Soft Inference and Posterior Marginals

September 19, 2013

Soft vs. Hard Inference

- Hard inference
  - "Give me a single solution"
  - Viterbi algorithm
  - Maximum spanning tree (Chu-Liu-Edmonds alg.)
- Soft inference
  - Task 1: Compute a distribution over outputs
  - Task 2: Compute functions on distribution
    - **marginal probabilities**, expected values, entropies, divergences

Why Soft Inference?

- Useful applications of posterior distributions
  - **Entropy**: how confused is the model?
  - **Entropy**: how confused is the model of its prediction at time *i*?
  - **Expectations**
    - What is the expected number of words in a translation of this sentence?
    - What is the expected number of times a word ending in –ed was tagged as something other than a verb?
  - **Posterior marginals**: given some input, how likely is it that some (***latent***) event of interest happened?

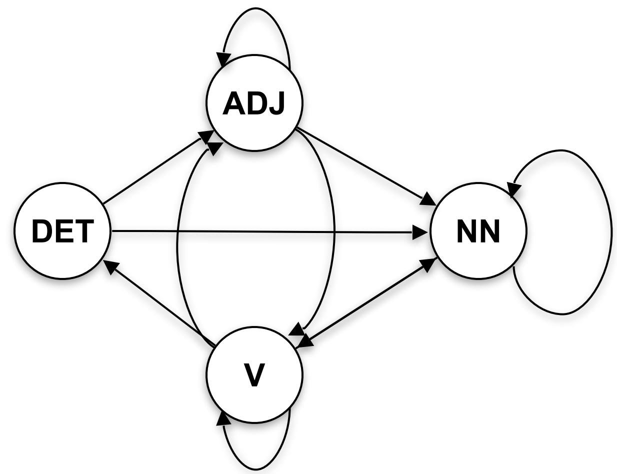
String Marginals

- Inference question for HMMs
  - What is the probability of a string **w**?
    Answer: generate all possible tag sequences and explicitly *marginalize*

$$O(|\Omega|^{|\mathbf{w}|}) \text{ time}$$

Initial Probabilities:

DET	ADJ	NN	V
0.5	0.1	0.3	0.1



η **Transition Probabilities:**

	DET	ADJ	NN	V
DET	0.0	0.0	0.0	0.5
ADJ	0.3	0.2	0.1	0.1
NN	0.7	0.7	0.3	0.2
V	0.0	0.1	0.4	0.1
	0.0	0.0	0.2	0.1

γ **Emission Probabilities:**

DET		**ADJ**		**NN**		**V**	
the	0.7	green	0.1	book	0.3	might	0.2
a	0.3	big	0.4	plants	0.2	watch	0.3
		old	0.4	people	0.2	watches	0.2
		might	0.1	person	0.1	loves	0.1
				John	0.1	reads	0.19
				watch	0.1	books	0.01

Examples:

John	might	watch
NN	**V**	**V**

the	old	person	loves	big	books
DET	**ADJ**	**NN**	**V**	**ADJ**	**NN**

John	**migh**	**watc**	$\Pr(x, y)$	**John**	**migh**	**watc**	$\Pr(x, y)$	**John**	**migh**	**watc**	$\Pr(x, y)$	**John**	**migh**	**watc**	$\Pr(x, y)$
DET	DET	DET	0.0	ADJ	DET	DET	0.0	NN	DET	DET	0.0	V	DET	DET	0.0
DET	DET	ADJ	0.0	ADJ	DET	ADJ	0.0	NN	DET	ADJ	0.0	V	DET	ADJ	0.0
DET	DET	NN	0.0	ADJ	DET	NN	0.0	NN	DET	NN	0.0	V	DET	NN	0.0
DET	DET	V	0.0	ADJ	DET	V	0.0	NN	DET	V	0.0	V	DET	V	0.0
DET	ADJ	DET	0.0	ADJ	ADJ	DET	0.0	NN	ADJ	DET	0.0	V	ADJ	DET	0.0
DET	ADJ	ADJ	0.0	ADJ	ADJ	ADJ	0.0	NN	ADJ	ADJ	0.0	V	ADJ	ADJ	0.0
DET	ADJ	NN	0.0	ADJ	ADJ	NN	0.0	**NN**	**ADJ**	**NN**	**0.0000042**	V	ADJ	NN	0.0
DET	ADJ	V	0.0	ADJ	ADJ	V	0.0	**NN**	**ADJ**	**V**	**0.0000009**	V	ADJ	V	0.0
DET	NN	DET	0.0	ADJ	NN	DET	0.0	NN	NN	DET	0.0	V	NN	DET	0.0
DET	NN	ADJ	0.0	ADJ	NN	ADJ	0.0	NN	NN	ADJ	0.0	V	NN	ADJ	0.0
DET	NN	NN	0.0	ADJ	NN	NN	0.0	NN	NN	NN	0.0	V	NN	NN	0.0
DET	NN	V	0.0	ADJ	NN	V	0.0	NN	NN	V	0.0	V	NN	V	0.0
DET	V	DET	0.0	ADJ	V	DET	0.0	NN	V	DET	0.0	V	V	DET	0.0
DET	V	ADJ	0.0	ADJ	V	ADJ	0.0	NN	V	ADJ	0.0	V	V	ADJ	0.0
DET	V	NN	0.0	ADJ	V	NN	0.0	***NN***	***V***	***NN***	***0.0000096***	V	V	NN	0.0
DET	V	V	0.0	ADJ	V	V	0.0	**NN**	**V**	**V**	**0.0000072**	V	V	V	0.0

John	**migh**	**watc**	$\Pr(x,y)$	**John**	**migh**	**watc**	$\Pr(x,y)$	**John**	**migh**	**watc**	$\Pr(x,y)$	**John**	**migh**	**watc**	$\Pr(x,y)$
DET	DET	DET	0.0	ADJ	DET	DET	0.0	NN	DET	DET	0.0	V	DET	DET	0.0
DET	DET	ADJ	0.0	ADJ	DET	ADJ	0.0	NN	DET	ADJ	0.0	V	DET	ADJ	0.0
DET	DET	NN	0.0	ADJ	DET	NN	0.0	NN	DET	NN	0.0	V	DET	NN	0.0
DET	DET	V	0.0	ADJ	DET	V	0.0	NN	DET	V	0.0	V	DET	V	0.0
DET	ADJ	DET	0.0	ADJ	ADJ	DET	0.0	NN	ADJ	DET	0.0	V	ADJ	DET	0.0
DET	ADJ	ADJ	0.0	ADJ	ADJ	ADJ	0.0	NN	ADJ	ADJ	0.0	V	ADJ	ADJ	0.0
DET	ADJ	NN	0.0	ADJ	ADJ	NN	0.0	**NN**	**ADJ**	**NN**	**0.0000042**	V	ADJ	NN	0.0
DET	ADJ	V	0.0	ADJ	ADJ	V	0.0	**NN**	**ADJ**	**V**	**0.0000009**	V	ADJ	V	0.0
DET	NN	DET	0.0	ADJ	NN	DET	0.0	NN	NN	DET	0.0	V	NN	DET	0.0
DET	NN	ADJ	0.0	ADJ	NN	ADJ	0.0	NN	NN	ADJ	0.0	V	NN	ADJ	0.0
DET	NN	NN	0.0	ADJ	NN	NN	0.0	NN	NN	NN	0.0	V	NN	NN	0.0
DET	NN	V	0.0	ADJ	NN	V	0.0	NN	NN	V	0.0	V	NN	V	0.0
DET	V	DET	0.0	ADJ	V	DET	0.0	NN	V	DET	0.0	V	V	DET	0.0
DET	V	ADJ	0.0	ADJ	V	ADJ	0.0	NN	V	ADJ	0.0	V	V	ADJ	0.0
DET	V	NN	0.0	ADJ	V	NN	0.0	***NN***	***V***	***NN***	***0.0000096***	V	V	NN	0.0
DET	V	V	0.0	ADJ	V	V	0.0	**NN**	**V**	**V**	**0.0000072**	V	V	V	0.0

$$p = 0.0000219$$

Weighted Logic Programming

- Slightly different notation than the textbook, but you will see it in the literature
- WLP is useful here because it lets us **build hypergraphs**

$$\frac{I_1 : w_1 \quad I_2 : w_2 \quad \cdots \quad I_k : w_k}{I : w} \quad \phi$$

Weighted Logic Programming

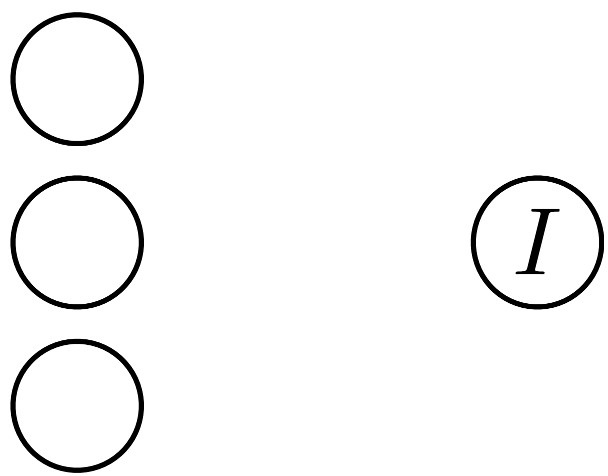
- Slightly different notation than the textbook, but you will see it in the literature
- WLP is useful here because it lets us **build hypergraphs**

$$\frac{I_1 : w_1 \quad I_2 : w_2 \quad \cdots \quad I_k : w_k}{I : w} \quad \phi$$

