Language and Statistics II

Lecture 10: Parsing
(Treebanks, Algorithms)

Noah Smith

PCFGs and HMMs

PCFG:

- Alphabet Σ
- Nonterminal set N
- Start nonterminal S
- Rules $X \xrightarrow{p} \alpha$

HMM (special case):

- Alphabet Σ
- State set N
- Start state S_0
- Rules
  - $X \xrightarrow{\eta(s \mid X)} s\ X'$
  - $X' \xrightarrow{\gamma(Y \mid X)} Y$
  - $X' \xrightarrow{\gamma(stop \mid X)} \varepsilon$

PCFGs and Log-Linear Models

Log-linear model:

- Set of inputs $\mathbb{X}$
- Set of outputs $\mathbb{Y}$
- Set of feature functions $f_i : (\mathbb{X}, \mathbb{Y}) \rightarrow \mathbb{R}_{\geq 0}$
- Set of weights θ_i corresponding to f_i

PCFG:

- Σ^*
- Derivable productions given the rules
- Counts of rules
- Logarithms of rule probabilities

Major Research Questions

✓ What’s the right **representation**?
✓ What’s the right **model**?

(We’ve talked about one representation and one model.)

- How to learn to parse **empirically**?
- How to make parsers **fast**?
- How to incorporate structure **downstream**?

Learning from Data

1. Where do the **rules** come from?
2. Where do the rule **probabilities** come from?

First answer: Look at a huge collection of trees (a treebank).

$X \rightarrow \alpha$ is in the grammar iff it's in the treebank.

$p(\alpha \mid X)$ is proportional to the count of $X \rightarrow \alpha$.

Penn Treebank (Marcus et al., 1993)

- A million words (40K sentences) of *Wall Street Journal* text (late 1980s).
- Parsed by experts; consensus parse for each sentence was published.
- The structure is basically what you'd expect from a PCFG.
  - Tends to be "flat" where there's controversy.
  - Some "traces" for extraposed elements.

Example Tree

```
( (S
    (NP-SBJ
      (NP (NNP Pierre) (NNP Vinken) )
      (, ,)
      (ADJP
        (NP (CD 61) (NNS years) )
        (JJ old) )
      (, ,) )
    (VP (MD will)
      (VP (VB join)
        (NP (DT the) (NN board) )
        (PP-CLR (IN as)
          (NP (DT a) (JJ nonexecutive) (NN director) ))
        (NP-TMP (NNP Nov.) (CD 29) )))
    (. .) ))
```

```
( (S
    (NP-SBJ-1
      (NP (NNP Rudolph) (NNP Agnew) )
      (, ,)
      (UCP
        (ADJP
          (NP (CD 55) (NNS years) )
          (JJ old) )
        (CC and)
        (NP
          (NP (JJ former) (NN chairman) )
          (PP (IN of)
            (NP (NNP Consolidated) (NNP Gold) (NNP Fields) (NNP PLC) ))))
      (, ,) )
    (VP (VBD was)
      (VP (VBN named)
        (S
          (NP-SBJ (-NONE- *-1) )
          (NP-PRD
            (NP (DT a) (JJ nonexecutive) (NN director) )
            (PP (IN of)
              (NP (DT this) (JJ British) (JJ industrial) (NN conglomerate)
  ))
))))
    (. .) ))
```

Evaluating Parsers

- Take a sentence from the test set.
- Use your parser to propose a **hypothesis** parse.
- Treebank gives you the **correct** parse.
- How to compare?
  - {unlabeled, labeled} × {precision, recall}
  - crossing brackets statistics
  - `evalb` (http://nlp.cs.nyu.edu/evalb)
- Significance testing …

The Dark Side

- This is **the** way to train and test an English parser.
- There are some inconsistencies.
- Other treebank builders haven't always been as diligent; often tag labels, nonterminal labels, conventions are assumed to **port** to other languages.
- Better way of handling disagreement: publish different annotators' trees (not consensus)?

Training Parsers In Practice

- Transformations on trees
  - Some of these are generally taken to be crucial
  - Some are widely debated
  - Lately, people have started **learning** these transformations
- Smoothing (crucial)
- We will come back to this as we explore some current state-of-the art parsers.
  - Collins (1999; 2003)
  - Charniak (2000)
  - Klein and Manning (2003)
  - McDonald, Pereira, Ribarov, and Hajic (2005)

Decoding Algorithms

- Suppose I have a PCFG and a sentence.
- What might I want to do?
  - Find the most likely tree (if it exists).
  - Find the *k* most likely trees.
  - Gather statistics on the **distribution** over trees.

- Should remind you of FS models!

Probabilistic CKY

Input: PCFG $G = (\Sigma, \mathbf{N}, S, \mathbf{R})$ in CNF and sequence $\mathbf{w} \in \boldsymbol{\Sigma}^*$

Output: most likely tree for **w**, if it exists, and its probability.

