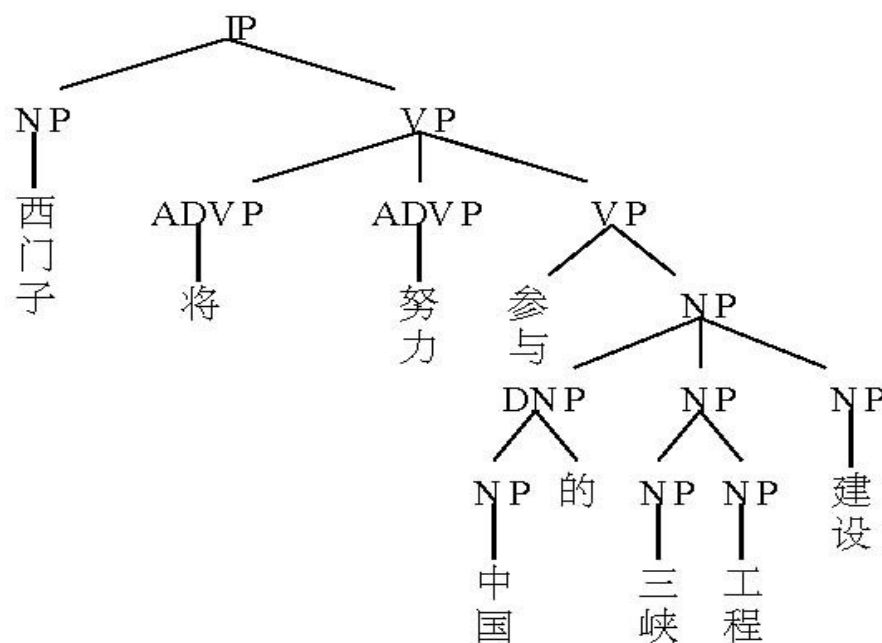
## ■移动后:

■Which **flight** **did** he **book** ?

# 树库 Treebanks

## ■ 句法树的集合

### ■ 西门子将努力参与中国的三峡工程建设

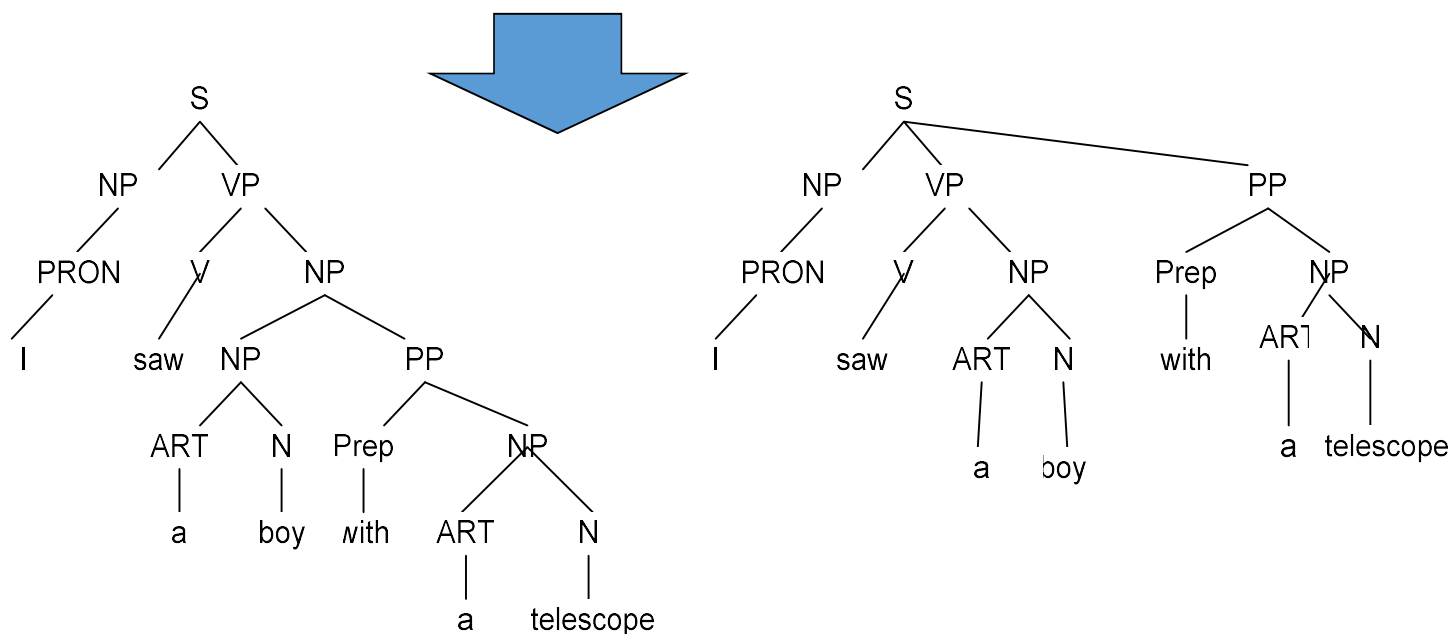


- 基于此可以提取CFG规则
  - $IP \rightarrow NP \quad VP$
  - ...

■基于CFG获得句子句法树可得到句子构成方式

■进一步的价值：结构消歧

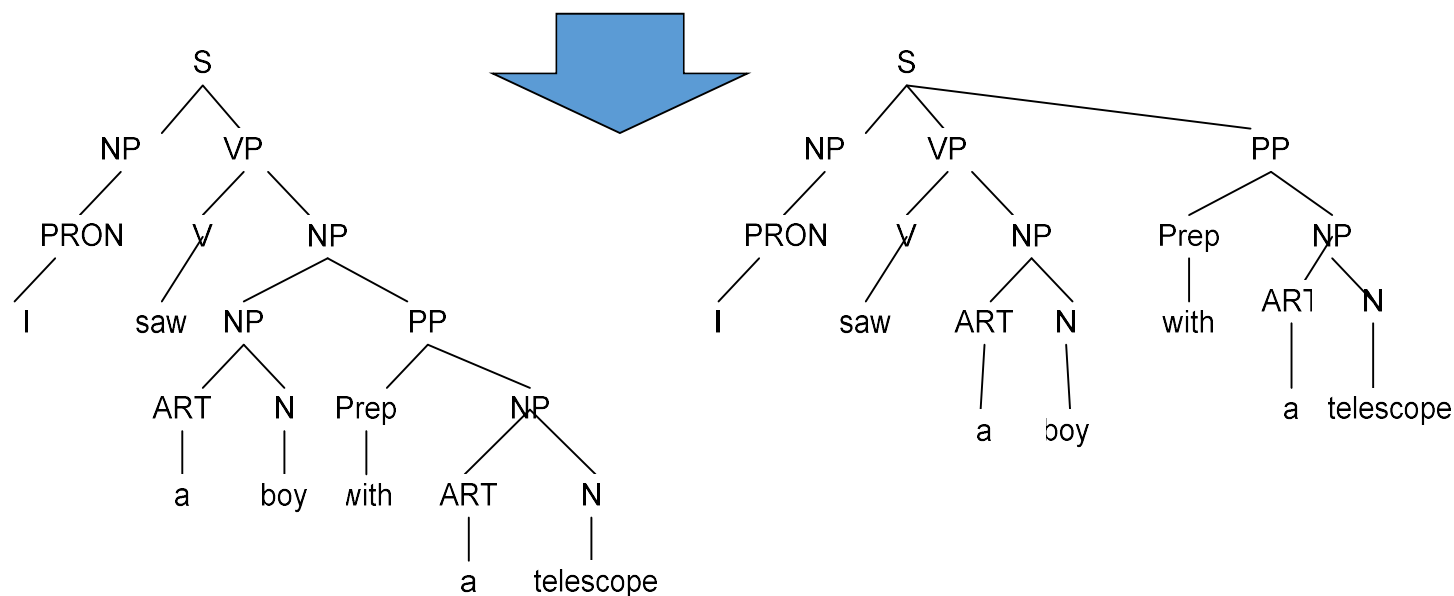
■例：I saw a boy with a telescope. (歧义)



■线性串存在的结构歧义可在句法树中得到消解

## ■线性词串 → 句法树

■例：I saw a boy with a telescope.



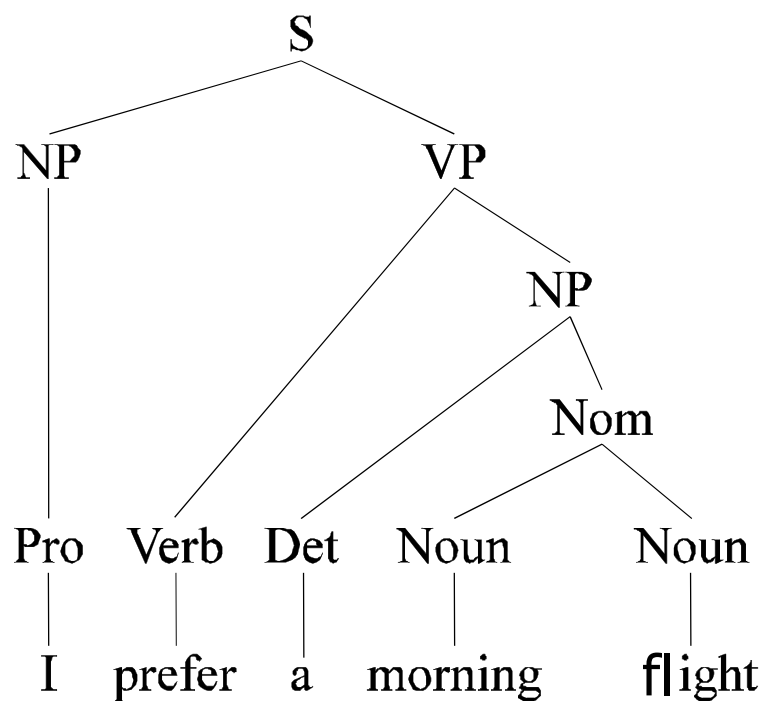
## ■如何自动进行？ 剖析parsing

# 大纲

- 上下文无关语法
- 基于搜索的句法分析方法
- CKY句法分析算法
- Earley句法分析算法
- 句法剖析算法的评估
- 部分句法分析(Partial parsing)



Remind: 句法树是由用来组成该句子所使用的CFG规则组合而成的



■ 所用规则:

$S \rightarrow NP VP$

$NP \rightarrow Pro$

$Pro \rightarrow I$

$VP \rightarrow Verb NP$

$Verb \rightarrow prefer$

$NP \rightarrow Det NOMINAL$

$Det \rightarrow a$

$NOMINAL \rightarrow Noun Noun$

$Noun \rightarrow morning$

$Noun \rightarrow flight$





■基于此，CFG的句法分析的目标是：

■→找出分析该句子所需使用的全部CFG规则及其使用次序



## ■基于蛮力搜索的Parsing

■I saw a boy

■找出分析该句子所需使用的全部  
CFG规则及其使用次序

$S \rightarrow NP VP$

$S \rightarrow Aux NP VP$

$S \rightarrow VP$

$NP \rightarrow Pronoun$

$NP \rightarrow Proper-Noun$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

