

CKY algorithm / PCFGs

CS 685, Fall 2021

Introduction to Natural Language Processing

<http://people.cs.umass.edu/~miyyer/cs685/>

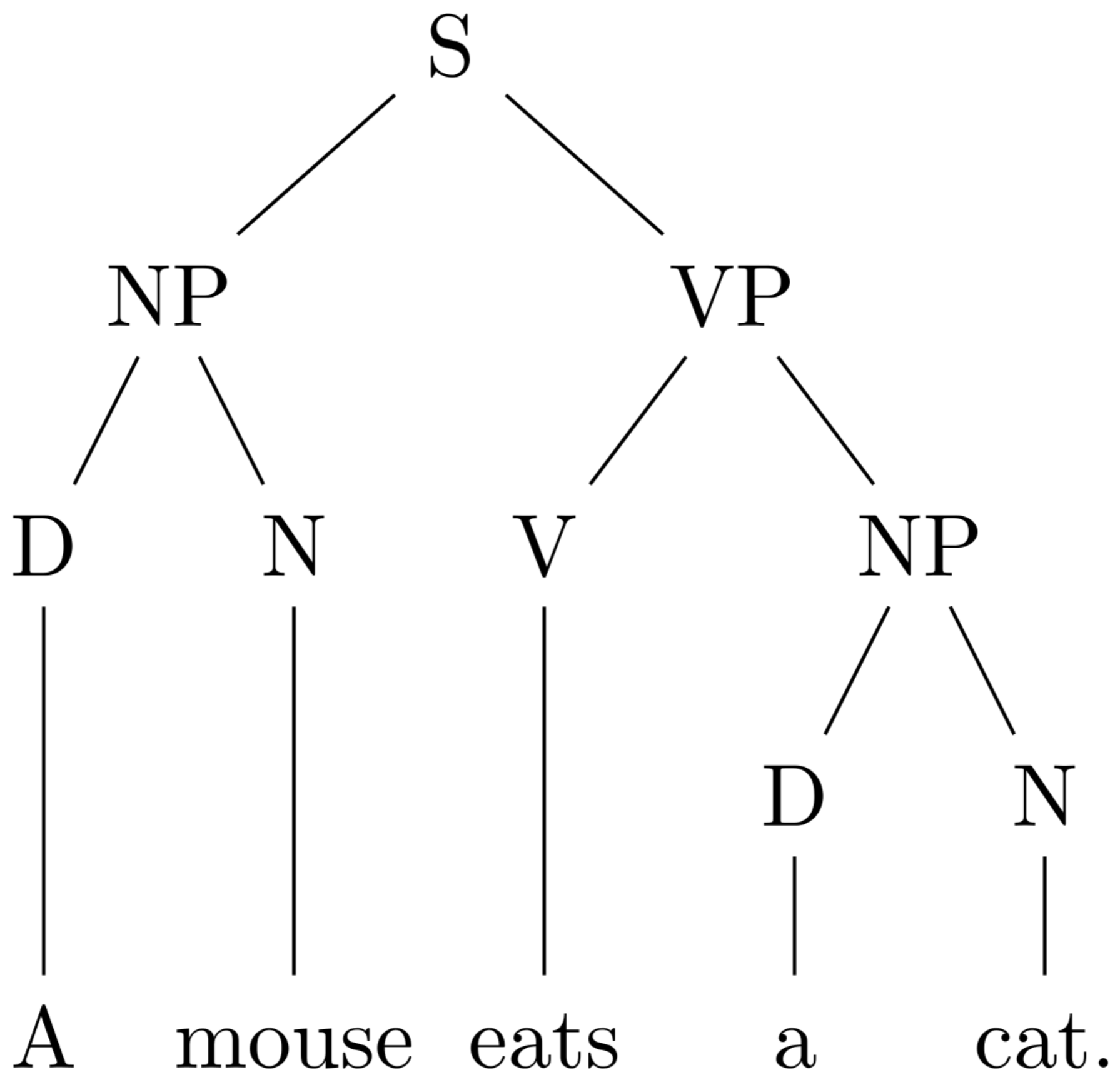
Mohit Iyyer

College of Information and Computer Sciences

University of Massachusetts Amherst

some slides from Brendan O'Connor

today we'll be doing *parsing*:
given a grammar, how do we
use it to parse a sentence?



why parsing?

- **historically:** good way to obtain features for downstream tasks
- **today:** can sometimes (not always) use syntax to improve neural models
- always useful for chunking text into phrases
- parsing makes for good *probe* tasks on top of neural models
- useful for psycholinguistics experiments

Formal Definition of Context-Free Grammar

- A context-free grammar G is defined by four parameters: N, Σ, R, S

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 $A \rightarrow \beta$,
where A is a non-terminal,

β is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$

S a designated **start symbol** and a member of N

let's start with a simple CFG

- $S \rightarrow NP VP$
- $NN \rightarrow \text{"dog"}$
- $NP \rightarrow DT JJ NN$

first, let's convert this to Chomsky Normal Form (CNF)

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 $A \rightarrow \beta$,
where A is a non-terminal,

β is a string of symbols from the infinite set of strings $(\Sigma \cup N)^*$

S a designated **start symbol** and a member of N

β is either a single terminal from Σ
or a pair of non-terminals from N

converting the simple CFG

- $S \rightarrow NP VP$
- $NN \rightarrow \text{"dog"}$
- $NP \rightarrow DT JJ NN$
 - $NP \rightarrow X NN$
 - $X \rightarrow DT JJ$

we can convert any CFG to a CNF.
this is a necessary preprocessing
step for the basic CKY alg.,
produces binary trees!

Parsing!