$$C(X,i,i) = \left\langle p(X \rightarrow w_i), \text{null} \right\rangle$$

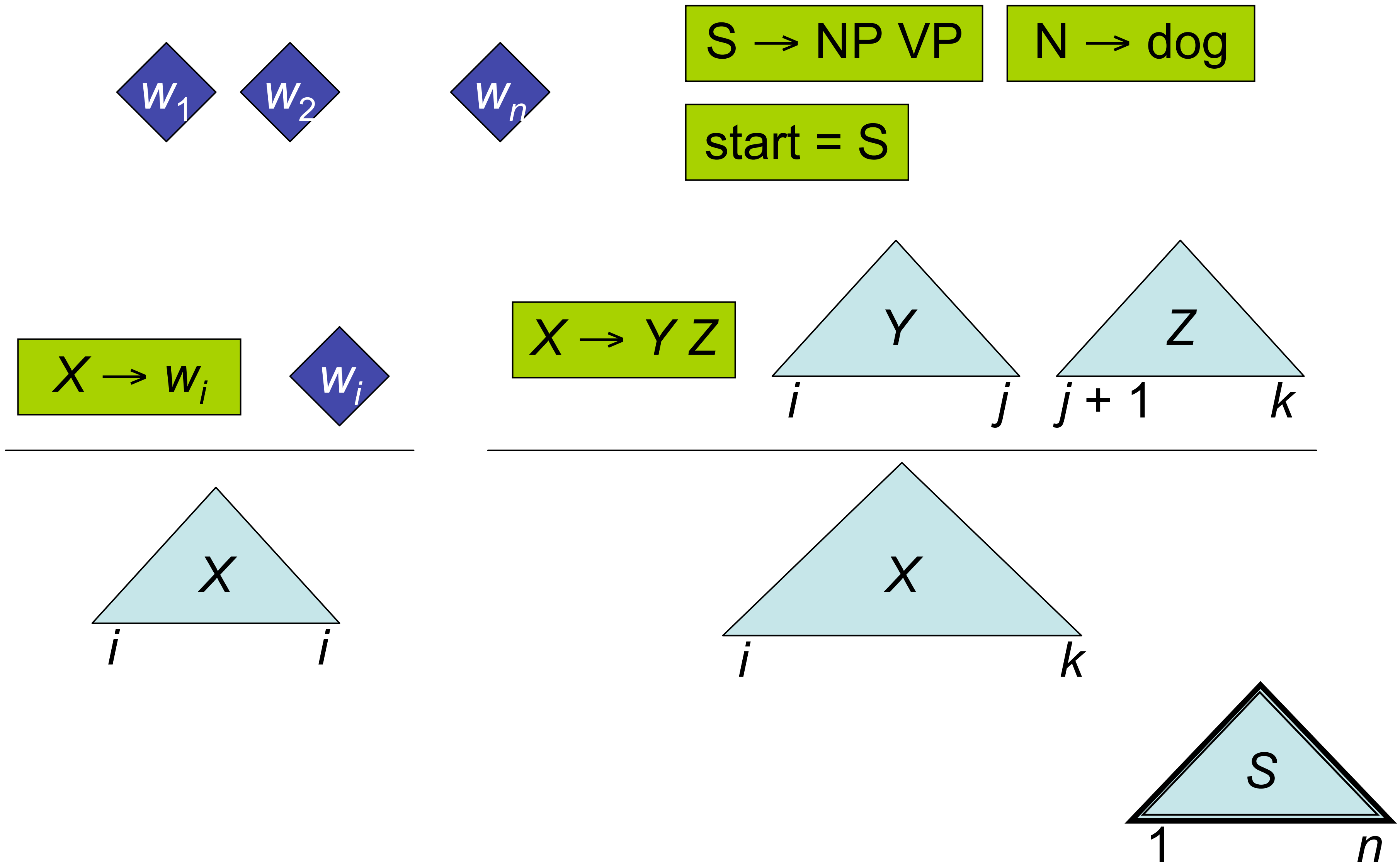
$$C(X,i,k) = \left\langle \begin{array}{c} \max\limits_{Y,Z \in \mathbf{N}, j \in [i+1,k-2]} C(Y,i,j) \cdot C(Z,j+1,k) \cdot p(X \rightarrow Y,Z), \\ \& \operatorname*{argmax}\limits_{Y,Z \in \mathbf{N}, j \in [i+1,k-2]} C(Y,i,j) \cdot C(Z,j+1,k) \cdot p(X \rightarrow Y,Z) \end{array} \right\rangle$$

$$\text{goal} = C(S,1,|\mathbf{w}|)$$

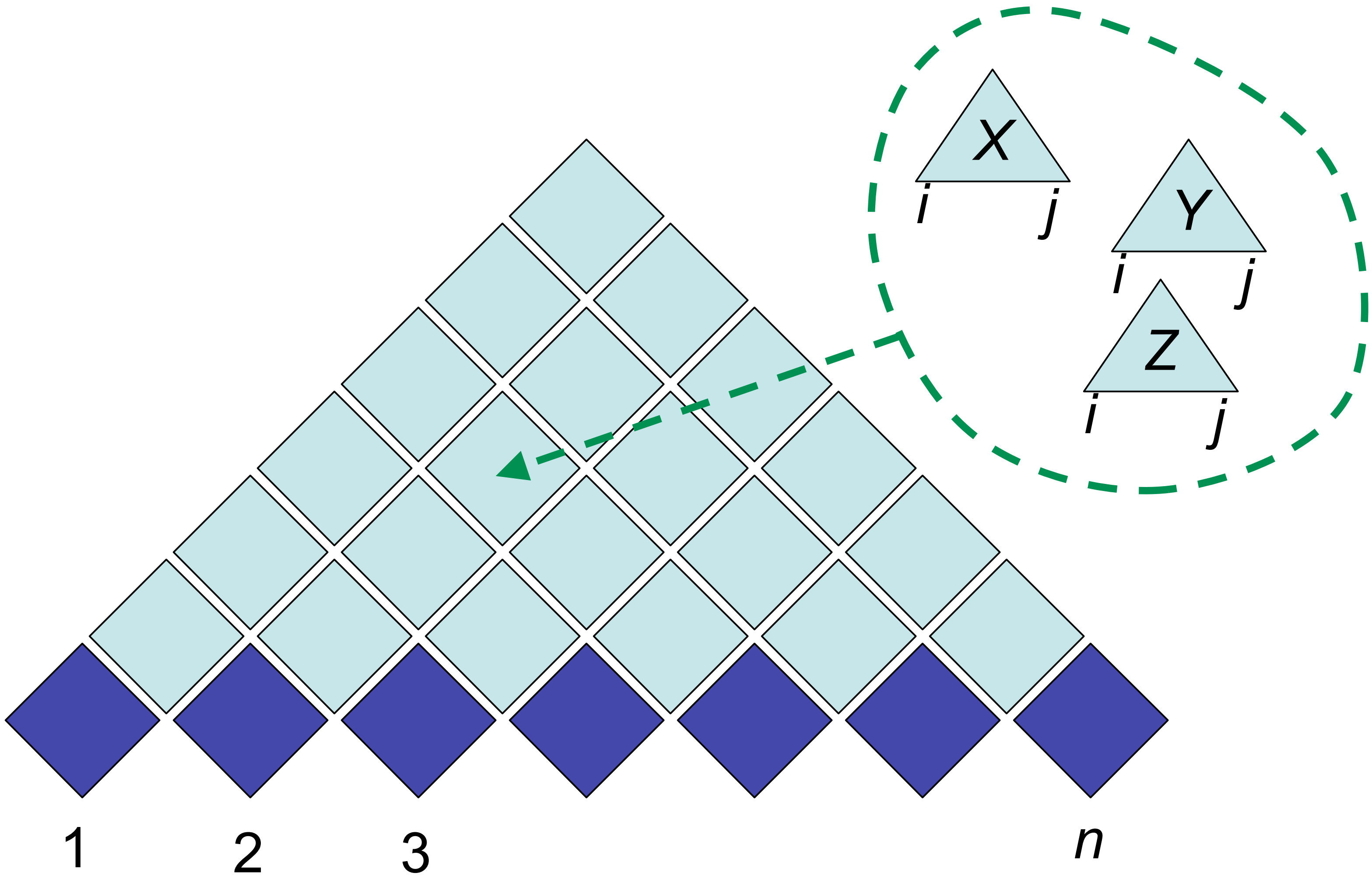
Resist This Temptation!

- CKY is not “building a tree” bottom-up.
- It is scoring partial hypotheses bottom-up.
- You can assume nothing about the tree until you get to the end!