$Nominal \rightarrow Nominal Noun$

$Nominal \rightarrow Nominal PP$

$VP \rightarrow Verb$

$VP \rightarrow Verb NP$

$VP \rightarrow Verb NP PP$

$VP \rightarrow Verb PP$

$VP \rightarrow VP PP$

$PP \rightarrow Preposition NP$

$Pron \rightarrow I$

$Verb \rightarrow saw$

$Det \rightarrow a$

$Noun \rightarrow boy$



- 自底向上的搜索
  - 从词汇开始
  - 匹配CFG规则的右边
  - 组成更大单元后
  - 继续以得到的大单元匹配CFG规则的右边
  - 直到组成S
- 以I saw a cat为例

## ■ 自底向上的搜索



I

saw

a

cat



## ■ 自底向上的搜索

Pron → I  
Verb → saw  
Det → a  
Noun → cat

Pron  
|  
I

Verb  
|  
saw

Det  
|  
a

Noun  
|  
cat



## ■ 自底向上的搜索

Pron → I

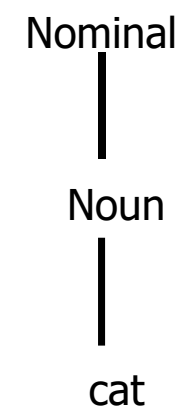
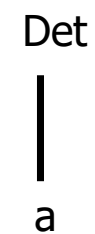
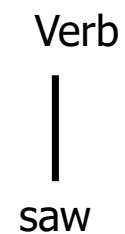
Verb → saw

Det → a

Noun → cat

Nominal → Noun

NP → Pron





## ■ 自底向上的搜索

Pron → I

Verb → saw

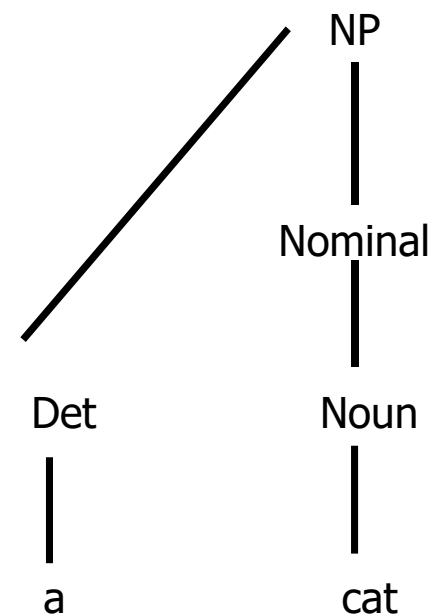
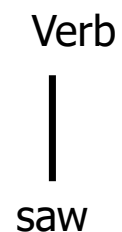
Det → a