- Given a sentence and a CNF, we want to **search** through the space of all possible parses for that sentence to find:
 - any valid parse for that sentence
 - all valid parses
 - the most probable parse
- Two approaches **Pros and cons of each?**
 - **bottom-up**: start from the words and attempt to construct the tree
 - **top-down**: start from START symbol and keep expanding until you can construct the sentence

Ambiguity in parsing

Syntactic ambiguity is endemic to natural language:¹

- ▶ Attachment ambiguity: *we eat sushi with chopsticks, I shot an elephant in my pajamas.*
- ▶ Modifier scope: *southern food store*
- ▶ Particle versus preposition: *The puppy tore up the staircase.*
- ▶ Complement structure: *The tourists objected to the guide that they couldn't hear.*
- ▶ Coordination scope: *“I see,” said the blind man, as he picked up the hammer and saw.*
- ▶ Multiple gap constructions: *The chicken is ready to eat*

¹Examples borrowed from Dan Klein

today: CKY algorithm

- Cocke-Kasami-Younger (independently discovered, also known as CYK)
- a *bottom-up* parser for CFGs (and PCFGs)

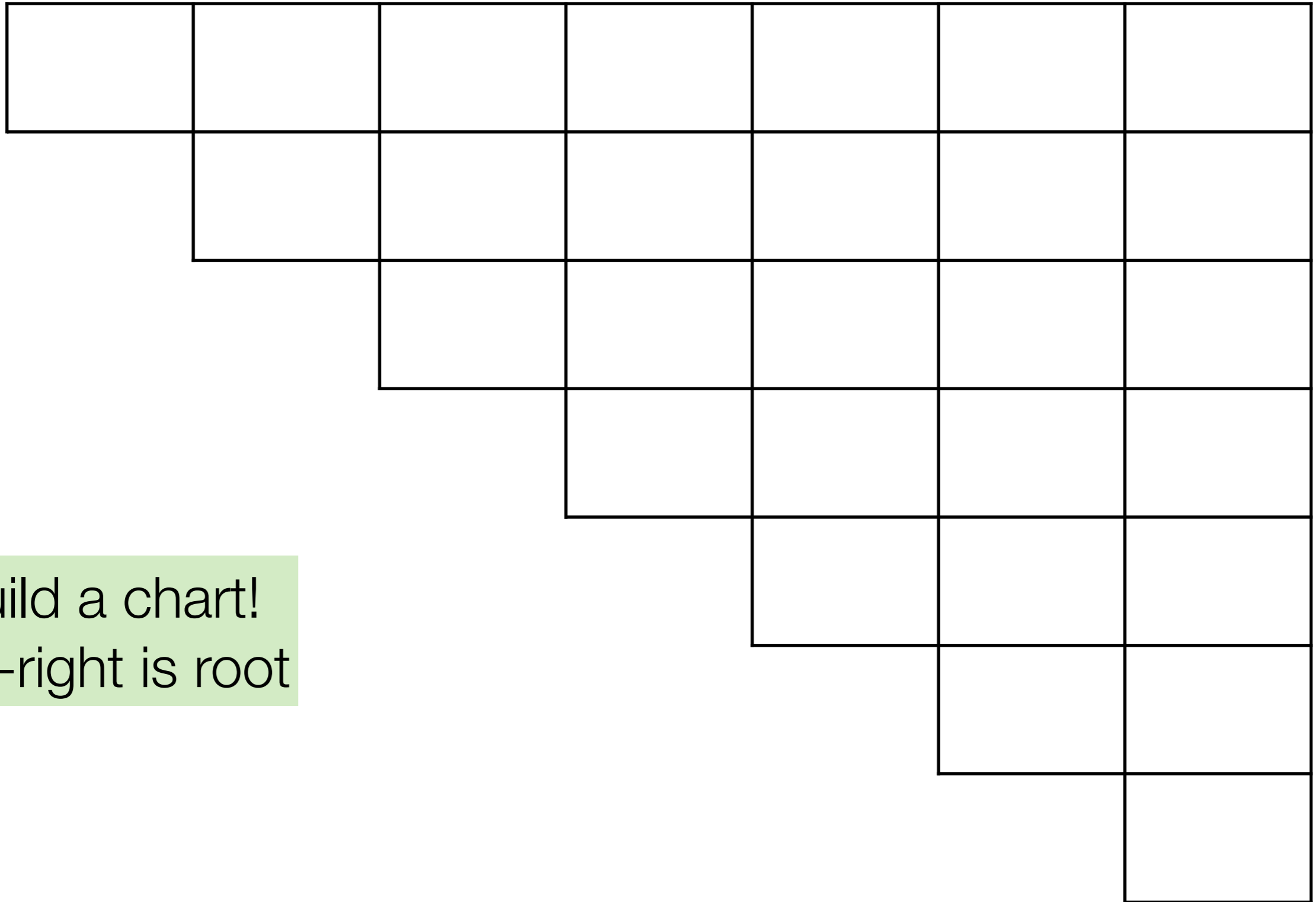
“I shot an elephant in my pajamas. How he got into my pajamas, I'll never know.”
— Groucho Marx

let's say I have this CNF

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

- DET ▶ “an”
- VBD ▶ “shot”
- NP ▶ “pajamas”
- NP ▶ “elephant”
- NP ▶ “I”
- PRP ▶ “I”
- IN ▶ “in”
- PRP\$ ▶ “my”

I shot an elephant in my pajamas



build a chart!
top-right is root

I shot an elephant in my pajamas

NP / PRP						
	VBD					
		DET				
			NP			
				IN		
					PRP\$	
						NP

fill in first level (words)
with possible derivations

I shot an elephant in my pajamas

NP / PRP						
	VBD					
		DET				
			NP			
				IN		
					PRP\$	
						NP

onto the second level!