Visualizing Probabilistic CKY

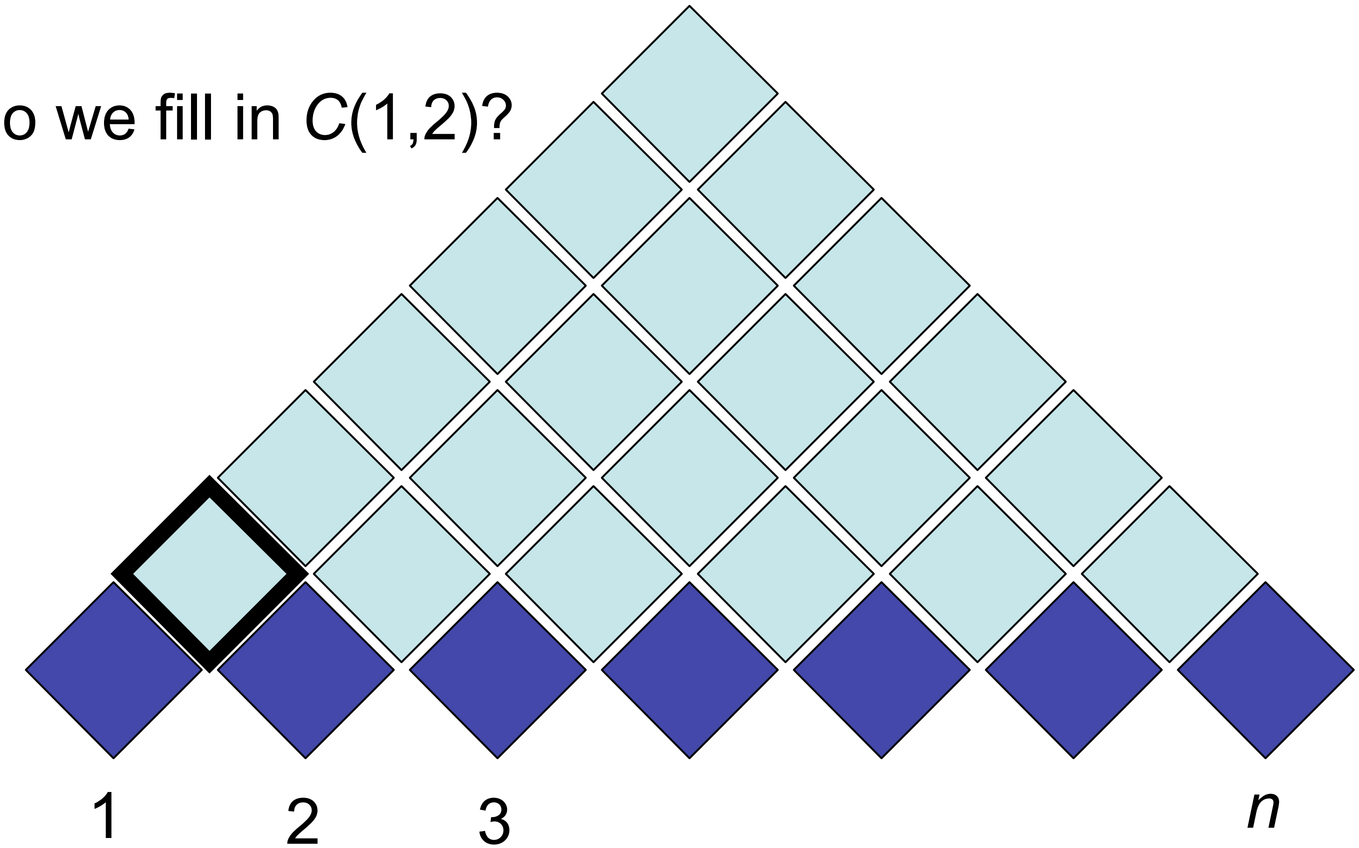


Visualizing Probabilistic CKY



Visualizing Probabilistic CKY

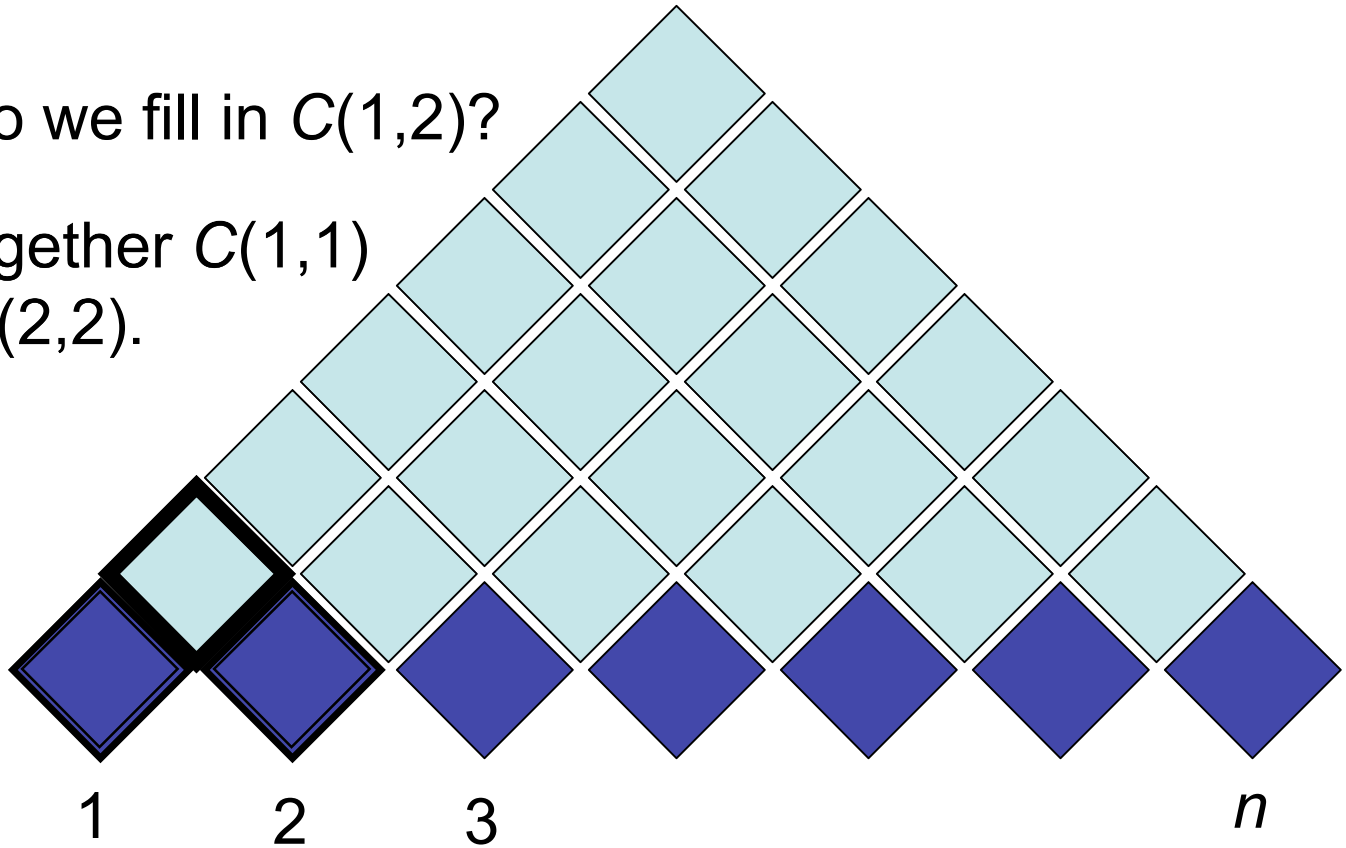
How do we fill in $C(1,2)$?