Noun → cat

Nominal → Noun

NP → Pron

NP → Det Nominal



## ■ 自底向上的搜索

Pron → I

Verb → saw

Det → a

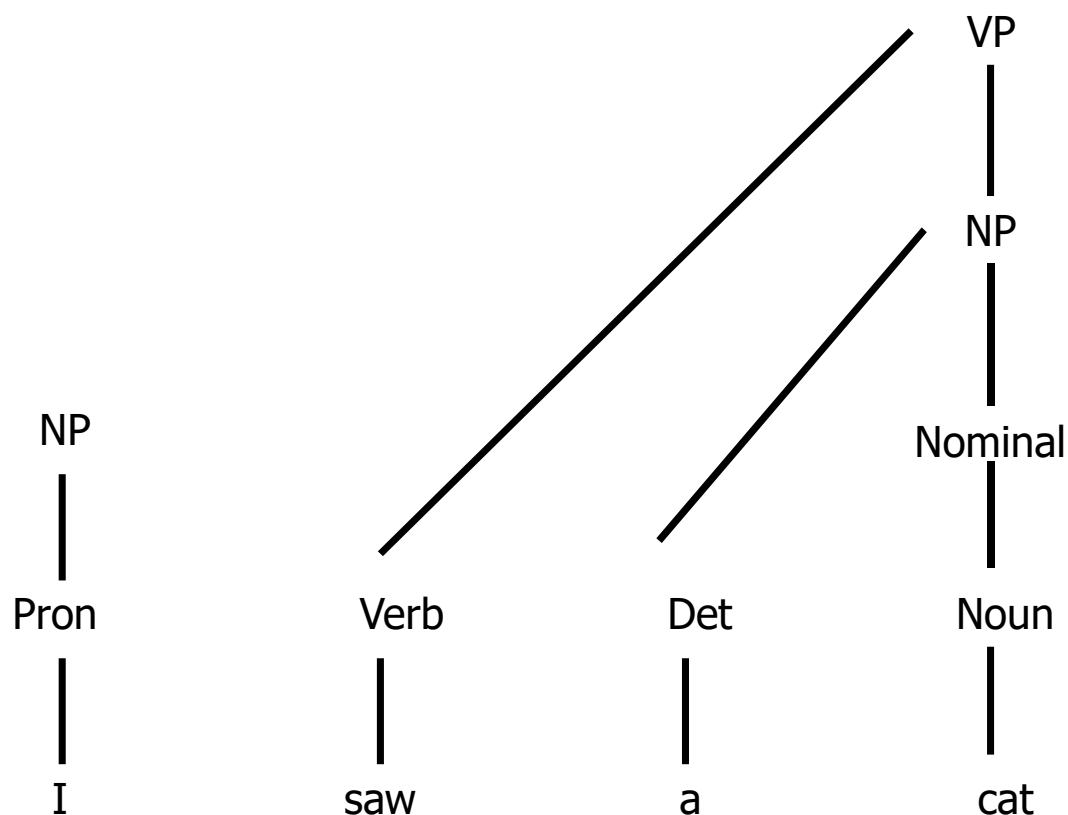
Noun → cat

Nominal → Noun

NP → Pron

NP → Det Nominal

VP → Verb NP





## ■ 自底向上的搜索

Pron  $\rightarrow$  I

Verb  $\rightarrow$  saw

Det  $\rightarrow$  a

Noun  $\rightarrow$  cat

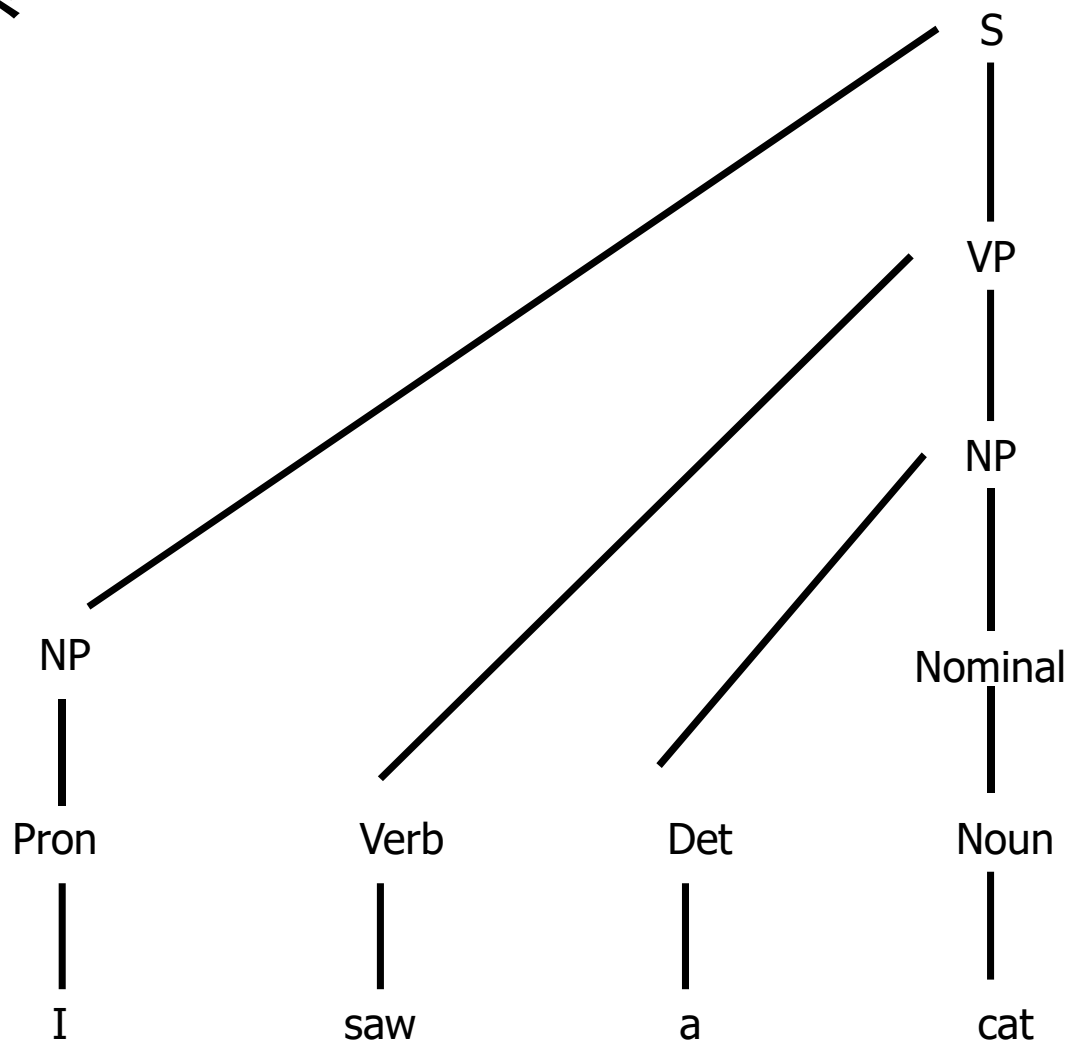
Nominal  $\rightarrow$  Noun

NP  $\rightarrow$  Pron

NP  $\rightarrow$  Det Nominal

VP  $\rightarrow$  Verb NP

**S  $\rightarrow$  NP VP**

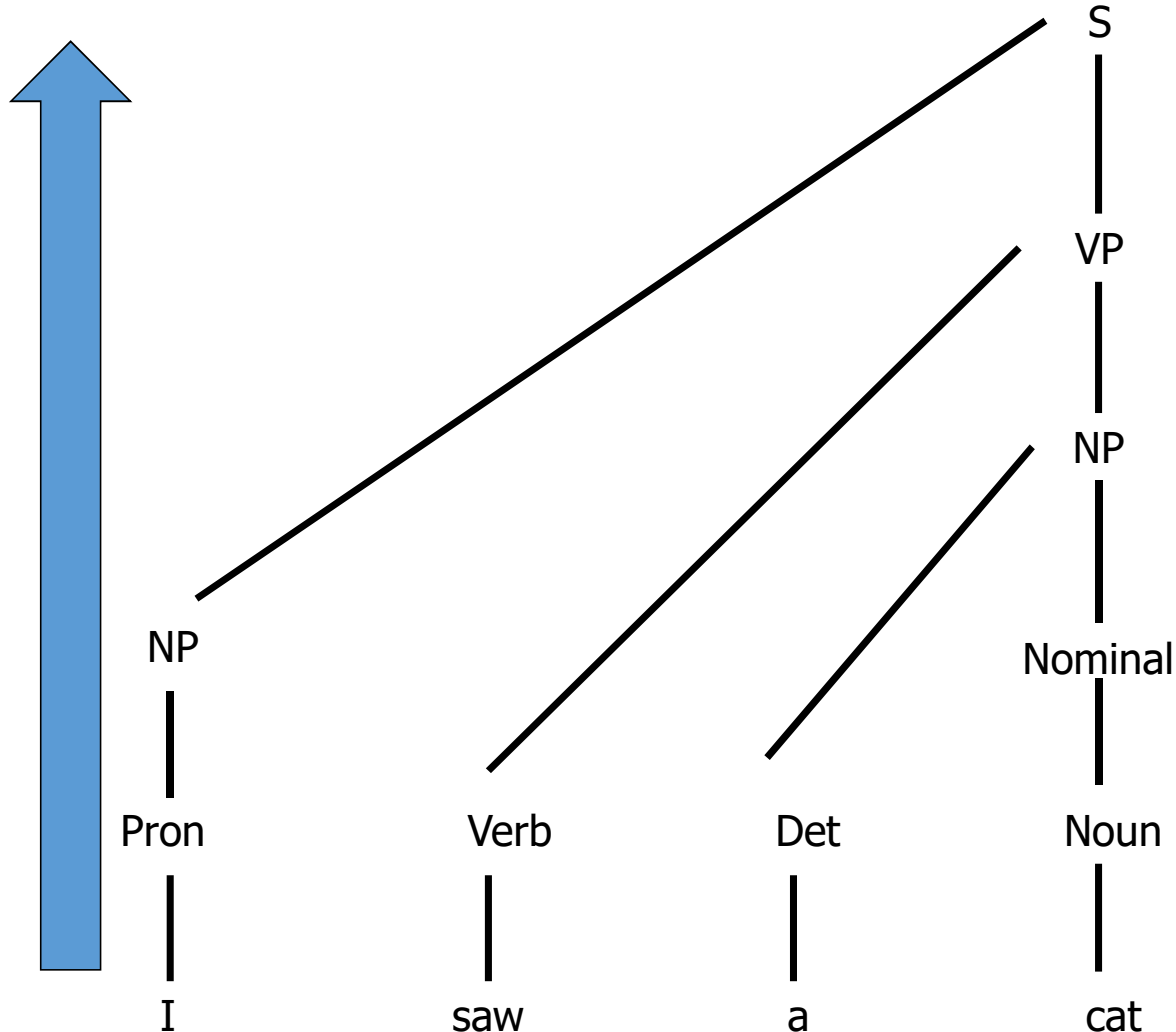


# ■从小单元逐层通过寻找CFG规则组成大单元



$S \rightarrow NP VP$   
 $S \rightarrow Aux NP VP$   
 $S \rightarrow VP$   
 $NP \rightarrow Pronoun$   
 $NP \rightarrow Proper-Noun$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow Noun$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow Verb$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow Verb NP PP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$

$Pron \rightarrow I$   
 $Verb \rightarrow saw$   
 $Det \rightarrow a$   
 $Noun \rightarrow boy$

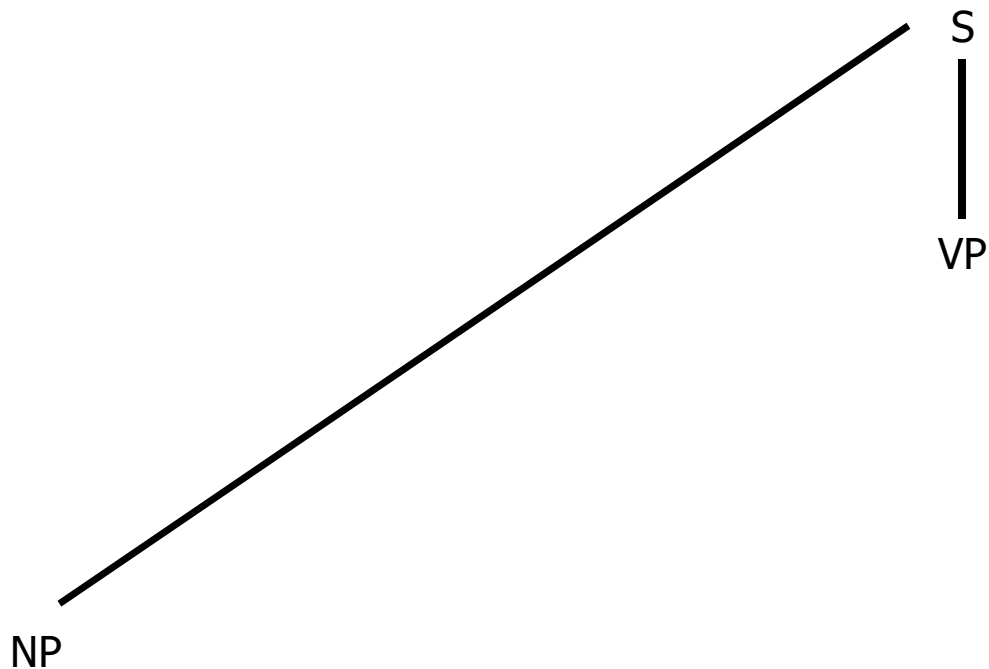




- 自顶向下的搜索
  - 从S开始
  - 匹配CFG规则的左边
  - 获得组成S的更小单元
  - 继续以获得的更小单元去匹配CFG规则的左边
  - 直到获得词汇序列
- 以I saw a cat为例