I shot an elephant in my pajamas

NP / PRP						
	VBD					
		DET				
			NP			
				IN		
					PRP\$	
						NP

onto the second level!

this cell spans
the phrase "I shot"

I shot an elephant in my pajamas

NP / PRP						
	VBD					
		DET				
			NP			
				IN		
					PRP\$	
						NP

onto the second level!

what does this cell span?

I shot an elephant in my pajamas

NP / PRP						
	VBD					
		DET				
					PRP\$	
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

onto the second level!

do any rules produce NP VBD or PRP VBD?

I shot an elephant in my pajamas

NP / PRP	∅					
	VBD					
		DET				
					PRP\$	
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

onto the second level!

do any rules produce
VBD DET?

I shot an elephant in my pajamas

NP / PRP	∅					
	VBD	∅				
		DET				
			NP			

onto the second level!

do any rules produce
DET NP?

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅					
	VBD	∅				
		DET	NP			
					PRP\$	
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

onto the second level!

do any rules produce
DET NP? Yes!

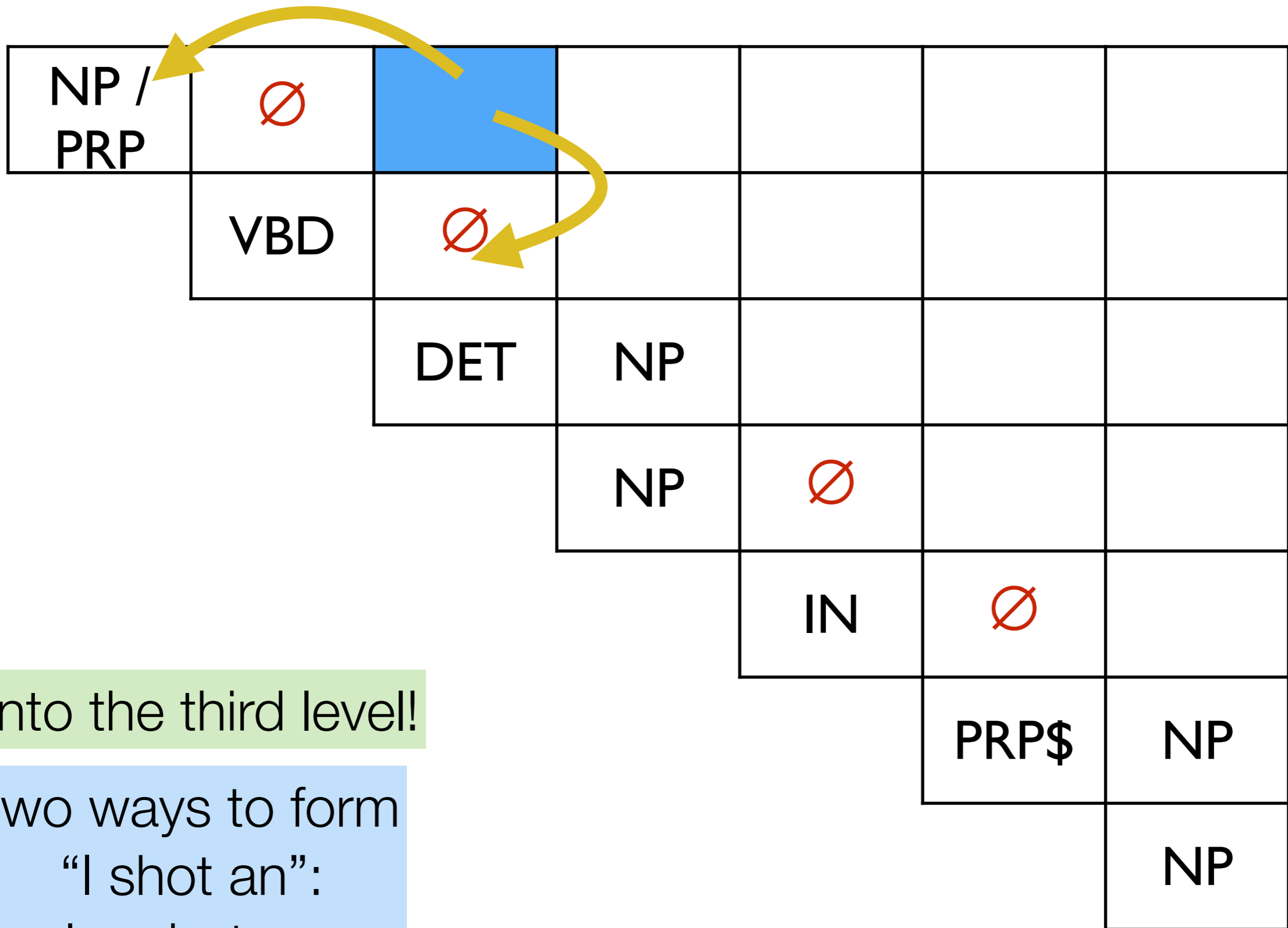
NP ▶ DET NP

I shot an elephant in my pajamas

NP / PRP	∅					
	VBD	∅				
		DET	NP			
			NP	∅		
				IN	∅	
					PRP\$	NP
						NP

onto the third level!

