

ECE59500NL Lecture 15: Top Down

Jeffrey Mark Siskind

School of Electrical and Computer Engineering

Spring 2021



© 2021 Jeffrey Mark Siskind. All rights reserved.

Outline

- ▶ Chomsky Normal Form
- ▶ CFG Parsing Algorithms
 - ▶ Top Down
 - ▶ Recursive Descent
 - ▶ Shift Reduce
- ▶ Complexity of Top Down Recognizer
- ▶ Questions to Think About

Chomsky Normal Form

$$A \rightarrow BC$$

$$A \rightarrow a$$

THEOREM: Any context-free grammar can be converted to a *weakly* equivalent grammar in Chomsky Normal Form

PROOF:

- Eliminate rules of the form $A \rightarrow \epsilon$.
- Eliminate rules of the form $A \rightarrow B$.
- Example: Translate $A \rightarrow BCDe$ to
 - ▶ $A \rightarrow BA_1$
 - ▶ $A_1 \rightarrow A_2A_3$
 - ▶ $A_2 \rightarrow c$
 - ▶ $A_3 \rightarrow DA_4$
 - ▶ $A_4 \rightarrow e$ □

Space of Parsing Algorithms

$$\begin{bmatrix} \text{Top Down} \\ \text{Recursive Descent} \\ \text{Shift Reduce} \end{bmatrix} \times \begin{bmatrix} \text{Partial Evaluation} \\ - \end{bmatrix} \times \begin{bmatrix} \text{Recognition} \\ \text{Parsing} \end{bmatrix} \times \begin{bmatrix} \text{Memoization} \\ - \end{bmatrix}$$

A Sample CFG

```
(define *rules*  
'((s ==> np vp)  
  (np ==> d nb)  
  (np ==> d n)  
  (nb ==> a nb)  
  (nb ==> a n)  
  (nb ==> n pp)  
  (pp ==> p np)  
  (vp ==> v np)  
  (vp ==> v pp)))
```

```
(define *lexicon*  
'(("square" n)  
  ("piece" n)  
  ("pawn" n)  
  ("bishop" n)  
  ("knight" n)  
  ("rook" n)  
  ("queen" n)  
  ("king" n)  
  ("rank" n)  
  ("file" n)  
  ("edge" n)  
  ("corner" n)  
  ("black" a)  
  ("white" a)  
  ("every" d)  
  ("some" d)  
  ("the" d)  
  ("is" v)  
  ("of" p)  
  ("on" p)))
```