## ■ 自顶向下的搜索

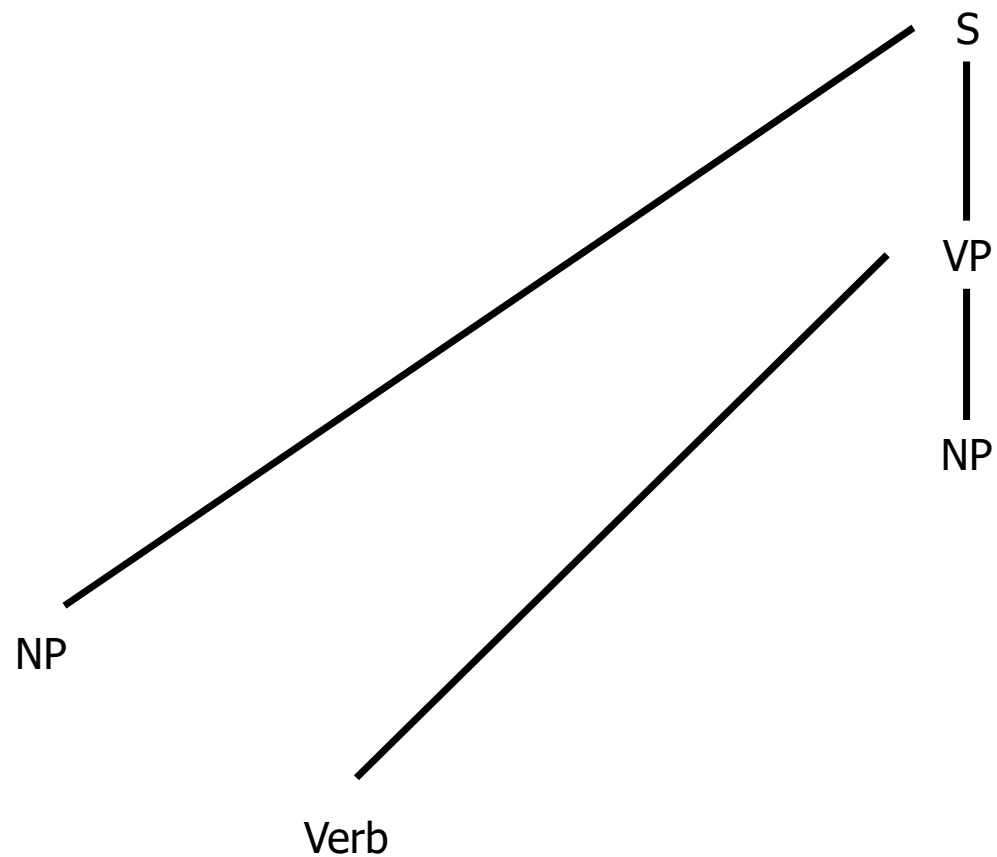
$S \rightarrow NP VP$



## ■ 自顶向下的搜索

$S \rightarrow NP VP$

$VP \rightarrow Verb NP$

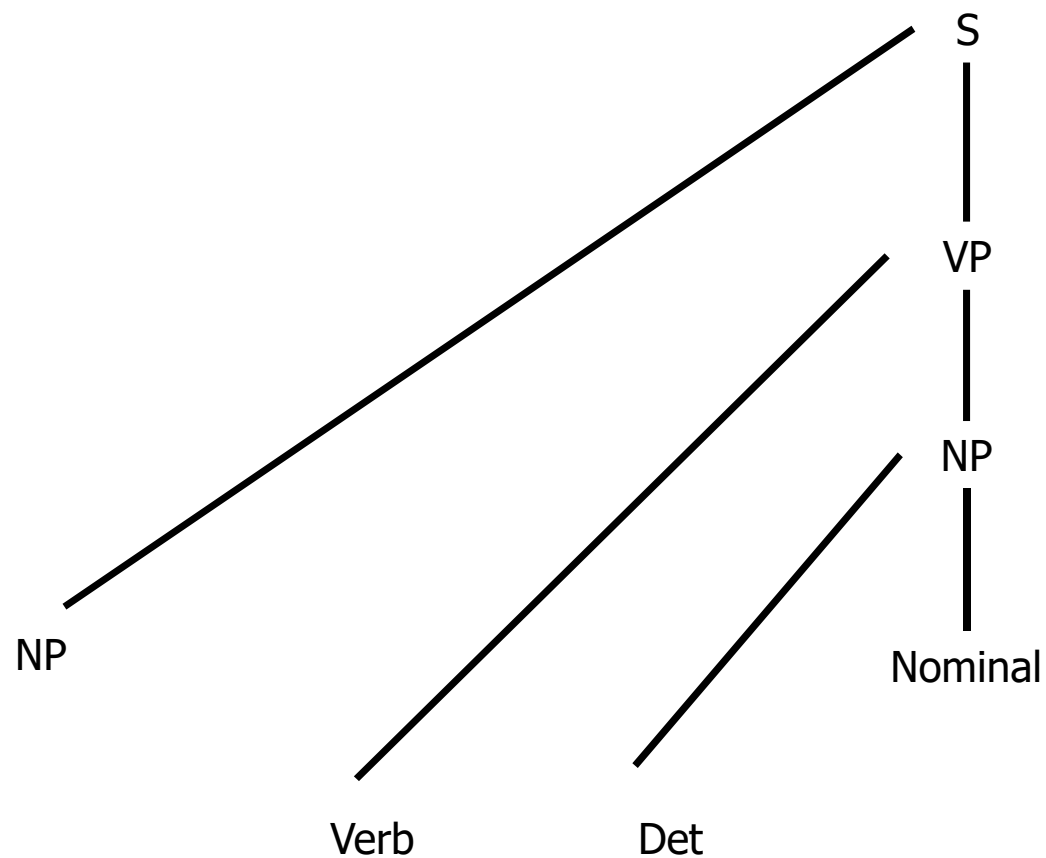


# ■ 自顶向下的搜索

$S \rightarrow NP VP$

$VP \rightarrow Verb NP$

$NP \rightarrow Det Nominal$



# ■ 自顶向下的搜索

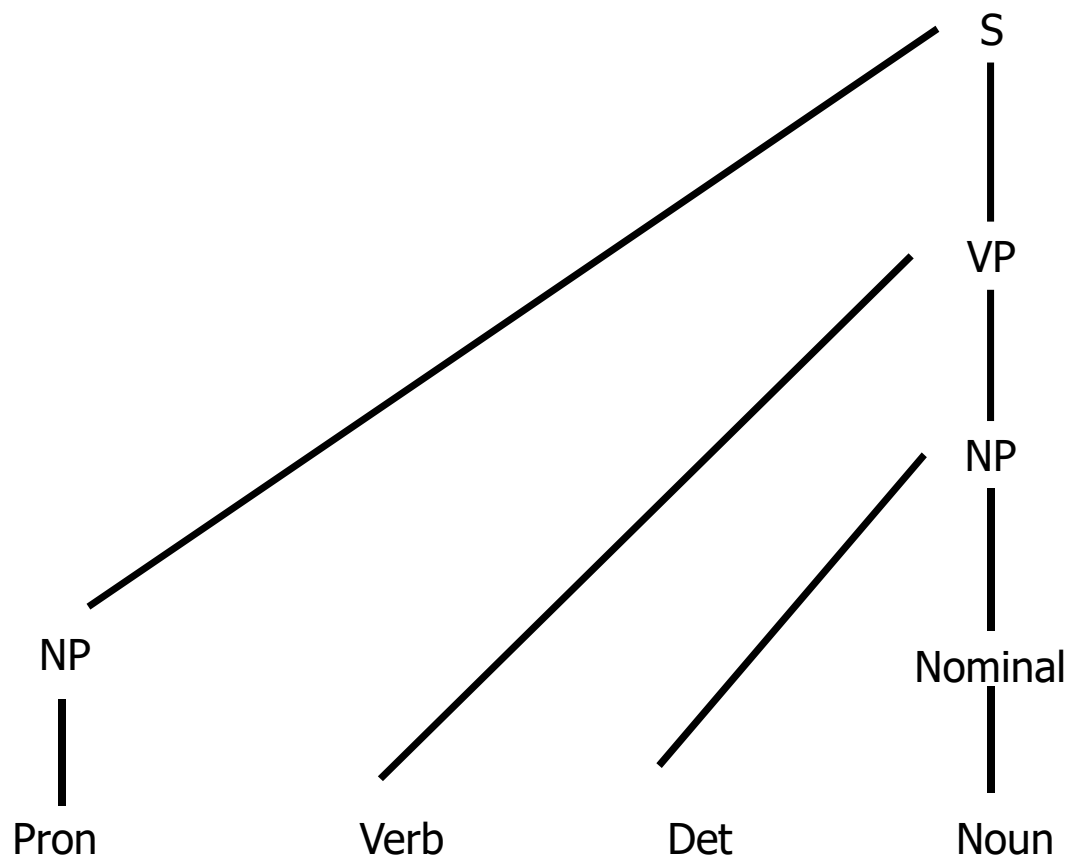
$S \rightarrow NP VP$

$VP \rightarrow Verb NP$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

$NP \rightarrow Pron$



# ■ 自顶向下的搜索



$S \rightarrow NP VP$

$VP \rightarrow Verb NP$

$NP \rightarrow Det Nominal$

$Nominal \rightarrow Noun$

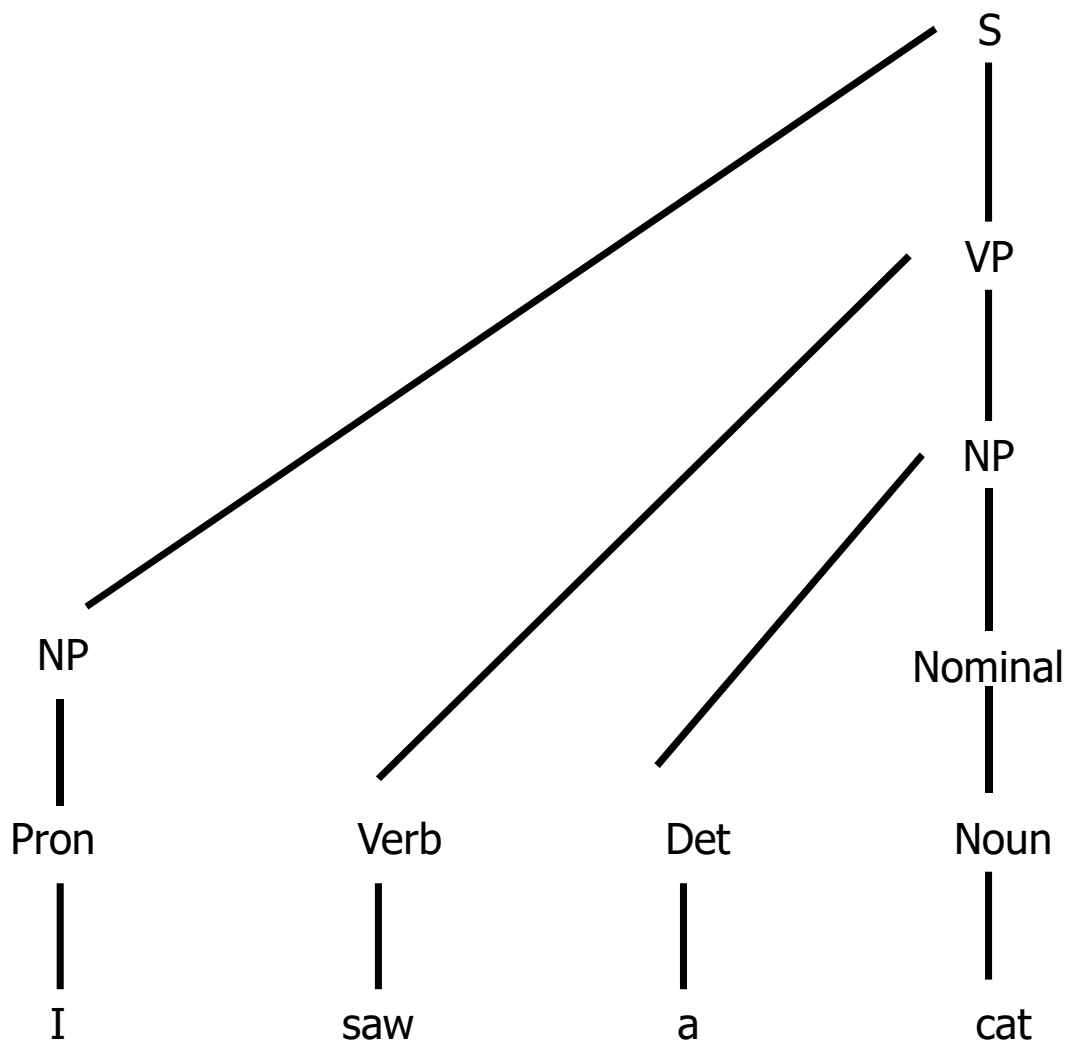
$NP \rightarrow Pron$

$Pron \rightarrow I$

$Verb \rightarrow saw$

$Det \rightarrow a$

$Noun \rightarrow cat$

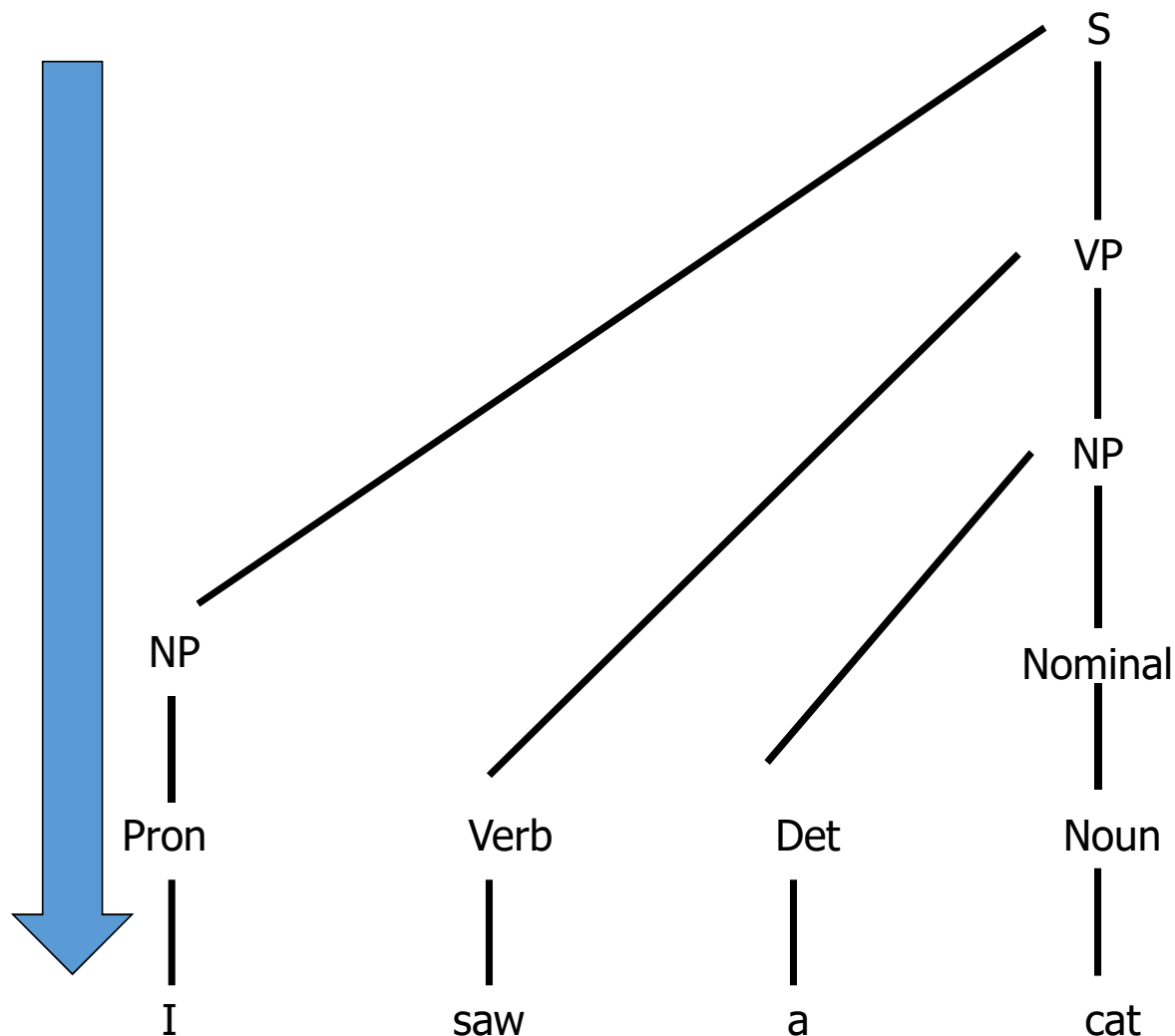




# ■从大单元逐层通过寻找CFG规则分解成小单元

$S \rightarrow NP VP$   
 $S \rightarrow Aux NP VP$   
 $S \rightarrow VP$   
 $NP \rightarrow Pronoun$   
 $NP \rightarrow Proper-Noun$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow Noun$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow Verb$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow Verb NP PP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$

$Pron \rightarrow I$   
 $Verb \rightarrow saw$   
 $Det \rightarrow a$   
 $Noun \rightarrow boy$



# 搜索中的问题—自顶向下

- 歧义导致的重复剖析

- 左递归





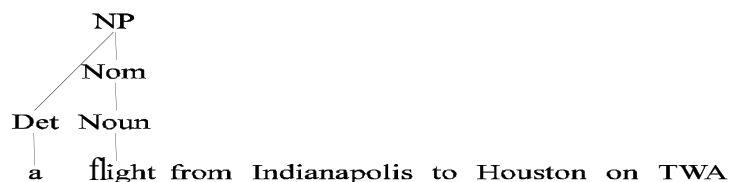
# 歧义导致的重复剖析

## ■ 自顶向下剖析如下NP

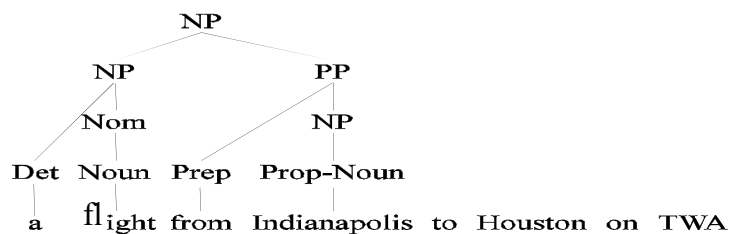
A flight from Indianapolis to Houston on TWA



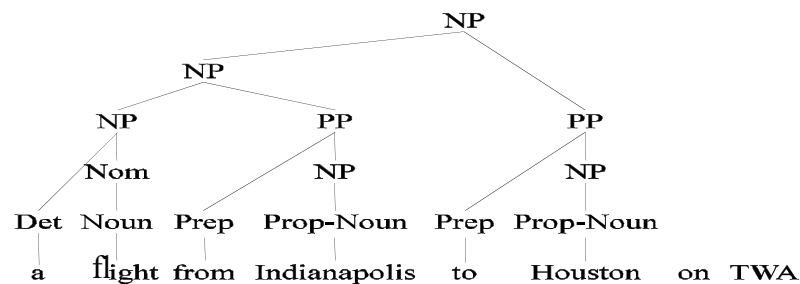
尝试1: NP → Det Nom  
 Nom → Noun  
 Det → a  
 Noun → flight



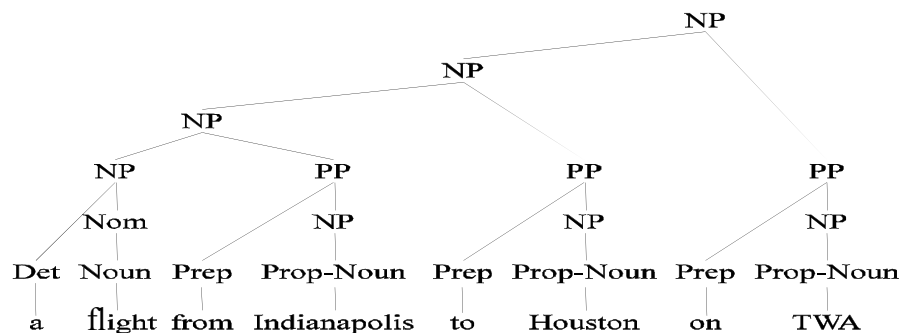
尝试2: NP → NP PP  
 NP → Det Nom  
 Nom → Noun  
 Det → a  
 Noun → flight



尝试3: NP → NP PP  
 NP → NP PP  
 NP → Det Nom  
 Nom → Noun  
 Det → a  
 Noun → flight



尝试4: NP → NP PP  
 NP → NP PP  
 NP → NP PP  
 NP → Det Nom  
 Nom → Noun  
 Det → a  
 Noun → flight



- 设有CFG:
- NP → Det Nominal
- NP → NP PP
- Nominal → Noun
- PP → Prep PP
- Det → a
- Noun → flight
- ...



# 剖析中的歧义多

## ■关于英语PP附着的歧义 (Church and Patil (1982))

■可能的不同剖析树数目与PP数目(n)的关系为:

$$C(n) = \frac{1}{n+2} \binom{2(n+1)}{n+1}$$

PPs	Parses
1	2
2	5
3	14
4	42
5	132
6	429

