Decoding and Inference with Syntactic Translation Models

April 8, 2014

CFGs

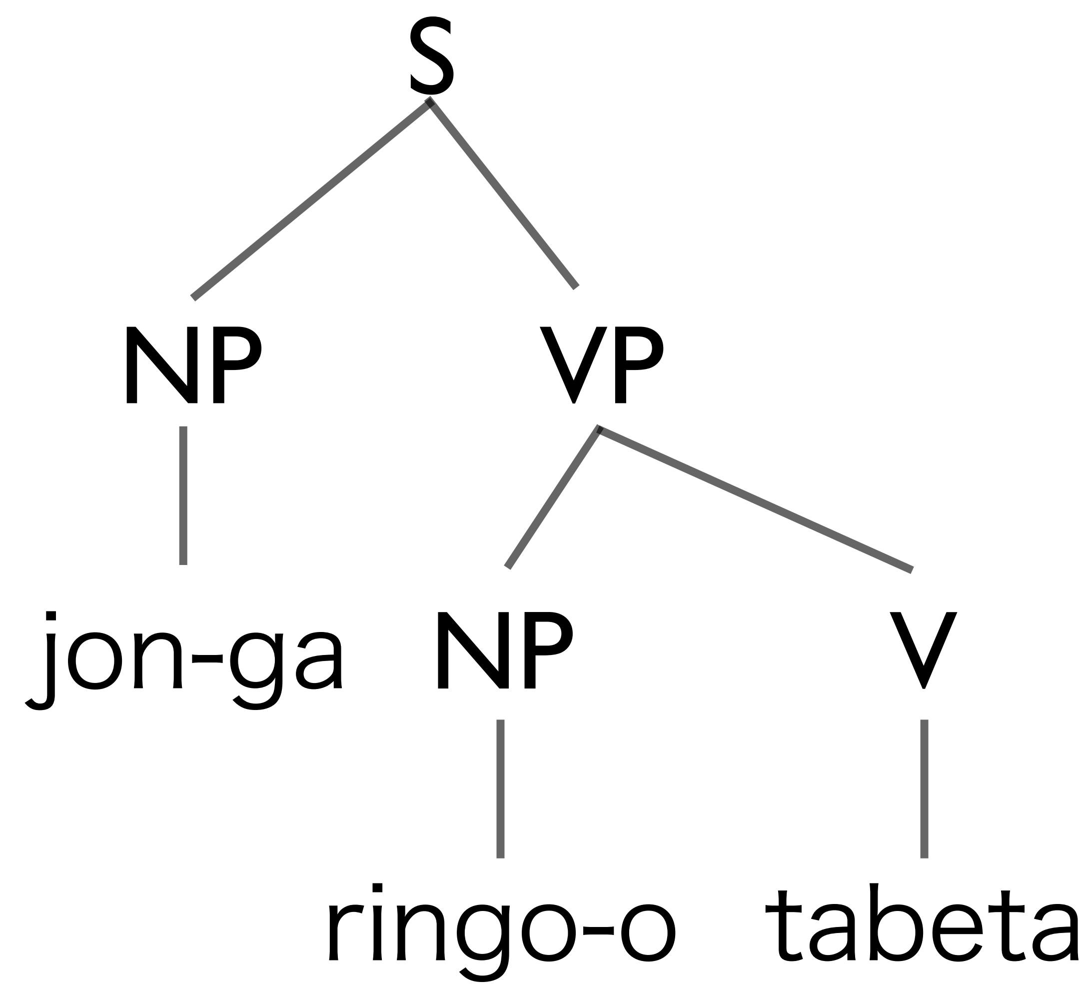
S → NP VP

VP → NP V

V → tabeta

NP → jon-ga

NP → ringo-o



Output: jon-ga ringo-o tabeta

Synchronous CFGs

S → NP VP

VP → NP V

V → tabeta

NP → jon-ga

NP → ringo-o

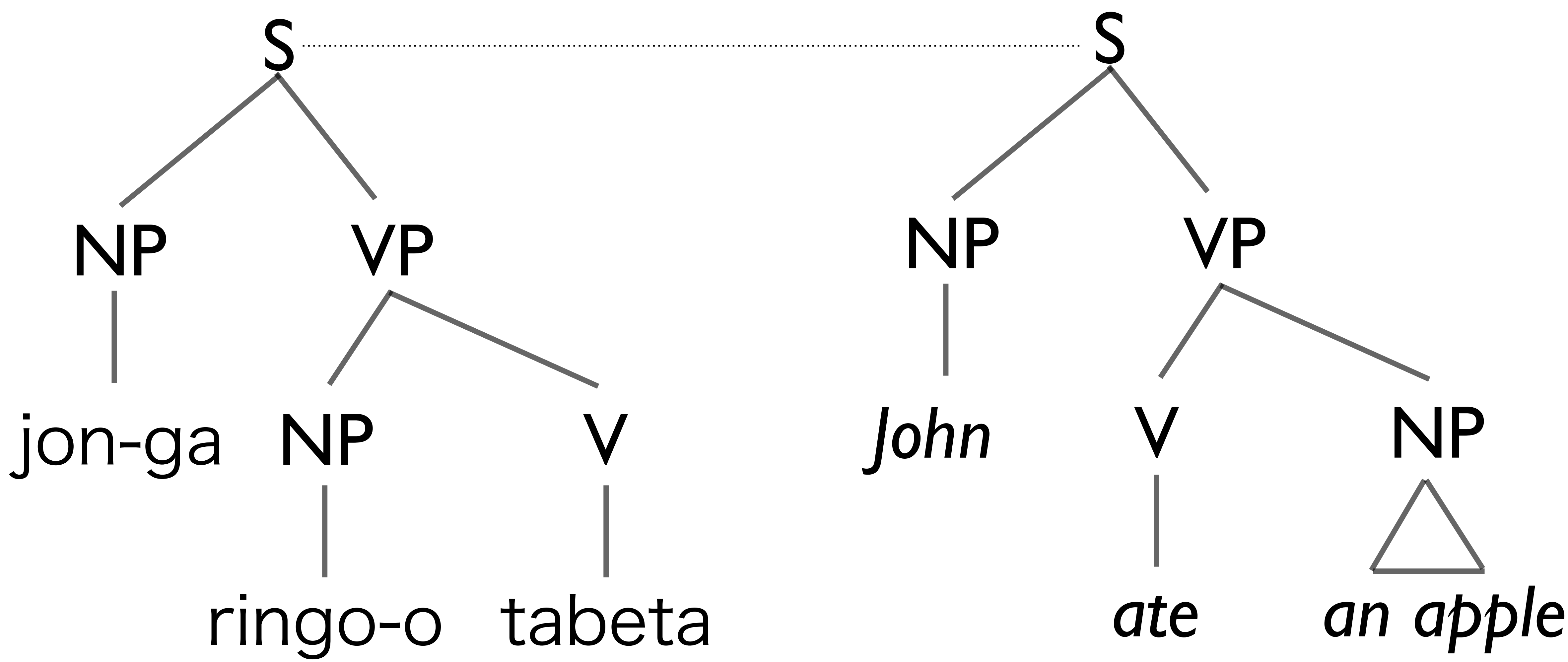
Synchronous CFGs

S	$\rightarrow$	NP VP	:	$\boxed{1}$ $\boxed{2}$	(monotonic)
VP	$\rightarrow$	NP V	:	$\boxed{2}$ $\boxed{1}$	(inverted)
V	$\rightarrow$	tabeta	:	*ate*	
NP	$\rightarrow$	jon-ga	:	*John*	
NP	$\rightarrow$	ringo-o	:	*an apple*	

Synchronous CFGs

S	→	NP VP	:	$\boxed{1}$ $\boxed{2}$	(monotonic)
VP	→	NP V	:	$\boxed{2}$ $\boxed{1}$	(inverted)
V	→	tabeta	:	*ate*	
NP	→	jon-ga	:	*John*	
NP	→	ringo-o	:	*an apple*	

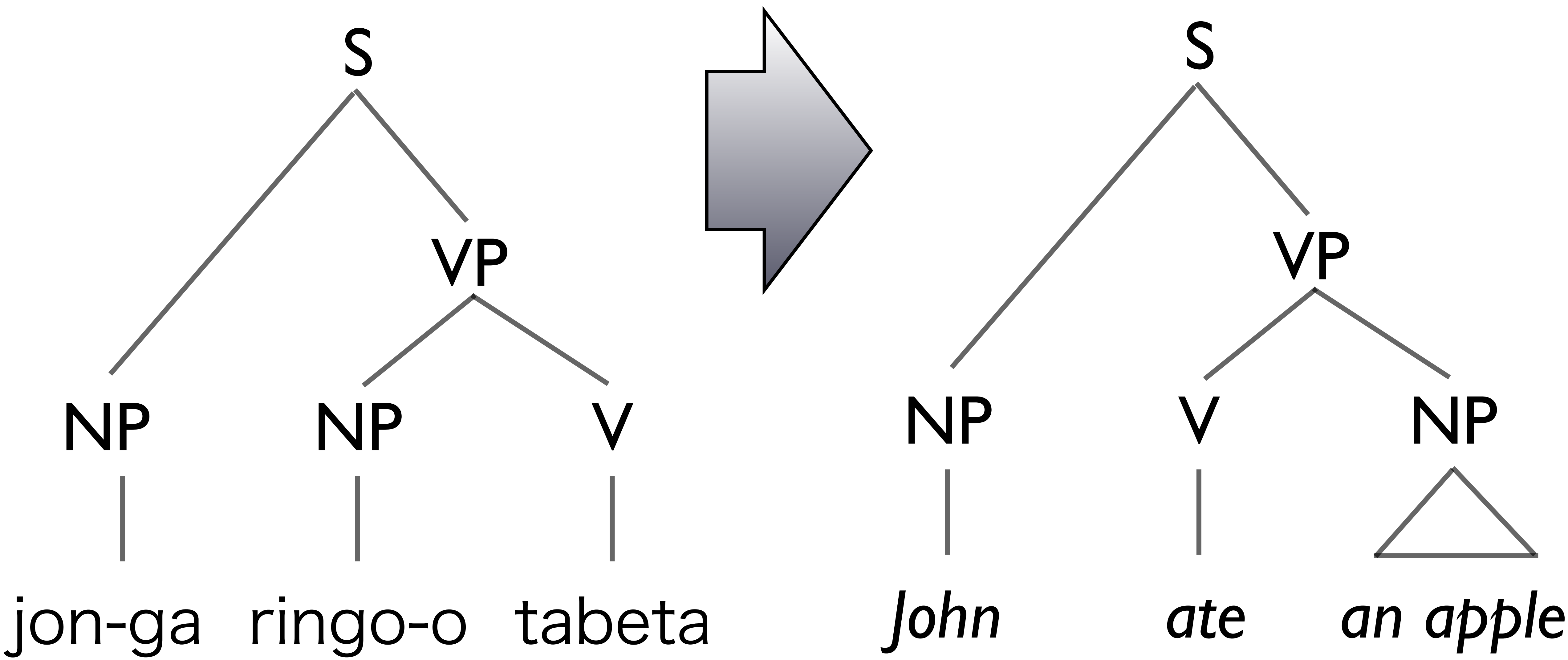
Synchronous generation



Output: (jon-ga ringo-o tabeta : *John ate an apple*)

Translation as parsing

Parse source **Project to target**



A closer look at parsing

- Parsing is usually done with dynamic programming
  - **Share common computations and structure**
  - Represent exponential number of alternatives in polynomial space
  - With SCFGs there are two kinds of ambiguity
    - source parse ambiguity
    - translation ambiguity
    - parse forests can represent both

A closer look at parsing

- Any monolingual parser can be used (most often: CKY / “dotted” CKY variants)
- Parsing complexity is O($|n^3|$)
  - cubic in the length of the sentence (n^3)
  - cubic in the number of non-terminals ($|G|^3$)
    - adding nonterminal types increases parsing complexity substantially!
    - With few NTs, exhaustive parsing is tractable

Parsing as deduction

Antecedents

Side conditions

$$\frac{A : u \quad B : v}{C : w} \quad \phi$$

Consequent

"If A and B are true with weights u and v, and phi is also true, then C is true with weight w."

Example: CKY

Inputs:

$$\mathbf{f} = \langle f_1, f_2, \ldots, f_\ell \rangle$$

G Context-free grammar in Chomsky normal form.

Item form:

$[X, i, j]$ A subtree rooted with NT type X spanning i to j has been recognized.

Example: CKY

Goal:

$$[S, 0, \ell]$$

Axioms:

$$\frac{}{[X, i-1, i] : w} \quad (X \xrightarrow{w} f_i) \in G$$

Inference rules:

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

S → PRP VP
VP → V NP
VP → V SBAR
SBAR → PRP V
NP → PRP NN
V → saw
NN → duck
V → duck
PRP → I
PRP → her

I **saw** **her** **duck**

0 1 2 3 4

S → PRP VP
VP → V NP
VP → V SBAR
SBAR → PRP V
NP → PRP NN
V → saw
NN → duck
V → duck
PRP → I
PRP → her

PRP,0,1

I **saw** **her** **duck**

0 1 2 3 4

$S \rightarrow PRP\ VP$

$VP \rightarrow V\ NP$

$VP \rightarrow V\ SBAR$

$SBAR \rightarrow PRP\ V$

$NP \rightarrow PRP\ NN$

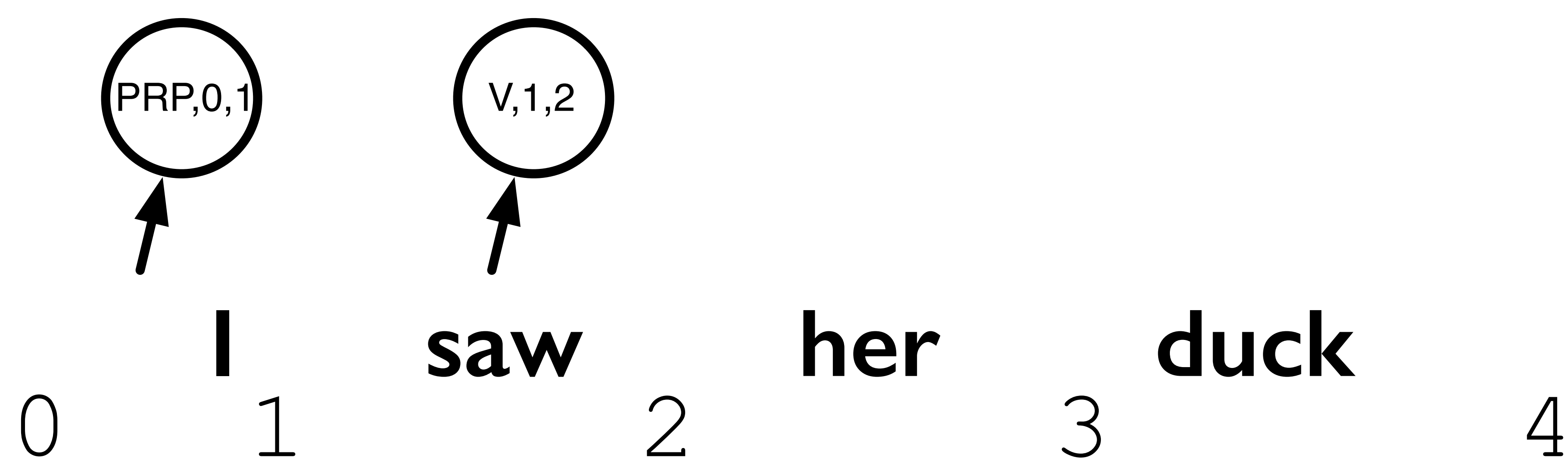
$V \rightarrow$ saw

$NN \rightarrow$ duck

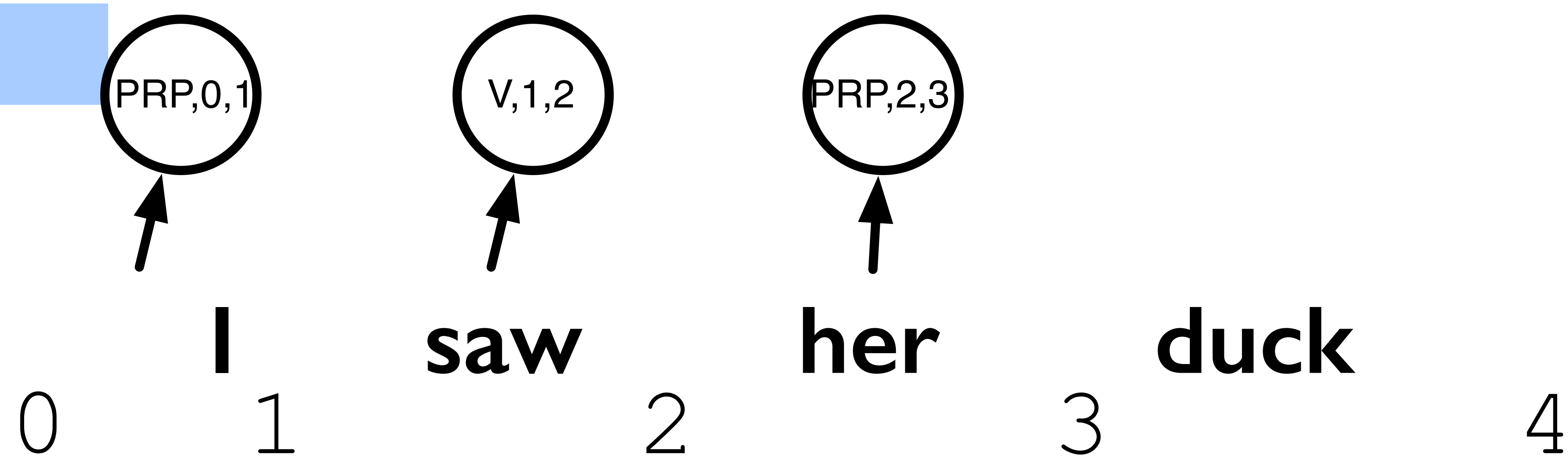
$V \rightarrow$ duck

$PRP \rightarrow$ I

$PRP \rightarrow$ her



S → PRP VP
VP → V NP
VP → V SBAR
SBAR → PRP V
NP → PRP NN
V → saw
NN → duck
V → duck
PRP → I
PRP → her