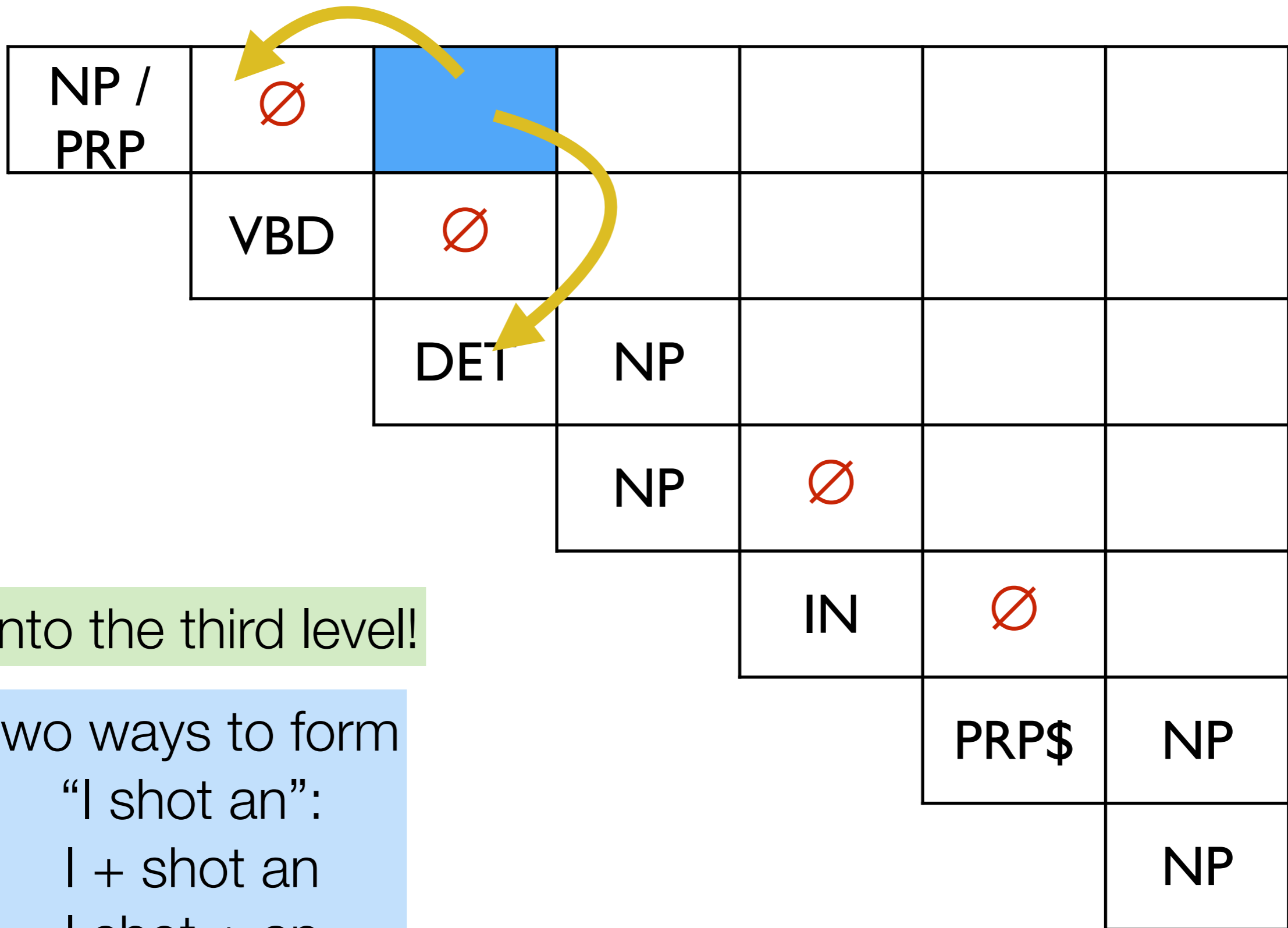
I shot an elephant in my pajamas



onto the third level!