■例如: Show me the meal on flight 286 from SF to Denver 可剖析出14个不同的结构树

# 左递归

## ■考虑如下情况

■S  $\rightarrow$  NP VP

■S  $\rightarrow$  Aux NP VP

■NP  $\rightarrow$  NP PP

■NP  $\rightarrow$  Det Nominal

■...

■剖析句子：

■Did the flight...





# 搜索中的问题—自底向上

## ■产生无意义的结构

- I saw a cat

  - Verb → saw

  - Noun → saw 无意义的结构

## ■重复产生相同结构



- 蛮力搜索出现大量重复

- 在搜索中结合动态规划，记录中间过程的结果



# 大纲

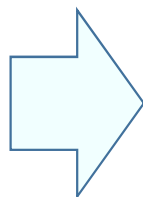
- 上下文无关语法
- 基于搜索的句法分析方法
- CKY句法分析算法
- Earley句法分析算法
- 句法剖析算法的评估
- 部分句法分析(Partial parsing)



# CKY parser: 1.转换: from CFG to CNF



$S \rightarrow NP VP$   
 $S \rightarrow Aux NP VP$  (1)  
 $S \rightarrow VP$  (2-1)  
 $S \rightarrow Verb NP$   
 $S \rightarrow Verb NP PP$  (3)  
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow Pronoun$  (4-1)  
 $NP \rightarrow Proper-Noun$  (5-1)  
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow Noun$  (6-1)  
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow Verb$  (2-2)  
 $VP \rightarrow Verb NP$   
 $VP \rightarrow Verb NP PP$  (7)  
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I | she | me$  (4-2)  
 $Proper-Noun \rightarrow Houston$  (5-2)  
 $Noun \rightarrow book | flight$  (6-2)  
 $Verb \rightarrow book | include$  (2-3)



$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$  (1-1)  
 $X1 \rightarrow Aux NP$  (1-2)  
 $S \rightarrow book | include$  (2)  
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$  (3-1)  
 $X2 \rightarrow Verb NP$  (3-2)  
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I | she | me$  (4)  
 $NP \rightarrow Houston$  (5)  
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book | flight$  (6)  
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book | include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$  (7-1)  
 $(X2 \rightarrow Verb NP)$  (3-2)  
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I | she | me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book | flight$   
 $Verb \rightarrow book | include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

# CKY parser: 2. 数据结构

0 Book 1 the 2 flight 3 through 4 Houston 5

(0,1)	(0,2)	(0,3)	(0,4)	(0,5)	
	(1,2)	(1,3)	(1,4)	(1,5)	
		(2,3)	(2,4)	(2,5)	
			(3,4)	(3,5)	
				(4,5)	