Visualizing Probabilistic CKY

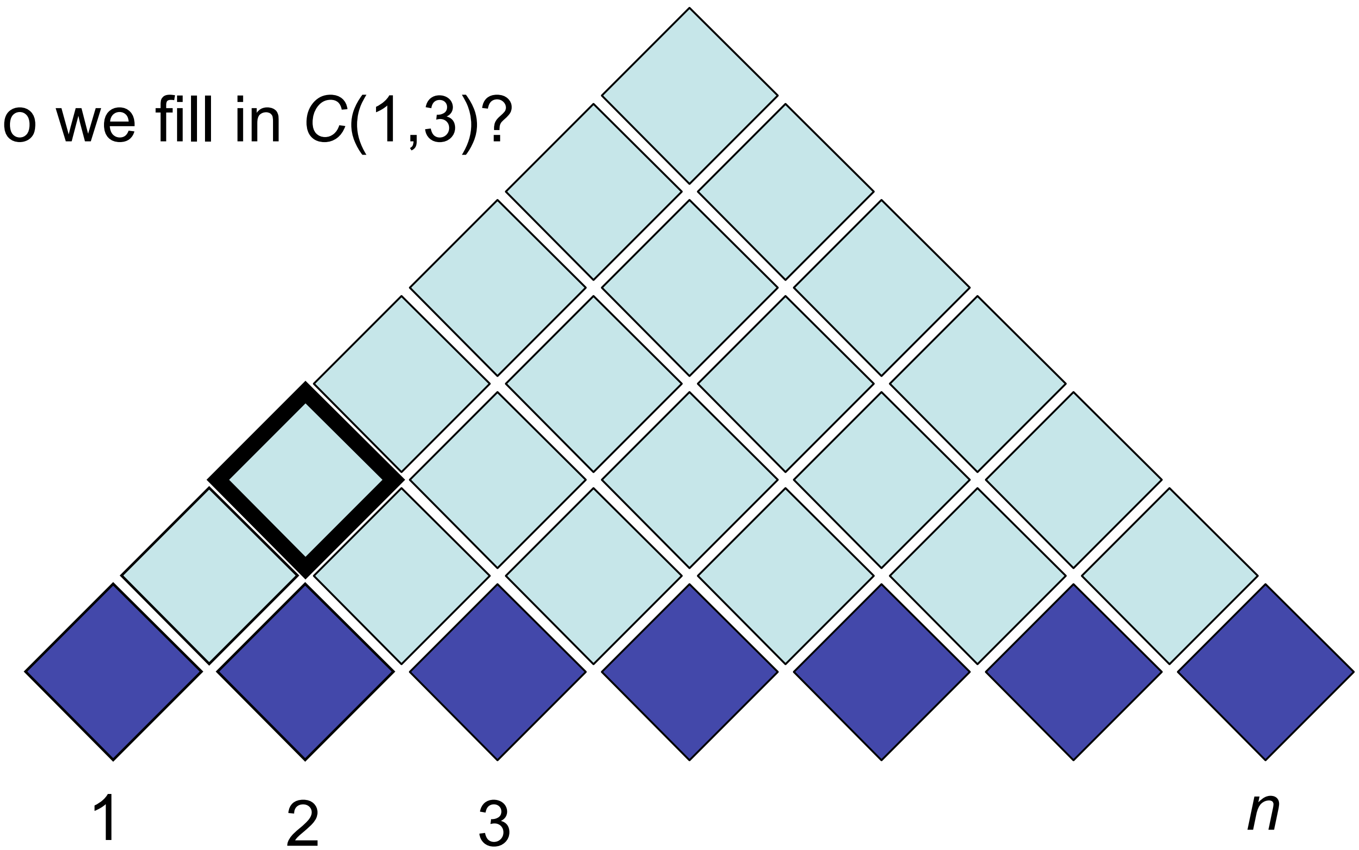
How do we fill in *C*(1,2)?

Put together *C*(1,1) and *C*(2,2).