S → PRP VP

VP → V NP

VP → V SBAR

SBAR → PRP V

NP → PRP NN

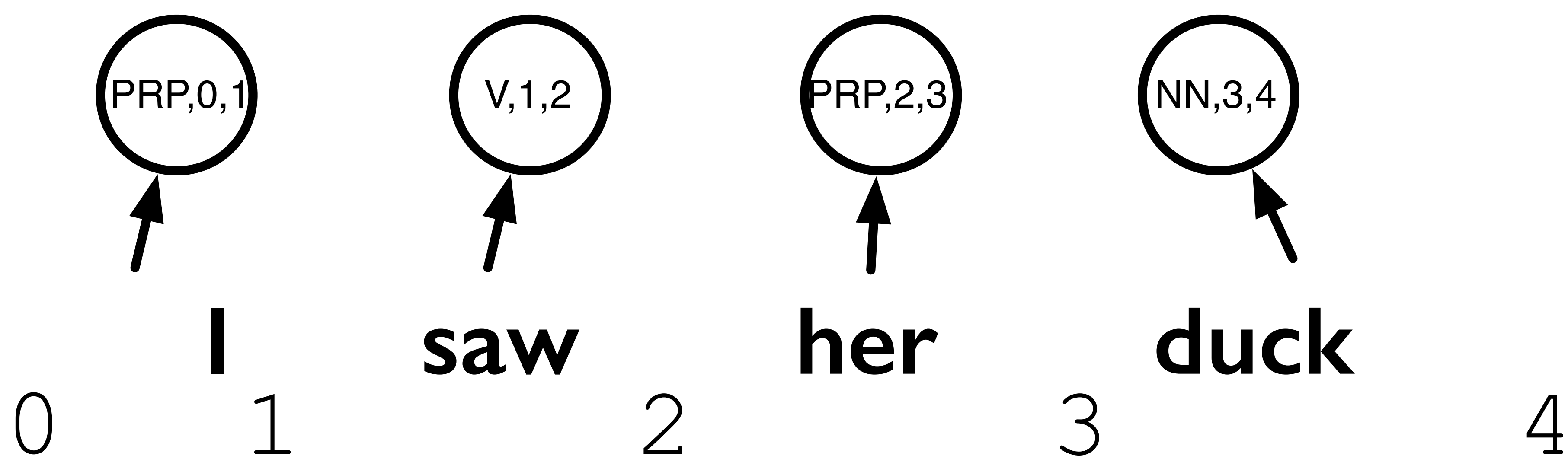
V → saw

NN → duck

V → duck

PRP → I

PRP → her



S → PRP VP

VP → V NP

VP → V SBAR

SBAR → PRP V

NP → PRP NN

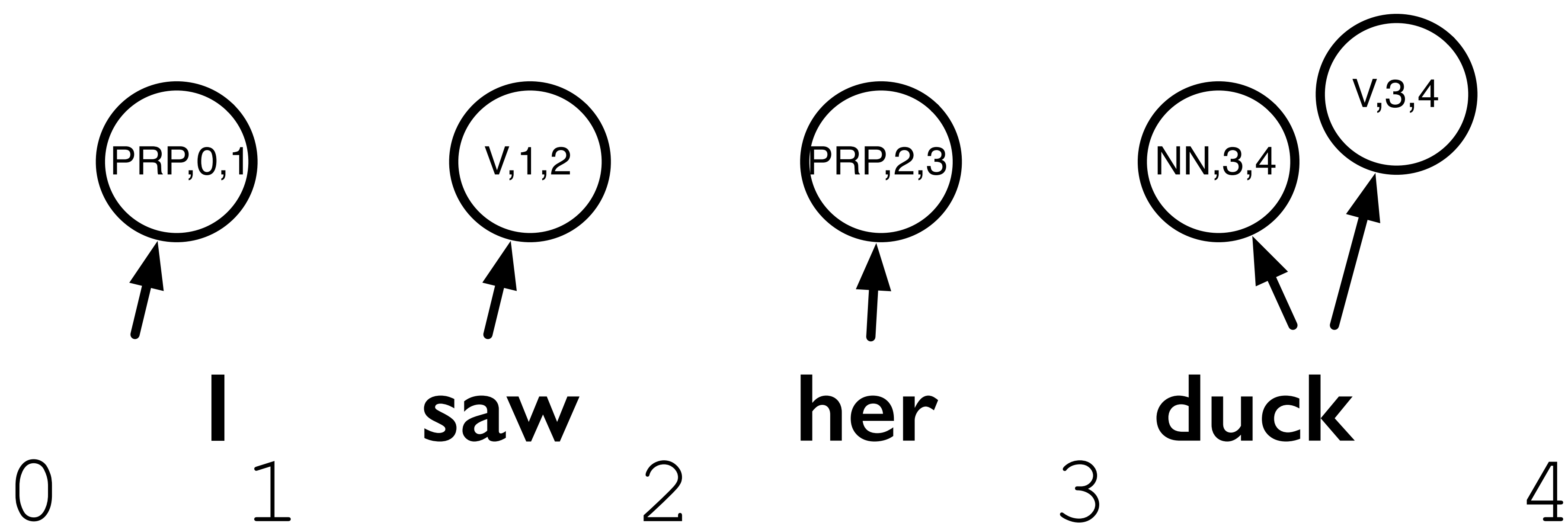
V → saw

NN → duck

V → duck

PRP → I

PRP → her



S → PRP VP

VP → V NP

VP → V SBAR

SBAR → PRP V

NP → PRP NN

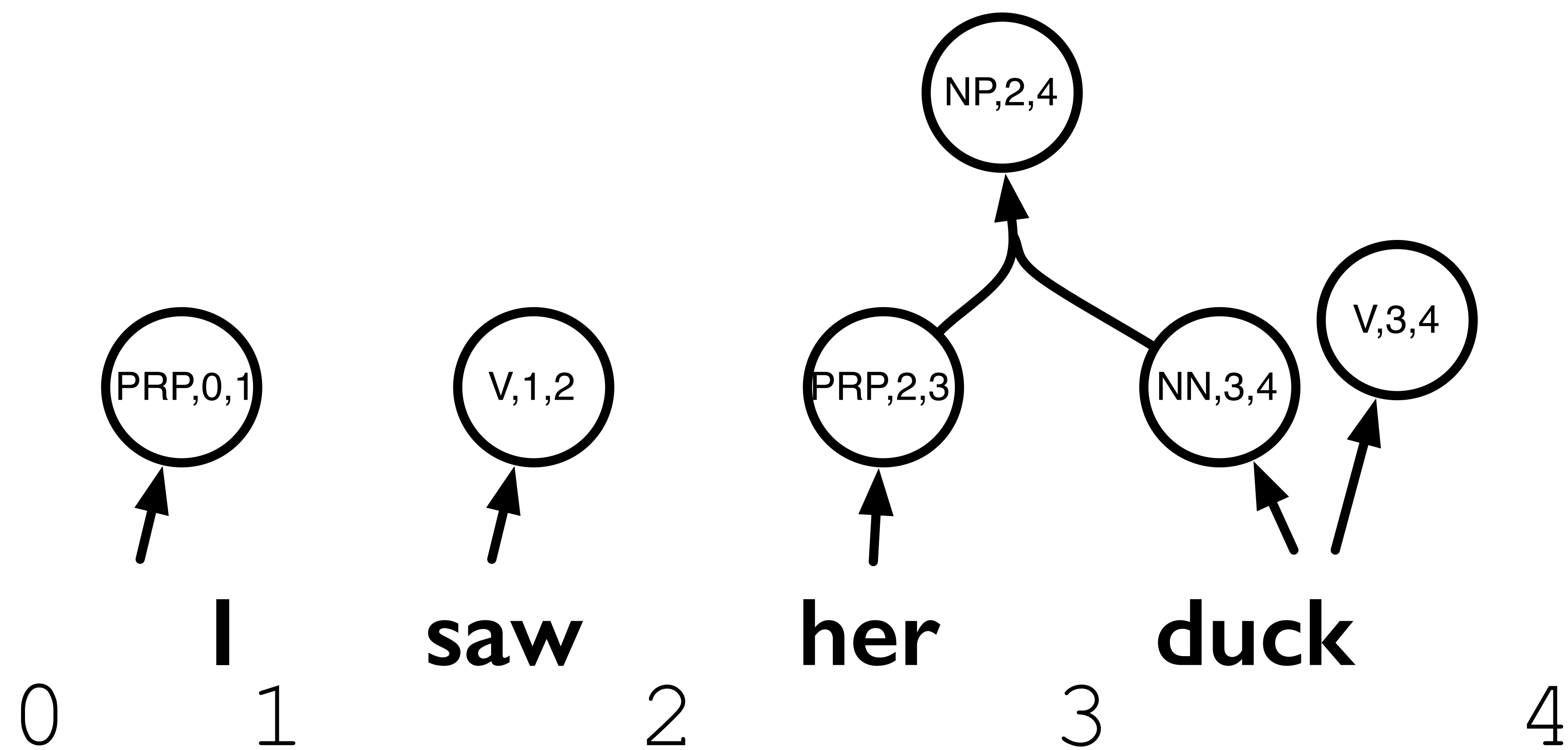
V → saw

NN → duck

V → duck

PRP → I

PRP → her



S → PRP VP

VP → V NP

VP → V SBAR

SBAR → PRP V

NP → PRP NN

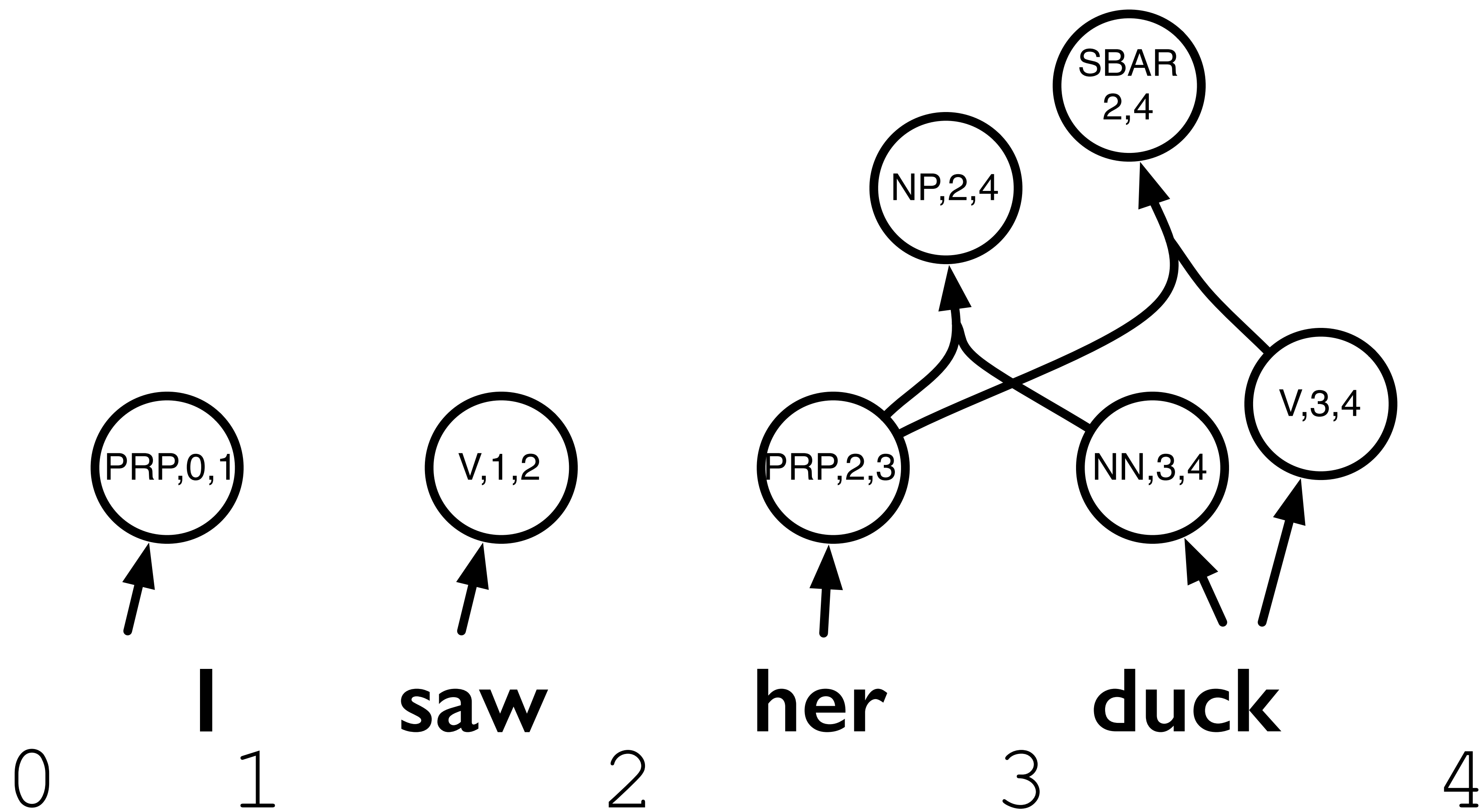
V → saw

NN → duck

V → duck

PRP → I

PRP → her



$S \rightarrow PRP\ VP$

$VP \rightarrow V\ NP$

$VP \rightarrow V\ SBAR$

$SBAR \rightarrow PRP\ V$

$NP \rightarrow PRP\ NN$

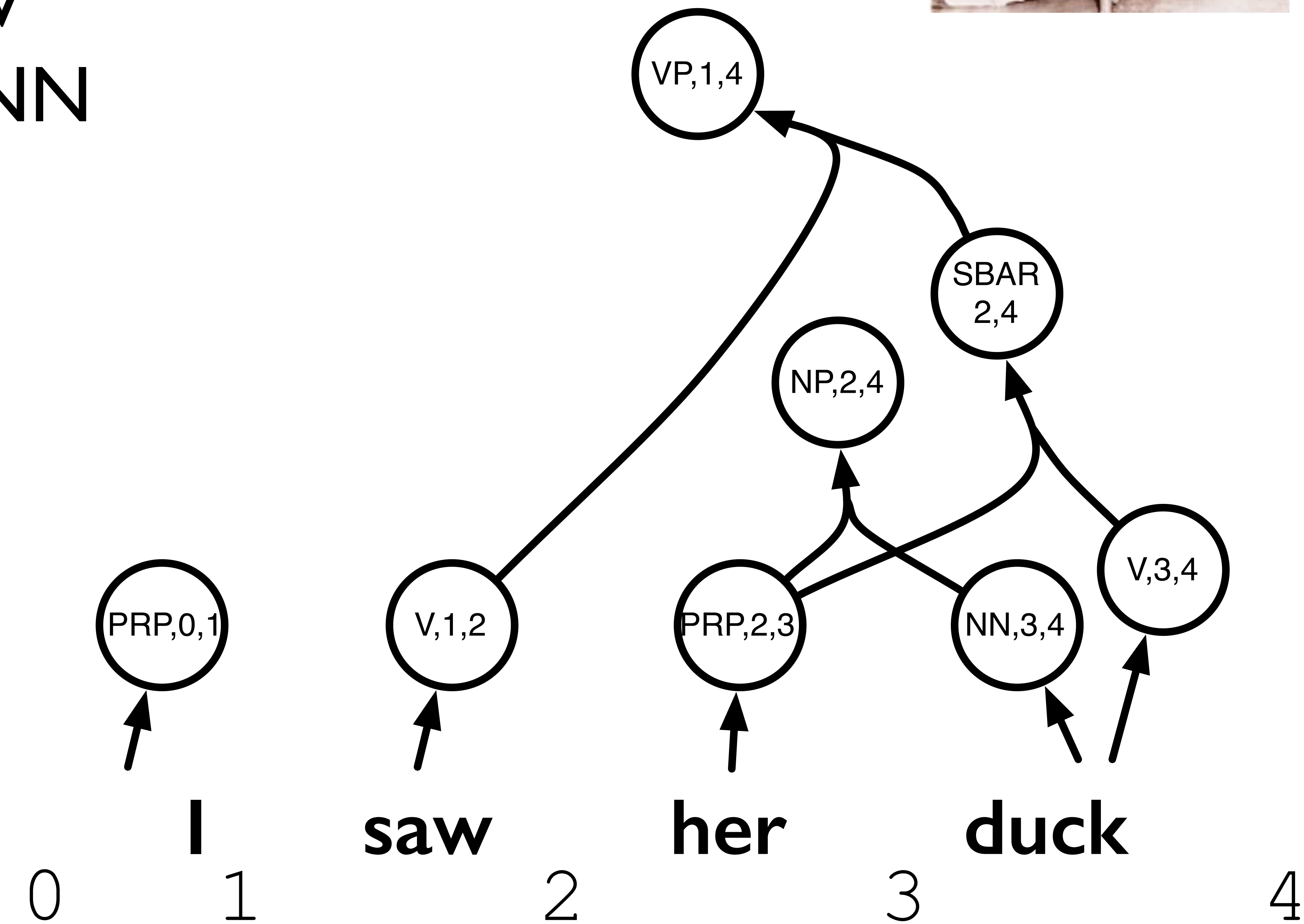
$V \rightarrow$ saw

$NN \rightarrow$ duck

$V \rightarrow$ duck

$PRP \rightarrow$ I

$PRP \rightarrow$ her



S → PRP VP

VP → V NP

VP → V SBAR

SBAR → PRP V

NP → PRP NN

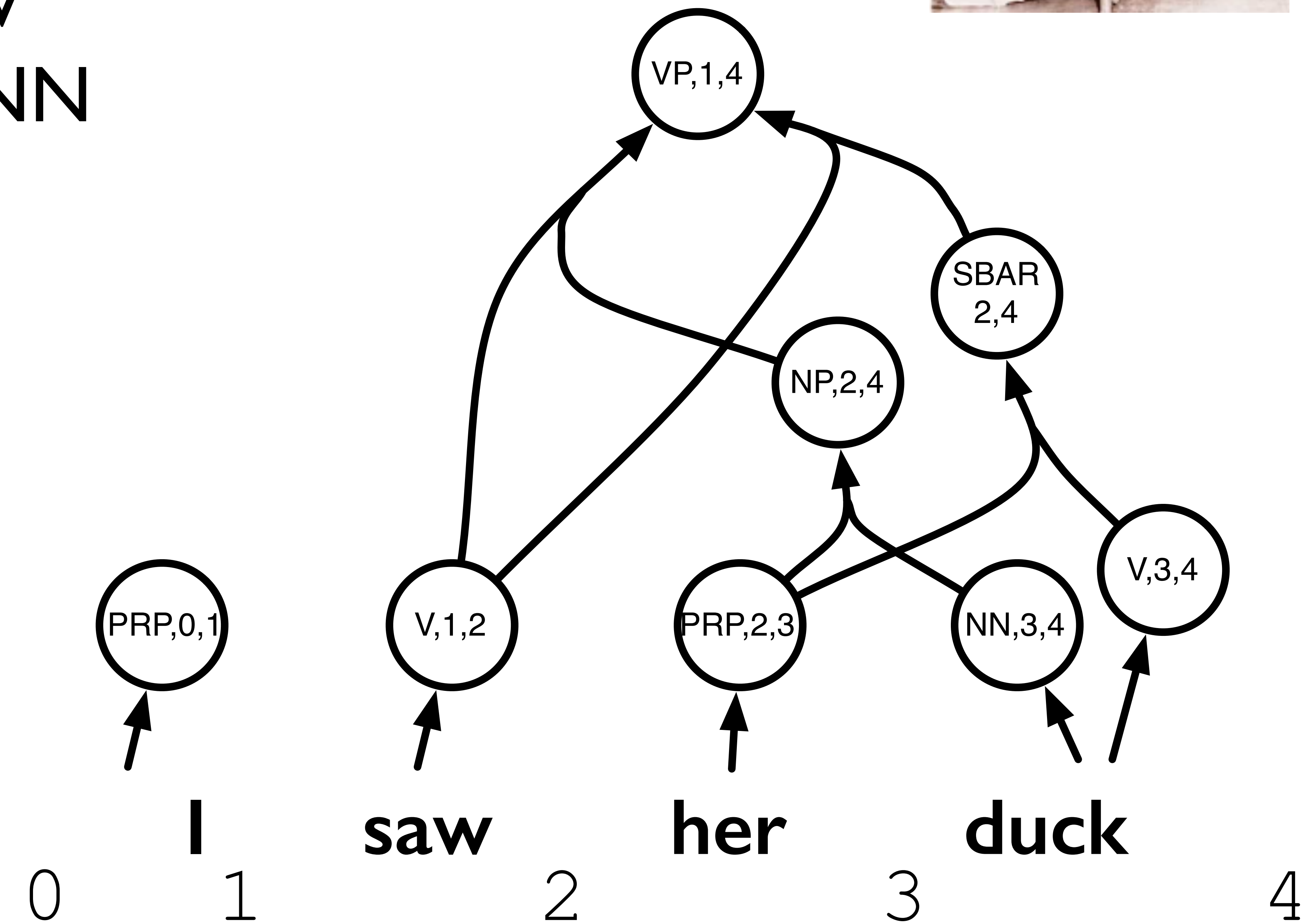
V → saw

NN → duck

V → duck

PRP → I

PRP → her



S → PRP VP

VP → V NP

VP → V SBAR

SBAR → PRP V

NP → PRP NN

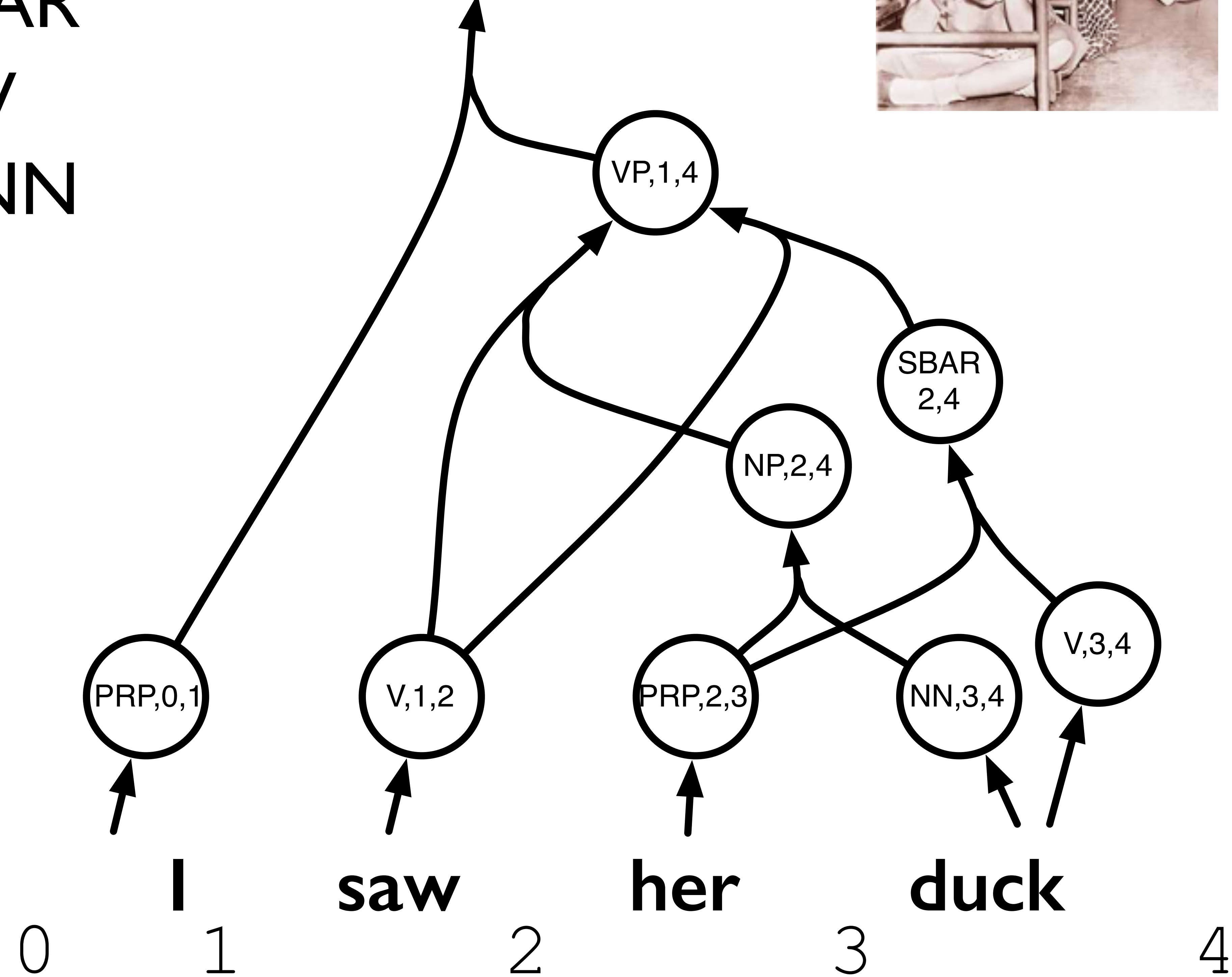
V → saw

NN → duck

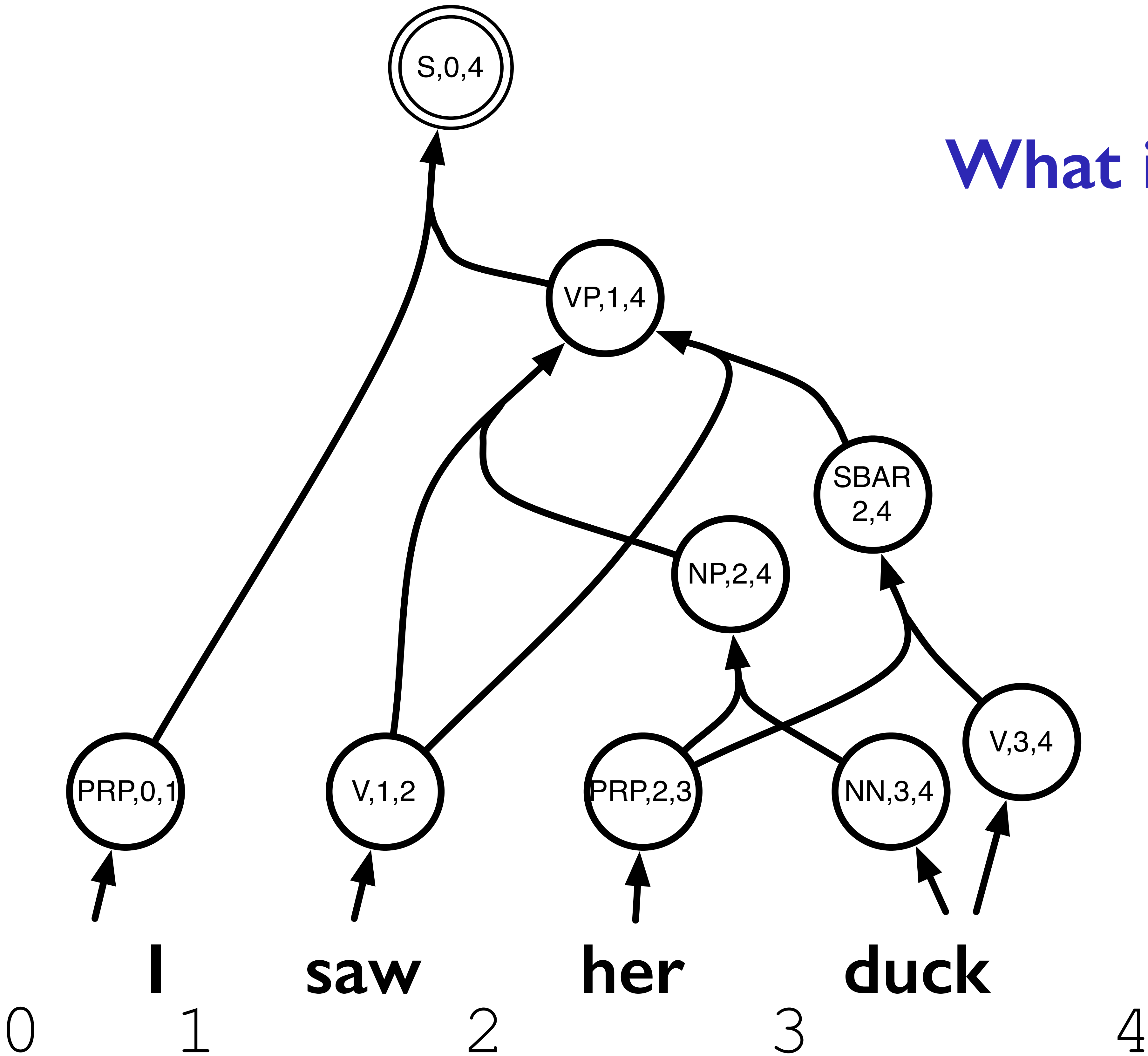
V → duck

PRP → I

PRP → her



What is this object?



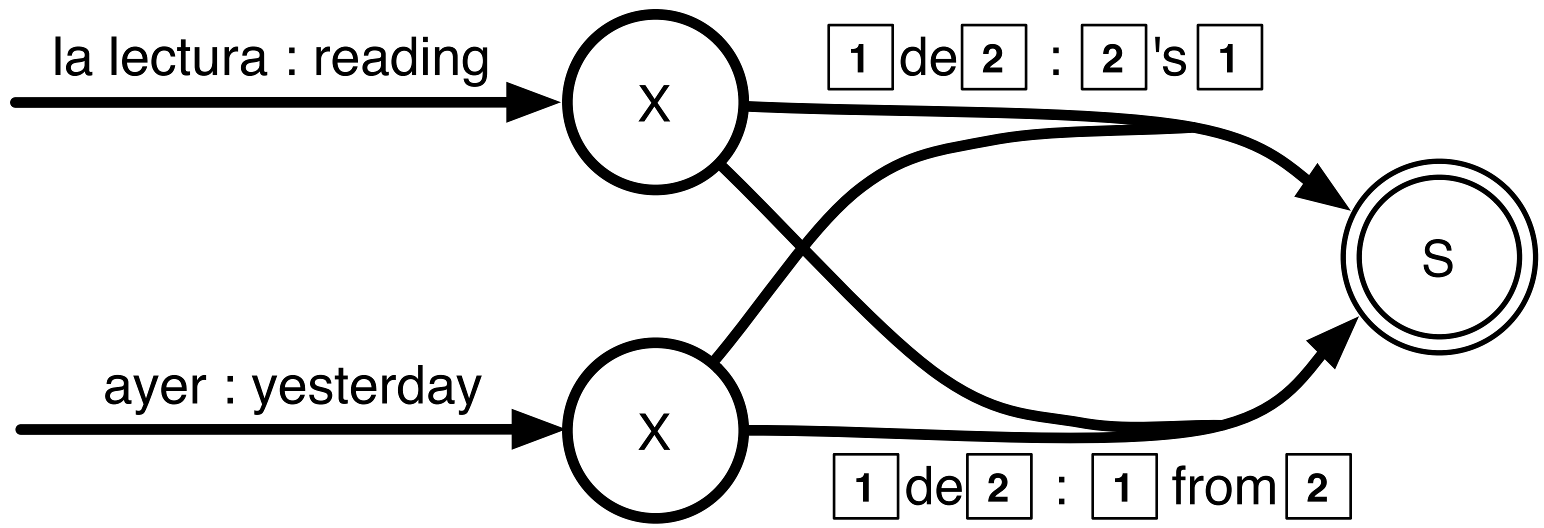
Semantics of hypergraphs

- Generalization of directed graphs
- Special node designated the “goal”
- Every edge has a single head and 0 or more tails (the **arity** of the edge is the number of tails)
- Node labels correspond to LHS’s of CFG rules
- A **derivation** is the generalization of the graph concept of **path** to hypergraphs
- Weights multiply along edges in the derivation, and add at nodes (cf. **semiring parsing**)

Edge labels

- Edge labels may be a mix of terminals and substitution sites (non-terminals)