two ways to form
“I shot an”:
I + shot an

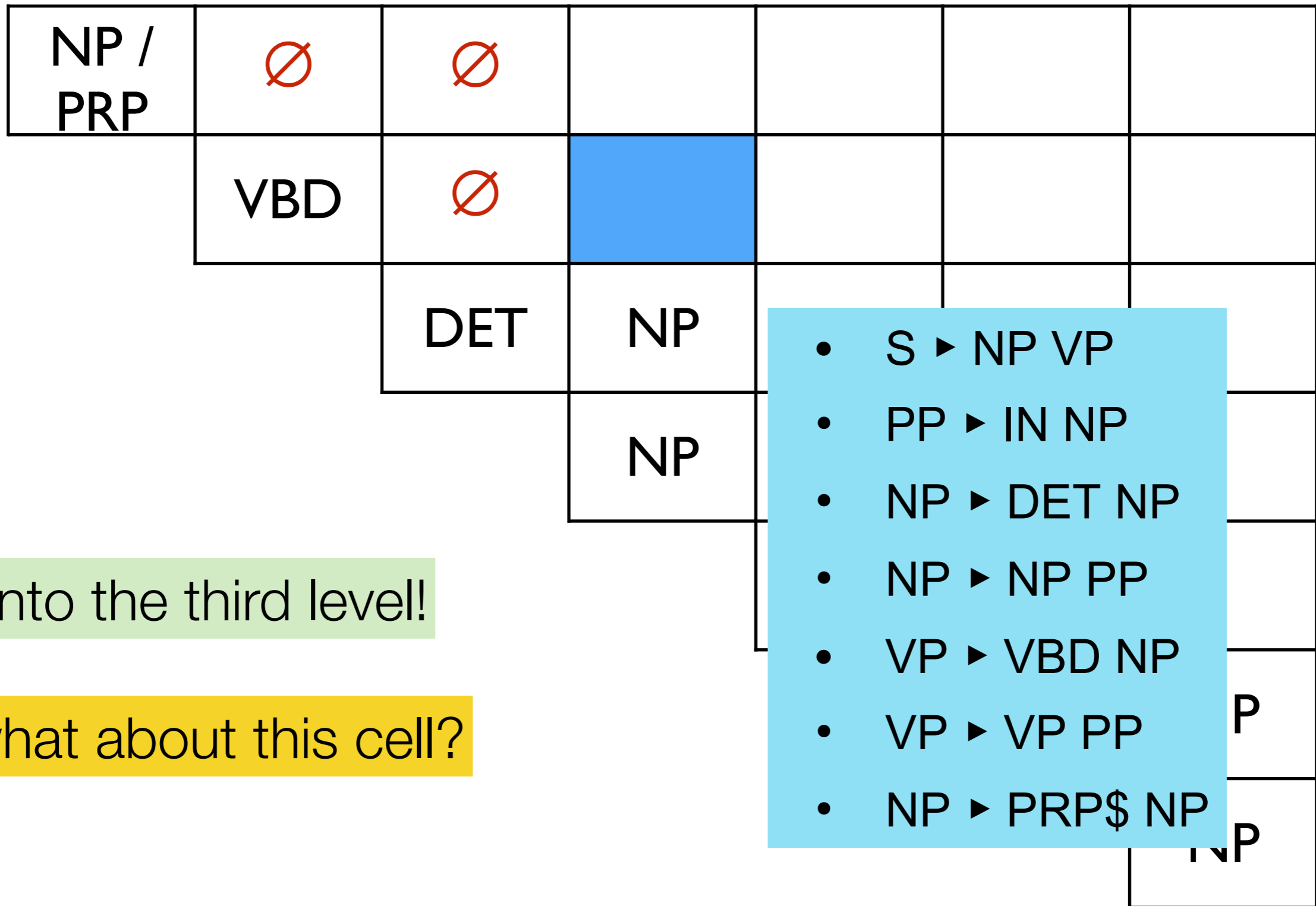
I shot an elephant in my pajamas



onto the third level!

two ways to form
 "I shot an":
 I + shot an
 I shot + an

I shot an elephant in my pajamas



- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

onto the third level!

what about this cell?

I shot an elephant in my pajamas

NP / PRP	∅	∅				
	VBD	∅	VP			
		DET	NP			
			NP	∅		
				IN	∅	
					PRP\$	NP
						NP

onto the third level!

I shot an elephant in my pajamas

NP / PRP	∅	∅				
	VBD	∅	VP			
		DET	NP	∅		
			NP	∅	∅	
				IN	∅	PP
					PRP\$	NP
						NP

I shot an elephant in my pajamas

NP / PRP	∅	∅				
	VBD	∅	VP			
		DET	NP	∅		
			NP	∅	∅	
				IN	∅	PP
					PRP\$	NP
						NP

onto the fourth level!

what are our options here?

I shot an elephant in my pajamas

NP / PRP	∅	∅				
	VBD	∅	VP			
		DET	NP	∅		
			NP			
						PP
						NP
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

onto the fourth level!

what are our options here?

NP VP
PRP VP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S			
	VBD	∅	VP	∅		
		DET	NP	∅	∅	
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

onto the fourth level!

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅		
	VBD	∅	VP	∅	∅	
		DET	NP	∅	∅	
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅		
	VBD	∅	VP	∅	∅	
		DET	NP	∅	∅	
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ **DET NP**
- NP ▶ **NP PP**
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅		
	VBD	∅	VP	∅	∅	
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅	∅	
	VBD	∅	VP	∅	∅	
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅	∅	
	VBD	∅	VP	∅	∅	
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅	∅	
	VBD	∅	VP	∅	∅	VP ₁ / VP ₂ / VP ₃
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅	∅	
	VBD	∅	VP	∅	∅	VP ₁ / VP ₂ / VP ₃
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

finally, the root!

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅	∅	
	VBD	∅	VP	∅	∅	VP₁ / VP₂ / VP₃
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

finally, the root!

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

S > NP VP₁
 S > NP VP₂
 S > NP VP₃

I shot an elephant in my pajamas

NP / PRP	∅	∅	S	∅	∅	S ₁ / S ₂ / S ₃
	VBD	∅	VP	∅	∅	VP ₁ / VP ₂ / VP ₃
		DET	NP	∅	∅	NP ₁ / NP ₂
			NP	∅	∅	NP
				IN	∅	PP
					PRP\$	NP
						NP

finally, the root!

- S ▶ NP VP
- PP ▶ IN NP
- NP ▶ DET NP
- NP ▶ NP PP
- VP ▶ VBD NP
- VP ▶ VP PP
- NP ▶ PRP\$ NP

S > NP VP₁

S > NP VP₂

S > NP VP₃

three valid parses!

how do we recover the full derivation
of the valid parses $S_1 / S_2 / S_3$?

CKY runtime?

```
function CKY-PARSE(words, grammar) returns table  
  
  for  $j \leftarrow$  from 1 to LENGTH(words) do  
    for all { $A \mid A \rightarrow \text{words}[j] \in \text{grammar}$ }  
       $\text{table}[j-1, j] \leftarrow \text{table}[j-1, j] \cup A$   
    for  $i \leftarrow$  from  $j-2$  downto 0 do  
      for  $k \leftarrow i+1$  to  $j-1$  do  
        for all { $A \mid A \rightarrow BC \in \text{grammar}$  and  $B \in \text{table}[i, k]$  and  $C \in \text{table}[k, j]$ }  
           $\text{table}[i, j] \leftarrow \text{table}[i, j] \cup A$ 
```

Figure 12.5 The CKY algorithm.

three nested loops, each $O(n)$ where n is # words

$O(n^3)$

how to find best parse?