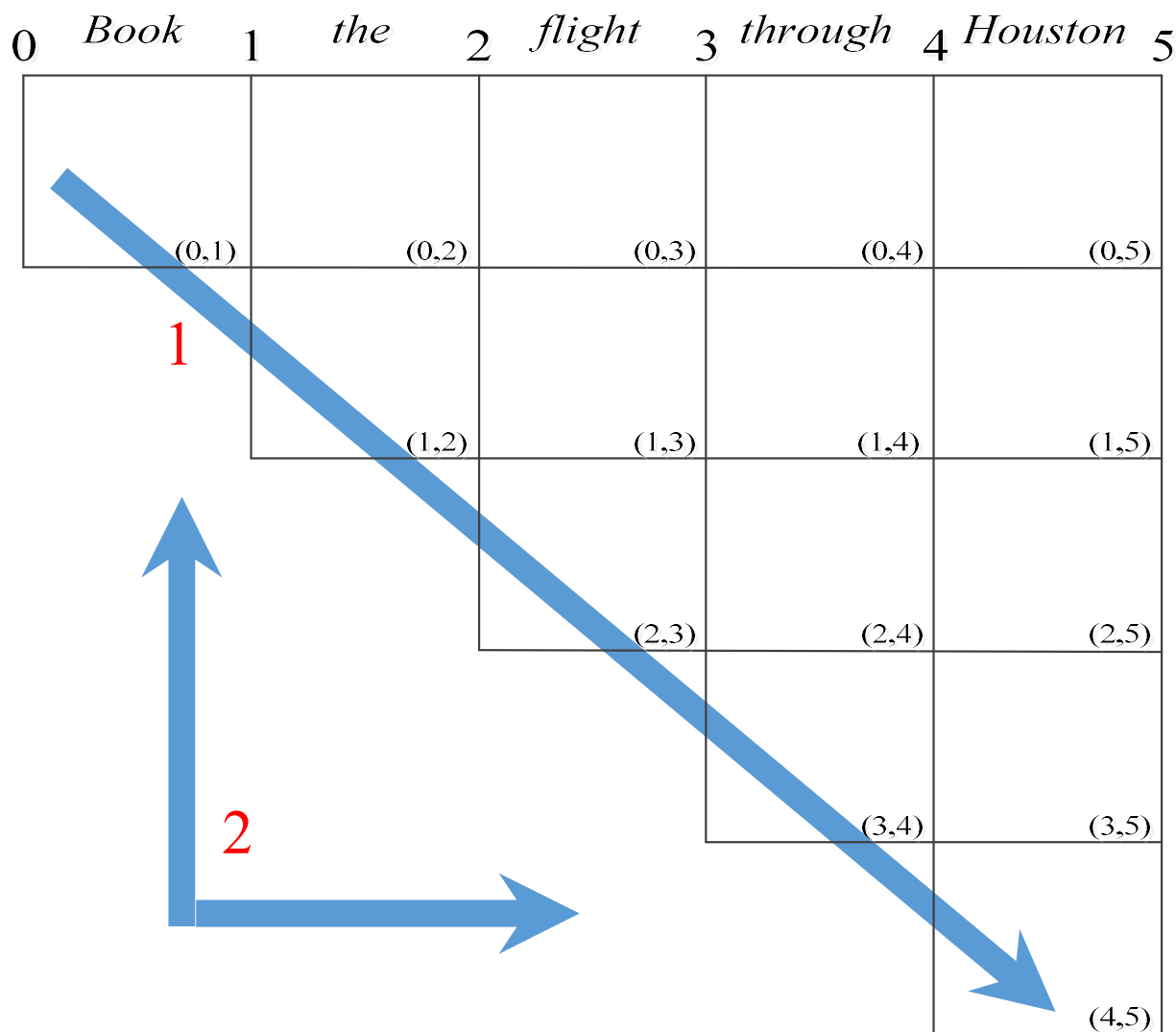
词编号: 0, 1, 2...

表格编号:

[i,j] 要填写i到j这个词串能构成什么句法单元, [0,1],[1,2]等对角线上是某个词, 再向左、向上 [0,2]\[1,3]\[0,3]等是包含更多词的单元, 例如 [0,2]是Book the, 最后右上角[0,5]包含所有词单元。



## CKY parser: 3. 算法框架



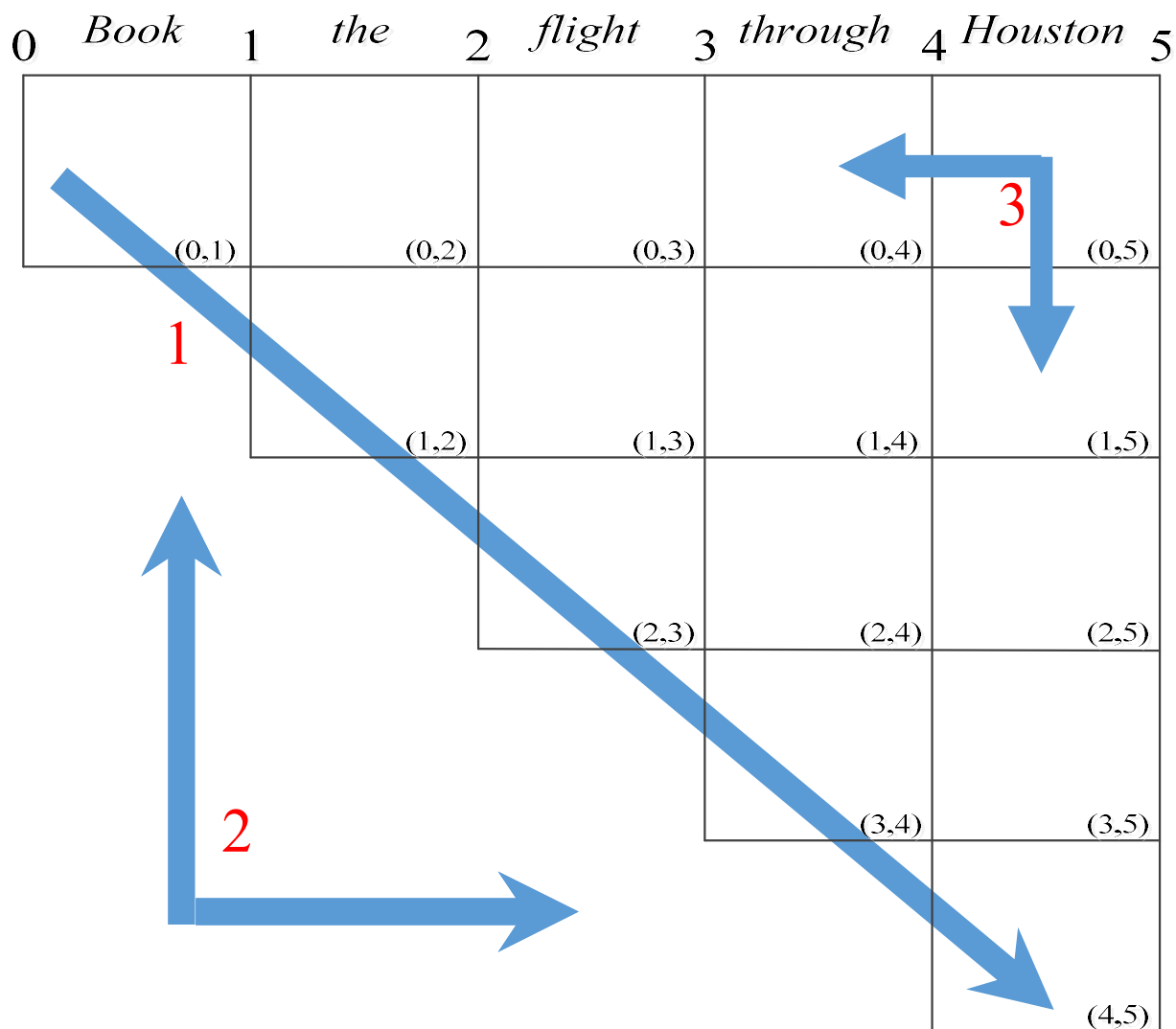
填表次序:

1、先对角线, 基本的每个单词的句法范畴

2、再从左到右、从下往上, 从小单元组合成大单元的次序(自底向上的算法), 并记录大单元由那两个小单元组成。



## CKY parser: 3. 算法框架



3、待填完[0,5]后，反向基于组合记录回溯获得组合，构建句法树。

前向填表：1、2  
反向建树：3

## CKY parser: 示例-前向计算



$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$   
 $X1 \rightarrow Aux NP$   
 $S \rightarrow book | include$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I | she | me$   
 $NP \rightarrow TWA | Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book | flight$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book | include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $(X2 \rightarrow Verb NP)$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I | she | me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book | flight$   
 $Verb \rightarrow book | include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

0	Book	1	the	2	flight	3	through	4	Huston	5
	(0,1)		(0,2)		(0,3)		(0,4)		(0,5)	
			(1,2)		(1,3)		(1,4)		(1,5)	
					(2,3)		(2,4)		(2,5)	
							(3,4)		(3,5)	
									(4,5)	

■以填[0,1]格为例：

■以[0,1]间词Book去匹配  
 规则集的右边：能匹配  
 上5个(右侧标红)，把这5  
 个规则的左边符号记录  
 在[0,1]中，即：  
 S,Nominal,VP,Noun,Verb

■类似的可以填满对角线





# CKY parser: 示例-前向计算

$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$   
 $X1 \rightarrow Aux NP$   
 $S \rightarrow book \mid include$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I \mid she \mid me$   
 $NP \rightarrow TWA \mid Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book \mid flight$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book \mid include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $(X2 \rightarrow Verb NP)$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I \mid she \mid me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book \mid flight$   
 $Verb \rightarrow book \mid include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

0 Book 1 the 2 flight 3 through 4 Houston 5

S, VP, Verb, Nominal, Noun (0,1)					
	Det (1,2)				
		Nominal, Noun (2,3)			
			Prep (3,4)		
				NP, Proper- Noun (4,5)	

■以填[0,1]格为例：

■以[0,1]间词Book去匹配  
 规则集的右边：能匹配  
 上5个(右侧标红)，把这5  
 个规则的左边符号记录  
 在[0,1]中，即：  
 S, Nominal, VP, Noun, Verb

■类似的可以填满对角线



■进一步填[0,2]，首先拆分[0,2]，即找[0,2]是如何从小单元构建、得到了什么大结构，

■只有只有一种方法[0,1]+[1,2],这两个单元是已经填好了的。

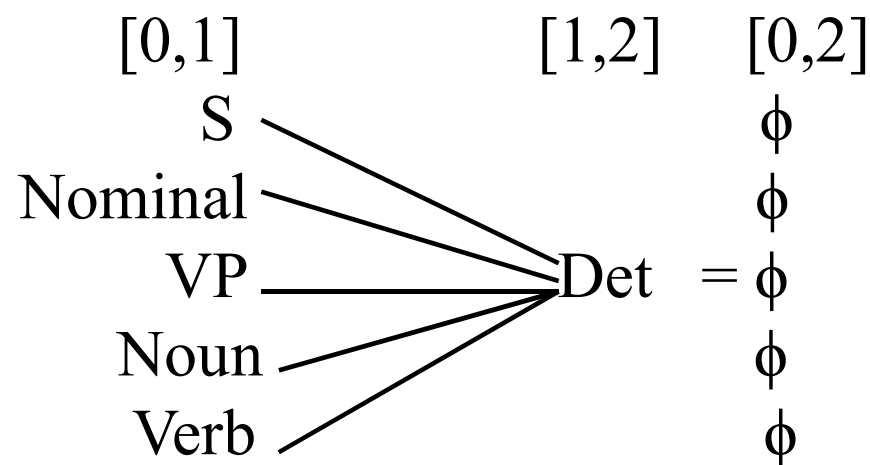
■右图：[0,1]中有：S、...