- In translation hypergraphs, edges are labeled in both the source and target languages
- The number of substitution sites must be equal to the arity of the edge and must be the same in both languages
- The two languages may have different orders of the substitution sites
- There is no restriction on the number of terminal symbols

Edge labels



$\{($ la lectura de ayer : *yesterday 's reading* $),$
$($ la lectura de ayer : *reading from yesterday* $)\}$

A Lingua Franca for MT

- Translation hypergraphs are a *lingua franca* for translation search spaces
  - Note that FST lattices are a special case
- **Decoding problem: how do I build a translation hypergraph?**
  - **For SCFG-translation: just parse**

Tree-to-string Translation

- How do we generate a hypergraph for a tree-to-string translation model?
  - Simple linear-time (given a fixed translation model) top-down matching algorithm
    - Recursively cover "uncovered" sites in tree
  - Each node in the input tree becomes a node in the translation forest
  - For details, Huang et al. (AMTA, 2006) and Huang et al. (EMNLP, 2010)

$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \to x_1\ x_2$$

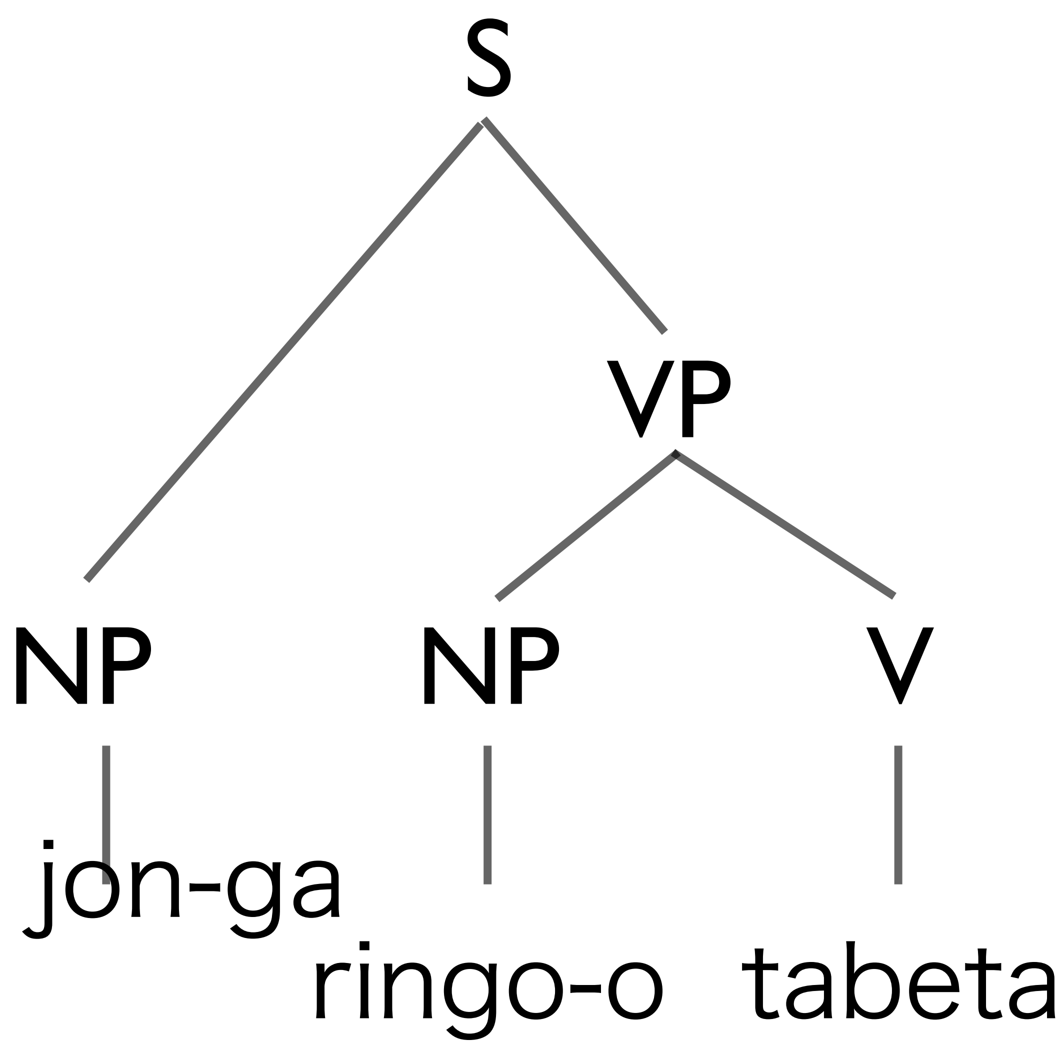
$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \to x_2\ x_1$$

$$tabeta \to ate$$

$$ringo\text{-}o \to an\ apple$$

$$jon\text{-}ga \to John$$

Tree-to-string grammar



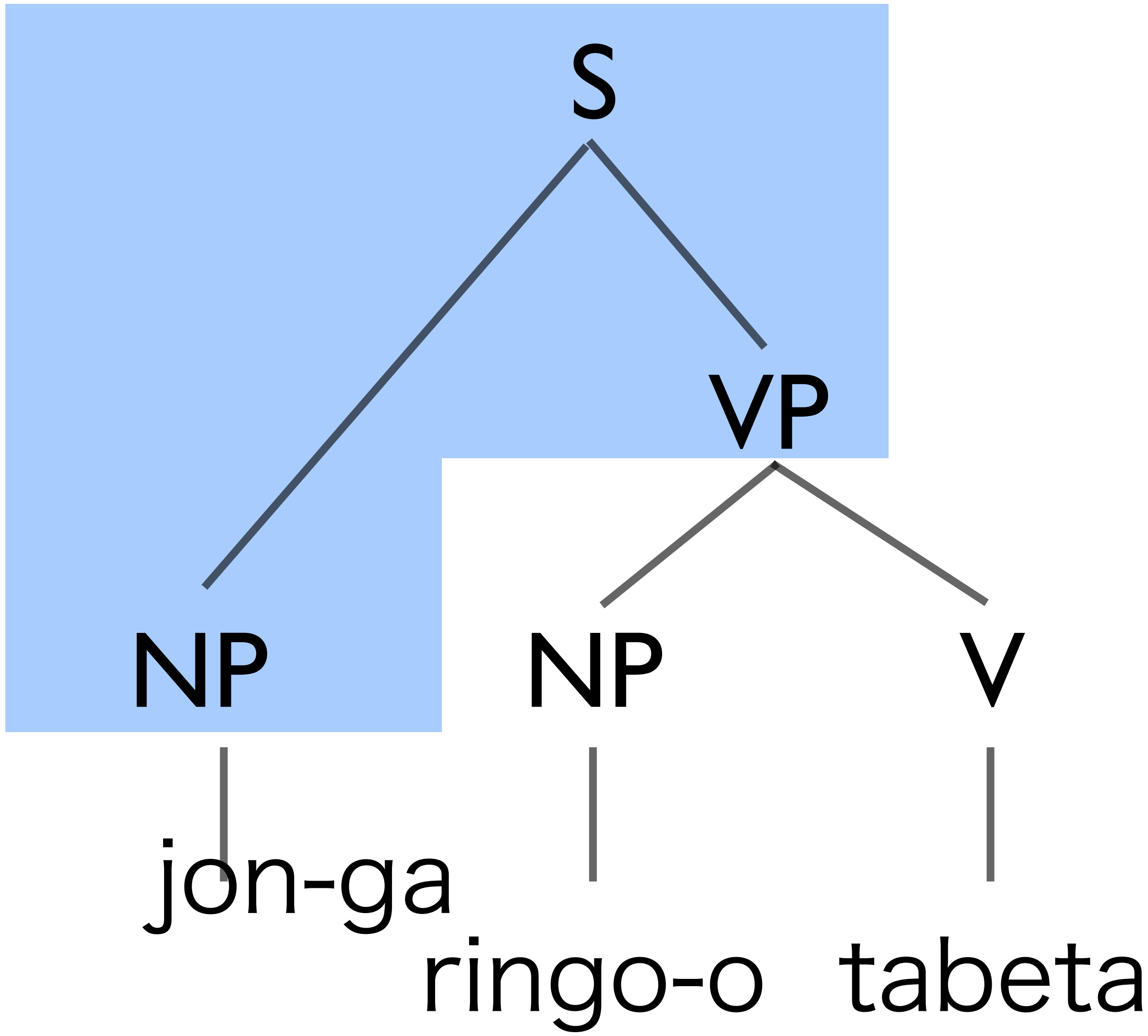
$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \rightarrow x_1\ x_2$$

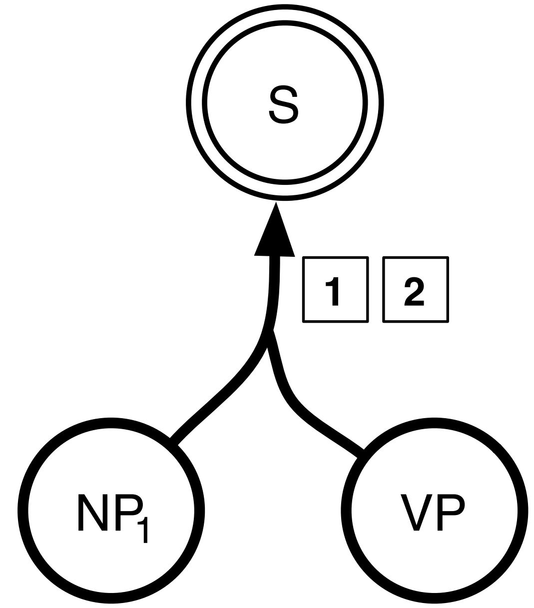
$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \rightarrow x_2\ x_1$$

$$tabeta \rightarrow ate$$

$$ringo\text{-}o \rightarrow an\ apple$$

$$jon\text{-}ga \rightarrow John$$





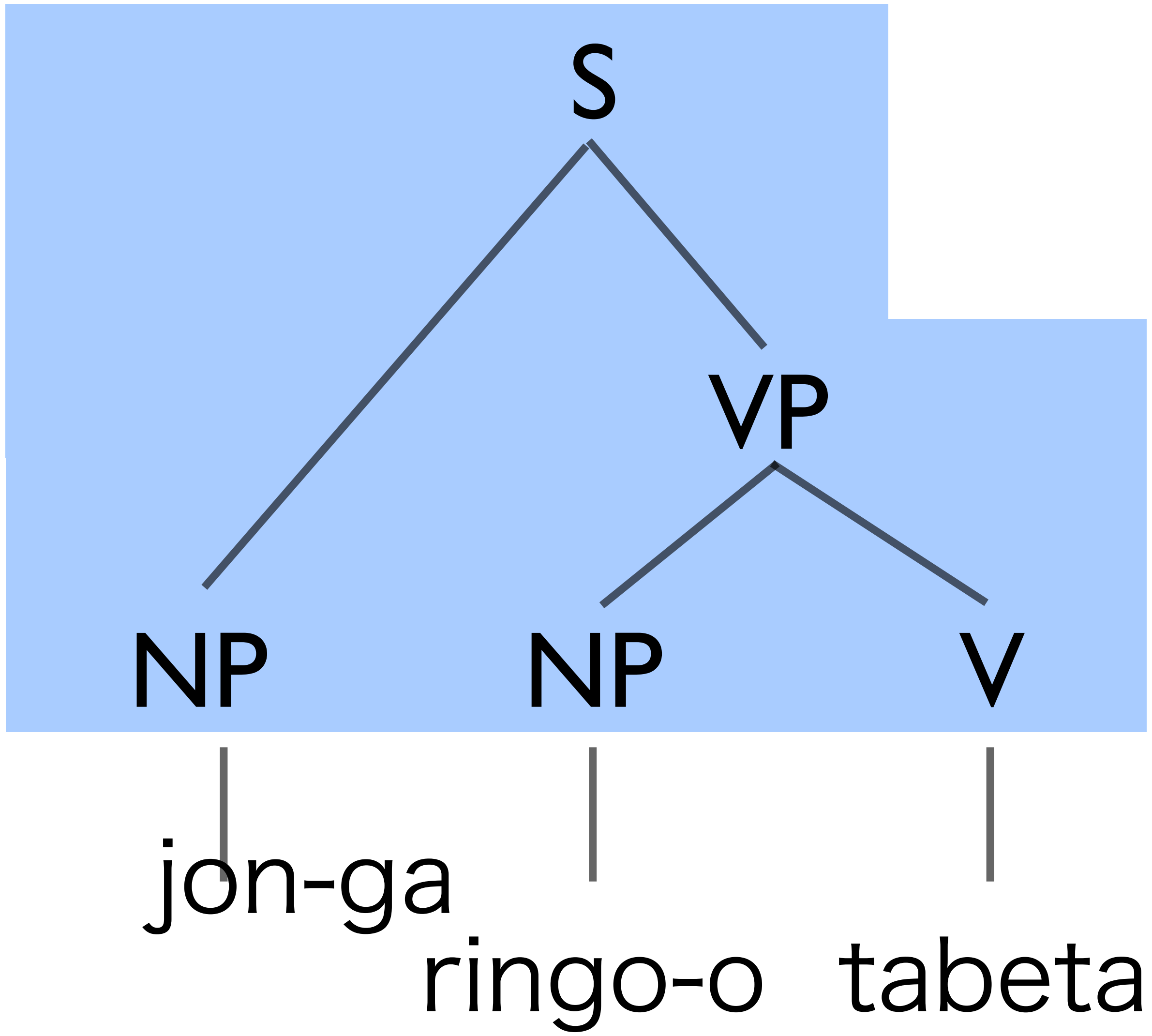
$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \rightarrow x_1\ x_2$$