Visualizing Probabilistic CKY

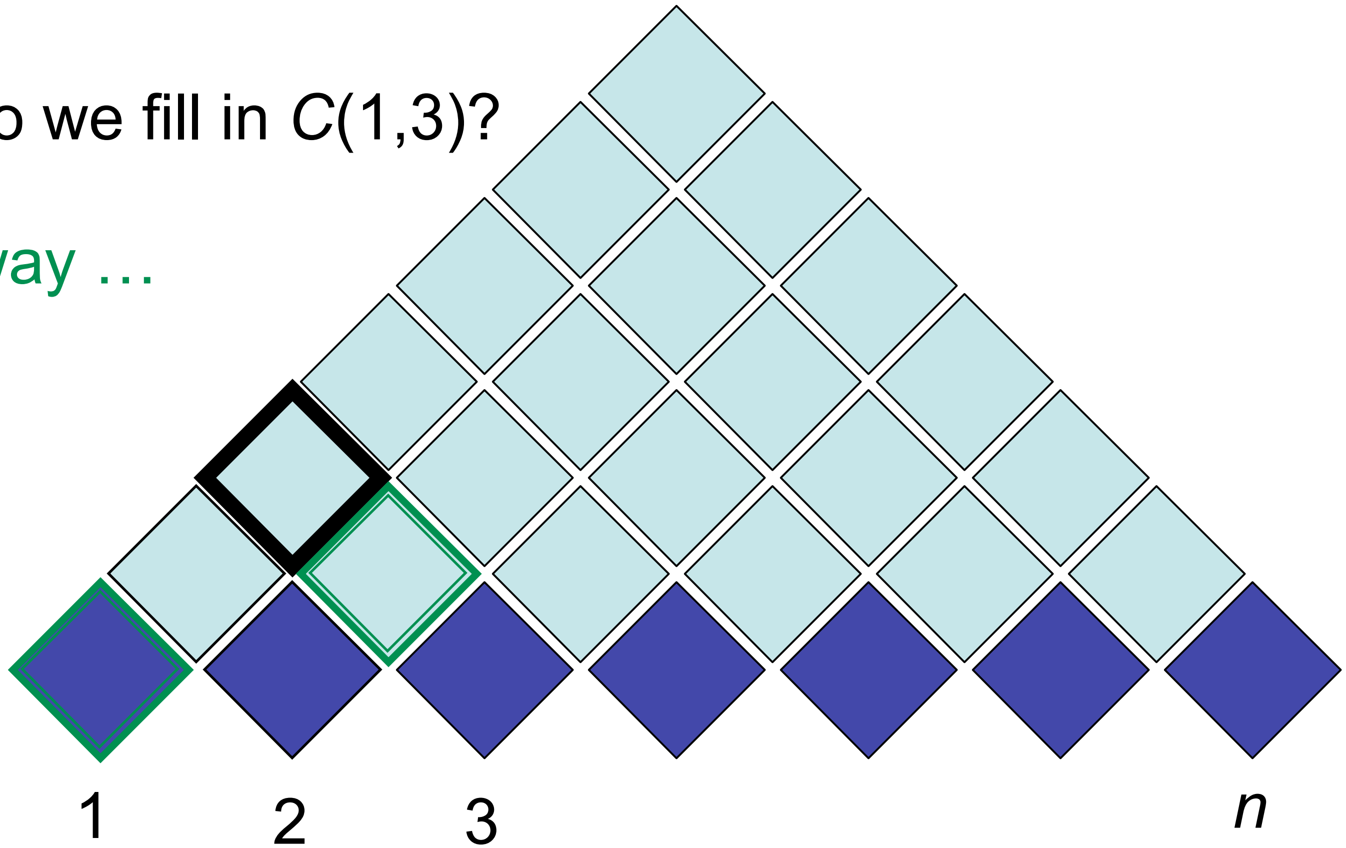
How do we fill in *C*(1,3)?



Visualizing Probabilistic CKY

How do we fill in *C*(1,3)?

