Syntax: CYK algorithm and PCFG

So far

- What is syntax?
- Grammars
- Parsing
- CFG
- Derivation- Top down and bottom up
- Ambiguity
- **Searching tree space- Dynamic programming- CYK algorithm**

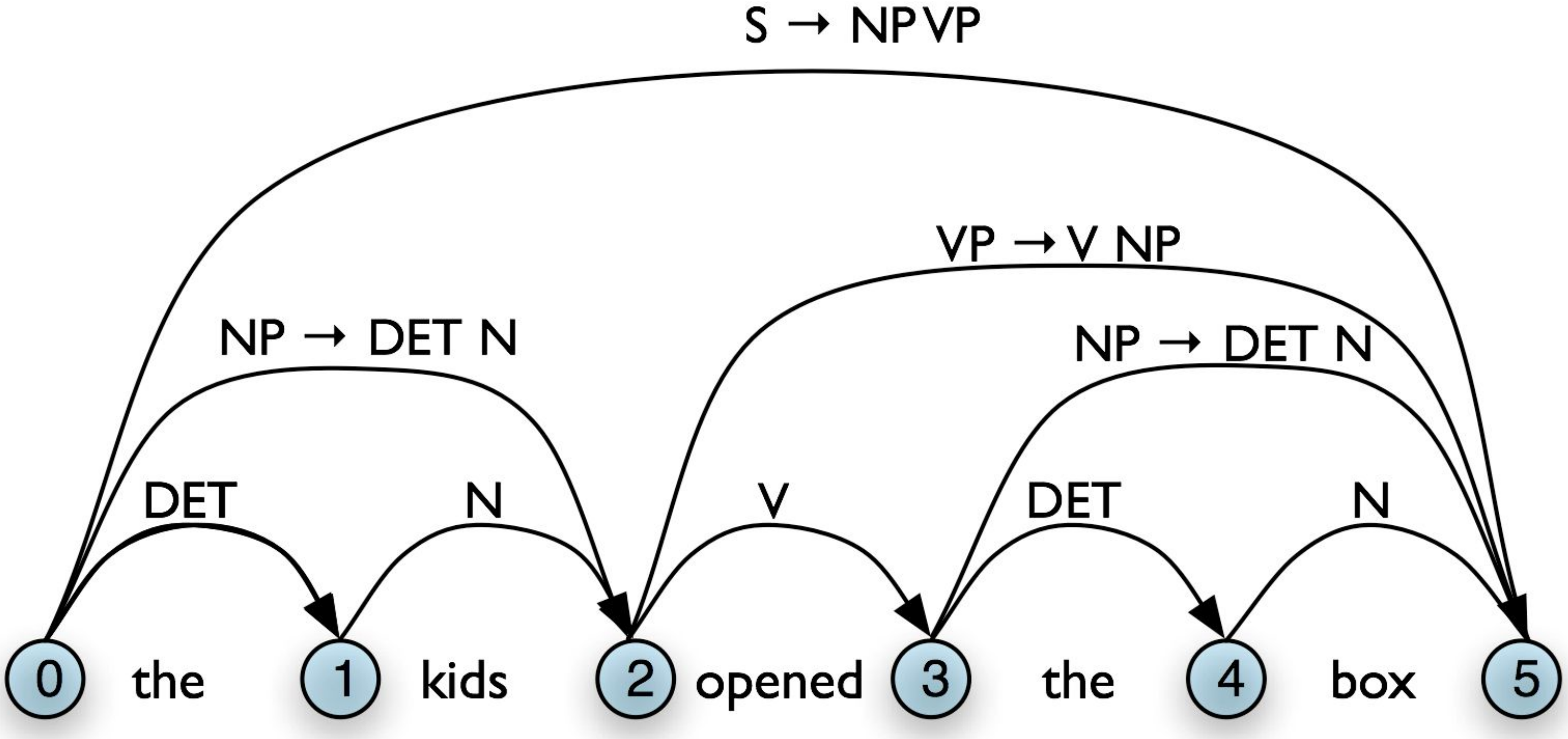
Chart Parsing

Dynamic programming **stores intermediate results** and re-uses them when appropriate, achieving significant efficiency gains.

This technique can be applied to syntactic parsing, allowing us to store partial solutions to the parsing task and then look them up as necessary in order to efficiently arrive at a complete solution.