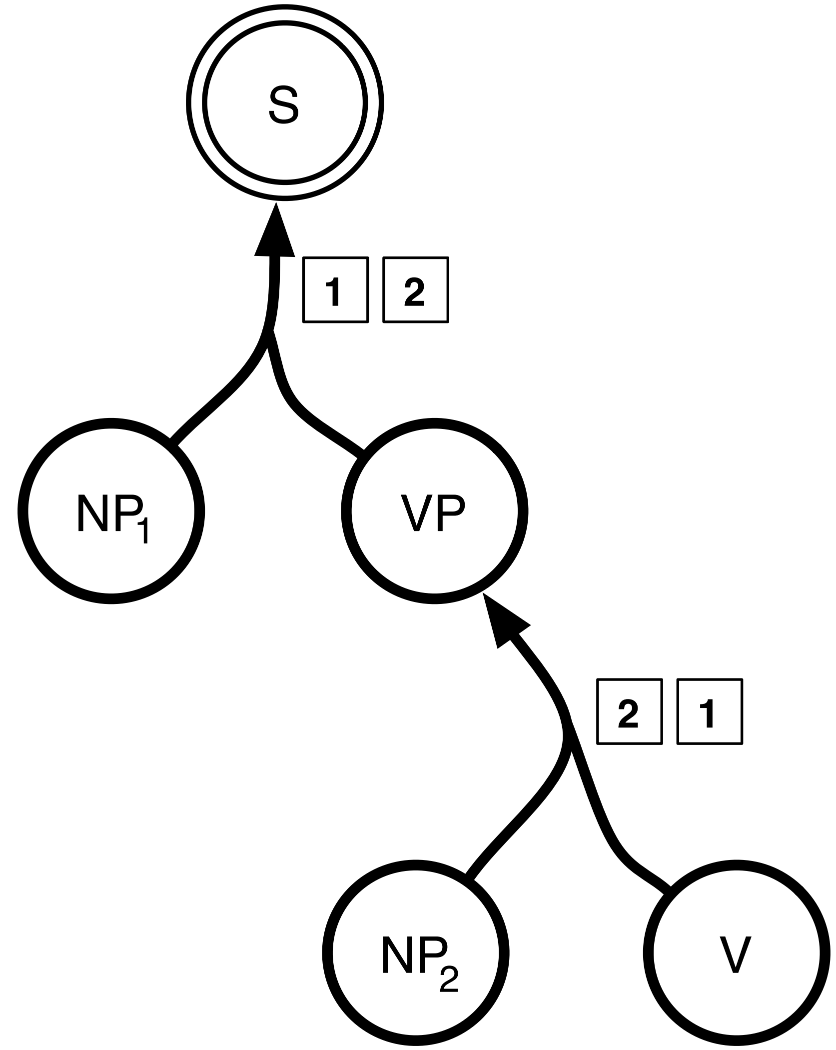
$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \rightarrow x_2\ x_1$$

$$\textit{tabeta} \rightarrow \textit{ate}$$

$$\textit{ringo-o} \rightarrow \textit{an apple}$$

$$\textit{jon-ga} \rightarrow \textit{John}$$





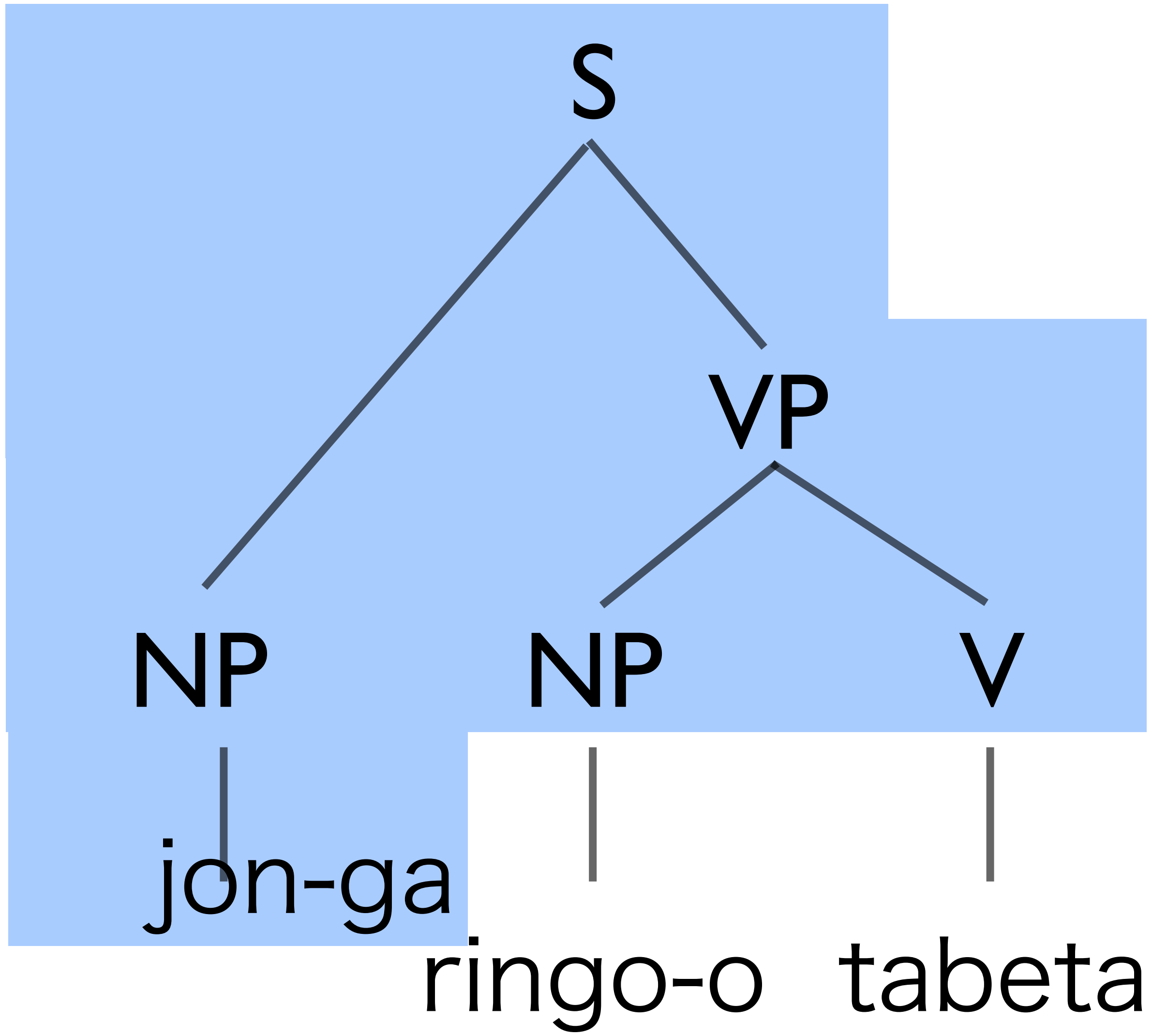
$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \to x_1\ x_2$$

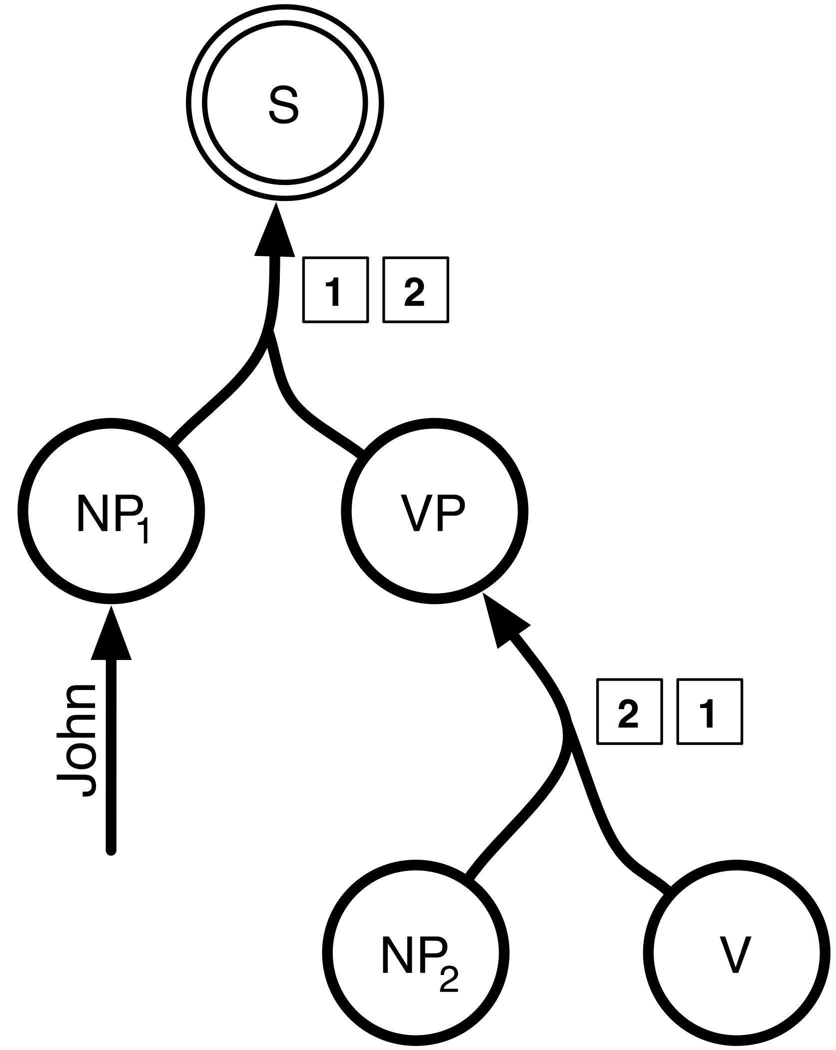
$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \to x_2\ x_1$$

$$\textit{tabeta} \to \textit{ate}$$

$$\textit{ringo-o} \to \textit{an apple}$$

$$\textit{jon-ga} \to \textit{John}$$





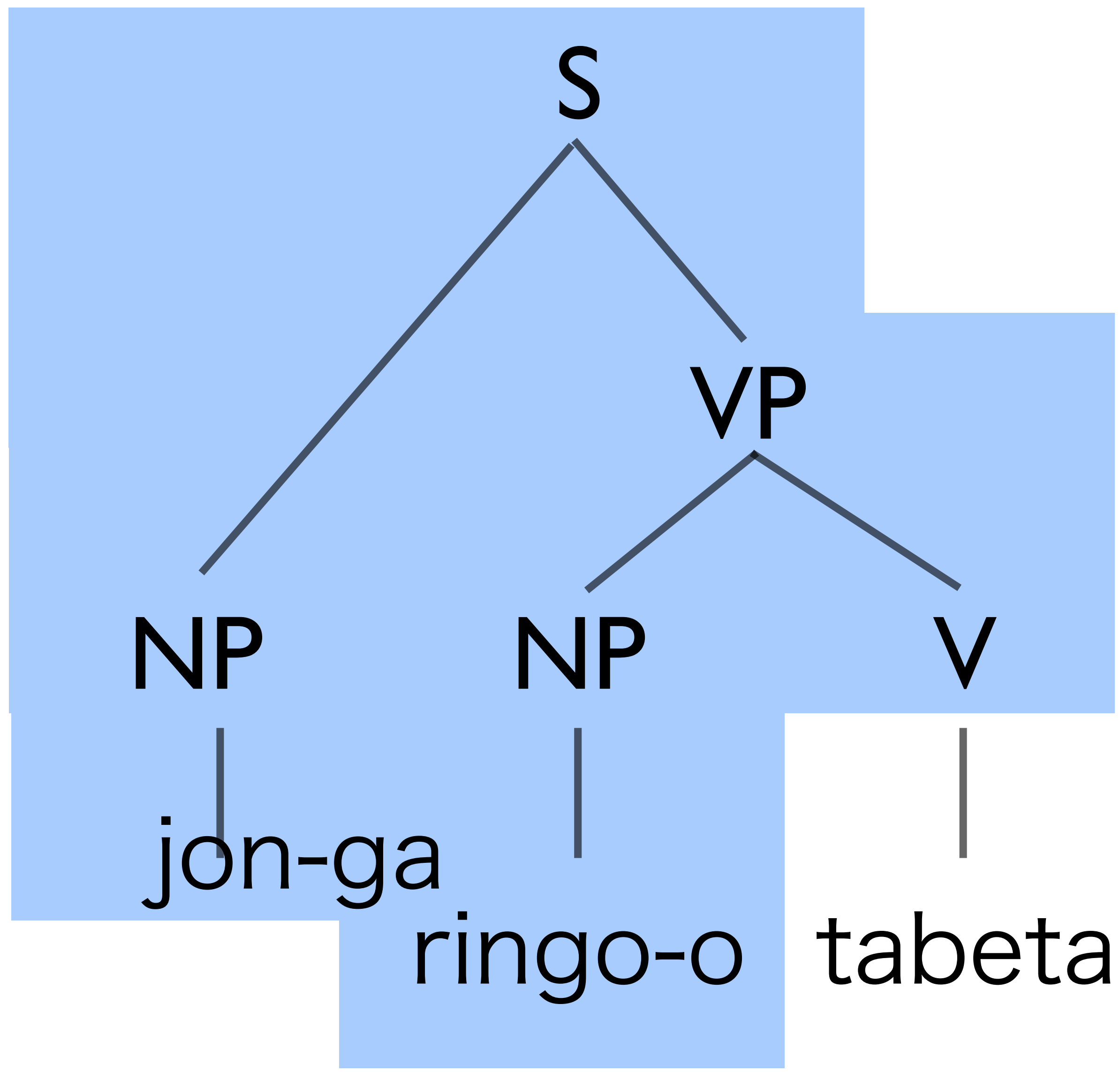
$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \to x_1\ x_2$$

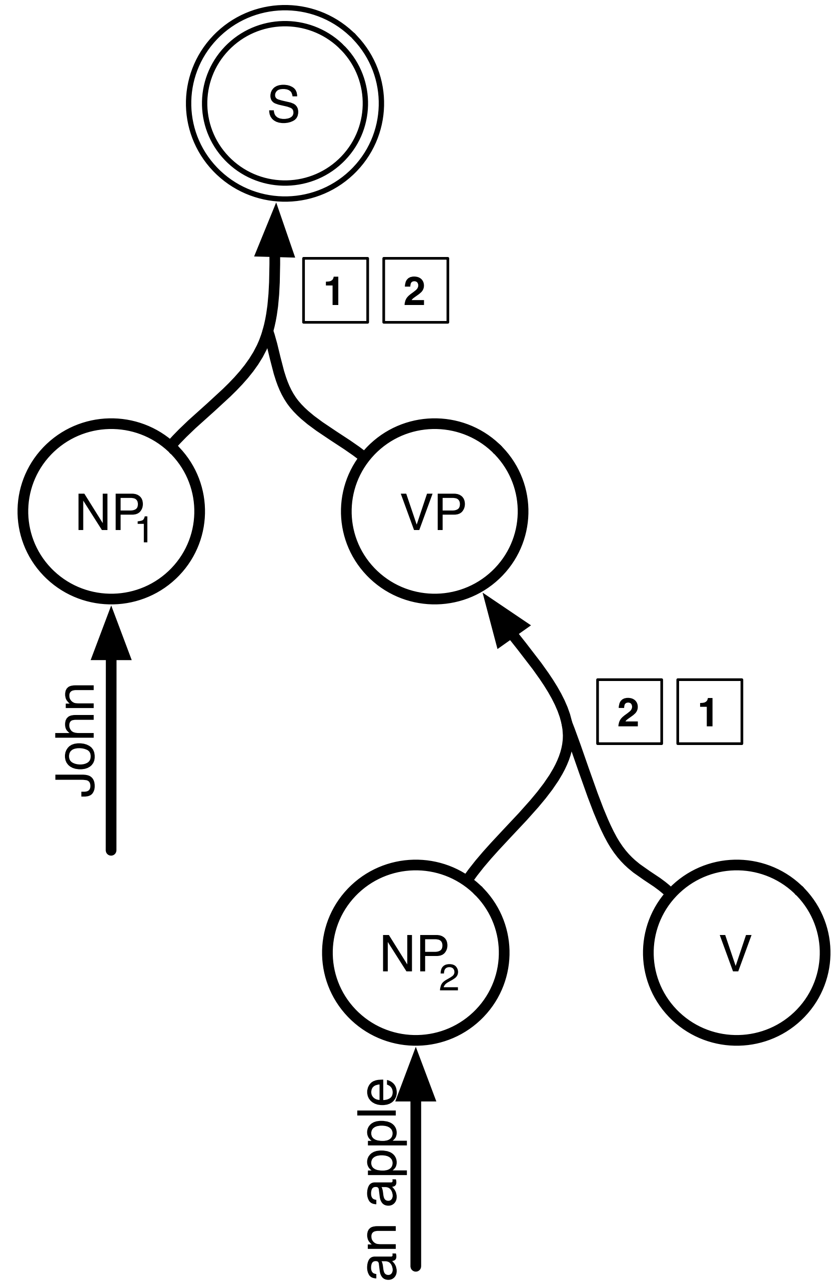
$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \to x_2\ x_1$$

$$\textit{tabeta} \to \textit{ate}$$

$$\textit{ringo-o} \to \textit{an apple}$$

$$\textit{jon-ga} \to \textit{John}$$





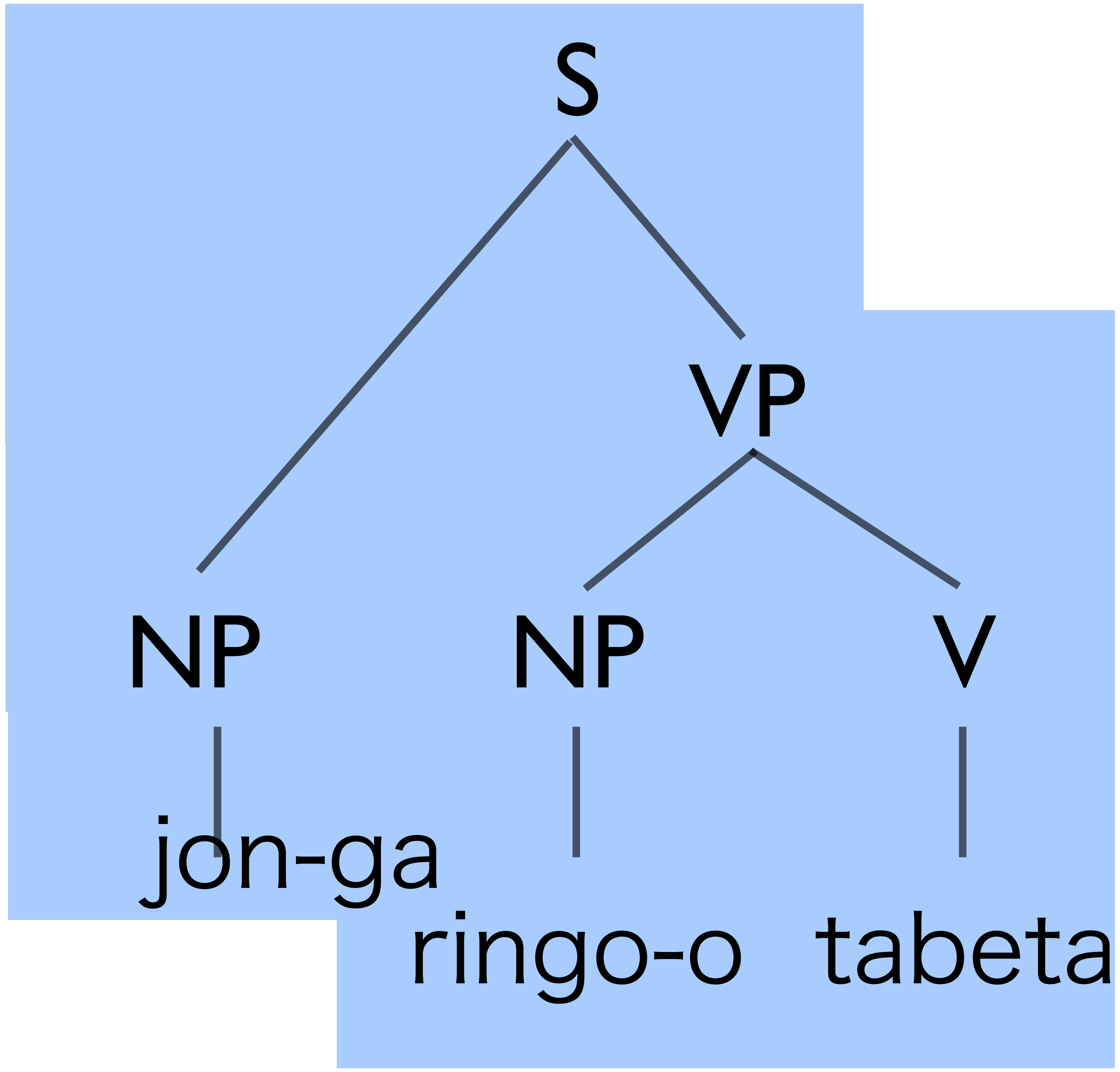
$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \rightarrow x_1\ x_2$$