$$\vdots$$

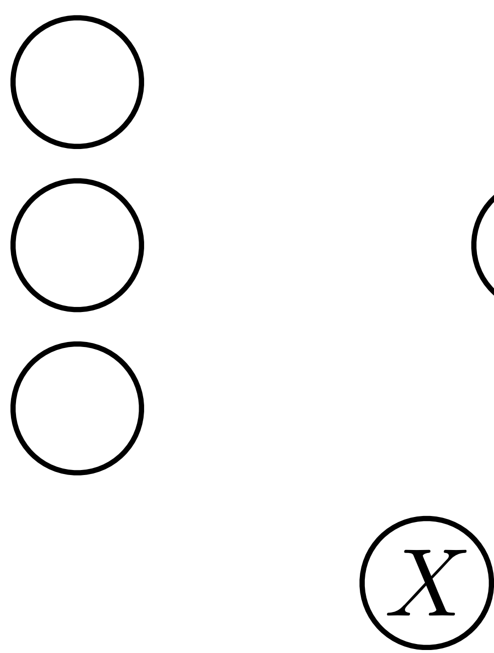
Hypergraphs

$$\frac{A \qquad B \qquad C}{I}$$



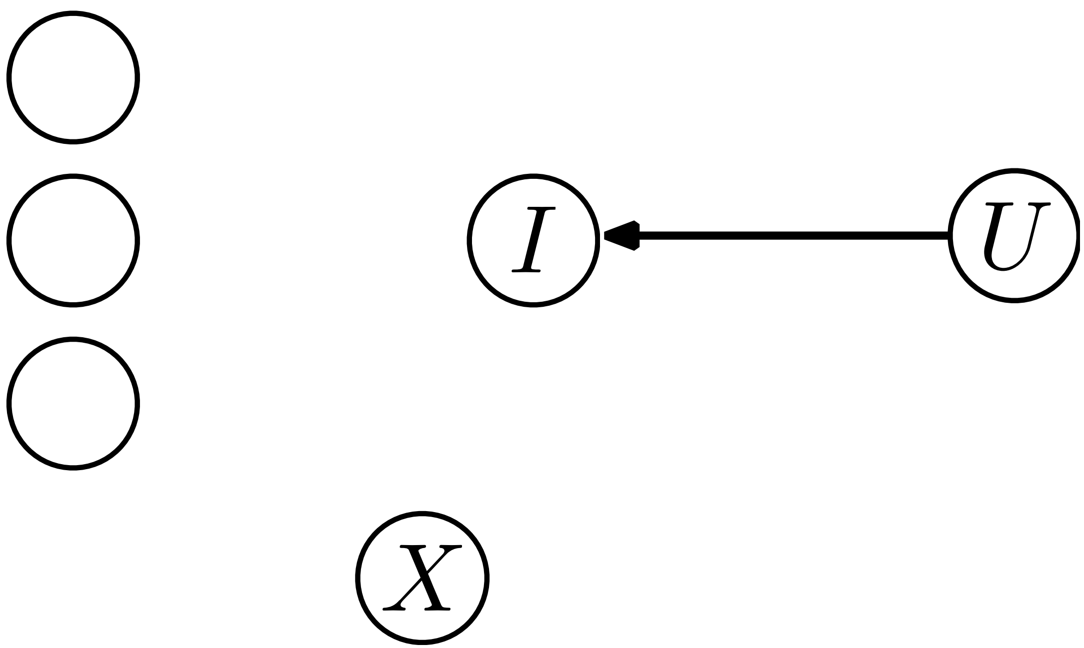
Hypergraphs

$$\frac{A \qquad B \qquad C}{I} \qquad\qquad \frac{X \qquad Y}{I}$$



Hypergraphs

$$\frac{A \qquad B \qquad C}{I} \qquad\qquad \frac{X \qquad Y}{I} \qquad\qquad \frac{U}{I}$$



Viterbi Algorithm

Item form

$$[q, i]$$

Viterbi Algorithm

Item form

$$[q, i]$$

Axioms

$$\frac{}{[\textsc{start}, 0] : 1}$$

Viterbi Algorithm

Item form

$$[q, i]$$

Axioms

$$\overline{[\text{START}, 0] : 1}$$

Goals

$$[\text{STOP}, |\mathbf{x}| + 1]$$

Viterbi Algorithm

Item form

$$[q, i]$$

Axioms

$$\overline{[\textsc{start}, 0] : 1}$$

Goals

$$[\textsc{stop}, |\mathbf{x}| + 1]$$

Inference rules

$$\frac{[q, i] : w}{[r, i+1] : w \otimes \eta(q \rightarrow r) \otimes \gamma(r \downarrow x_{i+1})}$$

Viterbi Algorithm

Item form

$$[q, i]$$

Axioms

$$\frac{}{[\text{START}, 0] : 1}$$

Goals

$$[\text{STOP}, |\mathbf{x}| + 1]$$

Inference rules

$$\frac{[q, i] : w}{[r, i + 1] : w \otimes \eta(q \rightarrow r) \otimes \gamma(r \downarrow x_{i+1})}$$

$$\frac{[q, |\mathbf{x}|] : w}{[\text{STOP}, |\mathbf{x}| + 1] : w \otimes \eta(q \rightarrow \text{STOP})}$$

Viterbi Algorithm

w=(John, might, watch) **Goal:** $[\mathrm{STOP}, 4]$

String Marginals

- Inference question for HMMs
  - What is the probability of a string **w**?
    Answer: generate all possible tag sequences and explicitly *marginalize*

    $$O(|\Omega|^{|\mathbf{w}|})$$ time

    Answer: use the **forward algorithm**

    $$O(|\Omega|^2 \times |\mathbf{w}|)$$ time

    $$O(|\Omega|)$$ space

Forward Algorithm

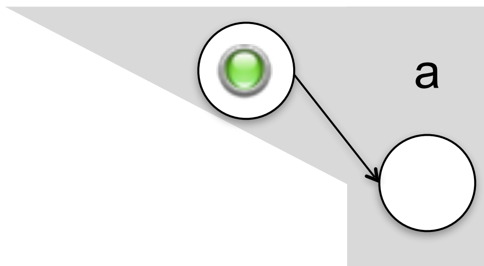
- Instead of computing a **max** of inputs at each node, use **addition**
- Same run-time, same space requirements
- Viterbi cell interpretation
  - What is the score of the best path through the lattice ending in state *q* at time *i*?
- **What does a forward node weight correspond to?**

Forward Algorithm Recurrence

$$\alpha_0(\text{START}) = 1$$

$$\alpha_t(y) = \sum_{q \in \Omega} \eta(q \to y) \times \gamma(y \downarrow x_i) \times \alpha_{t-1}(q)$$

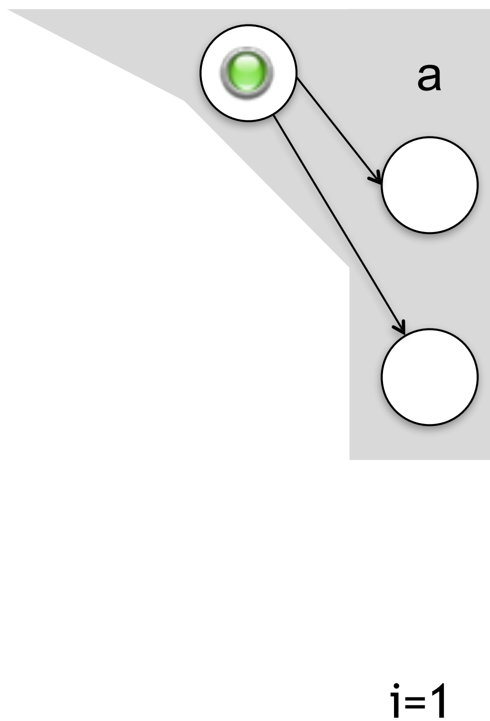
Forward Chart



i=1

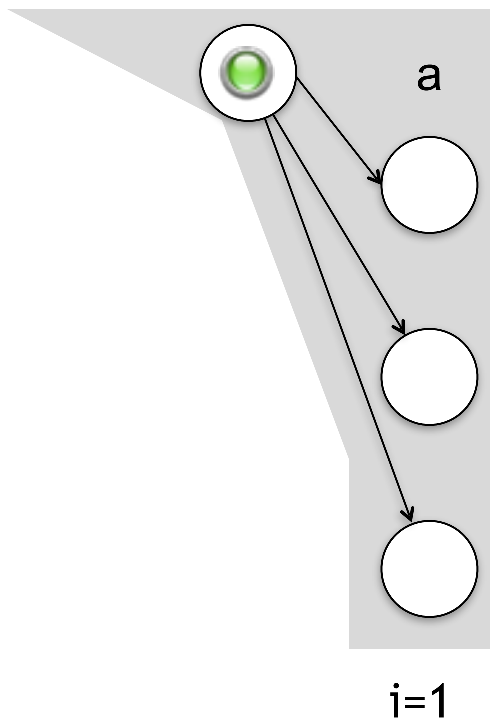
$$\alpha_t(q) = p(\mathrm{START}, x_1, \ldots, x_t, y_t = q)$$

Forward Chart



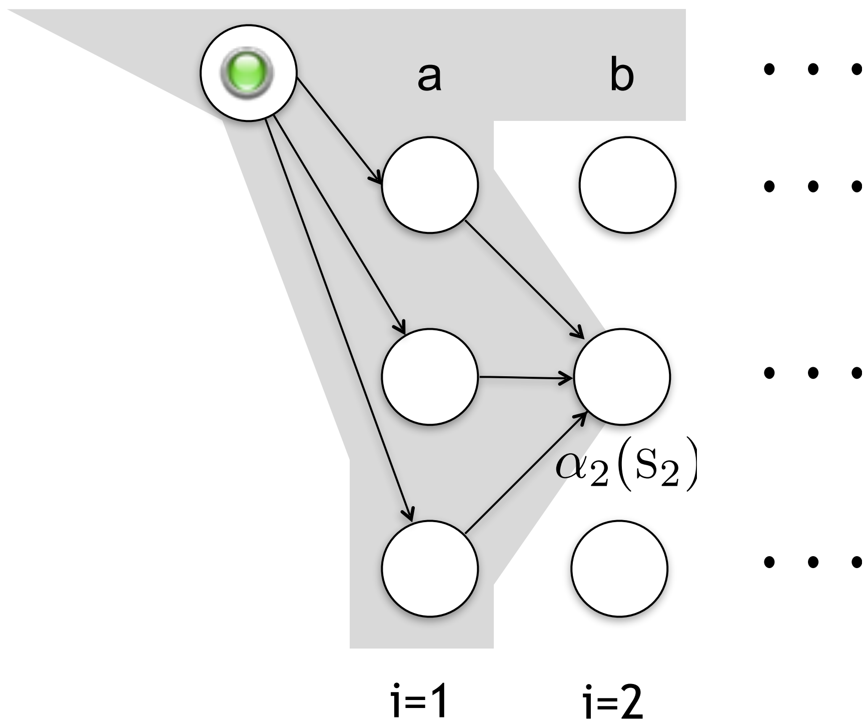
$$\alpha_t(q) = p(\text{START}, x_1, \ldots, x_t, y_t = q)$$