This approach to parsing is known as **chart parsing**

Well-Formed Substring Tables



Dynamic Programming Approaches for parsing

- **Cocke-Kasami-Younger (CKY) algorithm**
- Chart parsing (NLU, James Alen)
- Earley Algorithm (NLU, James Alen)

CKY Algorithm

- Dynamic Programming approach
- Divide the problem into subproblems
- Use tables to store subtrees for each of the various constituents in the input as they are discovered
- The efficiency gain arises from the fact that these subtrees are discovered once, stored, and then used in all parses calling for that constituent.
- This solves problems of
  - Reparsing sentence
  - Ambiguity problem

CYK Algorithm

- CFG should be in CNF
  - $A \rightarrow B\ C$ where A, B, C are Non-terminals
  - $A \rightarrow w$ where A is a non-terminal and w is a terminal

- So first step is convert all productions to CNF

- Example-1:
  - $A \rightarrow B$
  - $B \rightarrow w$

$A \rightarrow w$

- Example-2
  - $A \rightarrow BCD$

$A \rightarrow PD$
$P \rightarrow BC$

CKY Algorithm

- For sentence of length n (words), we need a table of (n+1)x(n+1)
- Each cell [i,j] contain set of non-terminals that represents constituents that span positions i to j
- $_0$**Book** $_1$**that** $_2$**flight**$_3$ ← word position
- Each entry i..j can be split into [i,k] and [k,j] for all i<k<j

Example: Book that flight through Houston

$\mathscr{L}_1$ **Grammar**	$\mathscr{L}_1$ **in CNF**
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

5					
4					
3					
2					
1					
0	Book	that	flight	through	Houston
	1	2	3	4	5

5	1,5	2,5	3,5	4,5	5,5
4	1,4	2,4	3,4	4,4	
3	1,3	2,3	3,3		
2	1,2	2,2			
1	1,1				
0	Book	the	flight	through	Houston
	1	2	3	4	5

5	1,5	2,5	3,5	4,5	NP, ProperNoun
4	1,4	2,4	3,4	Prep	
3	1,3	2,3	S, VP		
2	1,2	Det			
1	S, VP, Nominal,				
0	Book	the	flight	through	Houston
	1	2	3	4	5

i<k<j

5	1,5	2,5	3,5	PP	NP, ProperNoun	
4	1,4	2,4	3,4	Prep		
3	1,3	NP	S, VP, NP			
2	1,2	Det				
1	S, VP, Nominal,					
0	Book	the	flight	through	Houston	
	1	2	3	4	5	

5	S	2,5	3,5	PP	NP, ProperNoun
4	1,4	2,4	3,4	Prep	
3	1,3	NP	S, VP, NP		
2	1,2	Det			
1	S, VP, Nominal,				
0	Book	the	flight	through	Houston
	1	2	3	4	5

CKY Algorithm

function CKY-PARSE(*words, grammar*) **returns** *table*

for $j \leftarrow$ **from** 1 **to** LENGTH(*words*) **do**
  $table[j-1, j] \leftarrow \{A \mid A \rightarrow words[j] \in grammar\}$
  for $i \leftarrow$ **from** $j-2$ **downto** 0 **do**
    for $k \leftarrow i+1$ **to** $j-1$ **do**
      $table[i,j] \leftarrow table[i,j] \cup$
        $\{A \mid A \rightarrow BC \in grammar,$
          $B \in table[i,k],$
          $C \in table[k,j]\}$

Figure 13.10 The CKY algorithm

Statistical Parsing

- **Problem of Ambiguity**
  - Coordination ambiguity
  - Attachment ambiguity
- **Probabilistic Parsing**
  - Compute probability of each interpretation
  - Choose the most probable one
- **Probabilistic CFG (PCFG)**
  - **a probabilistic augmentation of context-free grammars**
  -

PCFG

N a set of **non-terminal symbols** (or **variables**)

Σ a set of **terminal symbols** (disjoint from N)

R a set of **rules** or productions, each of the form $A \rightarrow \beta$ $[p]$, where A is a non-terminal, β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$, and p is a number between 0 and 1 expressing $P(\beta|A)$

S a designated **start symbol**

That is, a PCFG differs from a standard CFG by augmenting each rule in R with a conditional probability:

$$A \rightarrow \beta \quad [p]$$

Here p expresses the probability that the given non-terminal A will be expanded to the sequence β. That is, p is the conditional probability of a given expansion β given the left-hand-side (LHS) non-terminal A. We can represent this probability as