■[1,2]中有：Det

■把二者中的单元逐一进行组合，

■搜索规则的右边，是否有这种组合，把匹配上的规则的左边写到[0,2]中

■如果没有任何可以匹配的，则填 $\phi$





## CKY parser: 示例-前向计算



$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$   
 $X1 \rightarrow Aux NP$   
 $S \rightarrow book \mid include$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I \mid she \mid me$   
 $NP \rightarrow TWA \mid Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book \mid flight$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book \mid include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $(X2 \rightarrow Verb NP)$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I \mid she \mid me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book \mid flight$   
 $Verb \rightarrow book \mid include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

0	Book	1	the	2	flight	3	through	4	Houston	5
S,VP, Verb, Nominal, Noun	$\phi$									
(0,1)	(0,2)	(0,3)	(0,4)	(0,5)						
	Det							NP		
	(1,2)	(1,3)	(1,4)	(1,5)						
		Nominal, Noun								
		(2,3)	(2,4)	(2,5)						
			Prep							
			(3,4)	(3,5)						
				NP, Proper- Noun						
				(4,5)						



■进一步填[1,3], 首先还是拆分[1,3], 也只有一种方法[1,2]+[2,3],这两个单元是已经填好了的

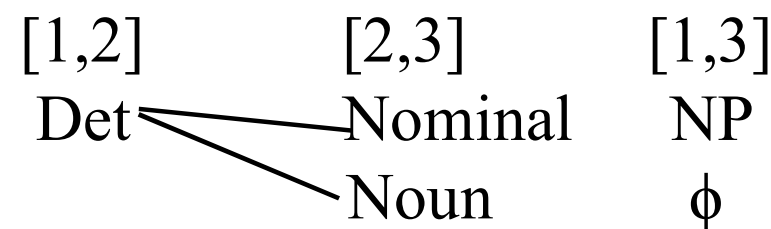
■右图: [1,2]中有: Det

■[2,3]中有: Nominal Noun

■把二者中的单元逐一进行组合,

■搜索规则的右边, 是否有这种组合, 把匹配上的规则的左边写到[1,3]中

■此时, Det Nominal 组合得到NP, 填入[1,3], 并记录组成源





## CKY parser: 示例-前向计算

$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$   
 $X1 \rightarrow Aux NP$   
 $S \rightarrow book \mid include$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I \mid she \mid me$   
 $NP \rightarrow TWA \mid Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book \mid flight$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book \mid include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $(X2 \rightarrow Verb NP)$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I \mid she \mid me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book \mid flight$   
 $Verb \rightarrow book \mid include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

0	Book	1	the	2	flight	3	through	4	Houston	5
S,VP, Verb, Nominal, Noun	$\phi$									
(0,1)	(0,2)	(0,3)	(0,4)	(0,5)						
	Det	NP						NP		
	(1,2)	(1,3)	(1,4)	(1,5)						
		Nominal, Noun								
		(2,3)	(2,4)	(2,5)						
			Prep							
			(3,4)	(3,5)						
				NP, Proper- Noun						
				(4,5)						



- 进一步填 $[0,3]$ , 首先还是拆分 $[0,3]$ , 这时有多种方法:
- 两分的:  $[0,1]+[1,3], [0,2]+[2,3]$
- 三分的:  $[0,1]+[1,2]+[2,3]$
- 拆分的单元是用来和规则右边匹配的, 三分意味着要匹配的右边有三个单元, 回忆: 已经把CFG转化为CNF, 右边不可能有三个单元, 因此, 三分不可能匹配上合适的规则, 因此, 只需要去进行两分的两种可能尝试, 把所有匹配的结果记录下来:

$[0,1] \quad + \quad [1,3]$

$[0,2] \quad + \quad [2,3]$

## CKY parser: 示例-前向计算



$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$   
 $X1 \rightarrow Aux NP$   
 $S \rightarrow book \mid include$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I \mid she \mid me$   
 $NP \rightarrow TWA \mid Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book \mid flight$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book \mid include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $(X2 \rightarrow Verb NP)$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I \mid she \mid me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book \mid flight$   
 $Verb \rightarrow book \mid include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

0	Book	1	the	2	flight	3	through	4	Houston	5
S,VP, Verb, Nominal, Noun	$\phi$	S, VP, X2								
(0,1)	(0,2)	(0,3)	(0,4)	(0,5)						
	Det	NP								
	(1,2)	(1,3)	(1,4)	(1,5)						
		Nominal, Noun								
		(2,3)	(2,4)	(2,5)						
			Prep							
			(3,4)	(3,5)						
				NP, Proper- Noun						
				(4,5)						

■如此继续：



$S \rightarrow NP VP$   
 $S \rightarrow X1 VP$   
 $X1 \rightarrow Aux NP$   
 $S \rightarrow book \mid include$   
 $S \rightarrow Verb NP$   
 $S \rightarrow X2 PP$   
 $X2 \rightarrow Verb NP$   
 $S \rightarrow Verb PP$   
 $S \rightarrow VP PP$   
 $NP \rightarrow I \mid she \mid me$   
 $NP \rightarrow TWA \mid Houston$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow book \mid flight$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow book \mid include$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow X2 PP$   
 $(X2 \rightarrow Verb NP)$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I \mid she \mid me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book \mid flight$   
 $Verb \rightarrow book \mid include$   
 $Preposition \rightarrow through$   
 $Det \rightarrow the$

# CKY parser: 示例-前向计算



0	Book	1	the	2	flight	3	through	4	Houston	5
S, VP, Verb, Nominal, Noun (0,1)	$\phi$			S, VP, X2 (0,2)	$\phi$			S1, VP1, S2, VP2, S3 (0,3)		
	Det (1,2)			NP (1,3)	$\phi$			NP (1,4)		
				Nominal, Noun (2,3)	$\phi$			Nominal (2,4)		
					Prep (3,4)			PP (3,5)		
								NP, Proper-Noun (4,5)		



## CKY parser: 示例-反向回溯



0	Book	1	the	2	flight	3	through	4	Houston	5
S, VP, Verb, Nominal, Noun (0,1)		φ	S, VP, X2 (0,2)		φ	S1, VP1, S2, VP2, S3 (0,3)		φ	(0,5)	
		Det (1,2)	NP (1,3)		φ	NP (1,4)		φ	(1,5)	
			Nominal, Noun (2,3)		φ	Nominal (2,4)		φ	(2,5)	
					Prep (3,4)	PP (3,5)		φ	(3,5)	
						NP, Proper-Noun (4,5)		φ	(4,5)	

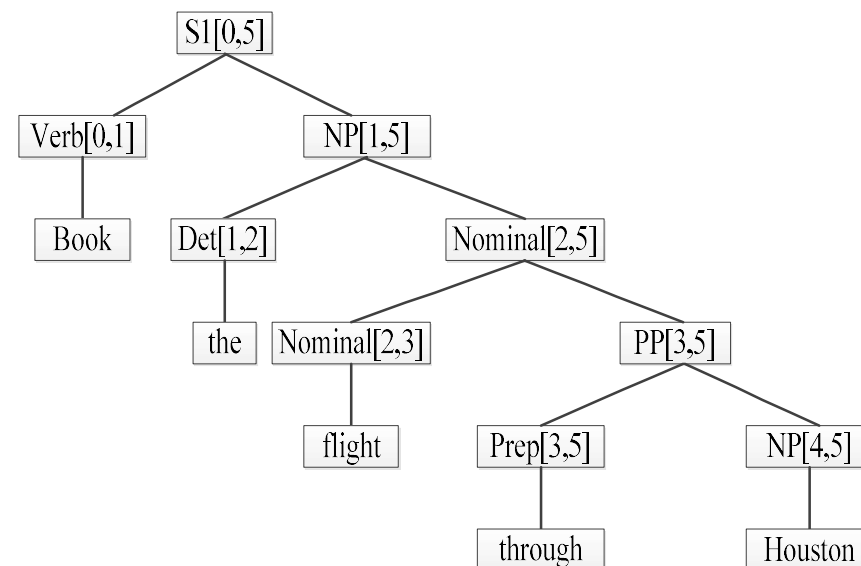
$S1[0,5] \rightarrow \text{Verb}[0,1] \text{ NP}[1,5]$

$\text{NP}[1,5] \rightarrow \text{Det}[1,2] \text{ Nominal}[2,5]$

$\text{Nominal}[2,5] \rightarrow \text{Nominal}[2,3] \text{ PP}[3,5]$

$\text{PP}[3,5] \rightarrow \text{Prep}[3,4] \text{ NP}[4,5]$

对应的树:



S2、S3可以类似得到





# CKY algorithm

```
function CKY-Parse(words, grammar) return table
  for j  $\leftarrow$  from 1 to Length(words) do
    table[j-1,j]  $\leftarrow$  {A | A  $\rightarrow$  words[j]  $\in$  grammar}
    for i  $\leftarrow$  from j-2 downto 0 do
      for k  $\leftarrow$  i+1 to j-1 do
        table[i,j]  $\leftarrow$  table[i,j]  $\cup$ 
          {A | A  $\rightarrow$  B C  $\in$  grammar, B  $\in$  table[i,k], C  $\in$  table[k,j]}
```



## ■几点说明

### ■单项写到最大

■ $V \rightarrow \text{can}$ ,  $VP \rightarrow V$ ,  $S \rightarrow VP$

■ $\rightarrow VP \rightarrow \text{can}$   $S \rightarrow \text{can}$

### ■同一格有多个相同名称的结构需要编号分别写出，并记录各自的来源

■NP1, NP2



# CKY 算法总结

## ■ 知识

- 词汇知识: 词典(POS)
- 句法知识: CFG语法(CNF, not necessary? )

## ■ 算法

- 自底向上: 从词到 S
- 两阶段DP
  - Forward
  - Backtrace

# 大纲

- 上下文无关语法
- 基于搜索的句法分析方法
- CKY句法分析算法
- Earley句法分析算法
- 句法剖析算法的评估
- 部分句法分析(Partial parsing)





# Earley 算法

- 自顶向下搜索 + 动态规划
- 状态 (dotted rule) 对比CKY状态 V,NP...
  - $S \rightarrow \bullet VP, [0,0]$
  - $NP \rightarrow Det \bullet Nominal, [1,2]$
  - $VP \rightarrow V NP \bullet, [0,3]$
- 数据结构：表
  - $N(\text{词数}) + 1$  列.
  - 每个词位置有一列状态，表示到当前位置为止产生的剖析结果.
    - For example:
 

	0	book	1	the	2	flight	3
■	state1		state3		state4		state6
■	state2				state5		state7

■ For example:	0	book	1	the	2	flight	3
■	state1		state3		state4		state6
■	state2				state5		state7



# Earley算法

## ■ 一个状态包括三种信息:

■  $A \rightarrow B \cdot C [m, n]$

■ 1. 一个对应于单个语法规则的子树:  $A \rightarrow B C$

■ 2. 完成这个子树的进程情况:

■ 已完成了B, 还希望找到一个C

■ 3. 对应的句子读取位置:

■  $[m, n]$ :

■ m: A这个单元开始时的句子位置

■ n: 当前处理状态的句子位置



## ■导致状态变化的操作：

- 预测(PREDICTOR)

- 完成(COMPLETER)

- 扫描(SCANNER)



## ■ 预测(PREDICTOR)

### ■ 操作的条件

- 状态中点的右边是一个 非终止符

### ■ 操作

- 在规则集中规则的左边匹配当前状态点右边的符号，匹配成功一个创建一个新状态增加到当前列：新状态由匹配成功的规则生成，点和[]的信息继承当前状态。

### ■ 例子

- 对于  $S \rightarrow \bullet VP$ , [0,0], VP 是非终止符, 使用预测操作:
  - $VP \rightarrow \bullet Verb$ , [0,0]
  - $VP \rightarrow \bullet Verb NP$ , [0,0]
  - $VP \rightarrow \bullet Verb NP PP$ , [0,0],
  - ...