Forward Chart

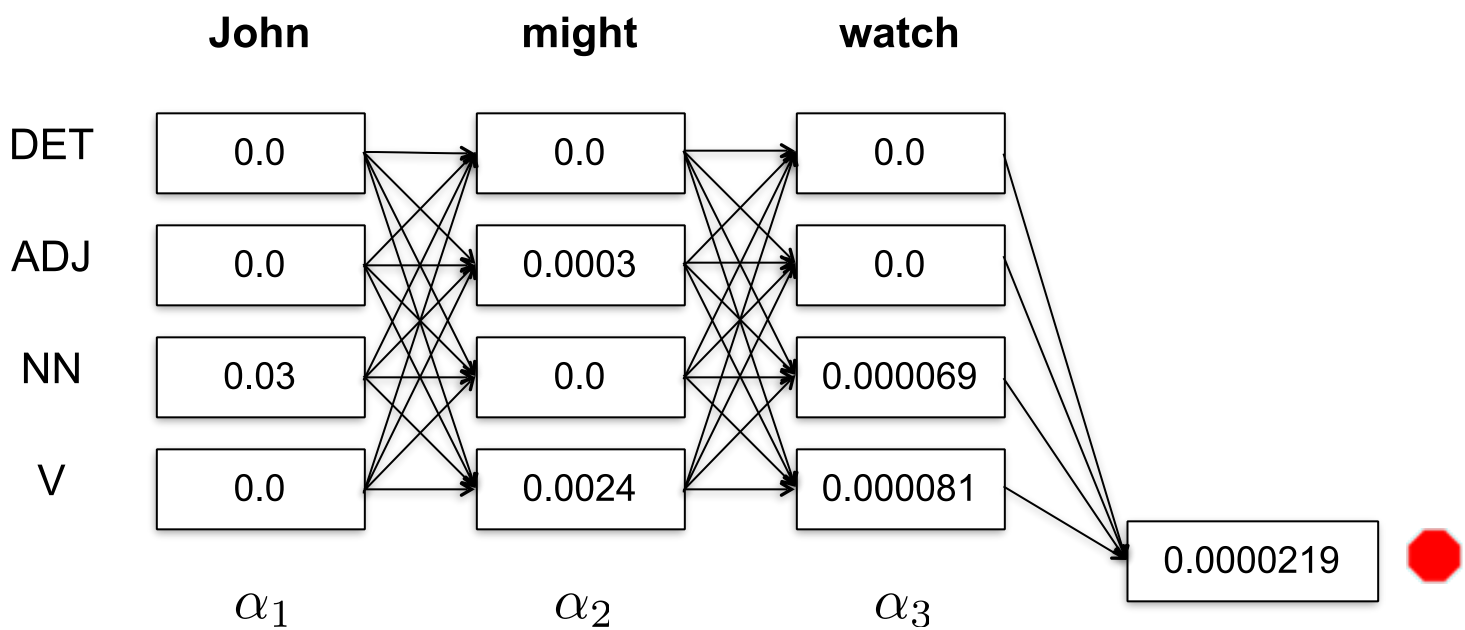


$$\alpha_t(q) = p(\mathrm{START}, x_1, \ldots, x_t, y_t = q)$$

Forward Chart



$$\alpha_t(q) = p(\mathrm{START}, x_1, \ldots, x_t, y_t = q)$$


John
might
watch
DET
ADJ
NN
V
0.0
0.0
0.03
0.0
0.0
0.0003
0.0
0.0024
0.0
0.0
0.000069
0.000081
0.0000219
α_1
α_2
α_3

$$p = 0.0000219$$

Posterior Marginals

- Marginal inference question for HMMs
  - Given **x**, what is the probability of being in a state *q* at time *i*?

$$p(x_1, \ldots, x_i, y_i = q \mid y_0 = \text{START}) \times$$

$$p(x_{i+1}, \ldots, x_{|\mathbf{x}|} \mid y_i = q)$$

  - Given **x**, what is the probability of transitioning from state *q* to *r* at time *i*?

$$p(x_1, \ldots, x_i, y_i = q \mid y_0 = \text{START}) \times$$

$$\eta(q \rightarrow r) \times \gamma(r \downarrow x_{i+1}) \times$$

$$p(x_{i+2}, \ldots, x_{|\mathbf{x}|} \mid y_{i+1} = r)$$

Posterior Marginals

- Marginal inference question for HMMs
  - Given **x**, what is the probability of being in a state *q* at time *i*?

$$\boxed{p(x_1, \ldots, x_i, y_i = q \mid y_0 = \mathrm{START}) \times}$$

$$p(x_{i+1}, \ldots, x_{|\mathbf{x}|} \mid y_i = q)$$

  - Given **x**, what is the probability of transitioning from state *q* to *r* at time *i*?

$$\boxed{p(x_1, \ldots, x_i, y_i = q \mid y_0 = \mathrm{START}) \times}$$

$$\eta(q \rightarrow r) \times \gamma(r \downarrow x_{i+1}) \times$$

$$p(x_{i+2}, \ldots, x_{|\mathbf{x}|} \mid y_{i+1} = r)$$

Posterior Marginals

- Marginal inference question for HMMs
  - Given **x**, what is the probability of being in a state *q* at time *i*?

$$p(x_1, \ldots, x_i, y_i = q \mid y_0 = \text{START}) \times$$

$$\boxed{p(x_{i+1}, \ldots, x_{|\mathbf{x}|} \mid y_i = q)}$$

  - Given **x**, what is the probability of transitioning from state *q* to *r* at time *i*?

$$p(x_1, \ldots, x_i, y_i = q \mid y_0 = \text{START}) \times$$

$$\eta(q \rightarrow r) \times \gamma(r \downarrow x_{i+1}) \times$$

$$\boxed{p(x_{i+2}, \ldots, x_{|\mathbf{x}|} \mid y_{i+1} = r)}$$

Backward Algorithm

- Start at the goal node(s) and work **backwards** through the hypergraph
- What is the probability in the goal node cell?
- What if there is more than one cell?
- What is the value of the axiom cell?

Backward Recurrence

$$\beta_{|\mathbf{x}|+1}(\text{STOP}) = 1$$

$$\beta_i(q) = \sum_{r \in \Omega} \beta_{i+1}(r) \times \gamma(r \downarrow x_{i+1}) \times \eta(q \rightarrow r)$$