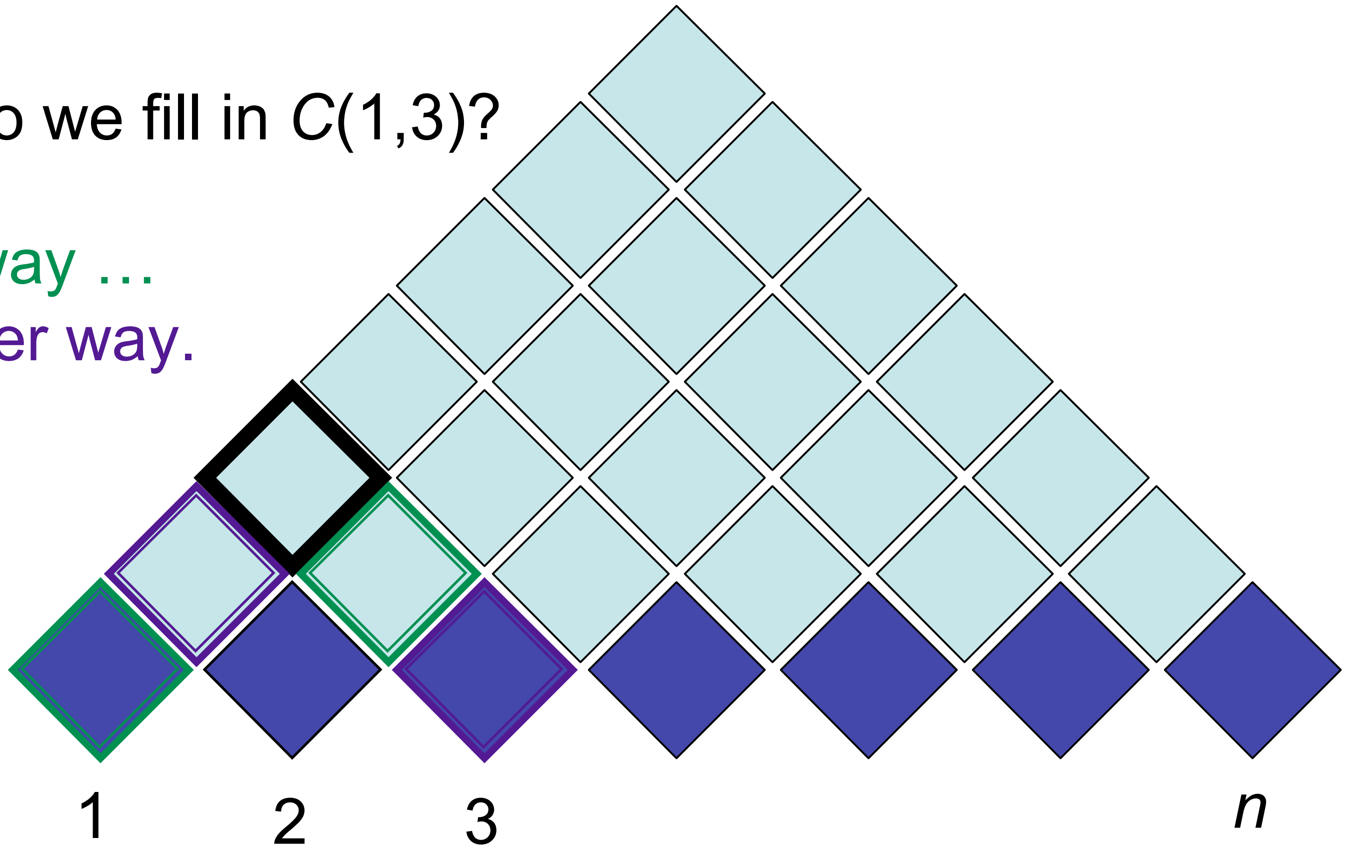
One way …



Visualizing Probabilistic CKY

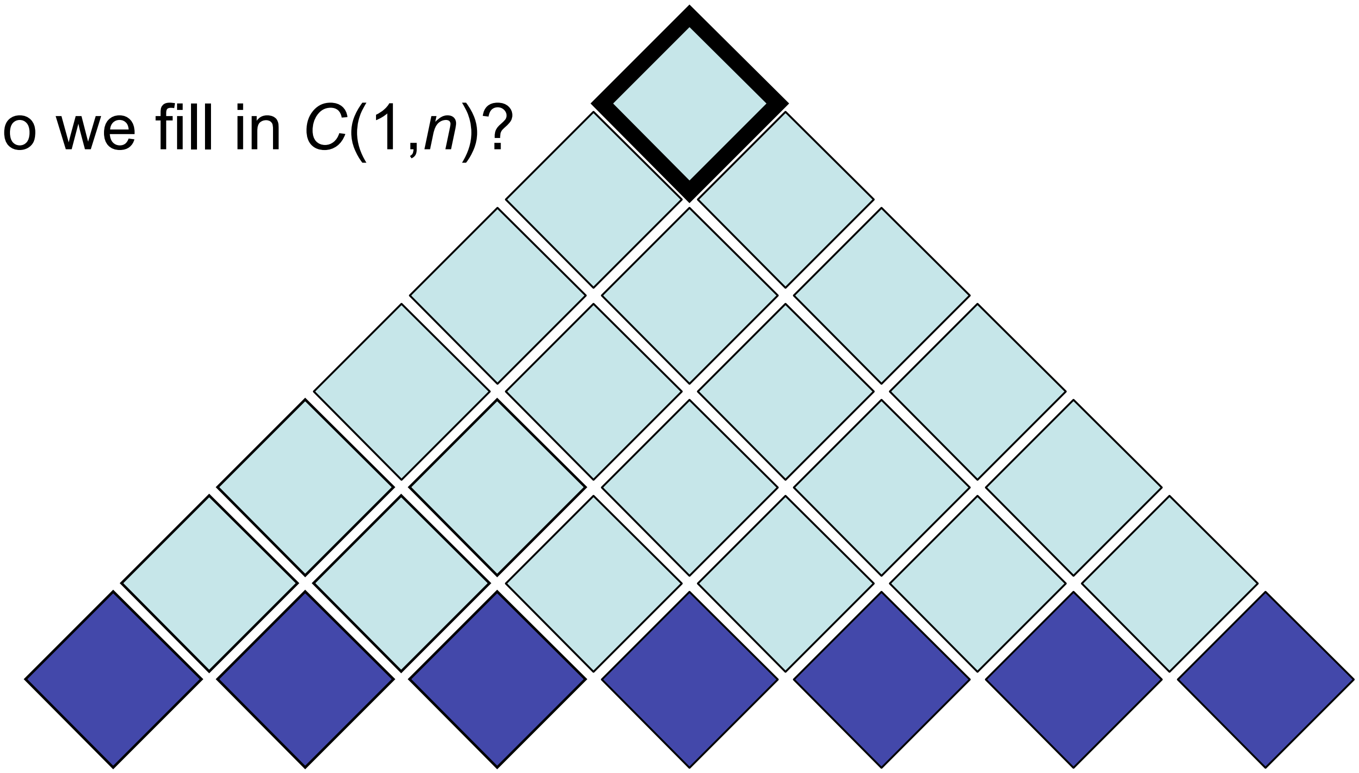
How do we fill in *C*(1,3)?

One way …
Another way.



Visualizing Probabilistic CKY

How do we fill in $C(1,n)$?