$$P(A \rightarrow \beta)$$

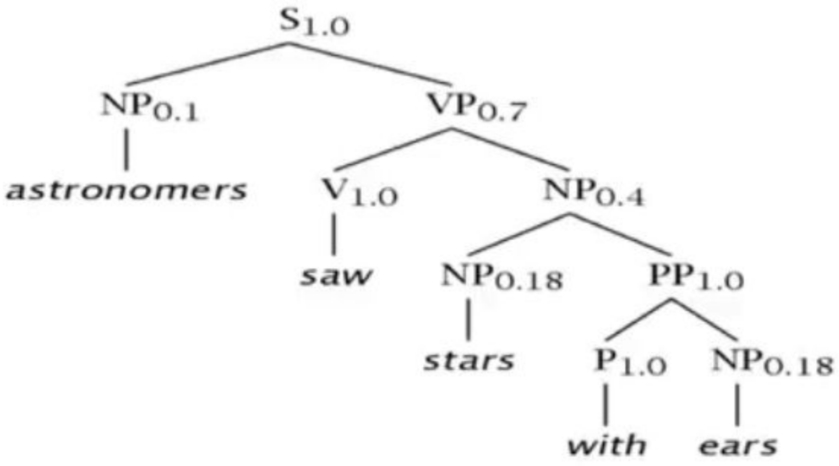
or as

$$P(A \rightarrow \beta|A)$$

Example

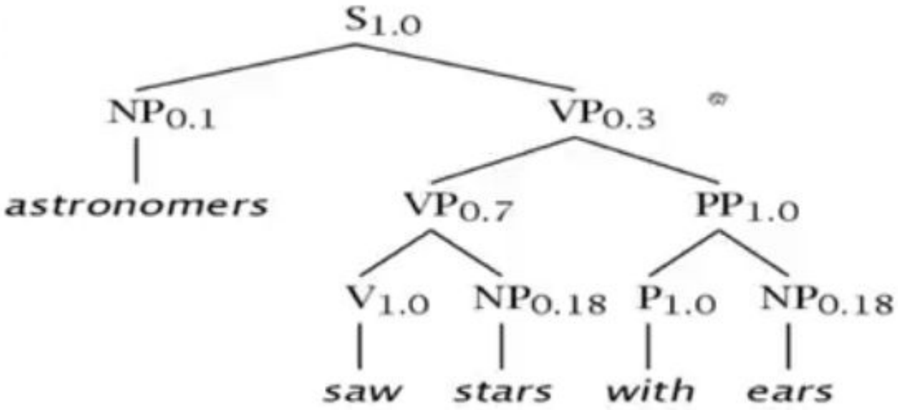
S	→	NP VP	1.0	NP	→	NP PP	0.4
VP	→	V NP	0.7	NP	→	*astronomers*	0.1
VP	→	VP PP	0.3	NP	→	*ears*	0.18
PP	→	P NP	1.0	NP	→	*saw*	0.04
P	→	*with*	1.0	NP	→	*stars*	0.18
V	→	*saw*	1.0	NP	→	*telescope*	0.1

t_1:



$S_{1.0}$
$NP_{0.1}$
$VP_{0.7}$
astronomers
$V_{1.0}$
$NP_{0.4}$
saw
$NP_{0.18}$
$PP_{1.0}$
stars
$P_{1.0}$
$NP_{0.18}$
with
ears

t_2:



$S_{1.0}$
$NP_{0.1}$
$VP_{0.3}$
astronomers
$VP_{0.7}$
$PP_{1.0}$
$V_{1.0}$
$NP_{0.18}$
$P_{1.0}$
$NP_{0.18}$
saw
stars
with
ears

- $P(t)$: The probability of tree is the product of the probabilities of the rules used to generate it
- $P(w_{1n})$: The probability of the string is the sum of the probabilities of the trees which have that string as their yield

$$
\begin{aligned}
w_{15} &= \textit{astronomers saw stars with ears} \\
P(t_1) &= 1.0 * 0.1 * 0.7 * 1.0 * 0.4 * 0.18 \\
&\quad * 1.0 * 1.0 * 0.18 \\
&= 0.0009072 \\
P(t_2) &= 1.0 * 0.1 * 0.3 * 0.7 * 1.0 * 0.18 \\
&\quad * 1.0 * 1.0 * 0.18 \\
&= 0.0006804 \\
P(w_{15}) &= P(t_1) + P(t_2) \\
&= 0.0009072 + 0.0006804 \\
&= 0.0015876
\end{aligned}
$$