## ■扫描(SCANNER)

### ■操作的条件

- 状态中点的右边是一个 POS

### ■操作

- 在规则集(POS集)中匹配当前状态的点的右边，匹配成功的话增加一个状态到下一列，新状态是由匹配成功的规则生成的：点右移，结束位置右移+1

### ■例子

- 对于  $VP \rightarrow \bullet \text{Verb NP}$ ,  $[0,0]$  verb 是POS, 用扫描:

- Verb  $\rightarrow$  book  $\bullet$ ,  $[0,1]$

- 词汇在预测中没有出现的POS不会用到。



## ■完成(COMPLETER)

### ■操作的条件

- 状态中点位于最右边

### ■操作

- 匹配前面状态的点的右边，匹配成功的话增加一个状态到当前列，新状态是匹配成功状态生成的：点右移，结束位置累加右移

### ■例子

- 对于  $NP \rightarrow \text{Det Nominal} \bullet$ ,  $[1,3]$ , and also have  $NP \rightarrow VP \bullet$   
•  $NP PP$ ,  $[0,1]$ .
- Then use COMPLETER
- $NP \rightarrow VP NP \bullet PP$ ,  $[0,3]$



## ■ 执行三个操作的条件：

### ■ 由 $\cdot$ 的右边符号决定

#### ■ PREDICTOR: $\cdot$ Non-terminal

■ E.g.:  $\cdot$  NP in  $S \rightarrow \cdot$  NP VP, [0,0],

#### ■ SCANNER: $\cdot$ Terminal

■ E.g.:  $\cdot$  verb in  $VP \rightarrow \cdot$  verb NP, [1,1],

#### ■ COMPLETER: $\cdot$ end

■ E.g.:  $\cdot$  in  $VP \rightarrow$  Verb VP  $\cdot$ , [2,3],



$S \rightarrow NP VP$   
 $S \rightarrow Aux NP VP$   
 $S \rightarrow VP$   
 $NP \rightarrow Pronoun$   
 $NP \rightarrow Proper-Noun$   
 $NP \rightarrow Det Nominal$   
 $Nominal \rightarrow Noun$   
 $Nominal \rightarrow Nominal Noun$   
 $Nominal \rightarrow Nominal PP$   
 $VP \rightarrow Verb$   
 $VP \rightarrow Verb NP$   
 $VP \rightarrow Verb NP PP$   
 $VP \rightarrow Verb PP$   
 $VP \rightarrow VP PP$   
 $PP \rightarrow Preposition NP$   
 $Pronoun \rightarrow I | she | me$   
 $Proper-Noun \rightarrow Houston$   
 $Noun \rightarrow book | flight$   
 $Verb \rightarrow book | include$   
 $Det \rightarrow that$

	0	book	1	that	2	flight.	3
	chart0		chart1		chart2		chart3



Chart0			
S0	$\gamma \rightarrow \cdot S$	[0,0]	Predictor
S1	$S \rightarrow \cdot NP VP$	[0,0]	Predictor
S2	$S \rightarrow \cdot Aux NP VP$	[0,0]	Scanner
S3	$S \rightarrow \cdot VP$	[0,0]	Predictor
S4	$NP \rightarrow \cdot Pronoun$	[0,0]	Predictor
S5	$NP \rightarrow \cdot Proper-Noun$	[0,0]	Predictor
S6	$NP \rightarrow \cdot Det Nominal$	[0,0]	Predictor
S7	$VP \rightarrow \cdot Verb$	[0,0]	Predictor
S8	$VP \rightarrow \cdot Verb NP$	[0,0]	Predictor
S9	$VP \rightarrow \cdot Verb NP PP$	[0,0]	Predictor
S10	$VP \rightarrow \cdot Verb PP$	[0,0]	Predictor
S11	$VP \rightarrow \cdot VP PP$	[0,0]	Predictor



Chart1			
S12	Verb $\rightarrow$ book $\cdot$	[0,1]	Scanner
S13	VP $\rightarrow$ verb $\cdot$	[0,1]	Completer
S14	VP $\rightarrow$ verb $\cdot$ NP	[0,1]	Completer
S15	VP $\rightarrow$ verb $\cdot$ NP PP	[0,1]	Completer
S16	VP $\rightarrow$ verb $\cdot$ PP	[0,1]	Completer
S17	S $\rightarrow$ VP $\cdot$	[0,1]	Completer
S18	VP $\rightarrow$ VP $\cdot$ PP	[0,1]	Completer
S19	NP $\rightarrow$ $\cdot$ Pronoun	[1,1]	Predictor
S20	NP $\rightarrow$ $\cdot$ Proper-Noun	[1,1]	Predictor
S21	NP $\rightarrow$ $\cdot$ Det Nominal	[1,1]	Predictor
S22	NP $\rightarrow$ $\cdot$ Prep NP	[1,1]	Predictor



Chart2			
S23	Det→ that ·	[1,2]	Scanner
S24	NP→ Det · Nominal	[1,2]	Completer
S25	Nominal→ · Noun	[2,2]	Predictor
S26	Nominal→ · Nominal Noun	[2,2]	Predictor
S27	Nominal→ · Nominal PP	[2,2]	Predictor



Chart3			
S28	Noun $\rightarrow$ flight $\cdot$	[2,3]	Scanner
S29	Nominal $\rightarrow$ Noun $\cdot$	[2,3]	Completer
S30	NP $\rightarrow$ Det Nominal $\cdot$	[1,3]	Completer
S31	Nominal $\rightarrow$ Nominal $\cdot$ Noun	[2,3]	Completer
S32	Nominal $\rightarrow$ Nominal $\cdot$ PP	[2,3]	Completer
S33	VP $\rightarrow$ Verb NP $\cdot$	[0,3]	Completer
S34	VP $\rightarrow$ Verb NP $\cdot$ PP	[0,3]	Completer
S35	PP $\rightarrow \cdot$ Prep NP	[3,3]	Predictor
S36	S $\rightarrow$ VP $\cdot$	[0,3]	Completer
S37	VP $\rightarrow$ VP $\cdot$ PP	[0,3]	Completer
	$\gamma \rightarrow$ S $\cdot$	[0,3]	





## ■回溯获得剖析子树

回溯				
Chart1	S12	Verb→ book ·	[0,1]	Scanner
Chart2	S23	Det→ that ·	[1,2]	Scanner
Chart3	S28	Noun→ flight ·	[2,3]	Scanner
	S29	Nominal→ Noun ·	[2,3]	(S28)
	S30	NP→ Det Nominal ·	[1,3]	(S23,S29)
	S33	VP→ Verb NP ·	[0,3]	(S12,S30)
	S36	S→ VP ·	[0,3]	(S33)

# 大纲

- 上下文无关语法
- 基于搜索的句法分析方法
- CKY句法分析算法
- Earley句法分析算法
- 句法剖析算法的评估
- 部分句法分析(Partial parsing)





## ■依据单元(constituent)的评估

### ■ $P \setminus R \setminus F_1$

■  $R = \# \text{正确剖析出的单元} / \# \text{所有标答单元}$

■  $P = \# \text{正确剖析出的单元} / \# \text{所有剖析出的单元}$

■  $F_1 = 2RP / (R + P)$

### ■cross-brackets

■  $((AB)C) \rightarrow (A(B\ C))$

# 大纲

- 上下文无关语法
- 基于搜索的句法分析方法
- CKY句法分析算法
- Earley句法分析算法
- 句法剖析算法的评估
- 部分句法分析(Partial parsing)





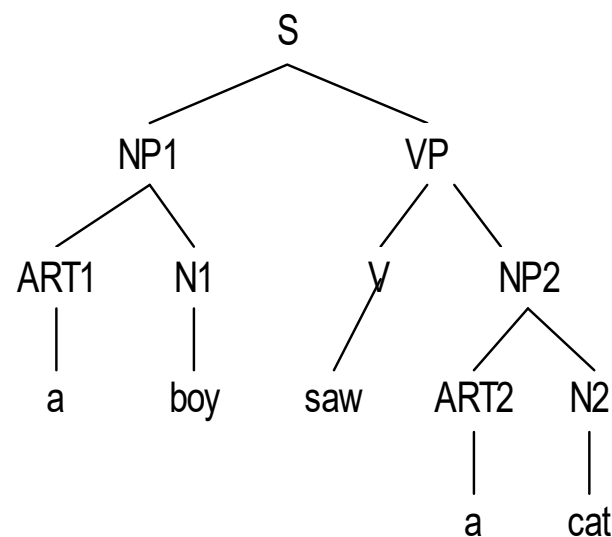
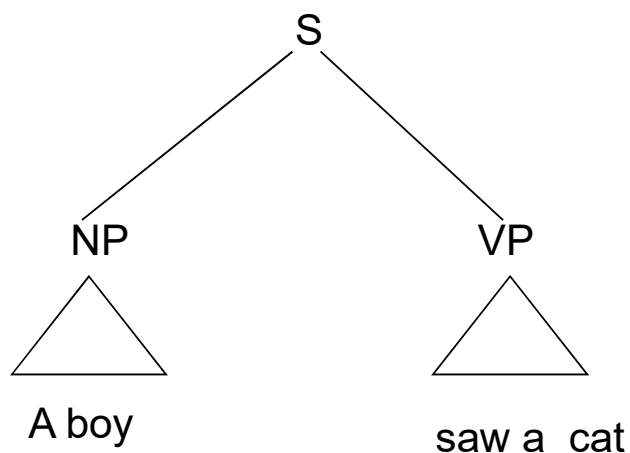
## 从实际应用来看

- Full-parsing 性能还不够好
  - 真实语料的复杂性
  - 语言知识的不足
    - CFG不够，需要进一步提升 CFG
- Full-parsing 可能有点浪费
  - 实际应用不需要那么详尽的树分析
  - 需要更快的分析



# Partial parsing

■ 例: a boy saw a cat



- Shallow parsing/ 浅层分析
- chunking / 组块分析



# Partial parsing方法

## ■从序标的视角来看

### ■BIO Tagging

■B: the beginning of a chunk

■I: internal parts of a chunk

■O: outside any chunk

例子

词序列: The morning flight from Denver has arrived

标序列: B\_NP I\_NP I\_NP B\_PP B\_NP B\_VP I\_VP

Partial structure: [The morning flight]NP [from]PP [Denver]NP [has arrived]VP



# Thank you !