Backward Chart

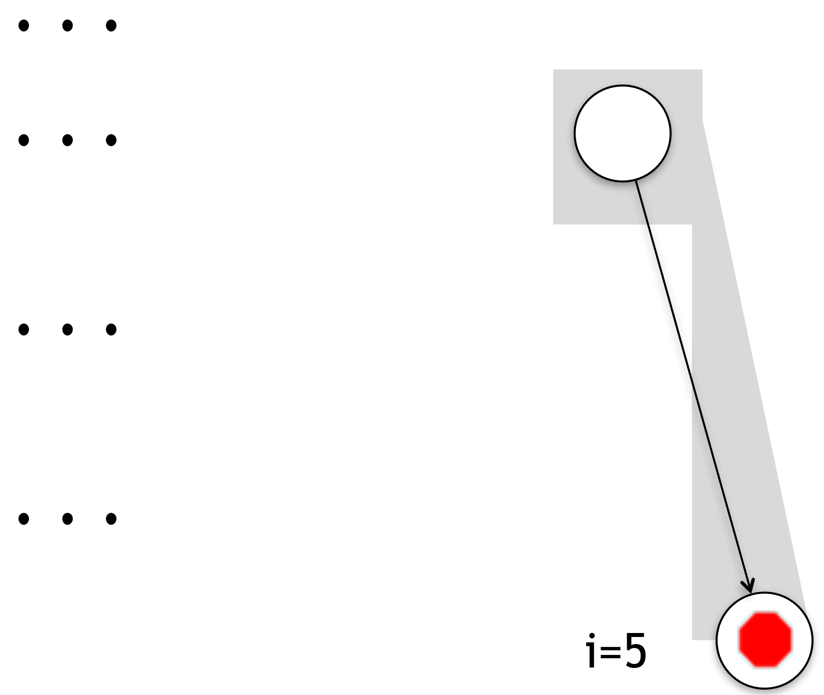
• • •

• • •

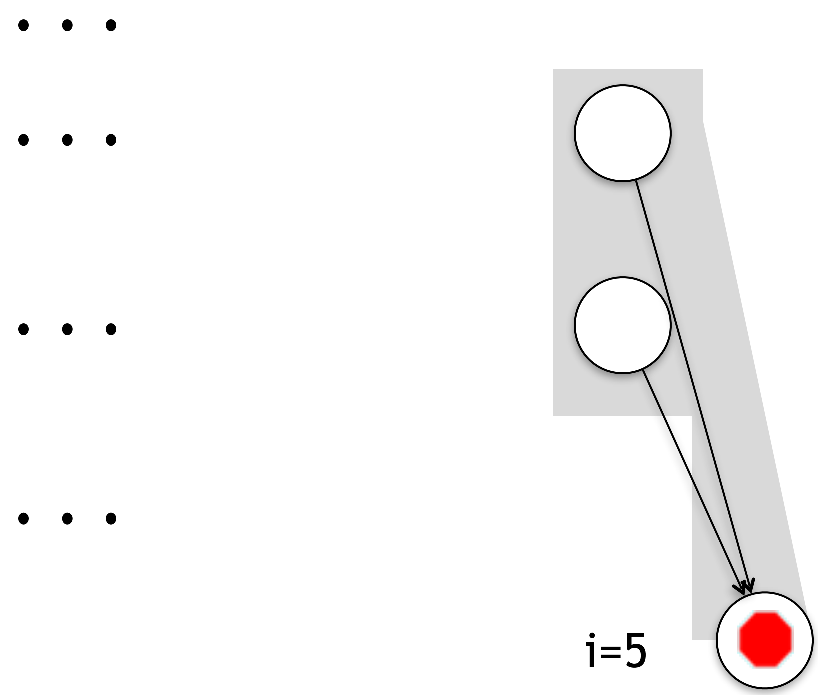
• • •

• • •

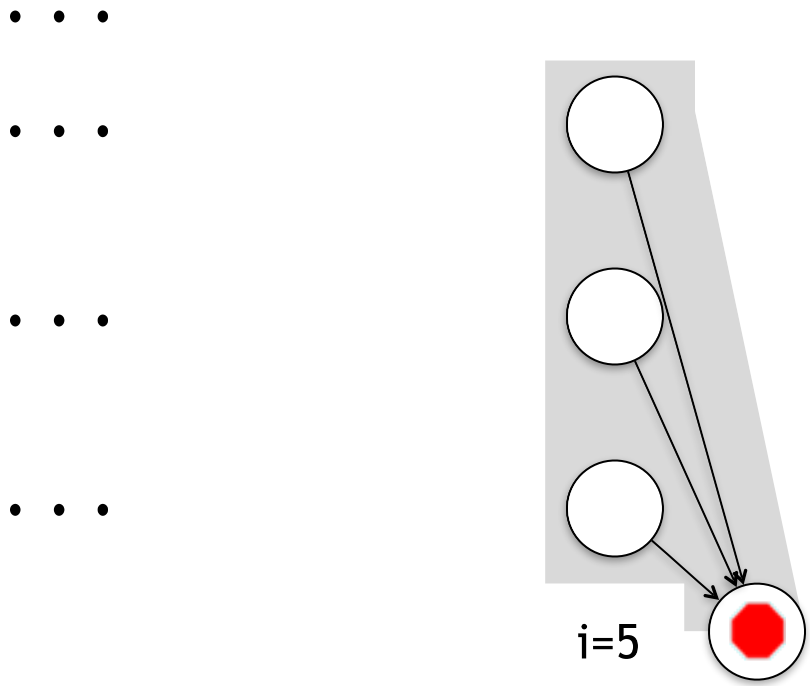
Backward Chart



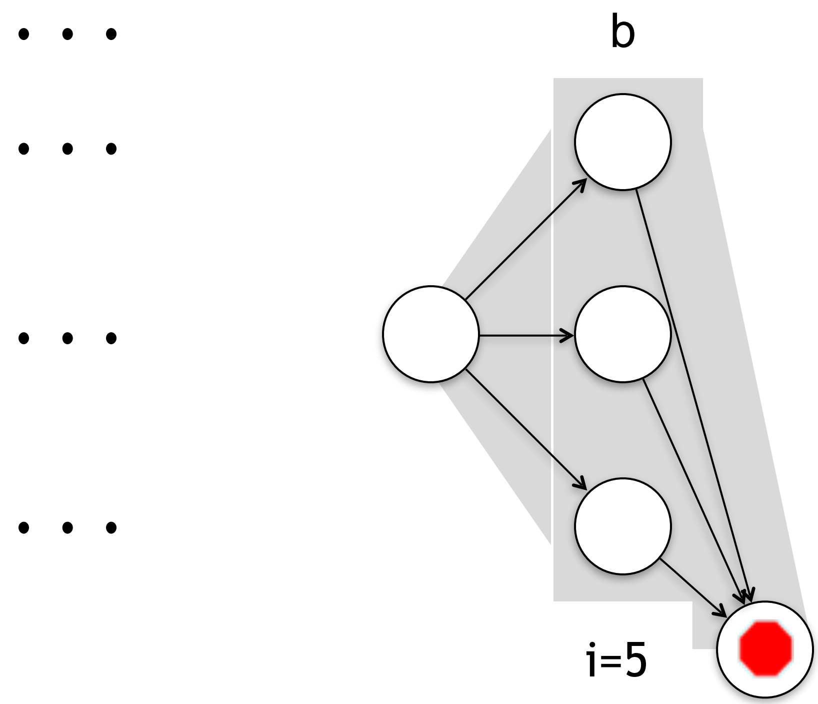
Backward Chart



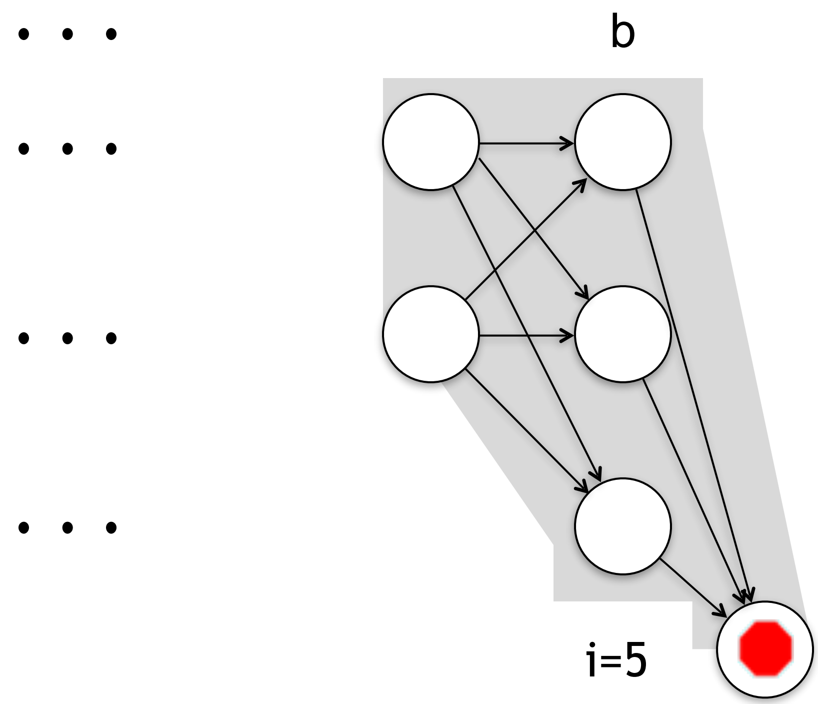
Backward Chart



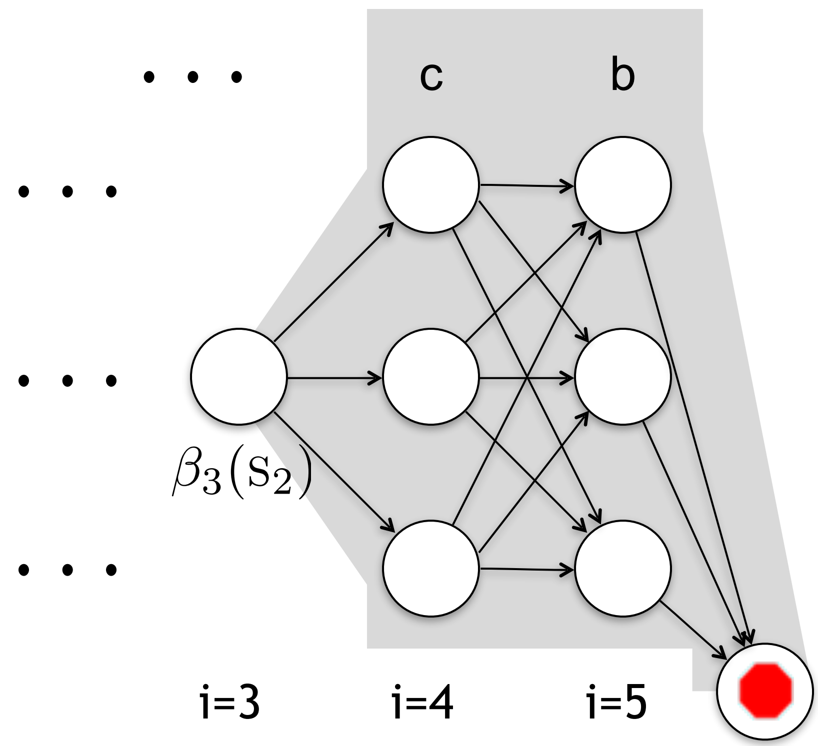
Backward Chart



Backward Chart



Backward Chart



$$\beta_t(q) = p(x_{t+1}, \ldots, x_{|\mathbf{x}|} \mid y_t = q)$$

Forward-Backward

- Compute forward chart

$$\alpha_t(q) = p(\text{START}, x_1, \ldots, x_t, y_t = q)$$

- Compute backward chart

$$\beta_t(q) = p(x_{t+1}, \ldots, x_{|\mathbf{x}|}, \text{STOP} \mid y_t = q)$$

What is $\alpha_t(q) \times \beta_t(q)$?

Forward-Backward

- Compute forward chart

$$\alpha_t(q) = p(\text{START}, x_1, \ldots, x_t, y_t = q)$$

- Compute backward chart

$$\beta_t(q) = p(x_{t+1}, \ldots, x_{|\mathbf{x}|}, \text{STOP} \mid y_t = q)$$

What is $\alpha_t(q) \times \beta_t(q)$ **?**

$$p(\mathbf{x}, y_t = q) = \alpha_t(q) \times \beta_t(q)$$

Edge Marginals

- What is the probability that **x** was generated and q -> r happened at time t?

$$
\begin{aligned}
p(x_1, \ldots, x_i, y_i = q \mid y_0 = \text{START}) \times \\
\eta(q \rightarrow r) \times \gamma(r \downarrow x_{i+1}) \times \\
p(x_{i+2}, \ldots, x_{|\mathbf{x}|} \mid y_{i+1} = r)
\end{aligned}
$$

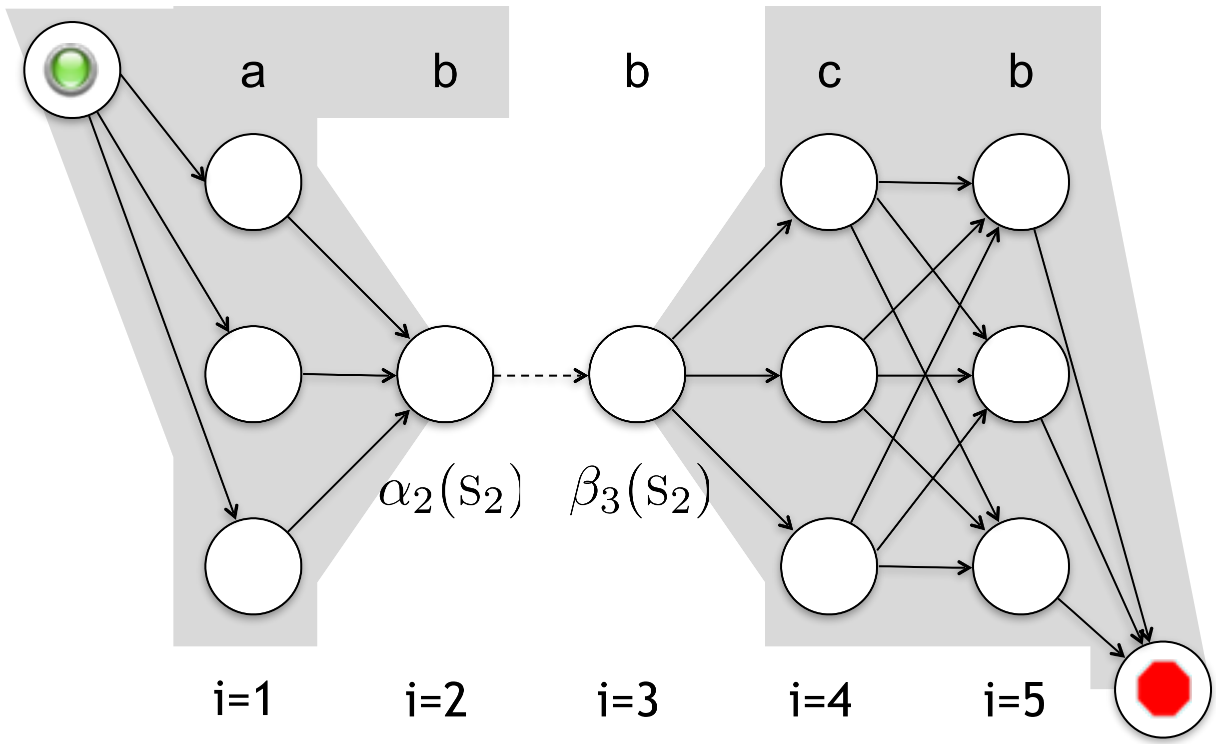
Edge Marginals

- What is the probability that **x** was generated and q -> r happened at time *t*?

$$p(x_1, \ldots, x_i, y_i = q \mid y_0 = \text{START}) \times$$
$$\eta(q \to r) \times \gamma(r \downarrow x_{i+1}) \times$$
$$p(x_{i+2}, \ldots, x_{|\mathbf{x}|} \mid y_{i+1} = r)$$

$$\alpha_t(q) \times$$
$$\eta(q \to r) \times \gamma(r \downarrow x_{t+1}) \times$$
$$\beta_{t+1}(r)$$

Forward-Backward



Generic Inference

- **Semirings** are useful structures in abstract algebra
  - Set of values
  - *Addition*, with additive identity 0: (a + 0 = a)
  - *Multiplication*, with mult identity 1: (a * 1 = a)
    - Also: a * 0 = 0
  - *Distributivity*: a * (b + c) = a * b + a * c
  - **Not required**: commutativity, inverses

So What?