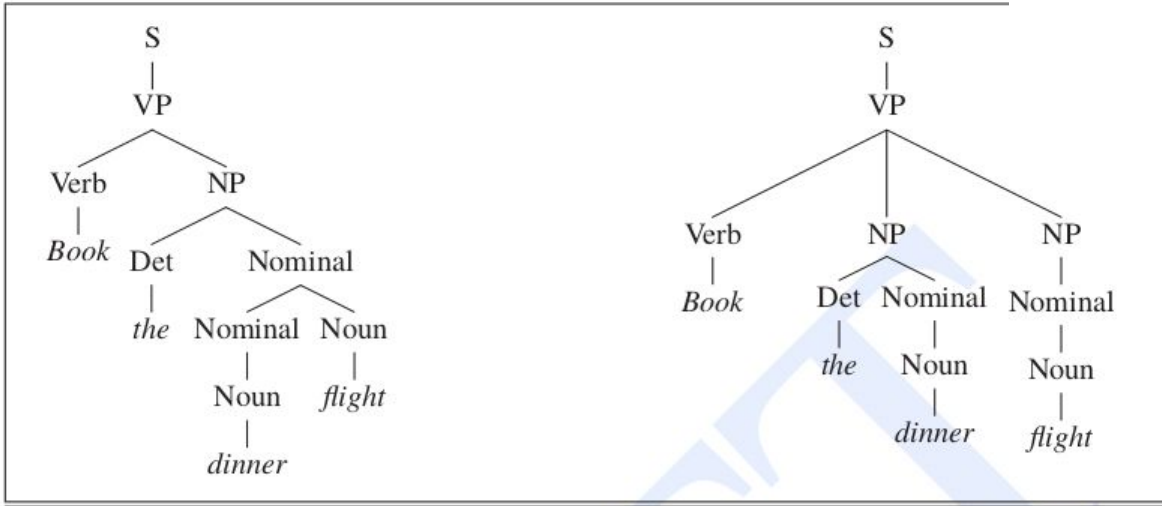
Example

Consistent: if
probability
sums to 1

Grammar		Lexicon
$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow that$ [.10] | a [.30] | the [.60]
$S \rightarrow Aux\ NP\ VP$	[.15]	$Noun \rightarrow book$ [.10] | $flight$ [.30]
$S \rightarrow VP$	[.05]	| $meal$ [.15] | $money$ [.05]
$NP \rightarrow Pronoun$	[.35]	| $flights$ [.40] | $dinner$ [.10]
$NP \rightarrow Proper\text{-}Noun$	[.30]	$Verb \rightarrow book$ [.30] | $include$ [.30]
$NP \rightarrow Det\ Nominal$	[.20]	| $prefer$; [.40]
$NP \rightarrow Nominal$	[.15]	$Pronoun \rightarrow I$ [.40] | she [.05]
$Nominal \rightarrow Noun$	[.75]	| me [.15] | you [.40]
$Nominal \rightarrow Nominal\ Noun$	[.20]	$Proper\text{-}Noun \rightarrow Houston$ [.60]
$Nominal \rightarrow Nominal\ PP$	[.05]	| TWA [.40]
$VP \rightarrow Verb$	[.35]	$Aux \rightarrow does$ [.60] | can [40]
$VP \rightarrow Verb\ NP$	[.20]	$Preposition \rightarrow from$ [.30] | to [.30]
$VP \rightarrow Verb\ NP\ PP$	[.10]	| on [.20] | $near$ [.15]
$VP \rightarrow Verb\ PP$	[.15]	| $through$ [.05]
$VP \rightarrow Verb\ NP\ NP$	[.05]	
$VP \rightarrow VP\ PP$	[.15]	
$PP \rightarrow Preposition\ NP$	[1.0]	

(where each rule i can be expressed as $LHS_i \rightarrow RHS_i$):

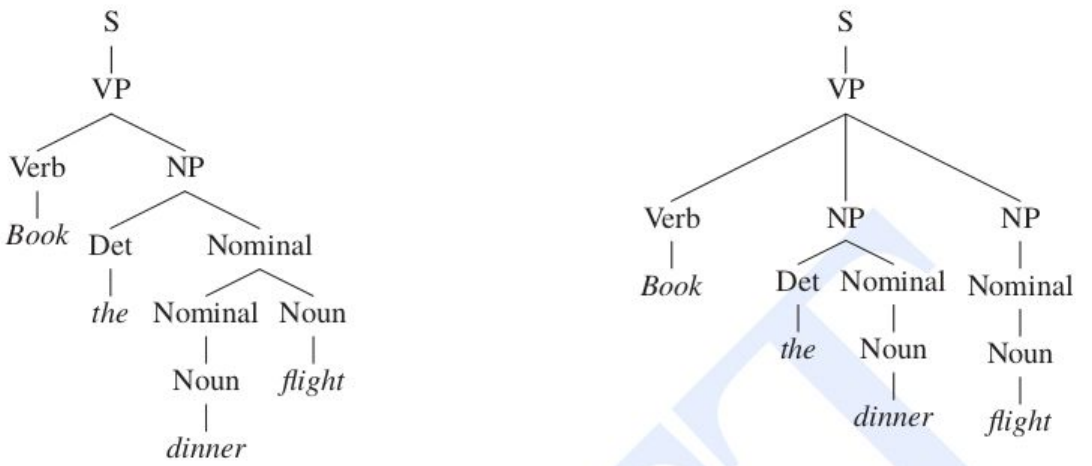
$$P(T,S) = \prod_{i=1}^{n} P(RHS_i|LHS_i)$$