- use PCFG (*probabilistic* CFG): same as CFG except each rule $A \rightarrow \beta$ in the grammar is associated with a probability $p(\beta \mid A)$
- can compute probability of a parse T by just multiplying rule probabilities of the rules r that make up T

$$p(T) = \prod_{r \in T} p(\beta_r \mid A_r)$$

- S ▶ NP VP, 0.4
- PP ▶ IN NP, 0.1
- NP ▶ DET NP, 0.3
- NP ▶ NP PP, 0.1
- VP ▶ VBD NP, 0.2
- VP ▶ VP PP, 0.3
- NP ▶ PRP\$ NP, 0.5

- DET ▶ “an”, 0.9
- VBD ▶ “shot”, 0.3
- NP ▶ “pajamas”, 0.8
- NP ▶ “elephant”, 0.9
- NP ▶ “I”, 0.2
- PRP ▶ “I”, 0.6
- IN ▶ “in”, 0.9
- PRP\$ ▶ “my”, 0.8

I shot an elephant in my pajamas

NP (0.2) / PRP (0.6)						
	VBD (0.3)					
		DET (0.9)				
			NP (0.8)			
				IN (0.9)		
					PRP\$ (0.8)	
						NP (0.8)

fill in first level (words)
with possible derivations
and probabilities

I shot an elephant in my pajamas

NP (0.2) / PRP (0.6)	∅					
	VBD (0.3)	∅				
		DET (0.9)	NP			
			NP (0.8)	∅		
				IN (0.9)	∅	
					PRP\$ (0.8)	NP
						NP (0.8)

how do we compute this cell's probability?

I shot an elephant in my pajamas

NP (0.2) / PRP (0.6)	∅					
	VBD (0.3)	∅				
		DET (0.9)	NP (0.22)			
			NP (0.8)	∅		
				IN (0.9)	∅	
					PRP\$ (0.8)	NP (0.32)
						NP (0.8)

how do we compute this cell's probability?

$$\begin{aligned}
 & p(\text{DET NP} \mid \text{NP}) * P(\text{cell}_{\text{DET}}) * \\
 & \quad P(\text{cell}_{\text{NP}}) \\
 & = 0.3 * 0.9 * 0.8 \\
 & = 0.22
 \end{aligned}$$

I shot an elephant in my pajamas

NP (-1.6) / PRP (-0.51)	∅	∅	S (-6.8)	∅	∅	
	VBD (-1.2)	∅	VP (-4.3)	∅	∅	
		DET (-0.11)	NP (-1.5)	∅	∅	NP ₁ / NP ₂
			NP (-0.22)	∅	∅	NP (-6.0)
				IN (-0.11)	∅	PP (-3.5)
					PRP\$ (-0.22)	NP (-1.1)
						NP (-0.22)

let's switch to log space and fill out the table some more

I shot an elephant in my pajamas

NP (-1.6) / PRP (-0.51)	∅	∅	S (-6.8)	∅	∅	
	VBD (-1.2)	∅	VP (-4.3)	∅	∅	
		DET (-0.11)	NP (-1.5)	∅	∅	NP ₁ / NP ₂
			NP (-0.22)	∅	∅	NP (-6.0)
				IN (-0.11)	∅	PP (-3.5)
					PRP\$ (-0.22)	NP (-1.1)
						NP (-0.22)

$$p(\text{NP}_1) = ?$$

$$p(\text{NP}_2) = ?$$

I shot an elephant in my pajamas

NP (-1.6) / PRP (-0.51)	∅	∅	S (-6.8)	∅	∅	
	VBD (-1.2)	∅	VP (-4.3)	∅	∅	
		DET (-0.11)	NP (-1.5)	∅	∅	NP ₁ (-7.31) / NP ₂ (-7.30)
			NP (-0.22)	∅	∅	NP (-6.0)
				IN (-0.11)	∅	PP (-3.5)
					PRP\$ (-0.22)	NP (-1.1)
						NP (-0.22)

do we have to store both NPs?

I shot an elephant in my pajamas

NP (-1.6) / PRP (-0.51)	∅	∅	S (-6.8)	∅	∅	
	VBD (-1.2)	∅	VP (-4.3)	∅	∅	VP ₁ / VP ₂
		DET (-0.11)	NP (-1.5)	∅	∅	NP (-7.3)
			NP (-0.22)	∅	∅	NP (-6.0)
				IN (-0.11)	∅	PP (-3.5)
					PRP\$ (-0.22)	NP (-1.1)
						NP (-0.22)