- You can unify Forward and Viterbi by changing the semiring

$$\text{FORWARD}(\mathcal{G}) = \bigoplus_{\pi \in \mathcal{G}} \bigotimes_{e \in \pi} w[e]$$

Table 2.1: Elements of common semirings.

semiring	$\mathbb{K}$	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$	notes
Boolean	$\{0,1\}$	$\vee$	$\wedge$	0	1	idempotent
count	$\mathbb{N}_0 \cup \{\infty\}$	$+$	$\times$	0	1	
probability	$\mathbb{R}_+ \cup \{\infty\}$	$+$	$\times$	0	1	
tropical	$\mathbb{R} \cup \{-\infty, \infty\}$	max	$+$	$-\infty$	0	idempotent
log	$\mathbb{R} \cup \{-\infty, \infty\}$	$\oplus_{\log}$	$+$	$-\infty$	0	

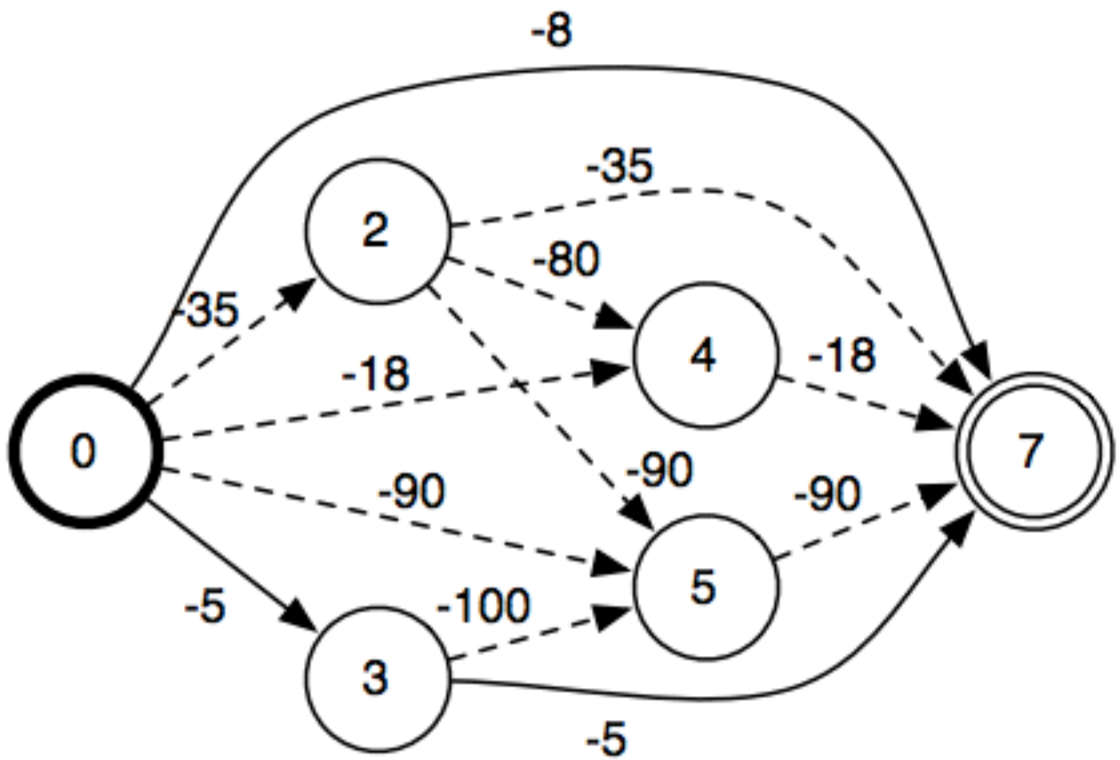
Semiring Inside

- **Probability semiring**
  – marginal probability of output
- **Counting semiring**
  – number of paths ("taggings")
- **Viterbi semiring**
  – best scoring derivation
- **Log semiring** $w[e] = \boldsymbol{w}^{\mathsf{T}}\boldsymbol{f}(e)$
  – $\log(Z)$ = log partition function

Semiring Edge-Marginals

- **Probability semiring**
  - posterior marginal probability of each edge
- **Counting semiring**
  - number of paths going through each edge
- **Viterbi semiring**
  - score of best path going through each edge
- **Log semiring**
  - log (sum of all exp path weights of all paths with e)
    = log(posterior marginal probability) + log(Z)

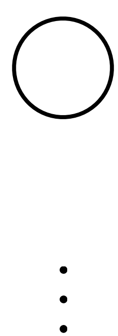
Max-Marginal Pruning



Weighted Logic Programming

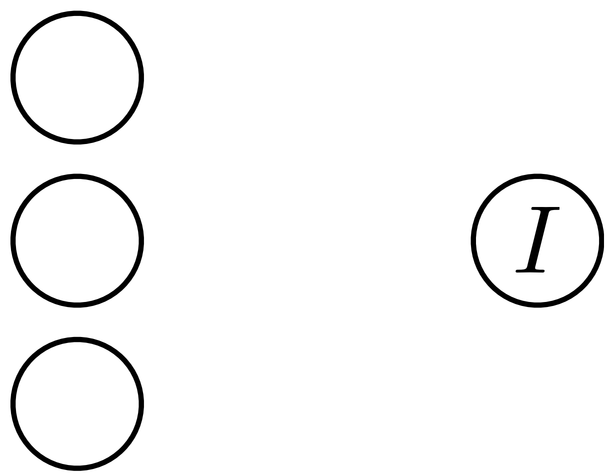
- Slightly different notation than the textbook, but you will see it in the literature
- WLP is useful here because it lets us **build hypergraphs**

$$\frac{I_1 : w_1 \quad I_2 : w_2 \quad \cdots \quad I_k : w_k}{I : w} \quad \phi$$



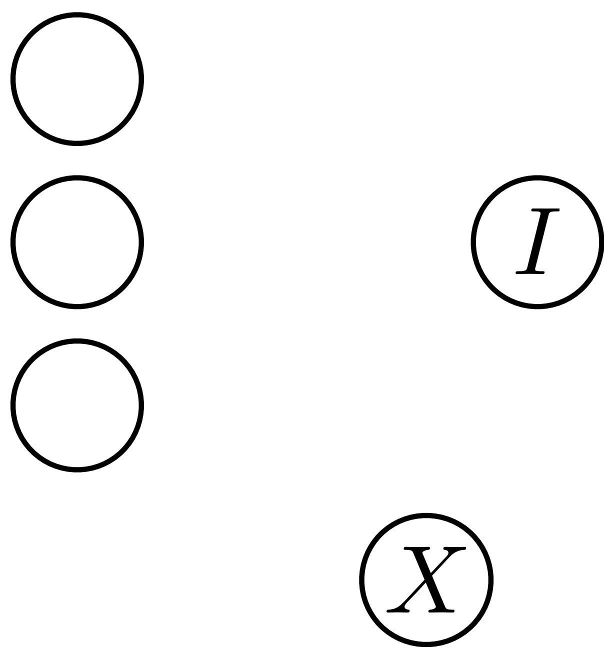
Hypergraphs

$$\frac{A \qquad B \qquad C}{I}$$



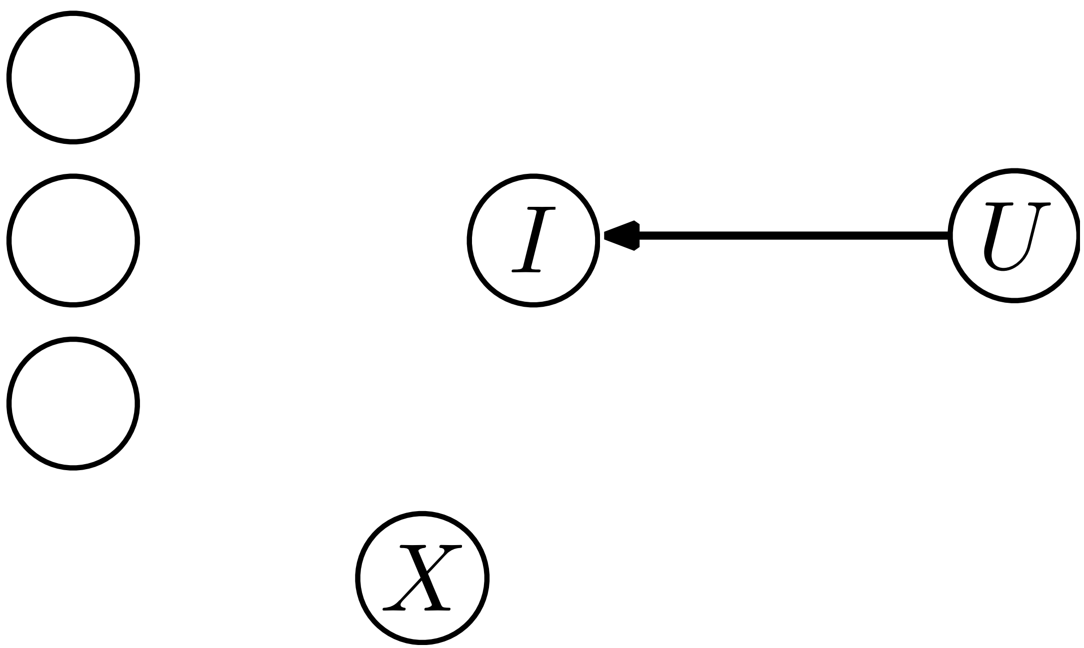
Hypergraphs

$$\frac{A \qquad B \qquad C}{I} \qquad\qquad \frac{X \qquad Y}{I}$$



Hypergraphs

$$\frac{A \quad B \quad C}{I} \qquad \frac{X \quad Y}{I} \qquad \frac{U}{I}$$



Generalizing Forward-Backward

- Forward/Backward algorithms are a special case of **Inside/Outside algorithms**
- It's helpful to think of I/O as algorithms on PCFG parse forests, but it's more general
  - Recall the 5 views of decoding: decoding is parsing
  - **More specifically, decoding is a weighted proof forest**