Rules			P	Rules			P
S	$\rightarrow$	VP	.05	S	$\rightarrow$	VP	.05
VP	$\rightarrow$	Verb NP	.20	VP	$\rightarrow$	Verb NP NP	.10
NP	$\rightarrow$	Det Nominal	.20	NP	$\rightarrow$	Det Nominal	.20
Nominal	$\rightarrow$	Nominal Noun	.20	NP	$\rightarrow$	Nominal	.15
Nominal	$\rightarrow$	Noun	.75	Nominal	$\rightarrow$	Noun	.75
				Nominal	$\rightarrow$	Noun	.75
Verb	$\rightarrow$	book	.30	Verb	$\rightarrow$	book	.30
Det	$\rightarrow$	the	.60	Det	$\rightarrow$	the	.60
Noun	$\rightarrow$	dinner	.10	Noun	$\rightarrow$	dinner	.10
Noun	$\rightarrow$	flights	.40	Noun	$\rightarrow$	flights	.40

(where each rule i can be expressed as $LHS_i \rightarrow RHS_i$):

$$P(T,S) = \prod_{i=1}^{n} P(RHS_i|LHS_i)$$



Rules			P	Rules			P
S	→	VP	.05	S	→	VP	.05
VP	→	Verb NP	.20	VP	→	Verb NP NP	.10
NP	→						
Nominal	→						
Nominal	→						
Verb	→						
Det	→	the	.60	Det	→	the	.60
Noun	→	dinner	.10	Noun	→	dinner	.10
Noun	→	flights	.40	Noun	→	flights	.40

$$P(T_{left}) = .05 * .20 * .20 * .20 * .75 * .30 * .60 * .10 * .40 = \mathbf{2.2 \times 10^{-6}}$$

$$P(T_{right}) = .05 * .10 * .20 * .15 * .75 * .75 * .30 * .60 * .10 * .40 = \mathbf{6.1 \times 10^{-7}}$$

PCFG

$$\hat{T}(S) = \underset{T s.t. S=\text{yield}(T)}{\operatorname{argmax}} P(T|S)$$

$$\hat{T}(S) = \underset{T s.t. S=\text{yield}(T)}{\operatorname{argmax}} \frac{P(T,S)}{P(S)}$$

$$\hat{T}(S) = \underset{T s.t. S=\text{yield}(T)}{\operatorname{argmax}} P(T,S)$$

$$\hat{T}(S) = \underset{T s.t. S=\text{yield}(T)}{\operatorname{argmax}} P(T)$$

$$P(T,S) = P(T)P(S|T)$$

But since a parse tree includes all the words of the sentence, $P(S|T)$ is 1. Thus:

$$P(T,S) = P(T)P(S|T) = P(T)$$

Features of PCFG

- As the number of possible trees for a given input grows, a PCFG gives some idea of the plausibility of a particular parse
- *But* the probability estimates are based purely on structural factors, and do not factor in lexical co-occurrence. Thus, PCFG does not give a very good idea of the plausibility of the sentence.
- Real text tends to have grammatical mistakes. PCFG avoids this problem by ruling out nothing, but by giving implausible sentences a low probability
- In practice, a PCFG is a worse language model for English than an n-gram model
- All else being equal, the probability of a smaller tree is greater than a larger tree

Important Questions

Let W_{1m} be a sentence, G a grammar, t a parse tree

- What is the most likely parse of sentence?

$$argmax_t P(t|w_{1m}, G)$$

- What is the probability of a sentence?

$$P(w_{1m}|G)$$

- How to learn the rule probabilities in the grammar G?

Important Questions

Let W_{1m} be a sentence, G a grammar, t a parse tree

- What is the most likely parse of sentence?

$$argmax_t P(t|w_{1m}, G)$$

- What is the probability of a sentence?

$$P(w_{1m}|G)$$

- How to learn the rule probabilities in the grammar G?

Next

- PCFG for LM
- Dependency Parsing