$p(VP_1) = ?$
 $p(VP_2) = ?$

I shot an elephant in my pajamas

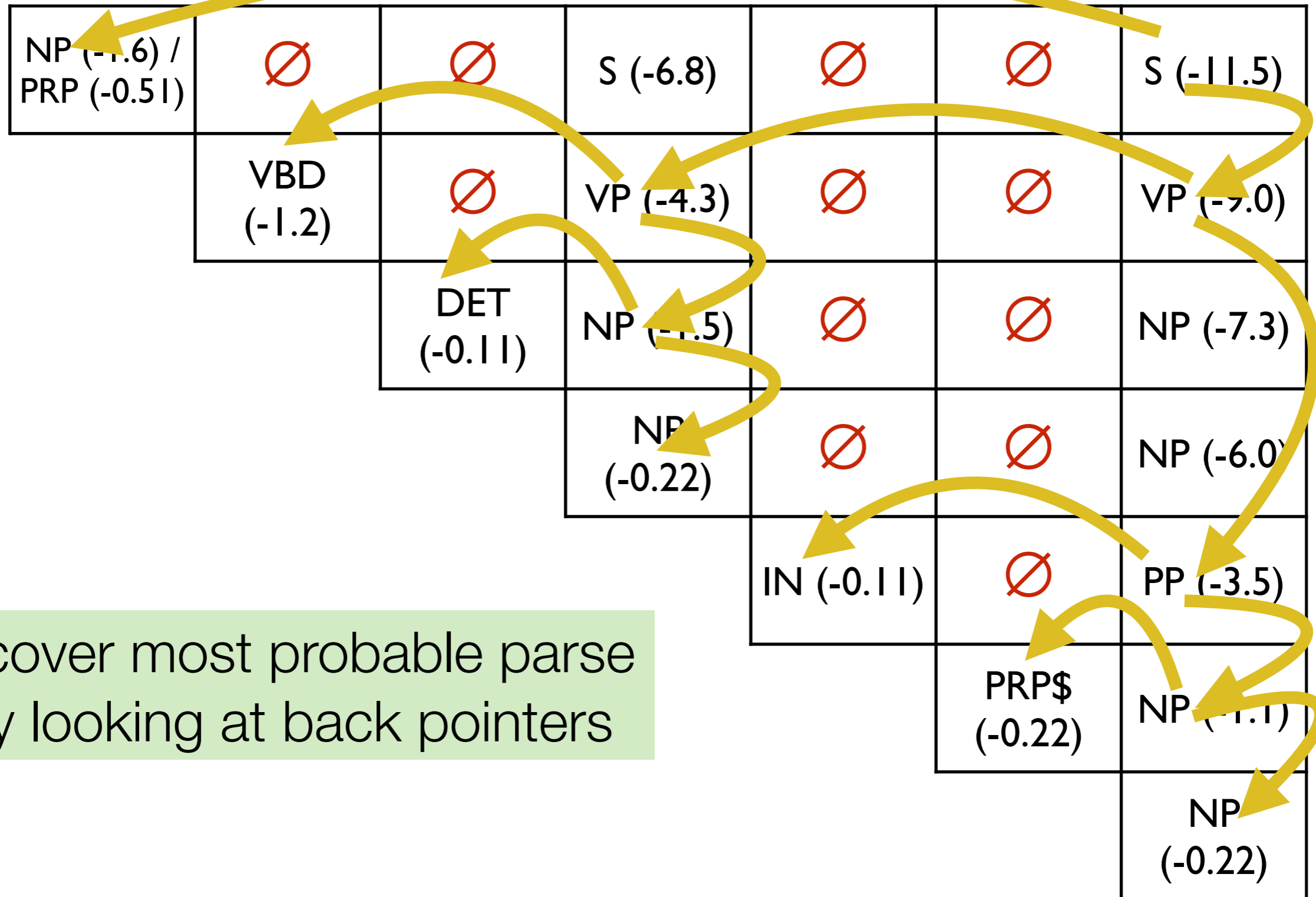
NP (-1.6) / PRP (-0.51)	∅	∅	S (-6.8)	∅	∅	
	VBD (-1.2)	∅	VP (-4.3)	∅	∅	VP ₁ (-10.1) /VP ₂ (-9.0)
		DET (-0.11)	NP (-1.5)	∅	∅	NP (-7.3)
			NP (-0.22)	∅	∅	NP (-6.0)
				IN (-0.11)	∅	PP (-3.5)
					PRP\$ (-0.22)	NP (-1.1)
						NP (-0.22)

do we need to store both VPs?

I shot an elephant in my pajamas

NP (-1.6) / PRP (-0.51)	∅	∅	S (-6.8)	∅	∅	S (-11.5)
	VBD (-1.2)	∅	VP (-4.3)	∅	∅	VP (-9.0)
		DET (-0.11)	NP (-1.5)	∅	∅	NP (-7.3)
			NP (-0.22)	∅	∅	NP (-6.0)
				IN (-0.11)	∅	PP (-3.5)
					PRP\$ (-0.22)	NP (-1.1)
						NP (-0.22)