CKY Algorithm

Item form

$$[X, i, j]$$

CKY Algorithm

Item form

$$[X, i, j]$$

Goals

$$[\mathrm{S}, 1, |\mathbf{x}| + 1]$$

CKY Algorithm

Item form

$$[X, i, j]$$

Goals

$$[\mathrm{S}, 1, |\mathbf{x}| + 1]$$

Axioms

$$\frac{}{[N, i, i+1] : w} \quad (N \xrightarrow{w} x_i) \in G$$

CKY Algorithm

Item form

$$[X, i, j]$$

Goals

$$[\mathrm{S}, 1, |\mathbf{x}| + 1]$$

Axioms

$$\frac{}{[N, i, i+1] : w} \quad (N \xrightarrow{w} x_i) \in G$$

Inference rules

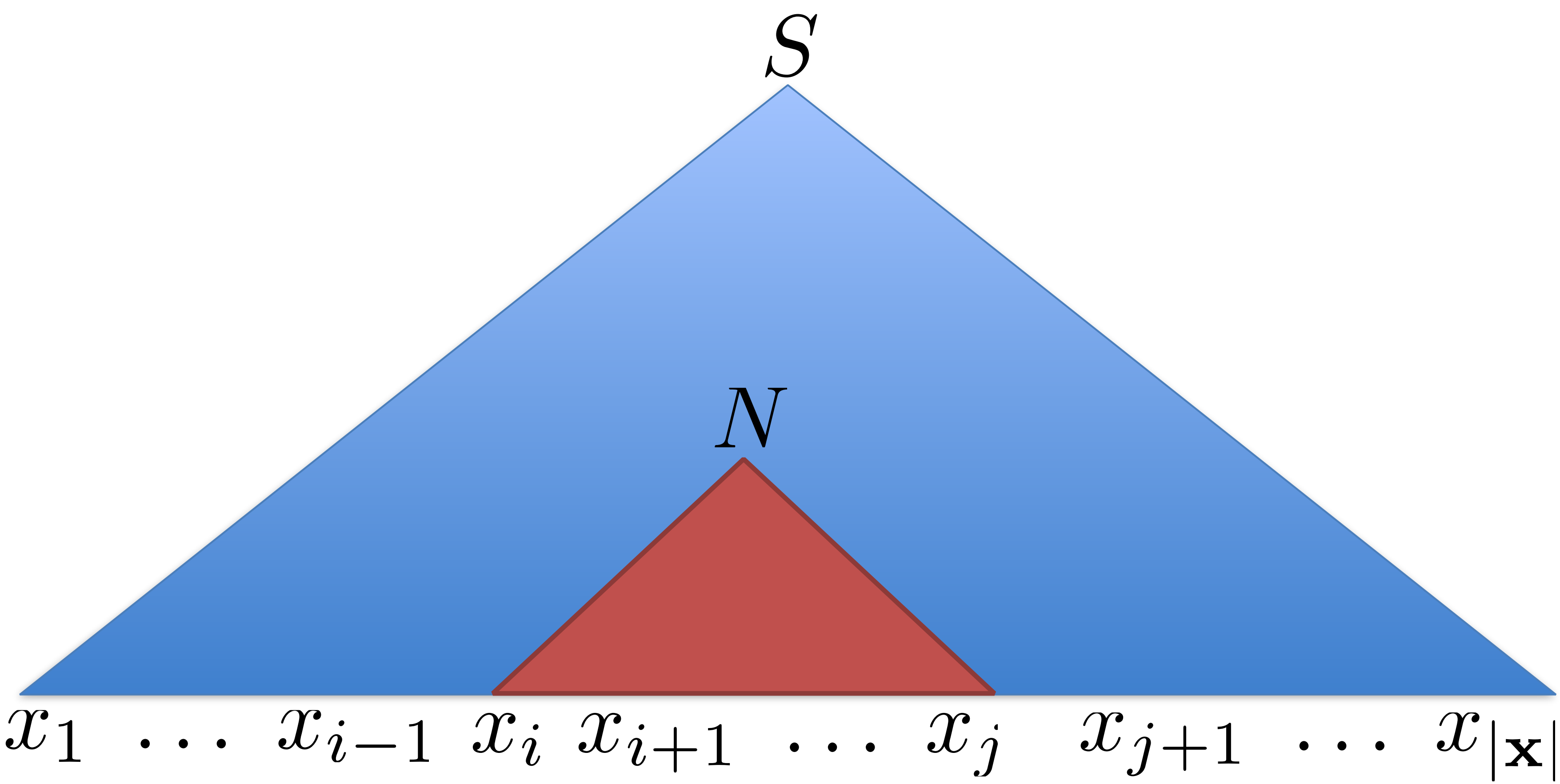
$$\frac{[X, i, k] : u \quad [Y, k, j] : v}{[Z, i, j] : u \otimes v \otimes w} \quad (Z \xrightarrow{w} X\ Y) \in G$$

Posterior Marginals

- Marginal inference question for PCFGs
  - Given **w**, what is the probability of having a constituent of type Z from *i* to *j*?
  - Given **w**, what is the probability of having a constituent of ***any*** type from *i* to *j*?
  - Given **w**, what is the probability of using rule Z -> XY to derive the span from *i* to *j*?