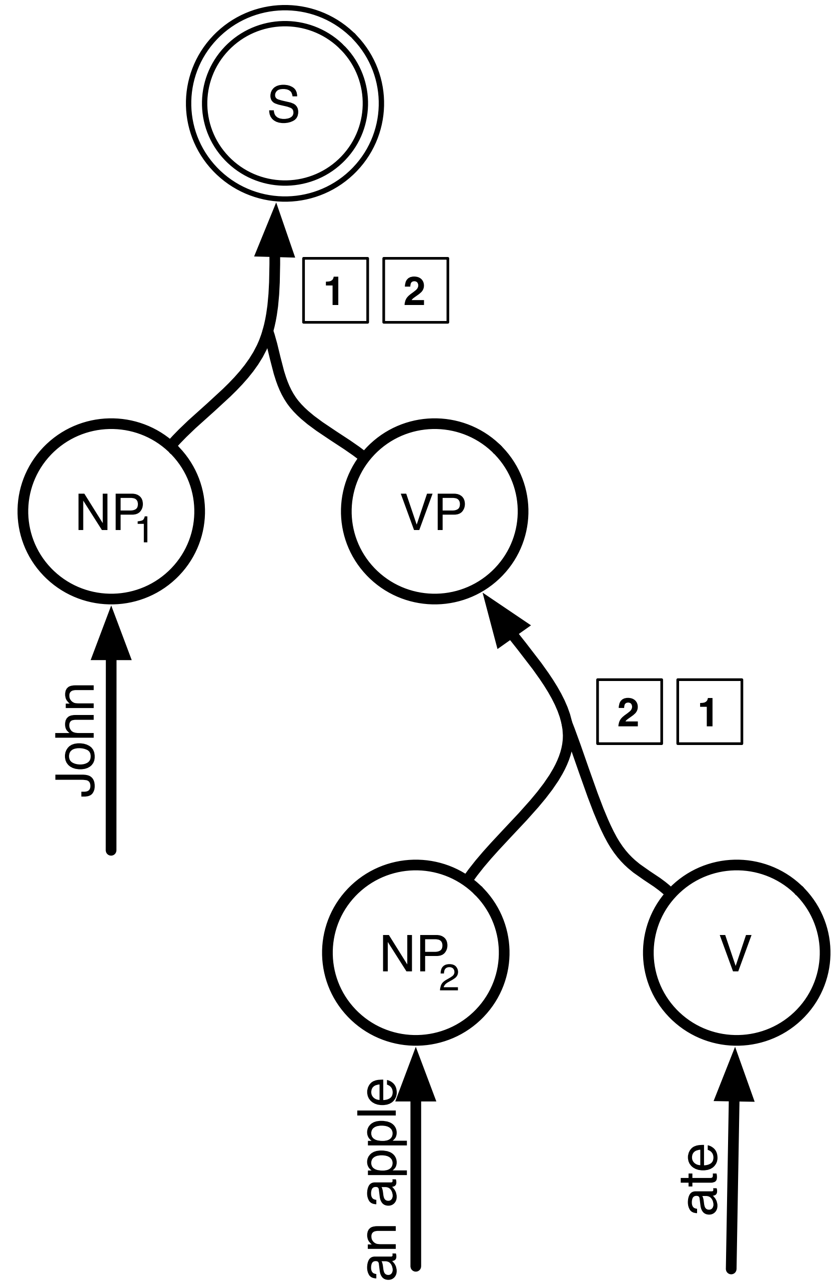
$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \rightarrow x_2\ x_1$$

$$\textit{tabeta} \rightarrow \textit{ate}$$

$$\textit{ringo-o} \rightarrow \textit{an apple}$$

$$\textit{jon-ga} \rightarrow \textit{John}$$





$$\mathrm{S}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{VP}) \rightarrow x_1\ x_2$$

$$\mathrm{VP}(x_1{:}\mathrm{NP}\ x_2{:}\mathrm{V}) \rightarrow x_2\ x_1$$

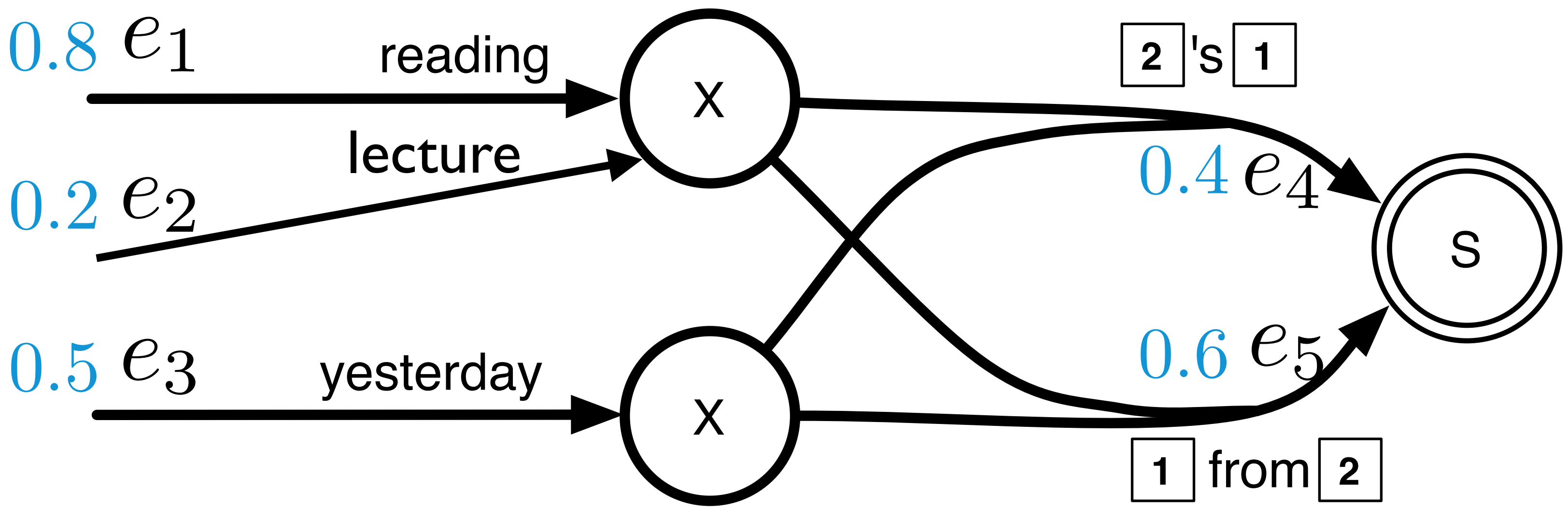
$$\textit{tabeta} \rightarrow \textit{ate}$$

$$\textit{ringo-o} \rightarrow \textit{an apple}$$

$$\textit{jon-ga} \rightarrow \textit{John}$$

Working With Hypergraphs

Derivations



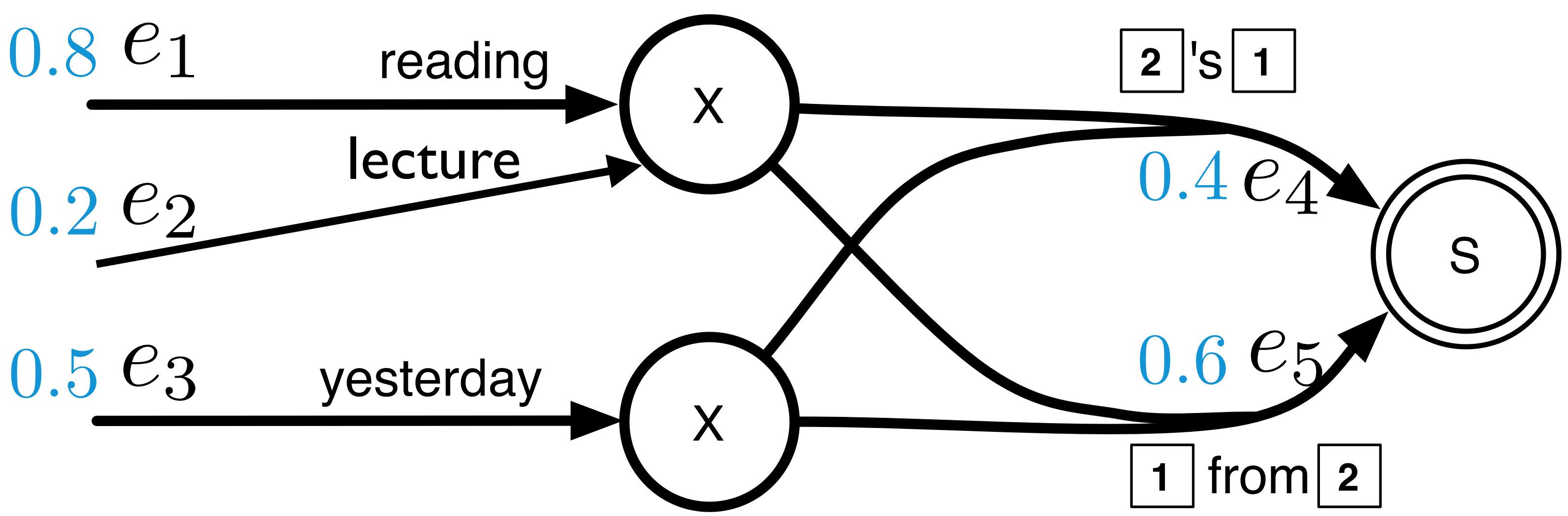
$$\boldsymbol{d}_1 = e_4 e_1 e_3 \qquad y(\boldsymbol{d}_1) = \textit{yesterday's reading}$$

$$\boldsymbol{d}_2 = e_5 e_1 e_3 \qquad y(\boldsymbol{d}_2) = \textit{reading from yesterday}$$

$$\boldsymbol{d}_3 = e_4 e_2 e_3 \qquad y(\boldsymbol{d}_3) = \textit{yesterday's lecture}$$

$$\boldsymbol{d}_4 = e_5 e_2 e_3 \qquad y(\boldsymbol{d}_4) = \textit{lecture from yesterday}$$

Derivations



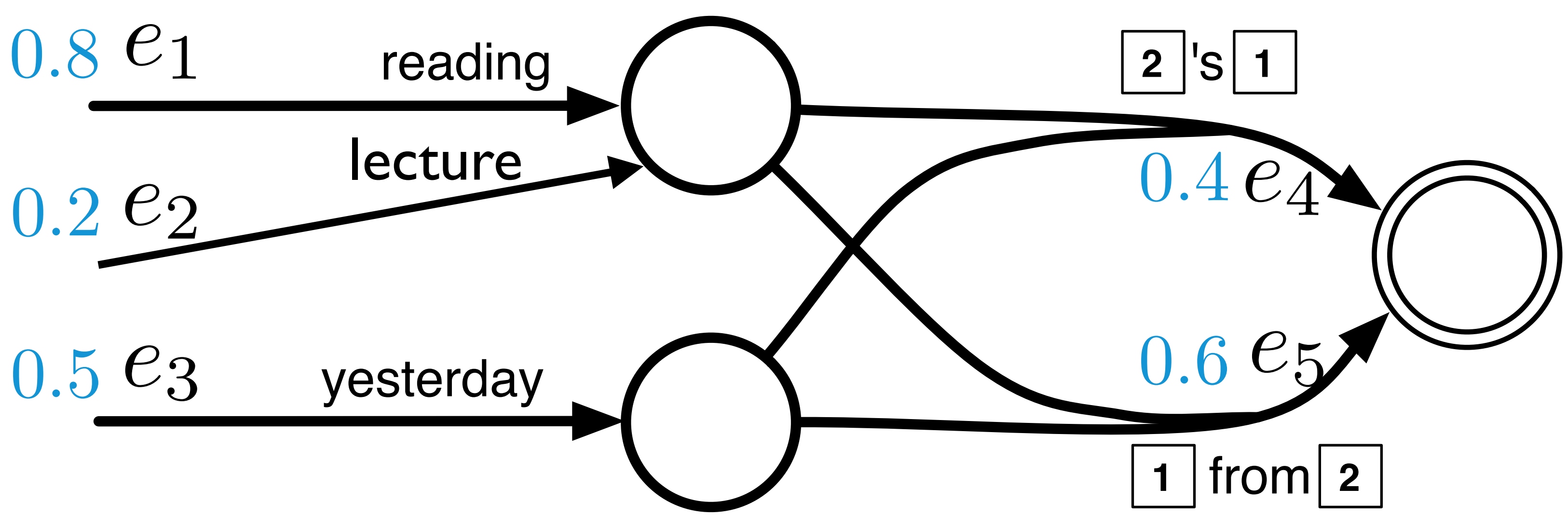
$$\boldsymbol{d}_1 = e_4 e_1 e_3 \qquad w[\boldsymbol{d}_1] = 0.4 \cdot 0.8 \cdot 0.5 = 0.16$$

$$\boldsymbol{d}_2 = e_5 e_1 e_3 \qquad w[\boldsymbol{d}_2] = 0.6 \cdot 0.8 \cdot 0.5 = 0.24$$

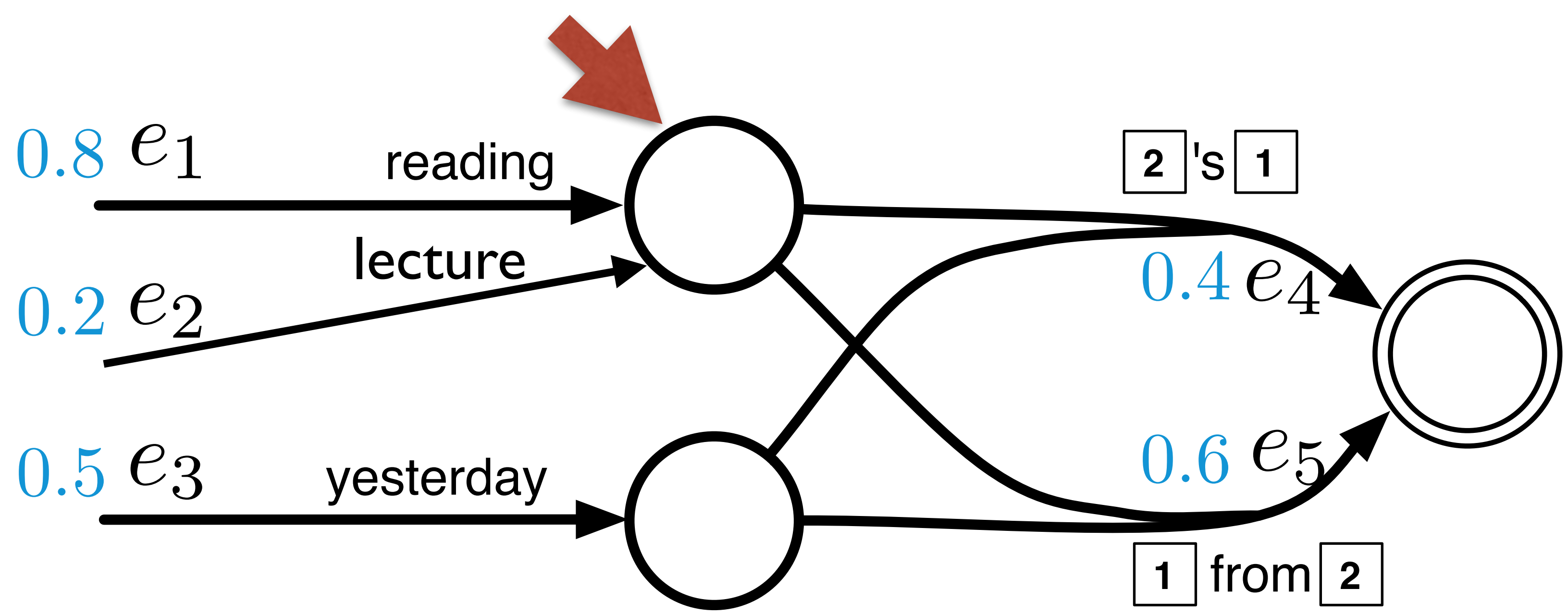
$$\boldsymbol{d}_3 = e_4 e_2 e_3 \qquad w[\boldsymbol{d}_3] = 0.4 \cdot 0.2 \cdot 0.5 = 0.04$$

$$\boldsymbol{d}_4 = e_5 e_2 e_3 \qquad w[\boldsymbol{d}_3] = 0.6 \cdot 0.2 \cdot 0.5 = 0.06$$

Best Path

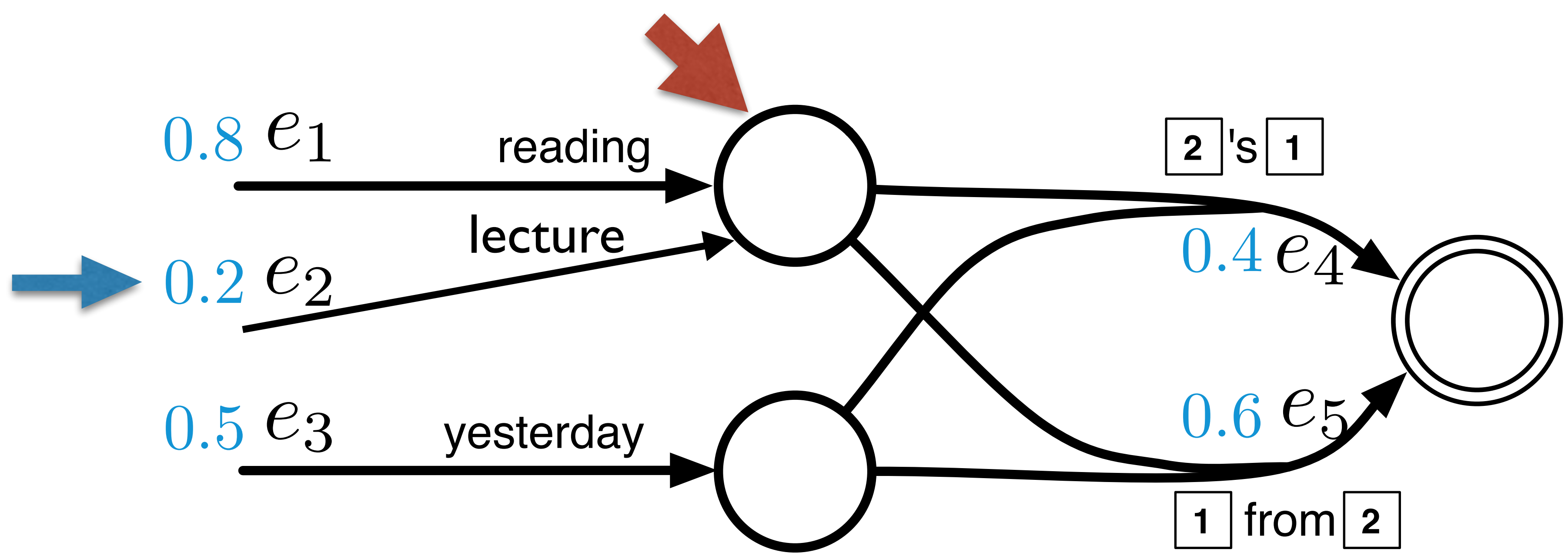

$0.8\ e_1$
reading
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



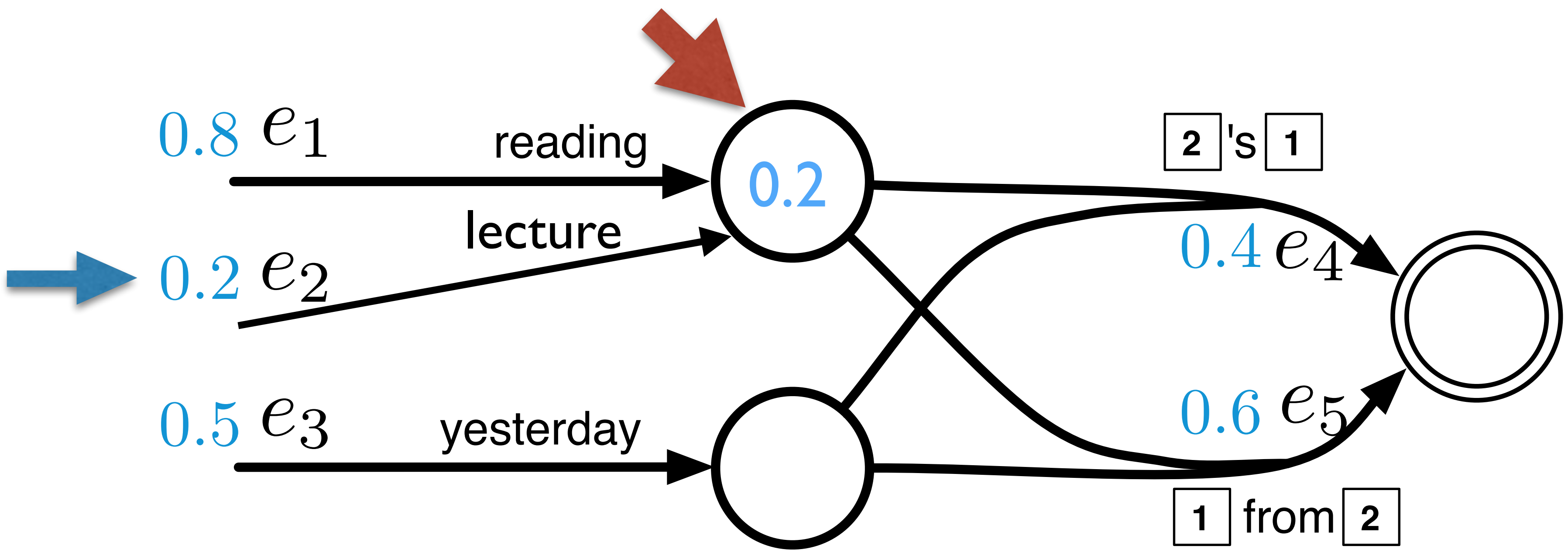
$0.8\ e_1$
reading
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