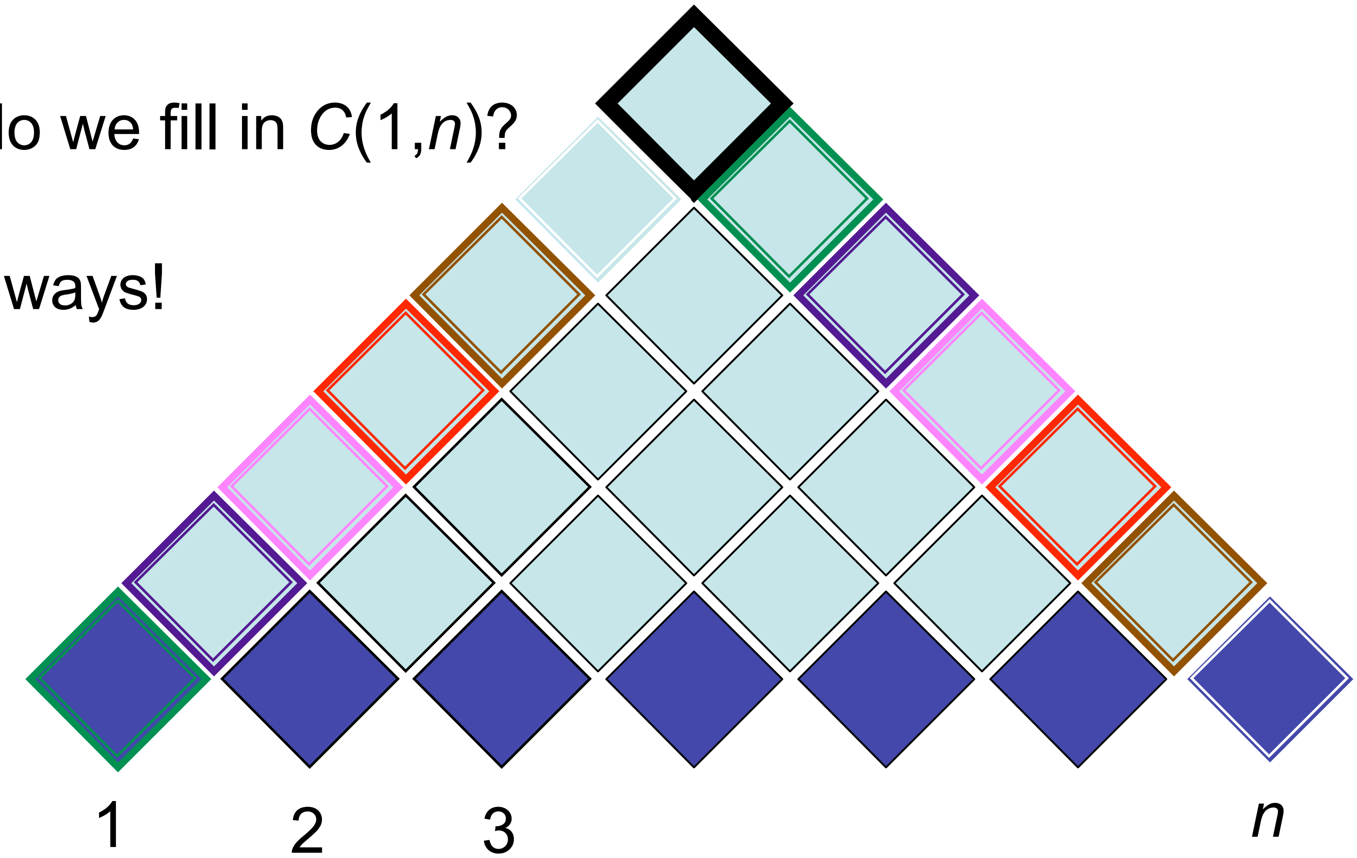


1 2 3 n

Visualizing Probabilistic CKY

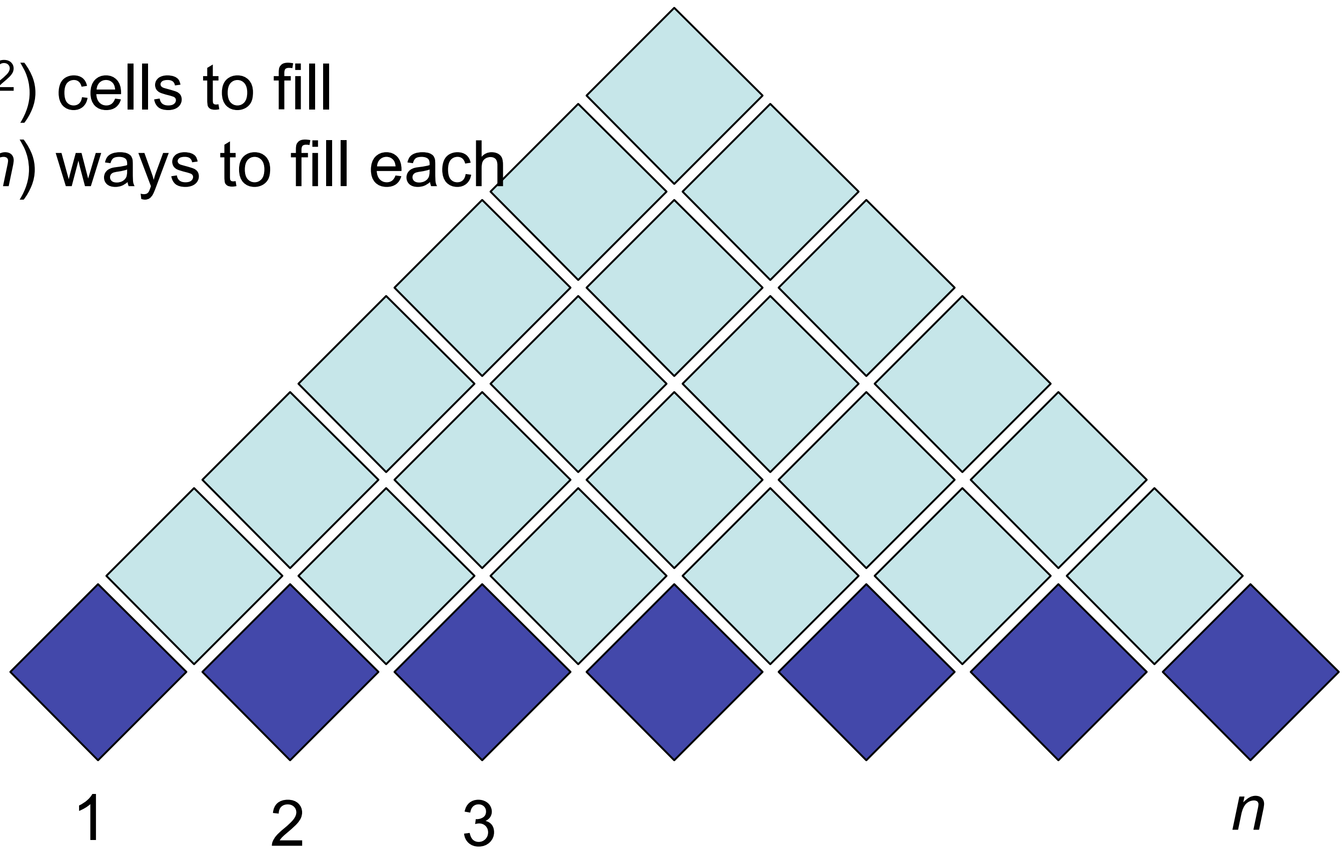
How do we fill in $C(1,n)$?

n - 1 ways!



Visualizing Probabilistic CKY

$O(|\mathbf{N}|n^2)$ cells to fill
$O(|\mathbf{N}|^2 n)$ ways to fill each



Probabilistic Earley's

Input: PCFG $G = (\Sigma, \mathbf{N}, S, \mathbf{R})$ and sequence $\mathbf{w} \in \boldsymbol{\Sigma}^*$

Output: most likely tree for **w**, if it exists, and its probability.

Probabilistic Earley's

$$C(X/\alpha,i,i) = \left\langle p\left(X \to \alpha\right),\text{null}\right\rangle$$

$$\text{if } \left(\left(\exists Z,h : C\left(Z/X,h,i\right) > 0\right) \vee \left(X = S \wedge i = 0\right)\right)$$

$$C\left(X/\alpha,i,j+1\right) = \left\langle C\left(X/w_j\alpha,i,j\right),\&C\left(X/w_j\alpha,i,j\right)\right\rangle$$

$$C\left(X/\alpha,i,k\right) = \left\langle \begin{array}{l} \max\limits_{j\in[i+1,k-2],Y\in\mathbf{N}} C\left(X/Y\alpha,i,j\right)\cdot C\left(Y/\varnothing,j+1,k\right) \\ \&\text{argmax}\ldots \end{array} \right\rangle$$

$$\text{goal} = C\left(S/\varnothing,0,|\mathbf{w}|\right)$$

Probabilistic Earley's

predict

$$C(X/\alpha,i,i) = \left\langle p\left(X \to \alpha\right),\mathrm{null}\right\rangle$$

$$\text{if } \Big(\big(\exists Z,h : C\big(Z/X,h,i\big) > 0\big) \vee \big(X = S \wedge i = 0\big)\Big)$$

$$C\left(X/\alpha,i,j+1\right) = \left\langle C\left(X/w_j\alpha,i,j\right),\&C\left(X/w_j\alpha,i,j\right)\right\rangle$$

$$C\left(X/\alpha,i,k\right) = \left\langle \begin{array}{l} \max\limits_{j\in[i+1,k-2],Y\in\mathbf{N}} C\left(X/Y\alpha,i,j\right)\cdot C\left(Y/\varnothing,j+1,k\right) \\ \&\mathrm{argmax}\ldots \end{array} \right\rangle$$

$$\mathrm{goal} = C\left(S/\varnothing,0,|\mathbf{w}|\right)$$

Probabilistic Earley's

$$C(X/\alpha,i,i)=\left\langle p\left(X\to\alpha\right),\mathrm{null}\right\rangle$$

$$\text{if }\left(\left(\exists Z,h: C\left(Z/X,h,i\right)>0\right)\vee\left(X=S\wedge i=0\right)\right)$$

scan

$$C\left(X/\alpha,i,j+1\right)=\left\langle C\left(X/w_j\alpha,i,j\right),\&C\left(X/w_j\alpha,i,j\right)\right\rangle$$

$$C\left(X/\alpha,i,k\right)=\left\langle \begin{array}{l}\max\limits_{j\in[i+1,k-2],Y\in\mathbf{N}} C\left(X/Y\alpha,i,j\right)\cdot C\left(Y/\varnothing,j+1,k\right)\\ \&\mathrm{argmax}\ldots\end{array}\right\rangle$$