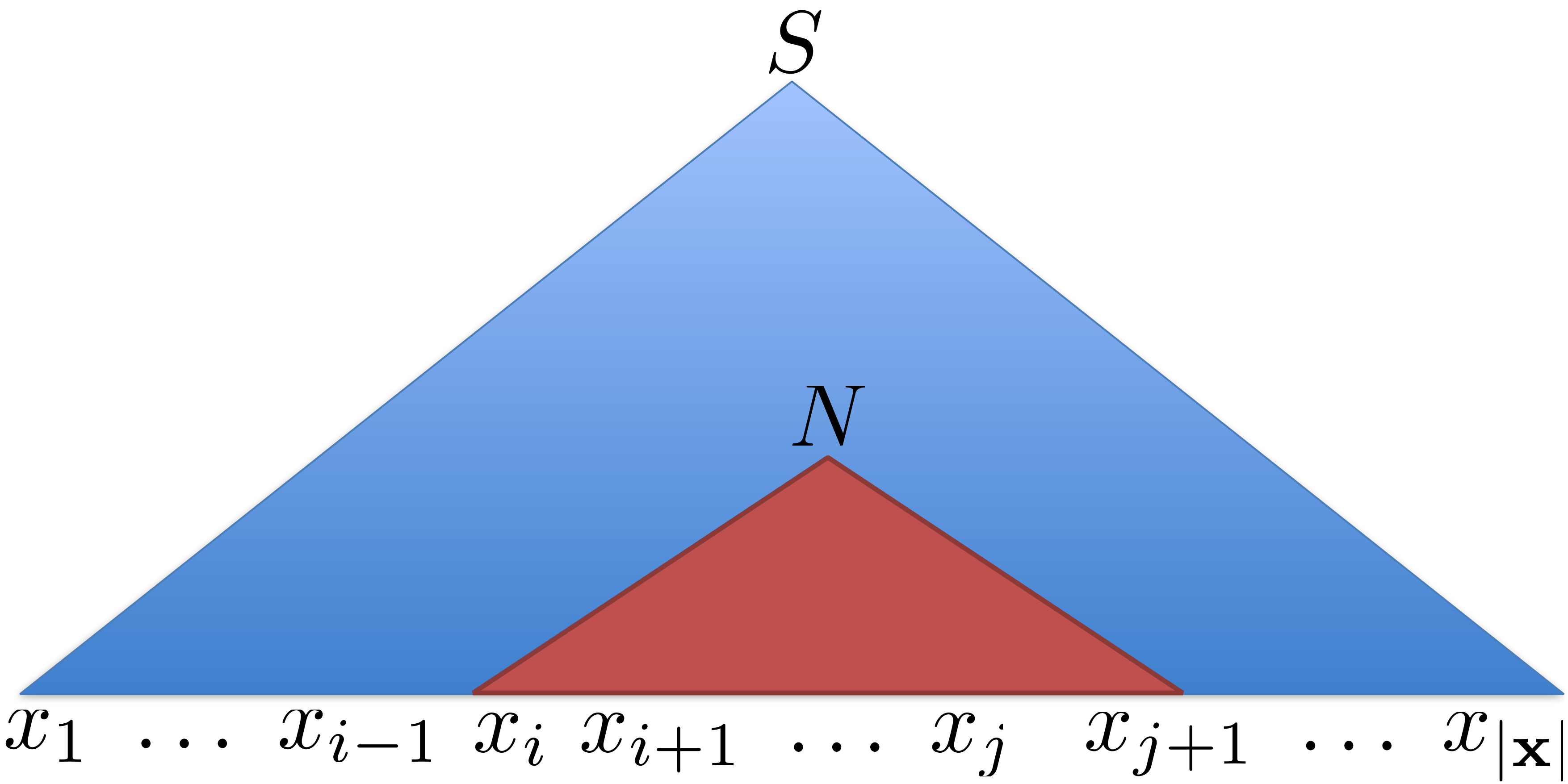
Inside Algorithm

$$\alpha_{[i,j]}(N) = p(x_i, x_{i+1}, \ldots, x_j \mid N; \mathcal{G})$$



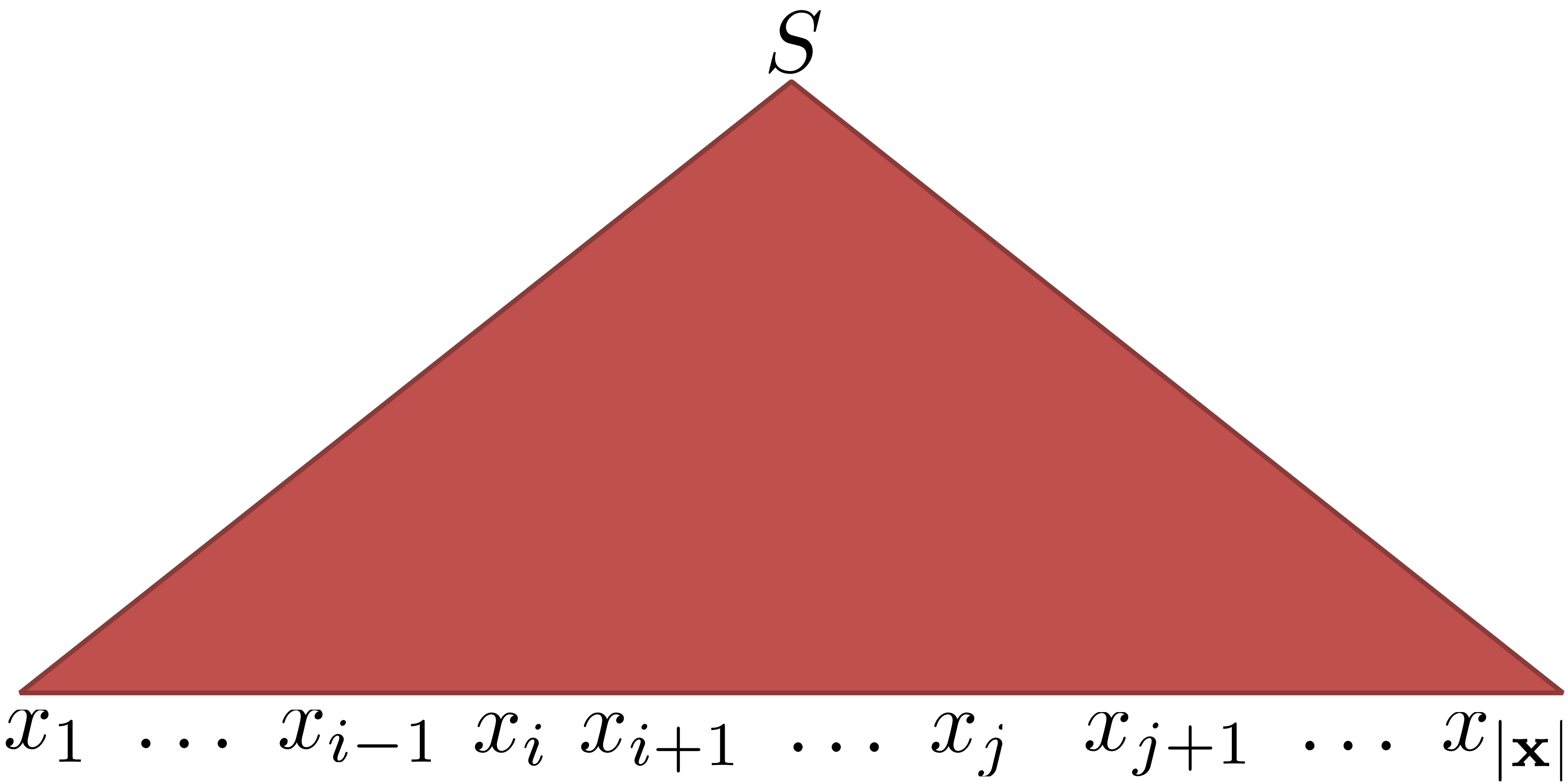
Inside Algorithm

$$\alpha_{[i,j]}(N) = p(x_i, x_{i+1}, \ldots, x_j \mid N; \mathcal{G})$$



Inside Algorithm

$$\alpha_{[i,j]}(N) = p(x_i, x_{i+1}, \ldots, x_j \mid N; \mathcal{G})$$



CKY Inside Algorithm

Base case(s)

$$\alpha_{[i,i+1]}(Z) = p(Z \to x_i)$$

Recurrence

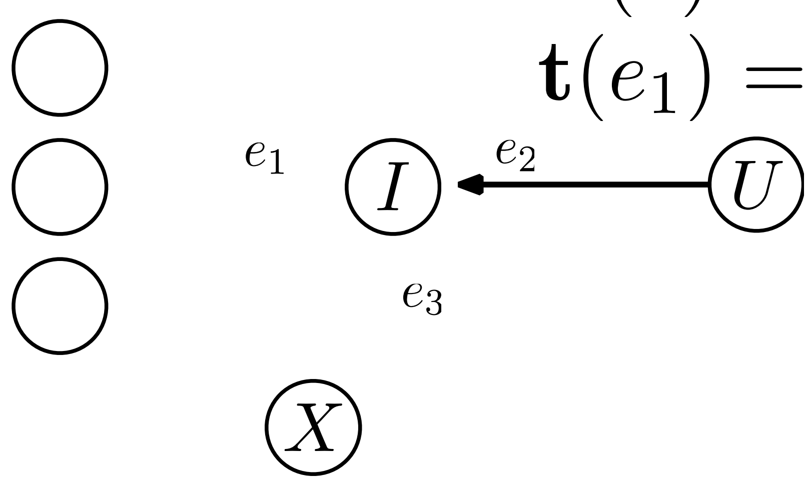
$$\alpha_{[i,j]}(Z) = \sum_{k=i+1}^{j-1} \sum_{(Z \to XY) \in G} \alpha_{[i,k]}(X) \times \alpha_{[k,j]}(Y) \times p(Z \to XY)$$

Generic Inside

```
function INSIDE(G, K)                              ▷ G is an acyclic hypergraph and K is a semiring
    for q in topological order in G do
        if B(q) = ∅ then
            α(q) ← 1̄                               ▷ assume states with no in-edges are axioms
        else
            α(q) ← 0̄
            for all e ∈ B(q) do                    ▷ all in-coming edges to node q
                k ← w(e)
                for all r ∈ t(e) do                ▷ all tail (previous) nodes of edge e
                    k ← k ⊗ α(r)
                α(q) ← α(q) ⊕ k
    return α
```

$$B(I) = \{e_1, e_2, e_3\}$$

$$\mathbf{t}(e_1) = \langle A, B, C \rangle$$

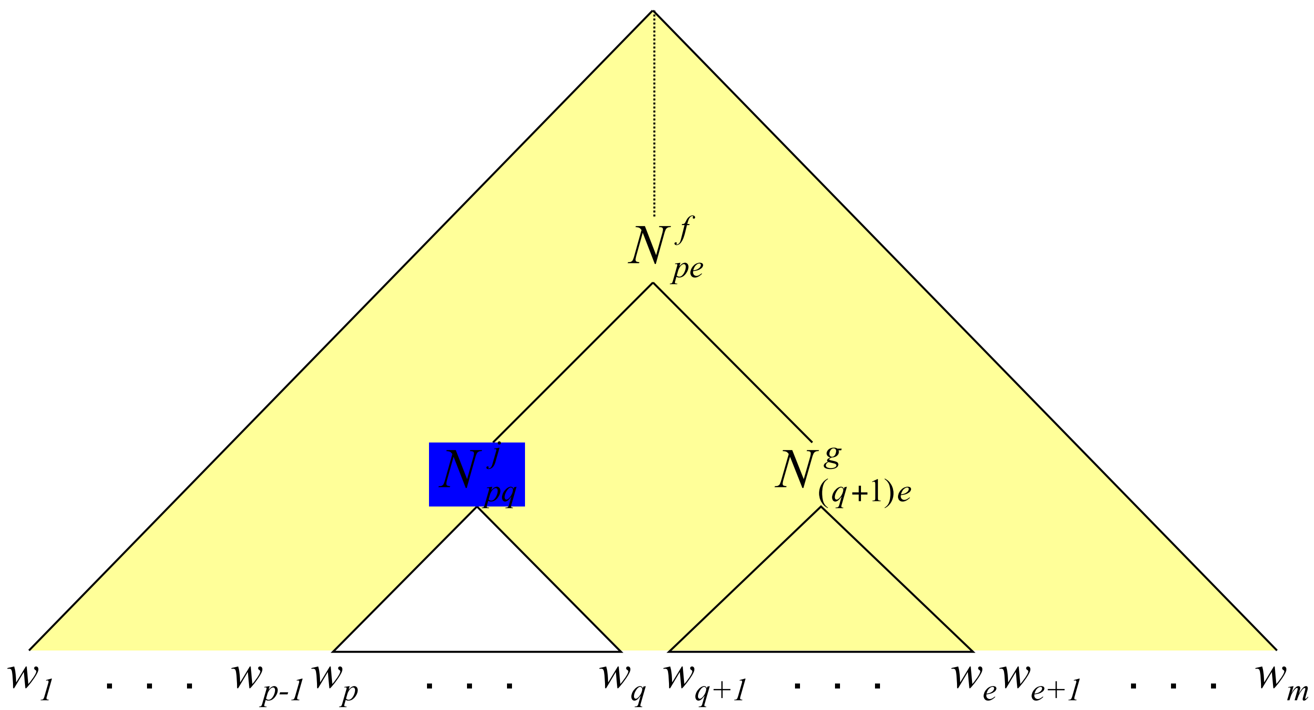


Questions for Generic Inside

- Probability semiring
  - Marginal probability of input
- Counting semiring
  - Number of paths (parses, labels, etc)
- Viterbi semiring
  - Viterbi probability (max joint probability)
- Log semiring
  - log Z(input)