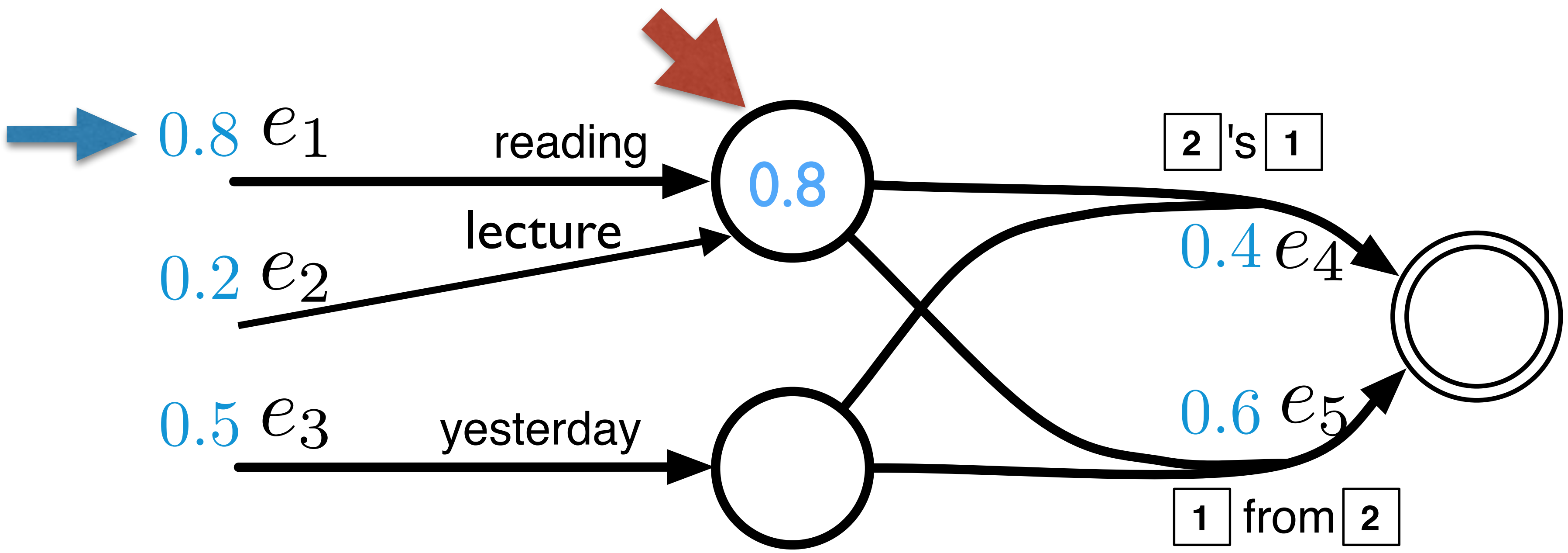
0.8 e_1
reading
0.2 e_2
lecture
0.5 e_3
yesterday
2 's 1
0.4 e_4
0.6 e_5
1 from 2

Best Path



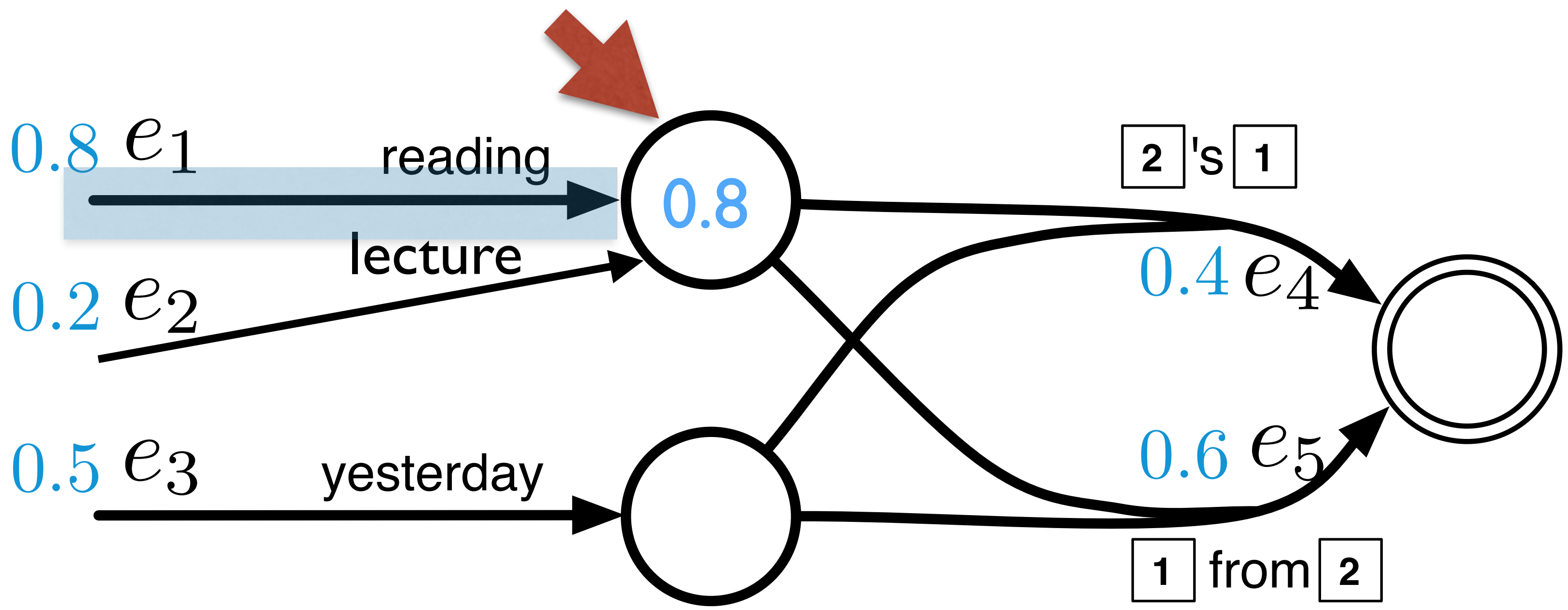
$0.8\ e_1$
reading
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
0.2
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



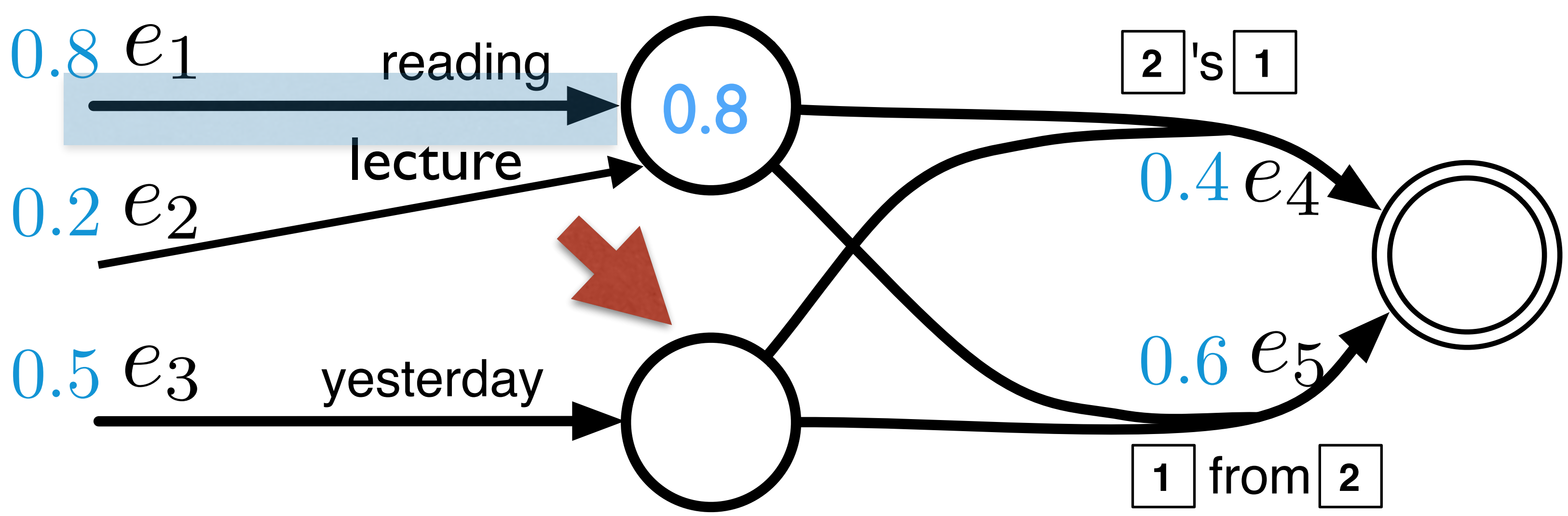
0.8 e_1
reading
0.2 e_2
lecture
0.5 e_3
yesterday
0.8
2 's 1
0.4 e_4
0.6 e_5
1 from 2

Best Path



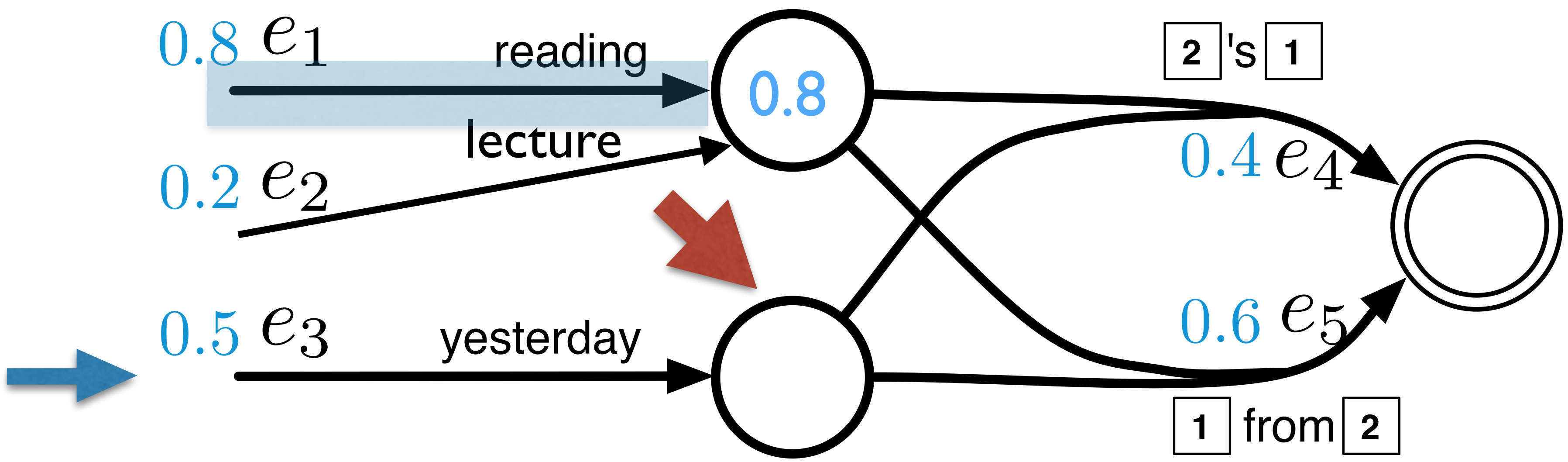
0.8 e_1
reading
0.8
0.2 e_2
lecture
0.5 e_3
yesterday
2 's 1
0.4 e_4
0.6 e_5
1 from 2

Best Path



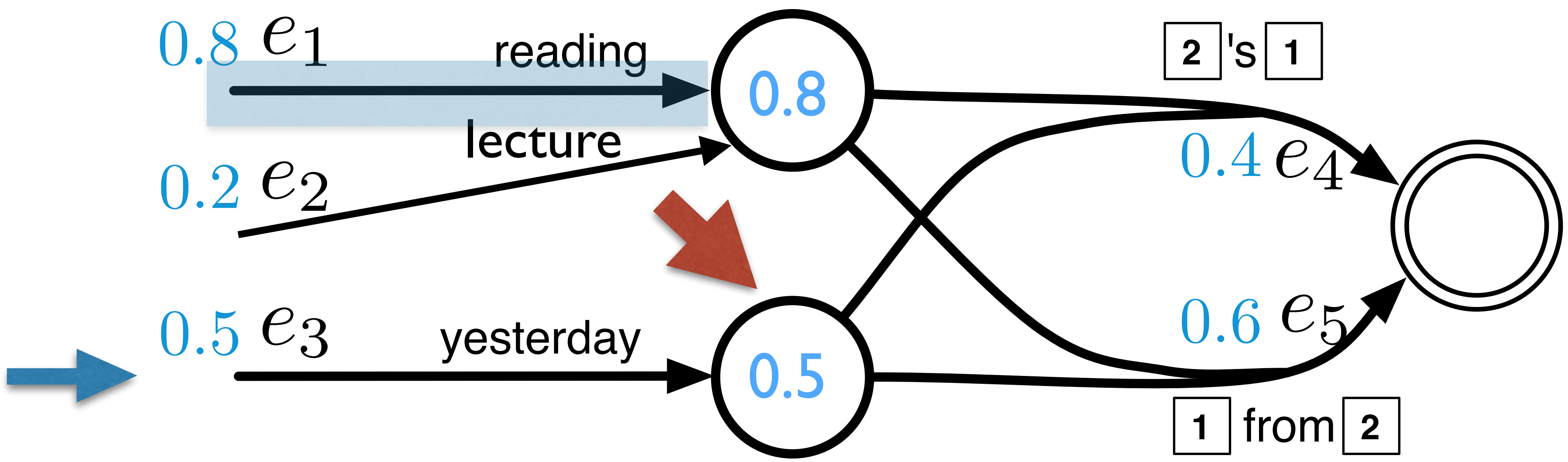
0.8 e_1
reading
0.8
0.2 e_2
lecture
0.5 e_3
yesterday
2 's 1
0.4 e_4
0.6 e_5
1 from 2

Best Path



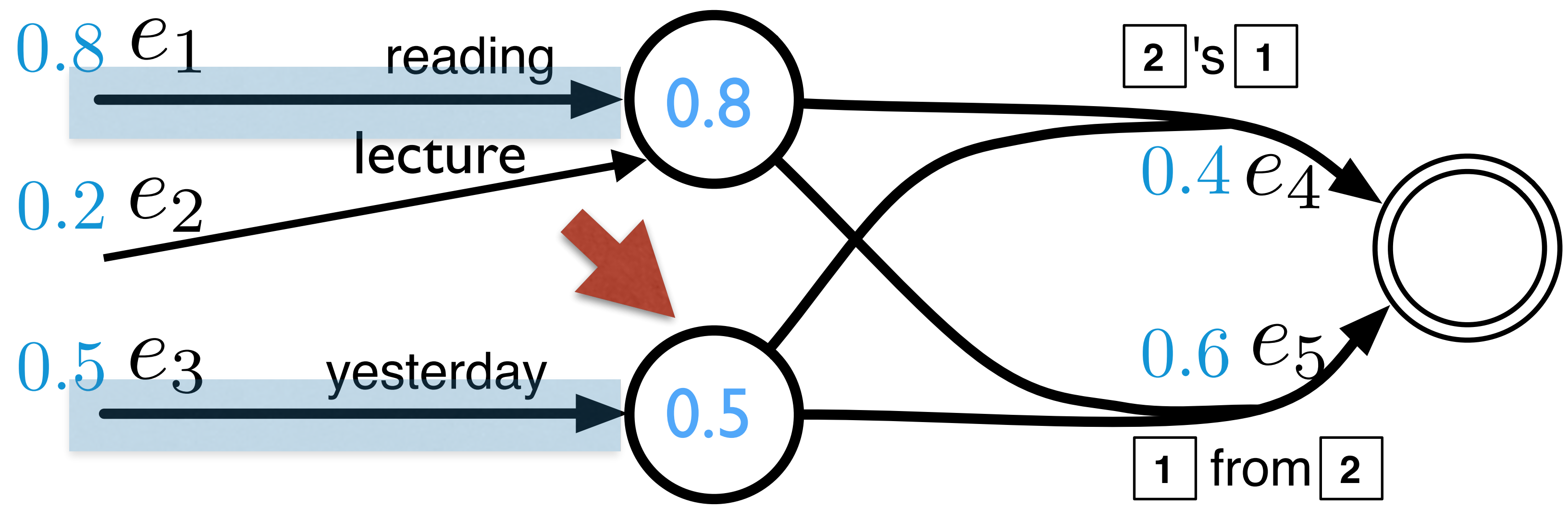
$0.8\ e_1$
reading
0.8
lecture
$0.2\ e_2$
$0.5\ e_3$
yesterday
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



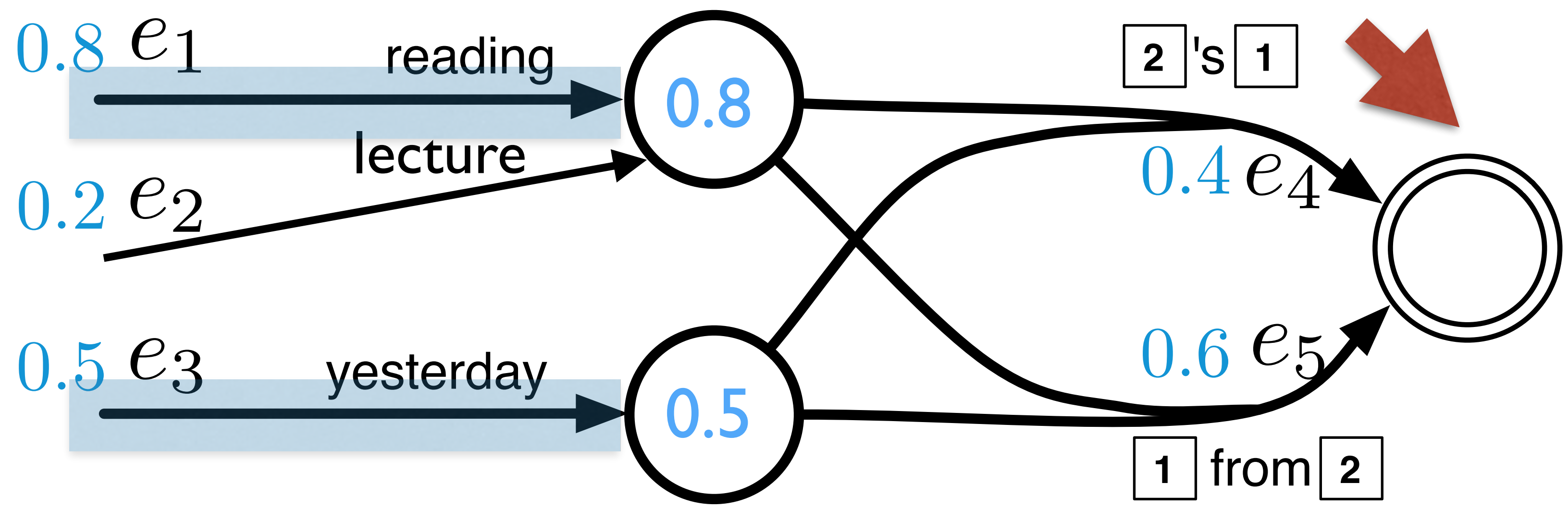
$0.8\ e_1$
reading
0.8
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
0.5
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