I shot an elephant in my pajamas



recover most probable parse
by looking at back pointers

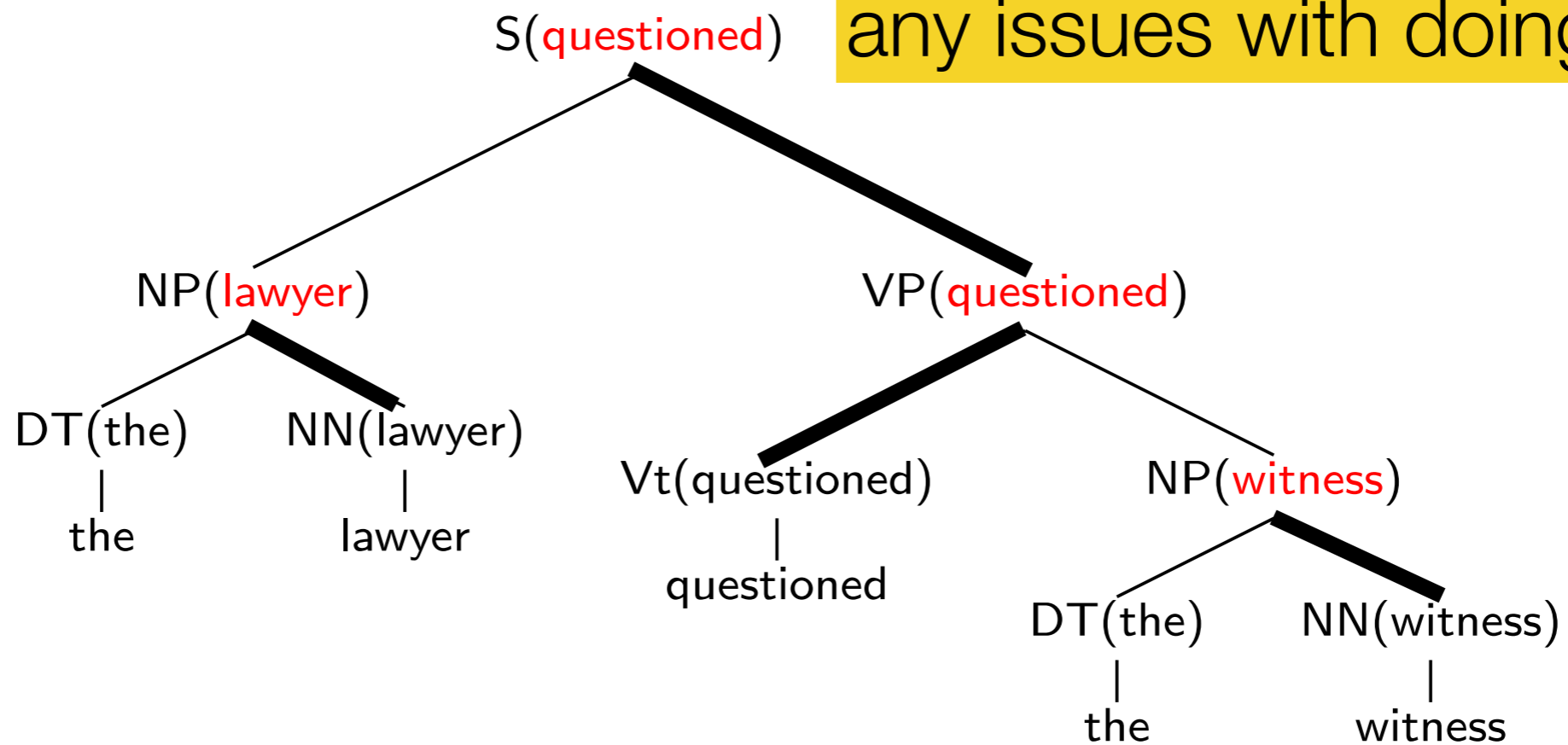
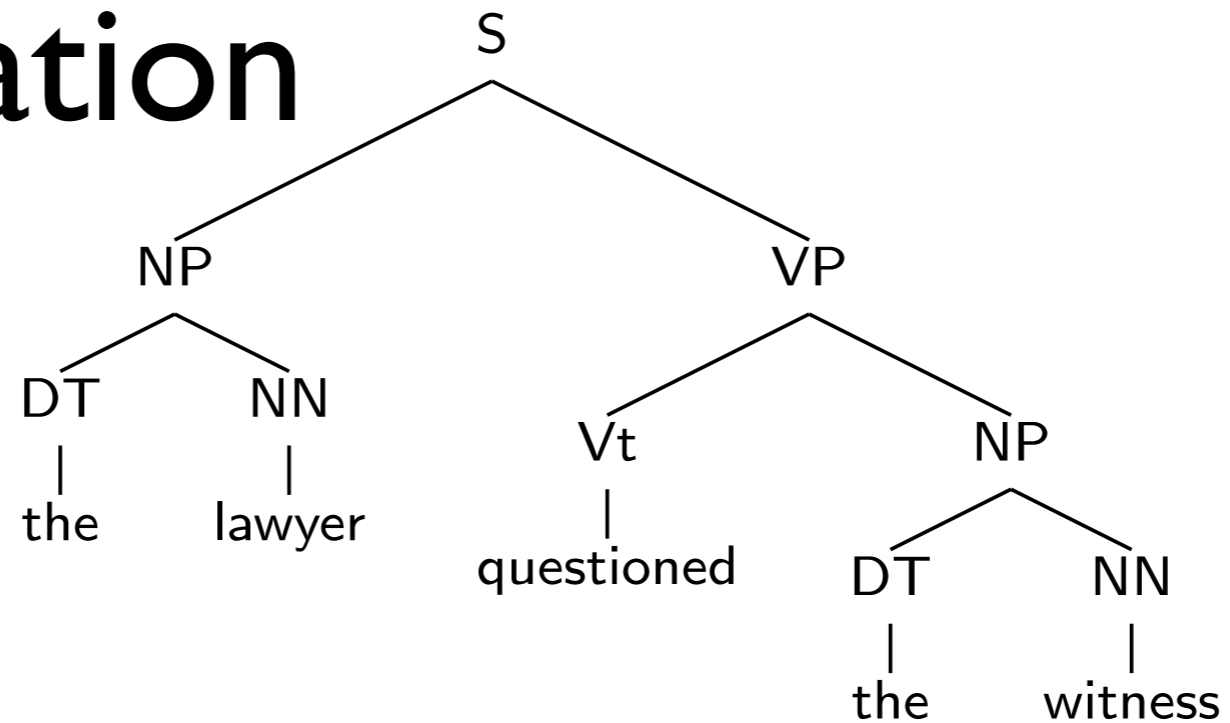
issues w/ PCFGs

- independence assumption: each rule's probability is independent of the rest of the tree!!!
- doesn't take into account location in the tree or what words are involved (for $A \rightarrow BC$)
 - John saw the man with the hat
 - John saw the moon with the telescope

add more info to PCFG!

- **How to make good attachment decisions?**
 - Enrich PCFG with
 - parent information: what's above me?
 - lexical information via head rules
 - VP[fight]: a VP headed by “fight”
 - (or better, word/phrase embedding-based generalizations: e.g. recurrent neural network grammars (RNNGs))

Lexicalization



where do we get the PCFG probabilities?

- given a treebank, we can just compute the MLE estimate by counting and normalizing

$$P(\alpha \rightarrow \beta | \alpha) = \frac{\text{Count}(\alpha \rightarrow \beta)}{\sum_{\gamma} \text{Count}(\alpha \rightarrow \gamma)} = \frac{\text{Count}(\alpha \rightarrow \beta)}{\text{Count}(\alpha)}$$

- without a treebank, we can use the *inside-outside algorithm* to estimate probabilities by
 1. randomly initializing probabilities
 2. computing parses
 3. computing expected counts for rules
 4. re-estimate probabilities
 5. repeat!