Natural Language Processing

Lecture 14:
Context-Free Recognition

Context-Free Grammars

- Using grammars
  - Recognition
  - Parsing
- Parsing algorithms
  - Top down
  - Bottom up
- CNF
- CKY Algorithm

Context-Free Grammars

- Vocabulary of terminal symbols, Σ
- Set of nonterminal symbols (a.k.a. variables), N
- Special start symbol $S \in N$
- Production rules of the form $X \rightarrow \alpha$

where

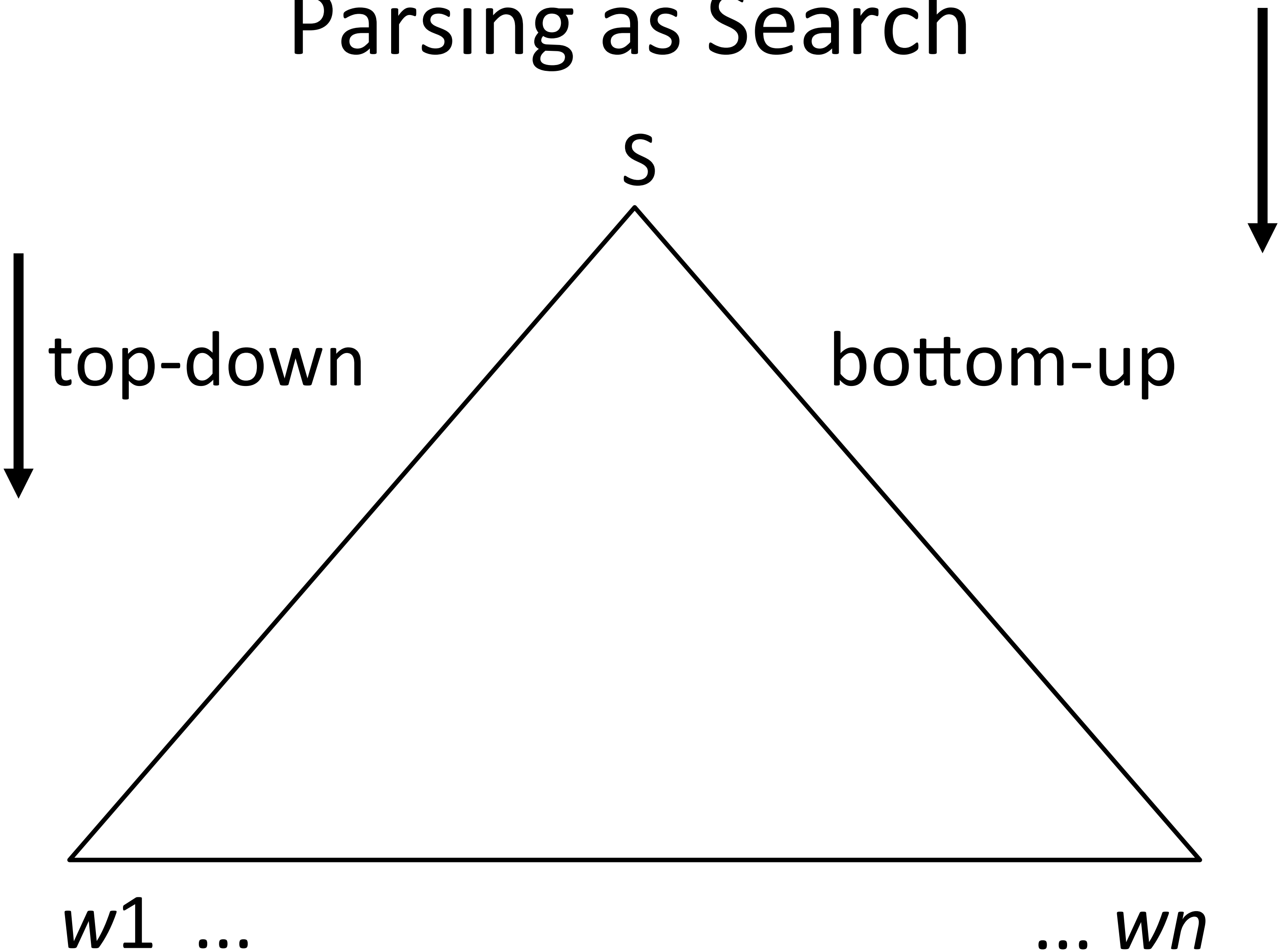
$X \in N$

$\alpha \in (N \cup \Sigma)^*$

Two Related Problems

- Input: sentence $\boldsymbol{w} = (w1, ..., wn)$ and CFG G
- Output (recognition): true iff $\boldsymbol{w} \in$ Language(G)
- Output (parsing): one or more derivations for $\boldsymbol{w}$, under G

Parsing as Search



Implementing Recognizers as Search

Agenda = { state0 }
while(Agenda not empty)
s = **pop** a state from Agenda
if s is a success-state **return** s // valid parse tree
else if s is not a failure-state:
generate new states from s
push new states onto Agenda
return nil // no parse!

Example Grammar and Lexicon

Grammar	Lexicon
$S \rightarrow NP\ VP$	$Det \rightarrow that \mid this \mid a$
$S \rightarrow Aux\ NP\ VP$	$Noun \rightarrow book \mid flight \mid meal \mid money$
$S \rightarrow VP$	$Verb \rightarrow book \mid include \mid prefer$
$NP \rightarrow Pronoun$	$Pronoun \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$Proper\text{-}Noun \rightarrow Houston \mid NWA$
$NP \rightarrow Det\ Nominal$	$Aux \rightarrow does$