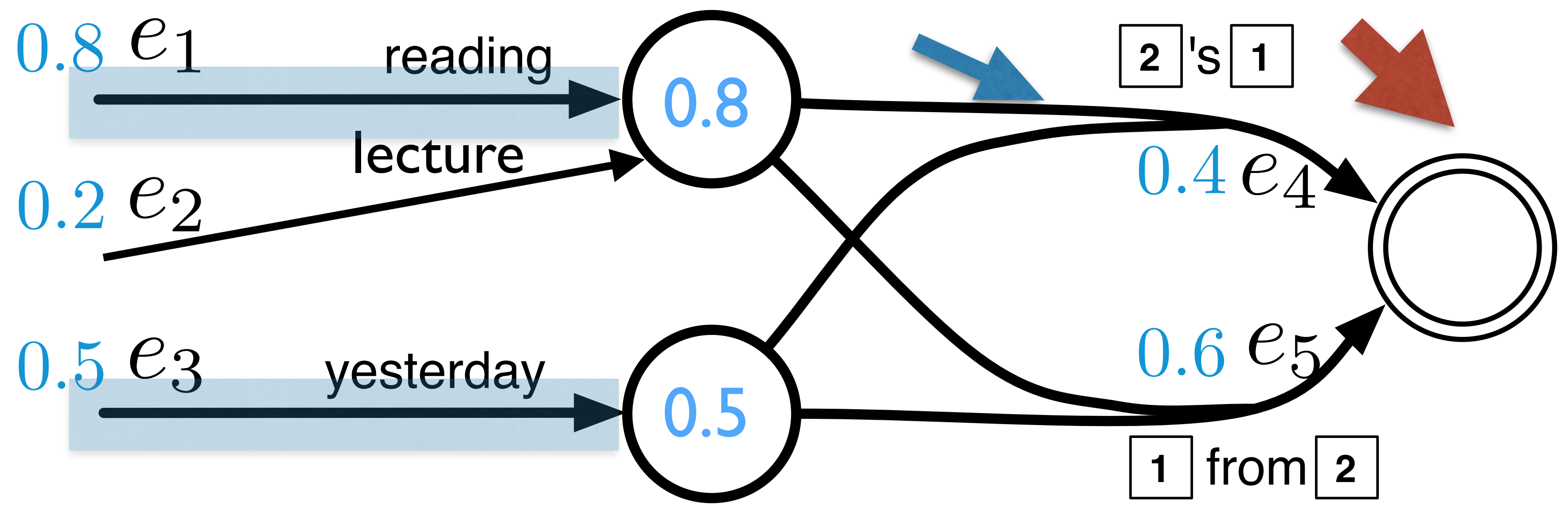
$0.8\ e_1$
reading
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
0.8
0.5
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



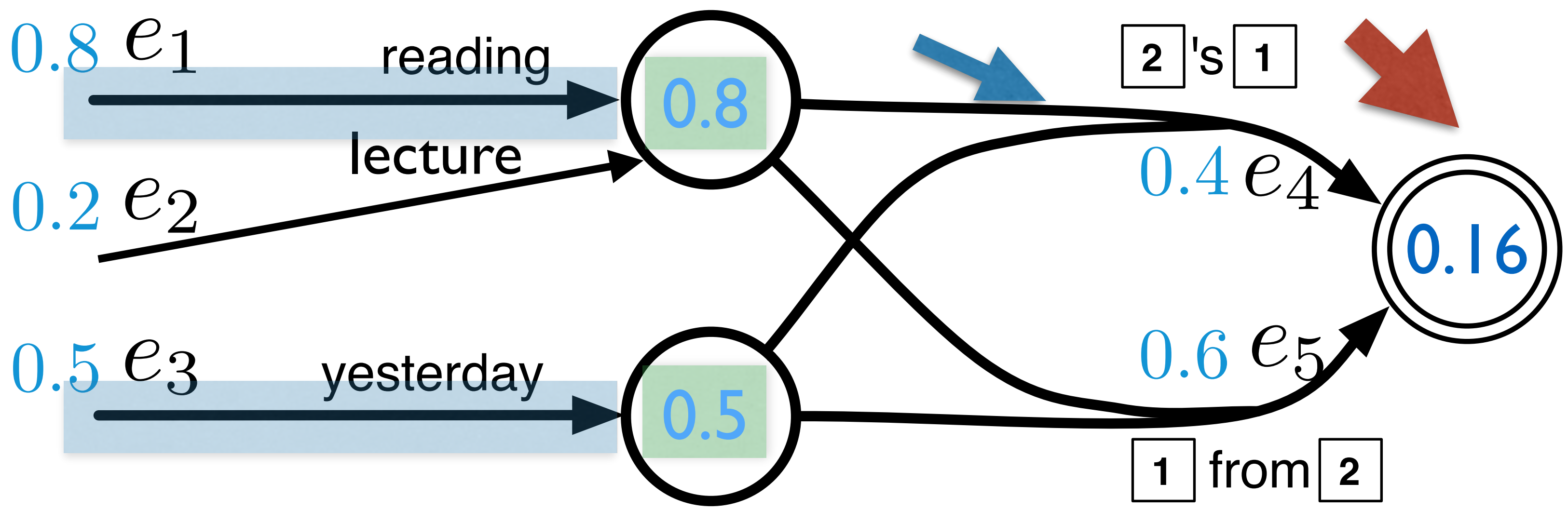
$0.8\ e_1$
reading
0.8
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
0.5
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2

Best Path



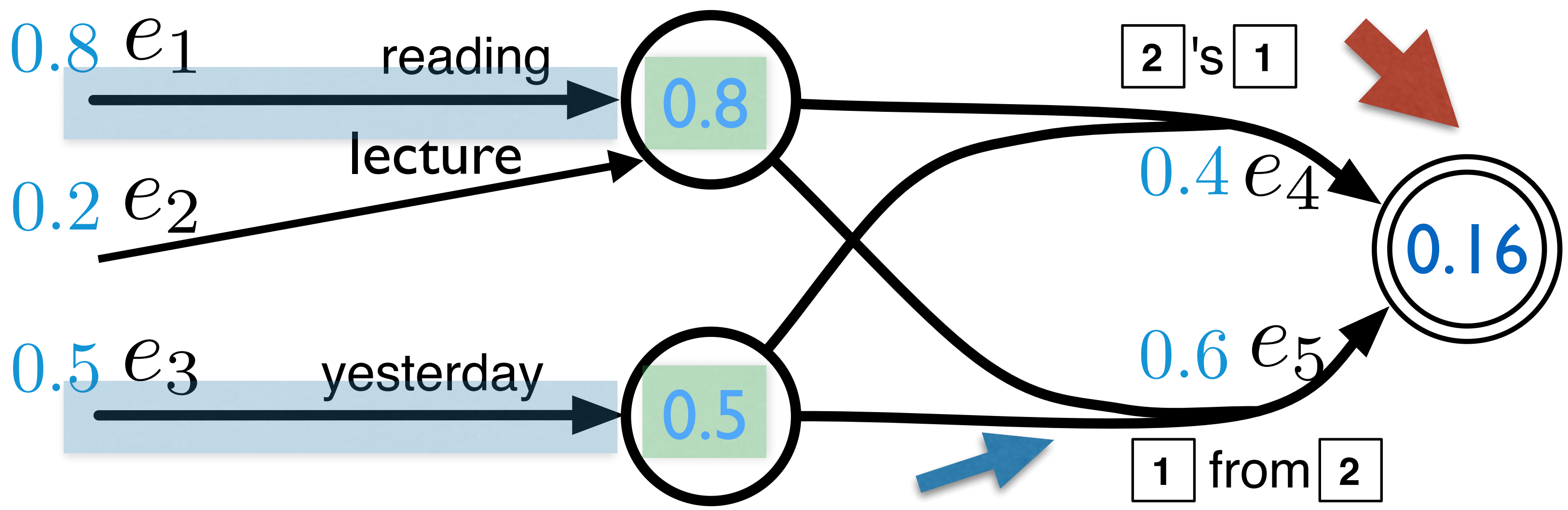
0.8 e_1
reading
0.2 e_2
lecture
0.5 e_3
yesterday
0.8
0.5
2 's 1
0.4 e_4
0.6 e_5
1 from 2

Best Path



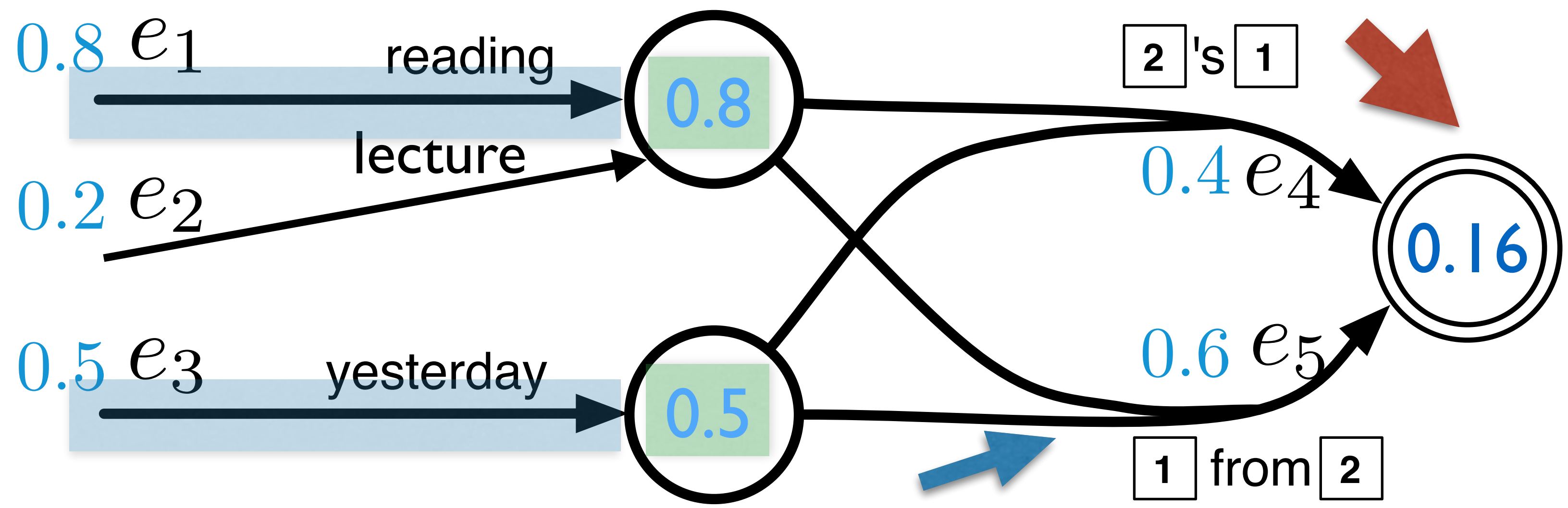
$0.8 \times 0.5 \times 0.4 = 0.16$

Best Path



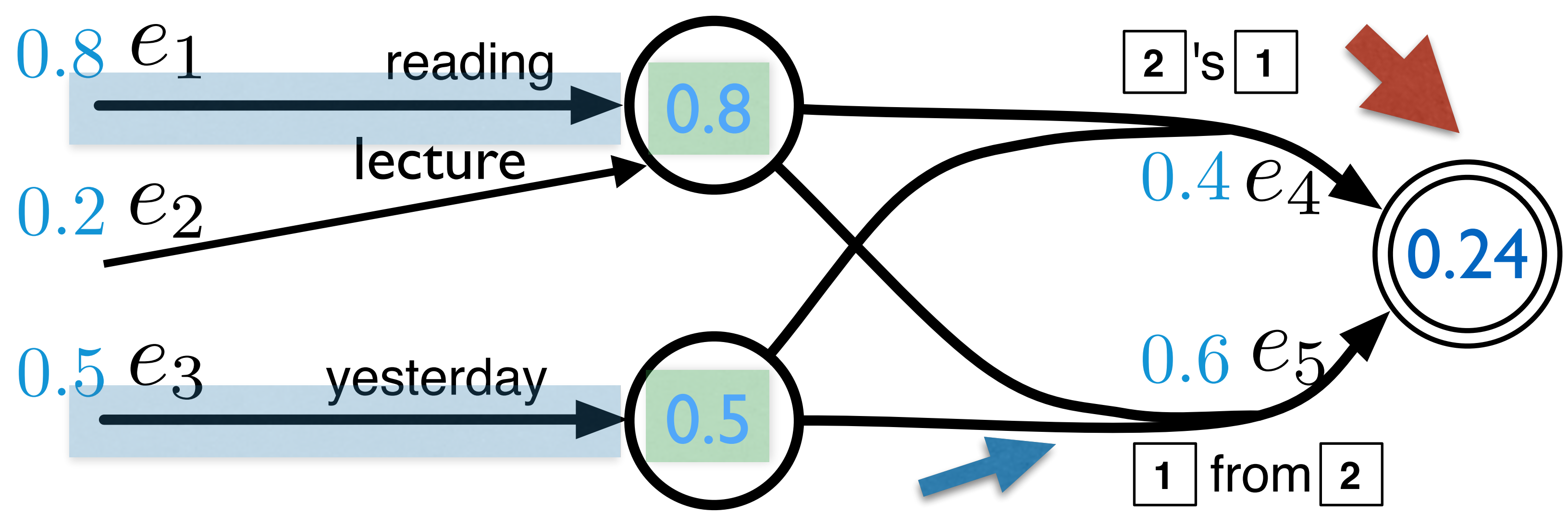
$0.8\ e_1$
reading
$0.2\ e_2$
lecture
$0.5\ e_3$
yesterday
0.8
0.5
2 's 1
$0.4\ e_4$
$0.6\ e_5$
1 from 2
0.16

Best Path



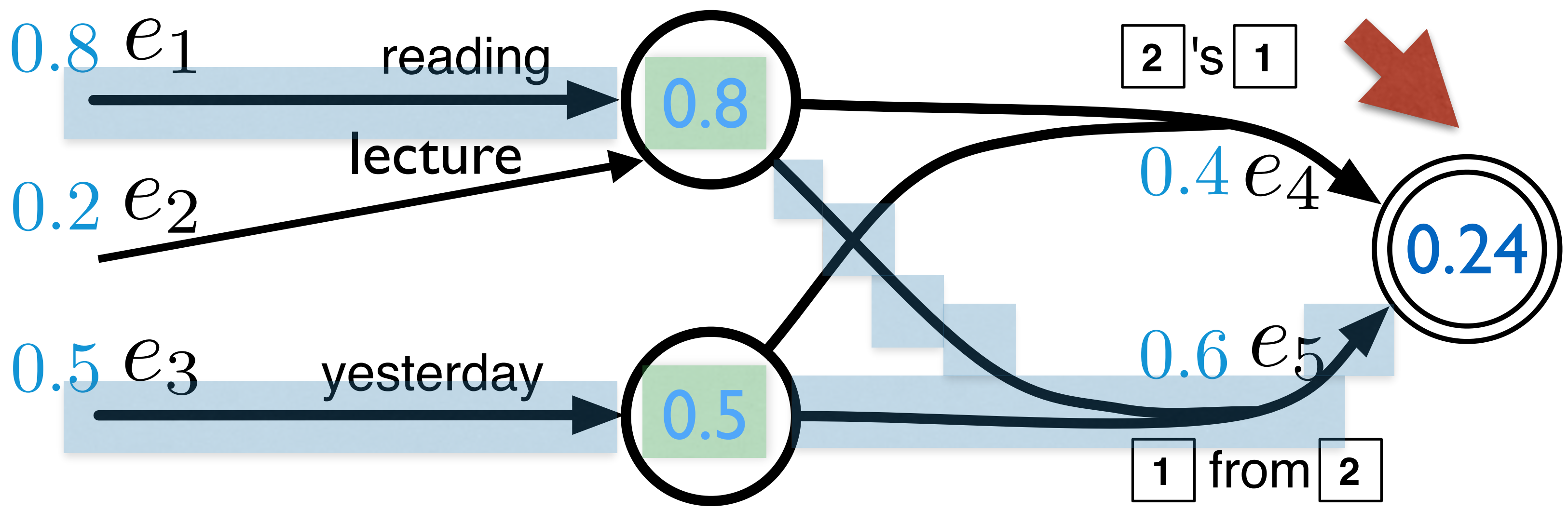
$0.8 \times 0.5 \times 0.6 = 0.24$

Best Path



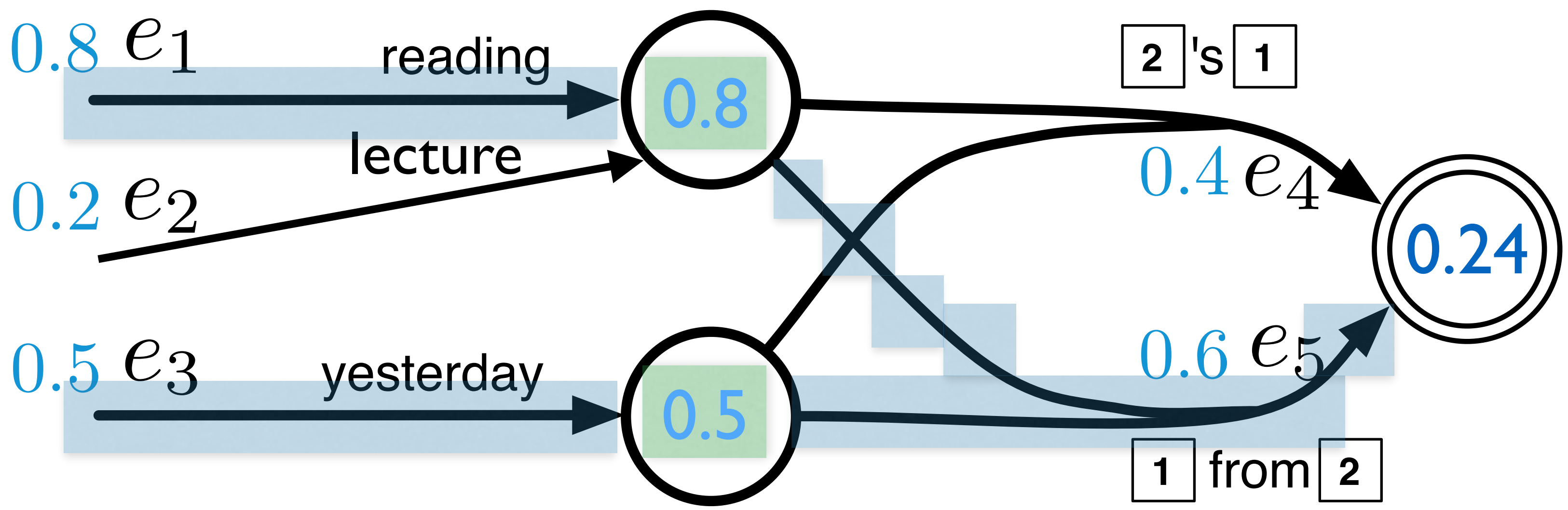
$0.8 \times 0.5 \times 0.6 = 0.24$

Best Path



0.8 e_1
reading
lecture
0.2 e_2
0.5 e_3
yesterday
0.8
0.5
2 's 1
0.4 e_4
0.6 e_5
1 from 2
0.24