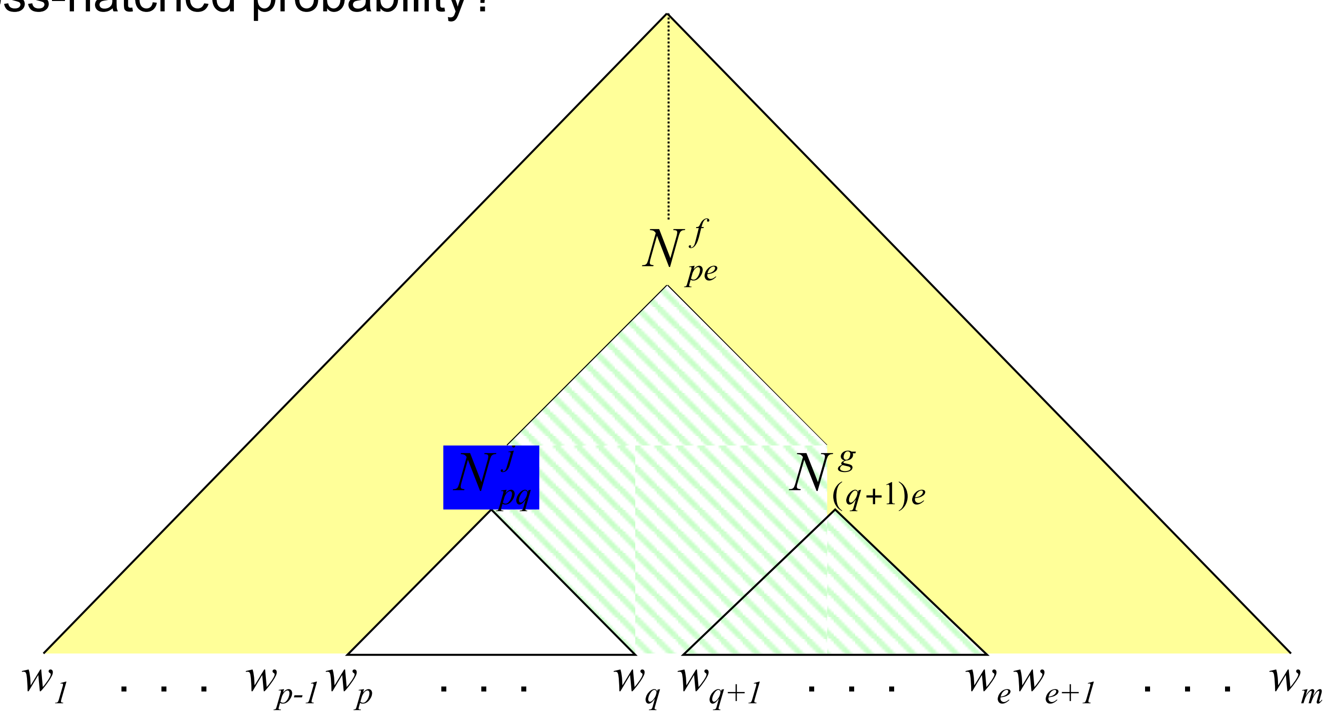
Outside probabilities: decomposing the problem

The shaded area represents the outside probability $\alpha_j(p,q)$ which we need to calculate. How can this be decomposed?



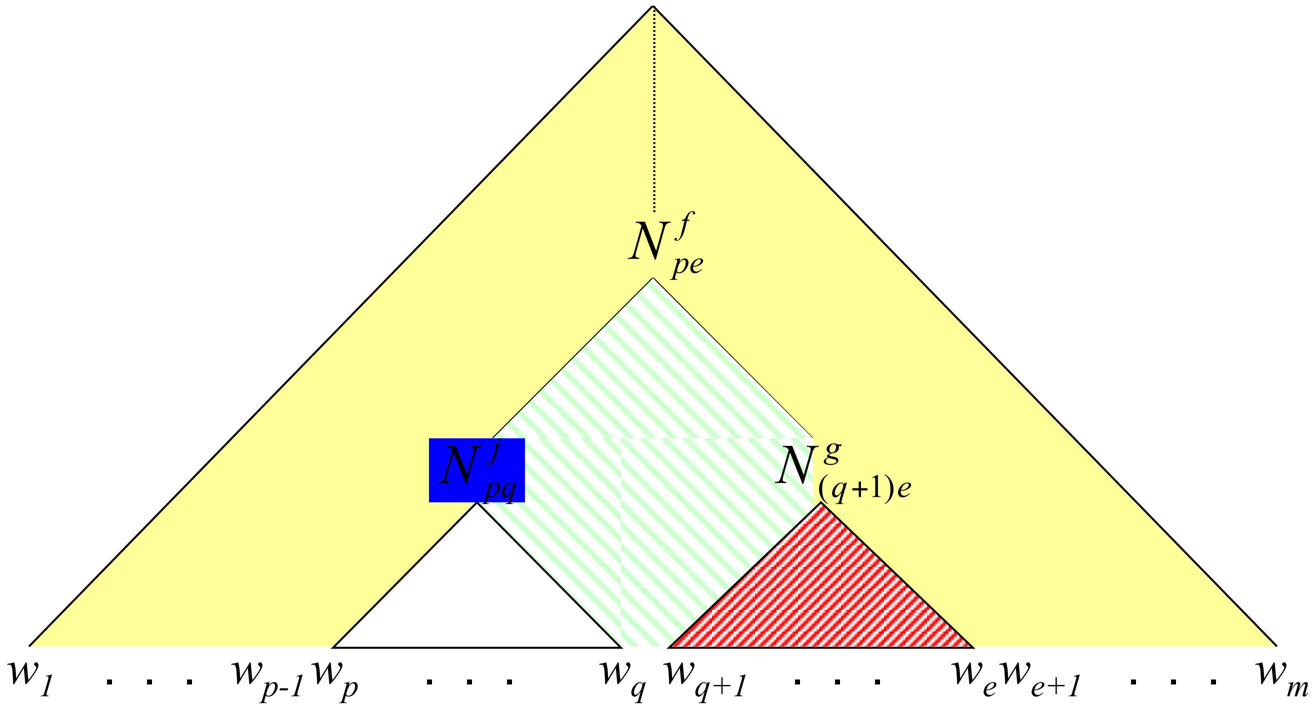
Outside probabilities: decomposing the problem

Step 1: We assume that N^f_{pe} is the parent of N^j_{pq}. Its outside probability, $\alpha_f(p,e)$, (represented by the yellow shading) is available recursively. How do we calculate the cross-hatched probability?



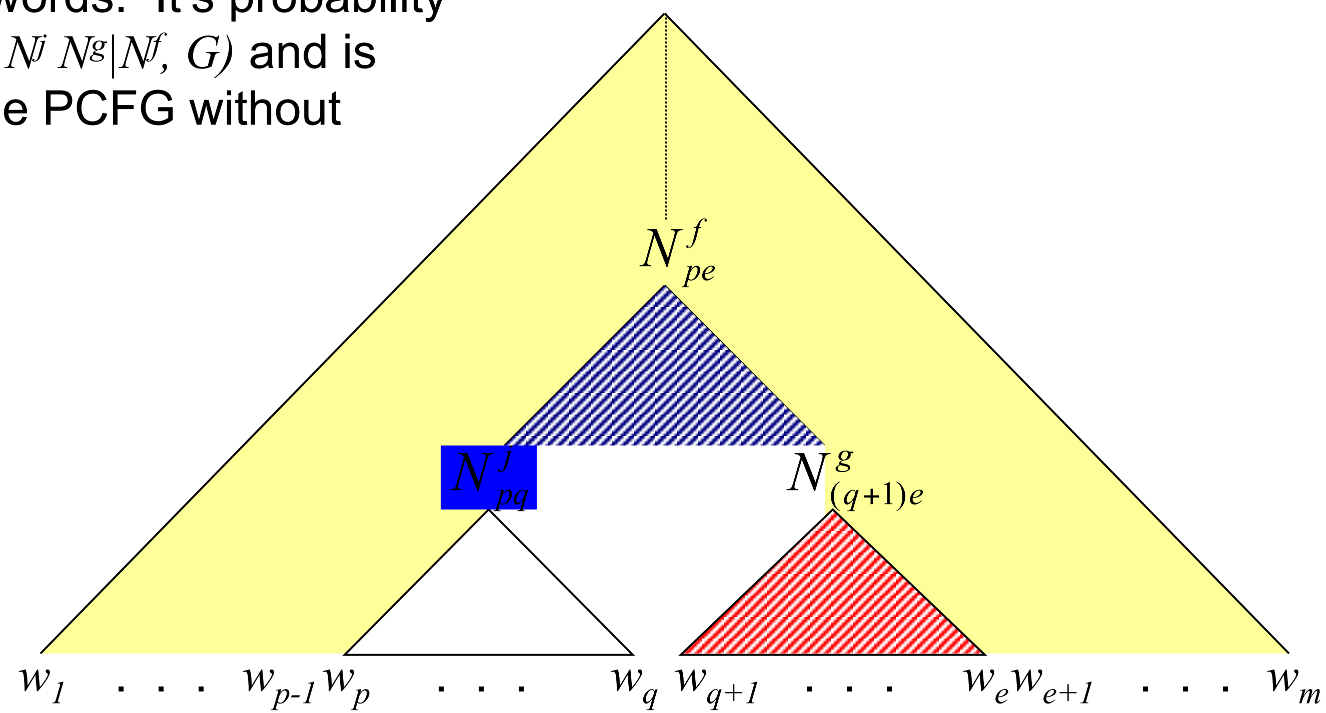
Outside probabilities: decomposing the problem

Step 2: The red shaded area is the inside probability of $N^g_{(q+1)e}$, which is available as $\beta_g(q+1,e)$.



Outside probabilities: decomposing the problem

Step 3: The blue shaded part corresponds to the production $N^f \rightarrow N^j\, N^g$, which because of the context-freeness of the grammar, is not dependent on the positions of the words. It's probability is simply $P(N^f \rightarrow N^j\, N^g | N^f, G)$ and is available from the PCFG without calculation.



Generic Outside

```
function OUTSIDE(G, K, α)                          ▷ α is the result of INSIDE(G, K)
    for all q ∈ G do
        β(q) ← 0̄
    β(q_goal) = 1̄
    for q in reverse topological order in G do
        for all e ∈ B(q) do                        ▷ all in-coming edges to node q
            for all r ∈ t(e) do                    ▷ all tail (previous) nodes of edge e
                k ← w(e) ⊗ β(q)
                for all s ∈ t(e) do                ▷ all tail (previous) nodes of edge e, again
                    if r ≠ s then
                        k ← k ⊗ α(s)               ▷ incorporate inside score
                β(r) ← β(r) ⊕ k
    return β
```

Generic Inside-Outside

function $\textsc{InsideOutside}(\mathcal{G}, K)$ ▷ compute edge marginals

  $\alpha \leftarrow \textsc{Inside}(\mathcal{G}, K)$

  $\beta \leftarrow \textsc{Outside}(\mathcal{G}, K, \alpha)$

  for edge e in $\mathcal{G}$ **do**

    $\gamma(e) \leftarrow w(e) \otimes \beta(n(e))$ ▷ edge weight and outside score of edge's head node

    for all $q \in \mathbf{t}(e)$ **do**

      $\gamma(e) \leftarrow \gamma(e) \otimes \alpha(q)$ ▷ inside score of tail nodes

  return γ ▷ $\gamma(e)$ is the edge marginal of e

Inside-Outside

- Inside probabilities are required to compute Outside probabilities
- Inside-Outside works where Forward-Backward does, but not vice-versa
- Implementation considerations
  - Building a hypergraph explicitly simplifies code, but it can be expensive in terms of memory