$$\mathrm{goal}=C\left(S/\varnothing,0,\left|\mathbf{w}\right|\right)$$

Probabilistic Earley's

$$C(X/\alpha,i,i) = \left\langle p\left(X \to \alpha\right),\text{null}\right\rangle$$

$$\text{if } \left(\left(\exists Z,h : C\left(Z/X,h,i\right) > 0\right) \vee \left(X = S \wedge i = 0\right)\right)$$

$$C\left(X/\alpha,i,j+1\right) = \left\langle C\left(X/w_j\alpha,i,j\right),\&C\left(X/w_j\alpha,i,j\right)\right\rangle$$

complete

$$C\left(X/\alpha,i,k\right) = \left\langle \begin{array}{l} \max\limits_{j\in[i+1,k-2],Y\in\mathbf{N}} C\left(X/Y\alpha,i,j\right)\cdot C\left(Y/\varnothing,j+1,k\right) \\ \&\text{argmax}\ldots \end{array}\right\rangle$$

$$\text{goal} = C\left(S/\varnothing,0,|\mathbf{w}|\right)$$

Probabilistic Earley's (Corrected!)

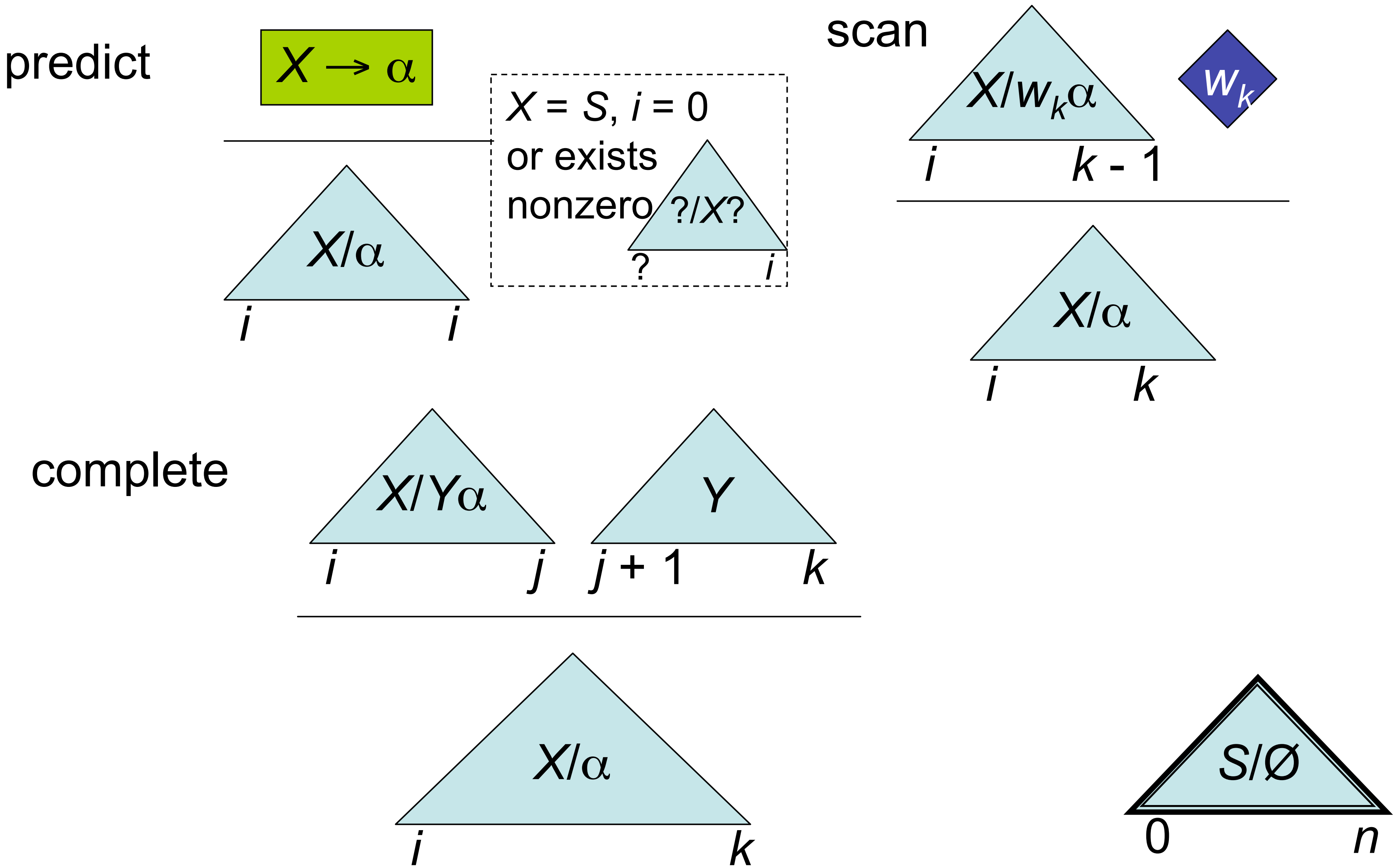
$$C(X/\alpha,i,i) = \left\langle p\left(X \rightarrow \alpha\right),\text{null}\right\rangle$$

$$\text{if } \left(\left(\exists Z,h : C\left(Z/X,h,i\right) > 0\right) \vee \left(X = S \wedge i = 0\right)\right)$$

$$C\left(X/\alpha,i,k\right) = \left\langle \begin{array}{l} \max\left(\begin{array}{l} \max\limits_{j\in[\,i+1,k-2],Y\in\mathbf{N}} C\left(X/Y\alpha,i,j\right)\cdot C\left(Y/\varnothing,j+1,k\right), \\ C\left(X/w_k\alpha,i,k-1\right) \end{array}\right) \\ \&\text{argmax}\ldots \end{array} \right\rangle$$

$$\text{goal} = C\left(S/\varnothing,0,\left|\mathbf{w}\right|\right)$$

Visualizing Probabilistic Earley's



CKY vs. Earley's

- Both $O(n^3)$ runtime, $O(n^2)$ space
- Earley's doesn't require the grammar to be in CNF
- Earley's usually moves left-to-right; CKY usually moves bottom-to-top.
- Earley's $\approx$ on-the-fly binarization + CKY

- Thought question: Does either remind you of Viterbi?

CKY and Earley’s vs. The World

- Tomita parsing - shift and reduce operations, with a stack - inspired by **search** in AI.
  - Can make it probabilistic.
  - No polynomial guarantees (could be exponential if lots of stack splitting).
  - In practice usually fast.
- CKY and Earley’s algorithms **can** be generalized to use an agenda, rather than filling in all cells.
  - “Best-first” tricks; sometimes optimality is not sacrificed!
- Remember the Forward algorithm?
  - We’ll come back to “inside” algorithms in a couple of weeks.