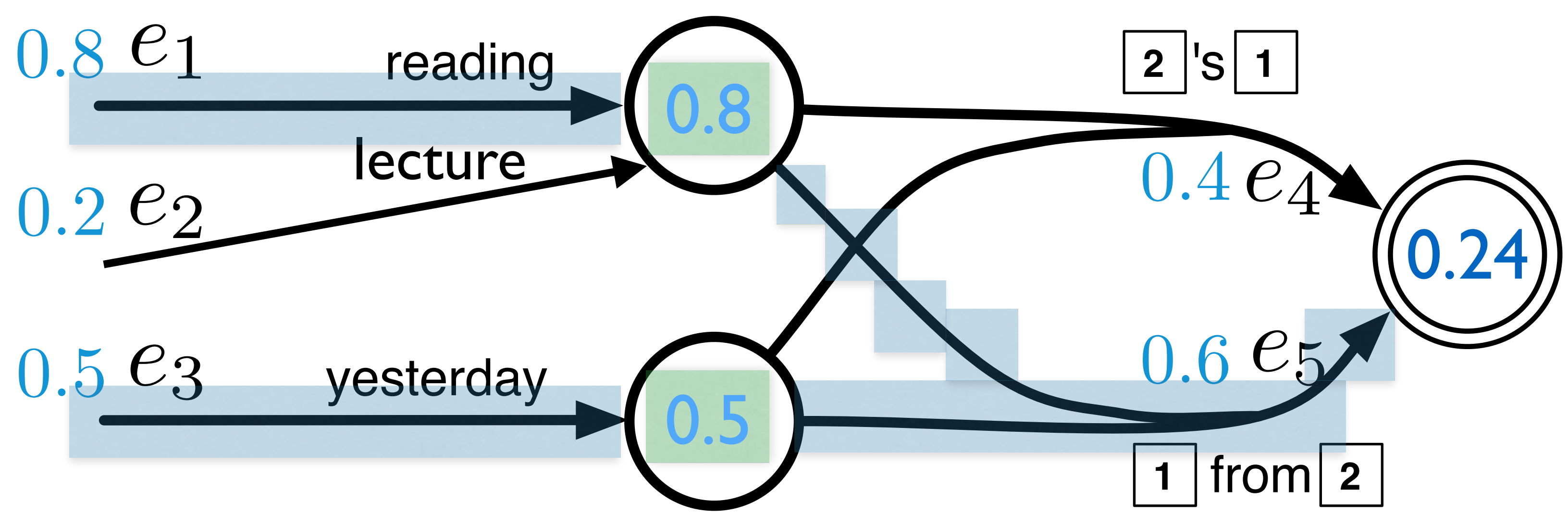
Best Path



Best yield: *reading from yesterday*

Best path: 0.24

Best Path



Best yield: ***reading from yesterday***

Best path: 0.24

$$\boldsymbol{d}_1 = e_4 e_1 e_3 \qquad w[\boldsymbol{d}_1] = 0.4 \cdot 0.8 \cdot 0.5 = 0.16$$

$$\boldsymbol{d}_2 = e_5 e_1 e_3 \qquad w[\boldsymbol{d}_2] = 0.6 \cdot 0.8 \cdot 0.5 = 0.24$$

$$\boldsymbol{d}_3 = e_4 e_2 e_3 \qquad w[\boldsymbol{d}_3] = 0.4 \cdot 0.2 \cdot 0.5 = 0.04$$

$$\boldsymbol{d}_4 = e_5 e_2 e_3 \qquad w[\boldsymbol{d}_3] = 0.6 \cdot 0.2 \cdot 0.5 = 0.06$$

Other Algorithms

- Given a weighted hypergraph
- In the Viterbi (Inside) algorithm, there are two operations
    - **Multiplication** (extend path)
    - **Maximization** (chose between paths)
- Semirings generalize these to compute other quantities

Semirings

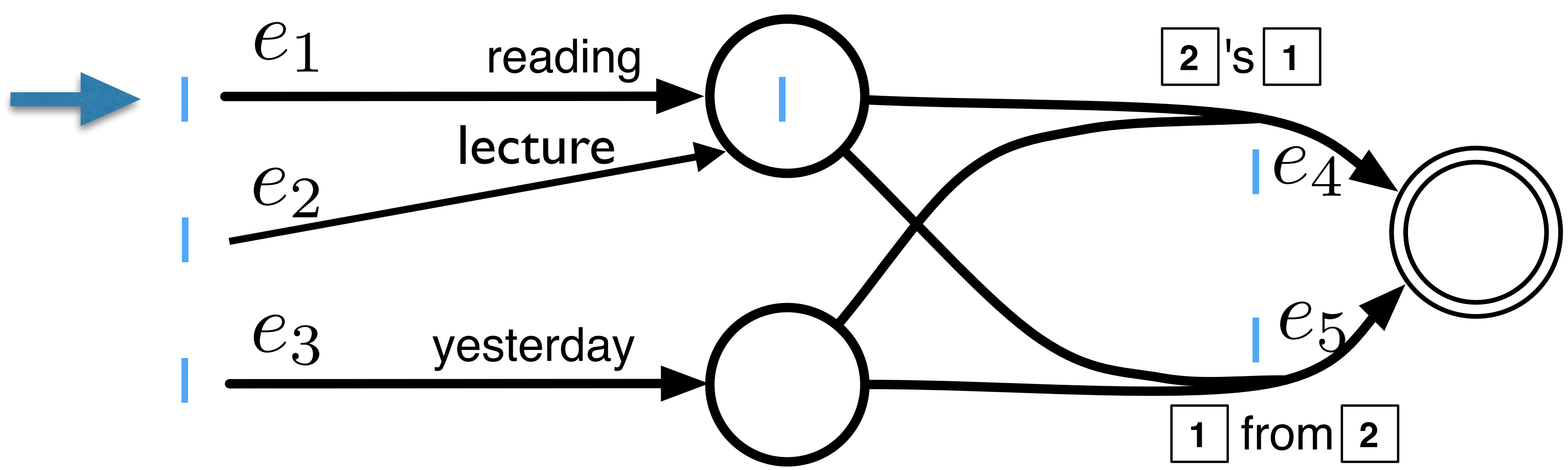
semiring	$\mathbb{K}$	$\oplus$	$\otimes$	$\bar{0}$	$\bar{1}$	notes
Boolean	$\{0,1\}$	$\vee$	$\wedge$	0	1	idempotent
count	$\mathbb{N}_0 \cup \{\infty\}$	$+$	$\times$	0	1	
probability	$\mathbb{R}_+ \cup \{\infty\}$	$+$	$\times$	0	1	
tropical	$\mathbb{R} \cup \{-\infty, \infty\}$	max	$+$	$-\infty$	0	idempotent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

Inside Algorithm

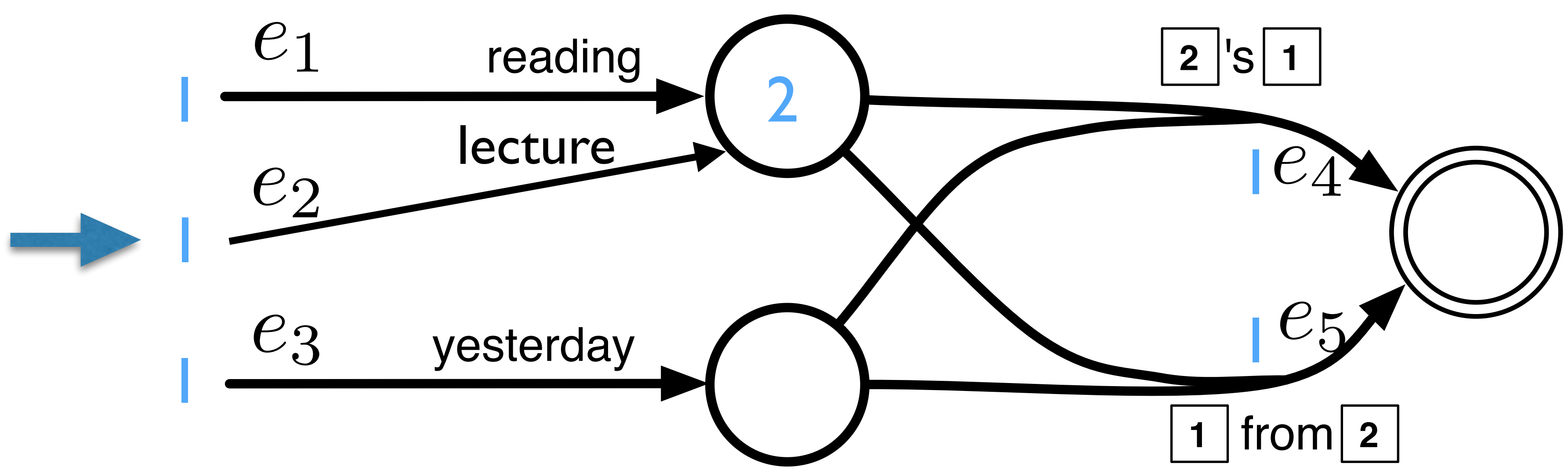
$$\alpha(q_{goal}) = \bigoplus_{\mathbf{d} \in \mathcal{G}} \bigotimes_{e \in \mathbf{d}} w(e)$$

function INSIDE($\mathcal{G}, K$) $\triangleright$ $\mathcal{G}$ is an acyclic hypergraph and K is a semiring
  for q in topological order in $\mathcal{G}$ **do**
    if $B(q) = \emptyset$ **then**
      $\alpha(q) \leftarrow \bar{1}$ $\triangleright$ assume states with no in-edges are axioms
    else
      $\alpha(q) \leftarrow \bar{0}$
      for all $e \in B(q)$ **do** $\triangleright$ all in-coming edges to node q
        $k \leftarrow w(e)$
        for all $r \in \mathbf{t}(e)$ **do** $\triangleright$ all tail (previous) nodes of edge e
          $k \leftarrow k \otimes \alpha(r)$
        $\alpha(q) \leftarrow \alpha(q) \oplus k$
  return α

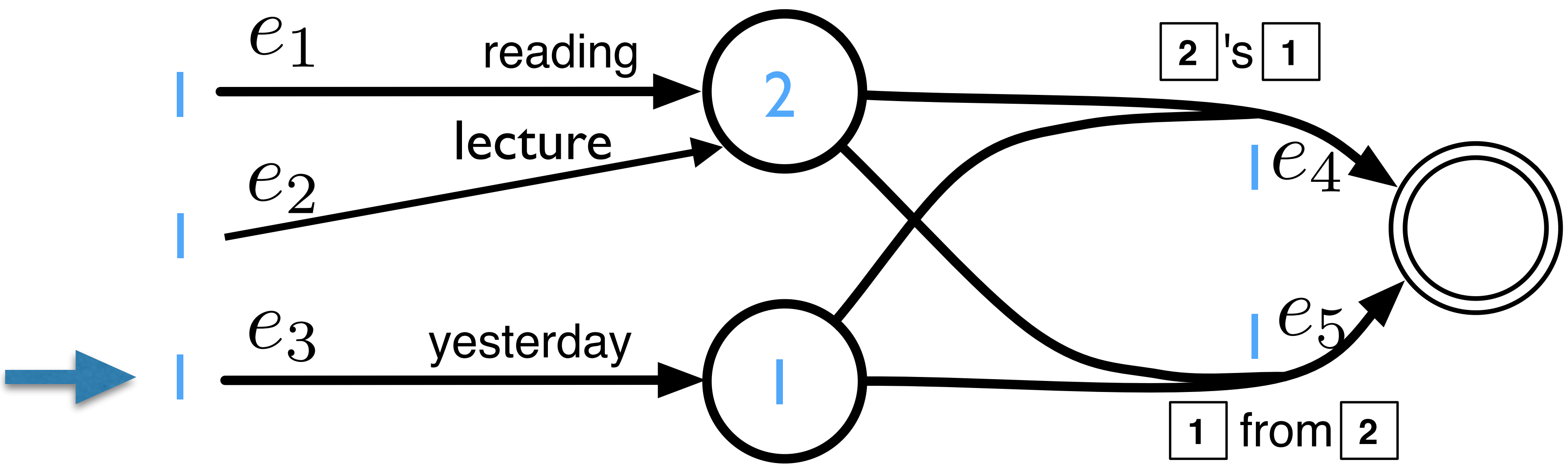
Count Derivations



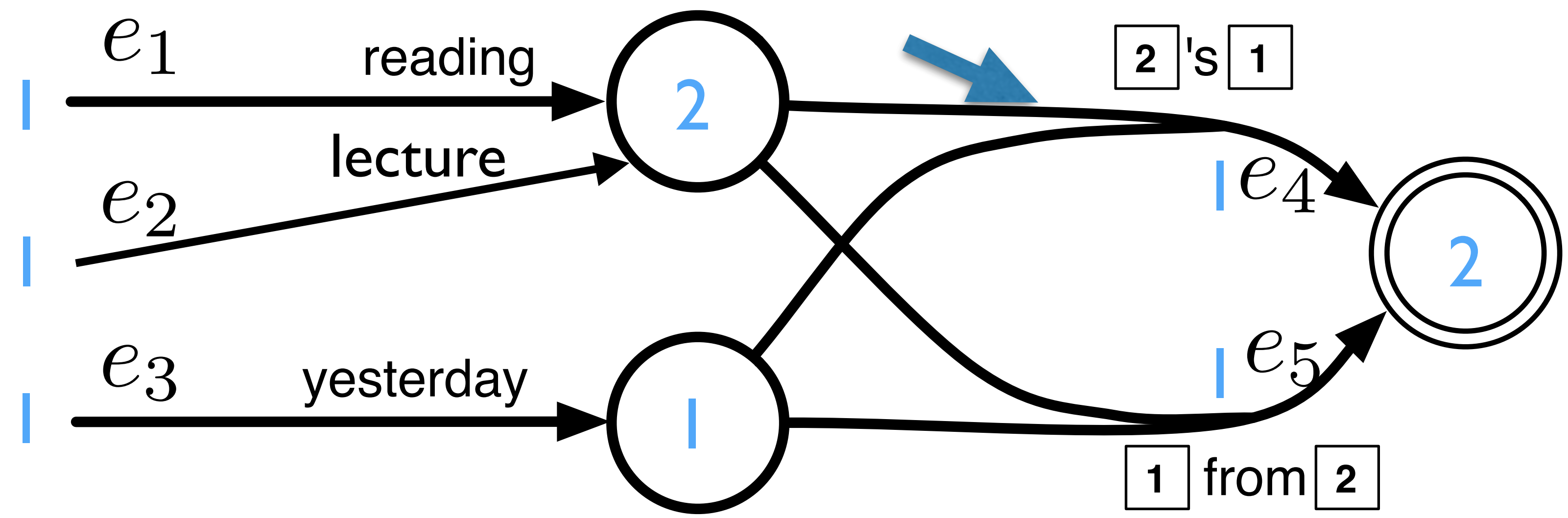
Count Derivations



Count Derivations

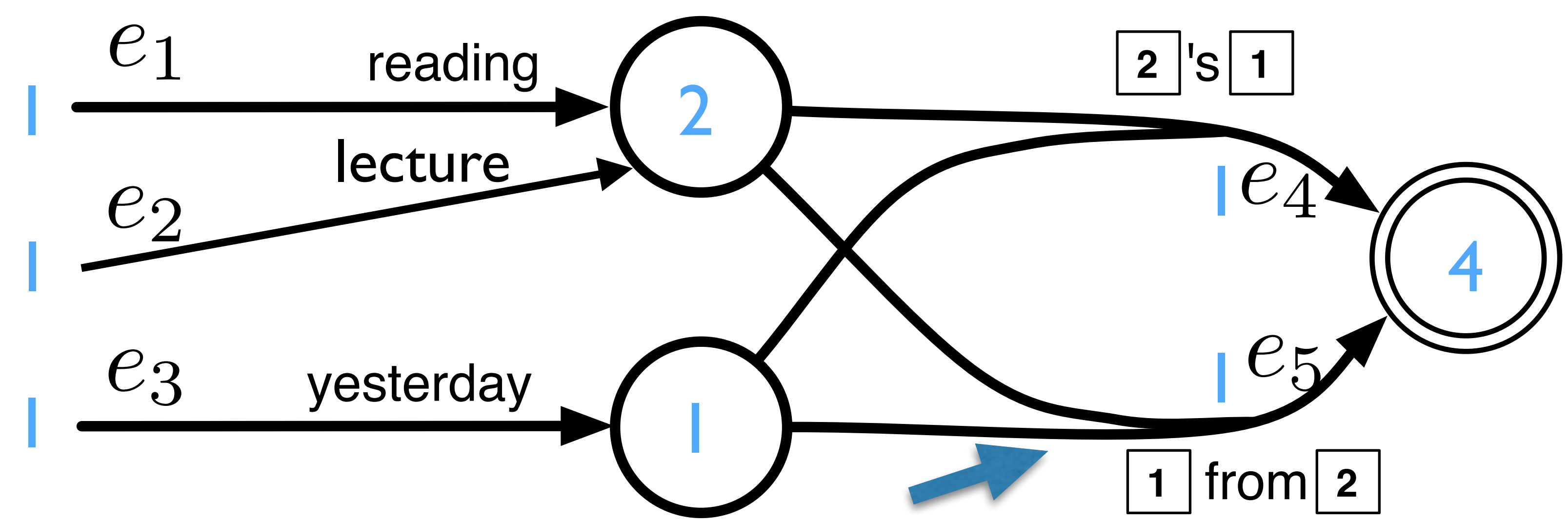


Count Derivations



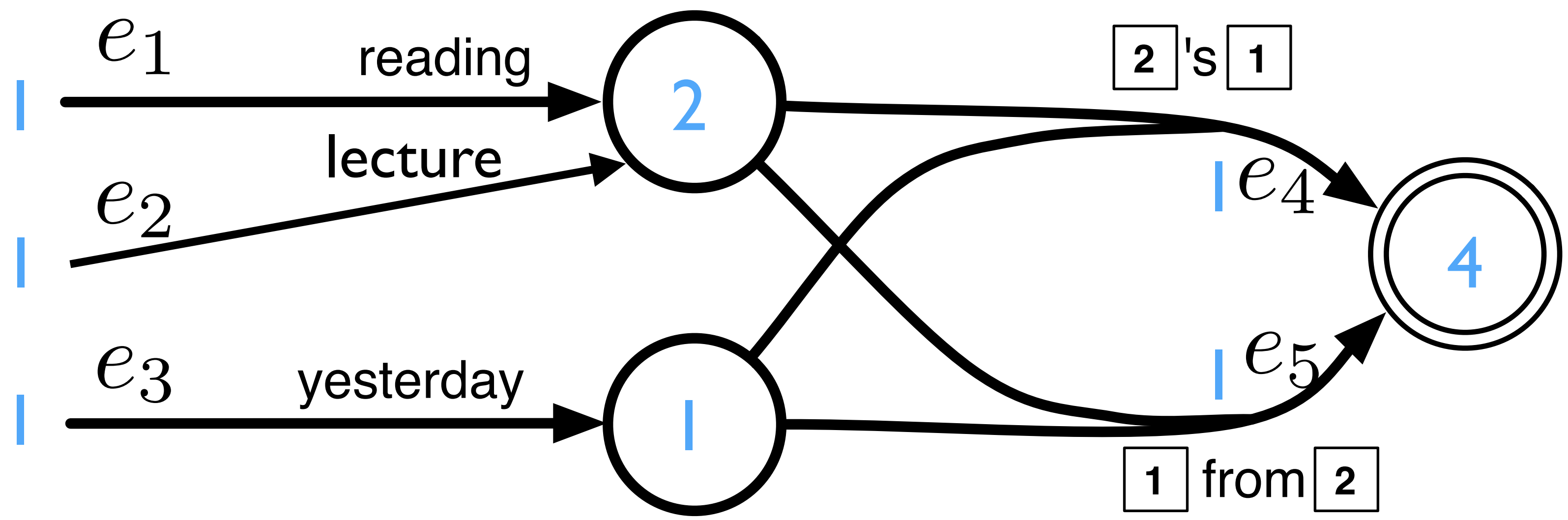
2 × 1 × 1 = 2

Count Derivations



2 × 1 × 1 = 2

Count Derivations



Inside-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
```

```
function INSIDEOUTSIDE(G, K)                 ▷ compute edge marginals
    α ← INSIDE(G, K)
    β ← OUTSIDE(G, K, α)
    for edge e in G do
        γ(e) ← w(e) ⊗ β(n(e))                ▷ edge weight and outside score of edge's head node
        for all q ∈ t(e) do
            γ(e) ← γ(e) ⊗ α(q)               ▷ inside score of tail nodes
    return γ                                 ▷ γ(e) is the edge marginal of e
```

Inside-Outside

- Compute lots of interesting quantities
  - The score of the best path through each edge
  - The total number of derivations that contain an edge
  - The total score of all derivations going through an edge

Inference algorithms

- **Viterbi** $O(|E| + |V|)$
  - Find the maximum weighted derivation
  - Requires a partial ordering of weights
- **Inside - outside** $O(|E| + |V|)$
  - Compute the marginal (sum) weight of all derivations passing through each edge/node
- **k-best derivations** $O(|E| + |D_{max}| k \log k)$
  - Enumerate the k-best derivations in the hypergraph
  - See IWPT paper by Huang and Chiang (2005)

Things to keep in mind

Bound on the number of edges (SCFG):

$$|E| \in O(n^3|G|^3)$$

Bound on the number of nodes:

$$|V| \in O(n^2|G|)$$

Next time

What about the LM?