$Nominal \rightarrow Noun$	$Preposition \rightarrow from \mid to \mid on \mid near \mid through$
$Nominal \rightarrow Nominal\ Noun$	
$Nominal \rightarrow Nominal\ PP$	
$VP \rightarrow Verb$	
$VP \rightarrow Verb\ NP$	
$VP \rightarrow Verb\ NP\ PP$	
$VP \rightarrow Verb\ PP$	
$VP \rightarrow VP\ PP$	
$PP \rightarrow Preposition\ NP$	

Figure 13.1 The $\mathscr{L}_1$ miniature English grammar and lexicon.

Recursive Descent
(A Top-Down Parser)

Start state: $(S, 0)$

Scan: From $(w_{j+1}\ \beta, j)$, you can get to $(\beta, j + 1)$.

Predict: If $Z \rightarrow \gamma$, then from $(Z\ \beta, j)$, you can get to $(\gamma\beta, j)$.

Final state: (ε, n)

Example Grammar and Lexicon

Grammar	Lexicon
$S \rightarrow NP\ VP$	$Det \rightarrow that \mid this \mid a$
$S \rightarrow Aux\ NP\ VP$	$Noun \rightarrow book \mid flight \mid meal \mid money$
$S \rightarrow VP$	$Verb \rightarrow book \mid include \mid prefer$
$NP \rightarrow Pronoun$	$Pronoun \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$Proper\text{-}Noun \rightarrow Houston \mid NWA$
$NP \rightarrow Det\ Nominal$	$Aux \rightarrow does$
$Nominal \rightarrow Noun$	$Preposition \rightarrow from \mid to \mid on \mid near \mid through$
$Nominal \rightarrow Nominal\ Noun$	
$Nominal \rightarrow Nominal\ PP$	
$VP \rightarrow Verb$	
$VP \rightarrow Verb\ NP$	
$VP \rightarrow Verb\ NP\ PP$	
$VP \rightarrow Verb\ PP$	
$VP \rightarrow VP\ PP$	
$PP \rightarrow Preposition\ NP$	

Figure 13.1 The $\mathscr{L}_1$ miniature English grammar and lexicon.

Shift-Reduce
(A Bottom-Up Parser)

- Start state: $(\varepsilon, 0)$
- **Shift**: From (α, j), you can get to $(\alpha\ w_{j+1}, j + 1)$.
- **Reduce**: If $Z \rightarrow \gamma$, then from $(\alpha\gamma, j)$ you can get to $(\alpha\ Z, j)$.
- Final state: (S, n)

Context-Free Grammars in Chomsky Normal Form

- Vocabulary of terminal symbols, Σ
- Set of nonterminal symbols (a.k.a. variables), N
- Special start symbol $S \in N$
- Production rules of the form $X \rightarrow \alpha$

where

$X \in N$

$\alpha \in N, N \cup \Sigma$

Context-Free Grammars in Chomsky Normal Form

- Vocabulary of terminal symbols, Σ
- Set of nonterminal symbols (a.k.a. variables), N
- Special start symbol $S \in N$
- Production rules of the form $X \rightarrow \alpha$

where

$X \in N$

$\alpha \in N, N \cup \Sigma$

Convert CFGs to CNF

- For each rule
  - $X \rightarrow A\ B\ C$
- Rewrite as
  - $X \rightarrow A\ X2$
  - $X2 \rightarrow B\ C$
- Introducing a new non-terminal

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

Figure 13.8 $\mathscr{L}_1$ Grammar and its conversion to CNF. Note that although they aren't shown here all the original lexical entries from $\mathscr{L}_1$ carry over unchanged as well.

CKY Algorithm: Chart

book					
	this				
		flight			
			through		
				Houston	

CKY Algorithm: Chart

	Noun				
book					
	this				
		flight			
			through		
				Houston	

CKY Algorithm: Chart

	Noun, Verb				
book					
	this				
		flight			
			through		
				Houston	

CKY Algorithm: Chart

	Noun, Verb				
book		Det			
	this		Noun		
		flight		Prep	
			through		PNoun
				Houston	

CKY Algorithm: Chart

	Noun, Verb				
book		Det			
	this		Noun		
		flight		Prep	
			through		Pnoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det			
	this		Noun		
		flight		Prep	
			through		PNoun
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP		
	this		Noun		
		flight		Prep	
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP		
	this		Noun		
		flight		Prep	
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP		
	this		Noun	-	
		flight		Prep	
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP	-	
	this		Noun	-	
		flight		Prep	
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP	-	
	this		Noun	-	
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP	-	
	this		Noun	-	-
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-			
book		Det	NP	-	NP
	this		Noun	-	-
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-	VP		
book		Det	NP	-	NP
	this		Noun	-	-
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-	VP,S		
book		Det	NP	-	NP
	this		Noun	-	-
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-	VP,S	-	
book		Det	NP	-	NP
	this		Noun	-	-
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm: Chart

	Noun, Verb	-	VP,S	-	S
book		Det	NP	-	NP
	this		Noun	-	-
		flight		Prep	PP
			through		PNoun, NP
				Houston	

CKY Algorithm

for $i = 1 \ldots n$

    $C[i\text{-}1, i] = \{ V \mid V \to wi \}$

for $\ell = 2 \ldots n$ // width

    for $i = 0 \ldots n - \ell$ // left boundary

        $k = i + \ell$ // right boundary

            for $j = i + 1 \ldots k - 1$ // midpoint

                $C[i, k] = C[i, k] \cup$

                    $\{ V \mid V \to YZ, Y \in C[i, j], Z \in$

$C[j, k] \}$

return true if $S \in C[0, n]$

CKY Equations

$$C[i-1, i, w_i] = \textsc{true}$$

$$C[i-1, i, V] = \begin{cases} \textsc{true} & \text{if } V \to w_i \\ \textsc{false} & \text{otherwise} \end{cases}$$

$$C[i, j, V] = \begin{cases} \textsc{true} & \text{if } \exists j, Y, Z \text{ such that} \\ & V \to Y\,Z \\ & \quad \text{and } C[i, k, Y] \\ & \quad \text{and } C[k, j, Z] \\ & \quad \text{and } i < k < j \\ \textsc{false} & \text{otherwise} \end{cases}$$

$$\text{goal} = C[0, n, S]$$

CKY Complexity

- CKY worst case is O(n^3 . G)
- Best in worst case
- (Others better in average case)

CFG Grammars

- Parsing and Recognition
- Bottom up and Top down
- CKY (for CNF)
- Chart Parsing (Earley's Algorithm)