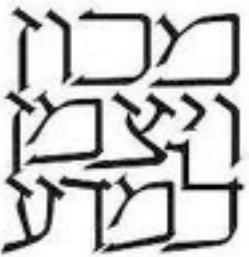




UPPSALA
UNIVERSITET



Parsing Morphologically Rich Languages (PMRL)

Class 2: Phrase-Structure Parsing

Reut Tsarfaty
reut.tsarfaty@weizmann.ac.il

Yesterday@PMRL

- Day 1: Introduction
 - Introduction to Parsing
 - Introduction to Morphology
- Day 2: Phrase-structure
- Day 3: Dependency-structure
- Day 4: Relational-Realizational
- Day 5: Evaluation and Multilinguality

Yesterday@PMRL

- ✓ Day 1: Introduction
- ✓ Introduction to Parsing
- ✓ Introduction to Morphology
- Day 2: Phrase-structure
- Day 3: Dependency-structure
- Day 4: Relational-Realizational
- Day 5: Evaluation and Multilinguality

Yesterday@PMRL



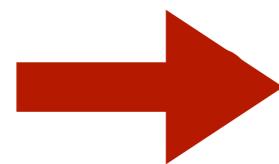
Day 1: Introduction



Introduction to Parsing



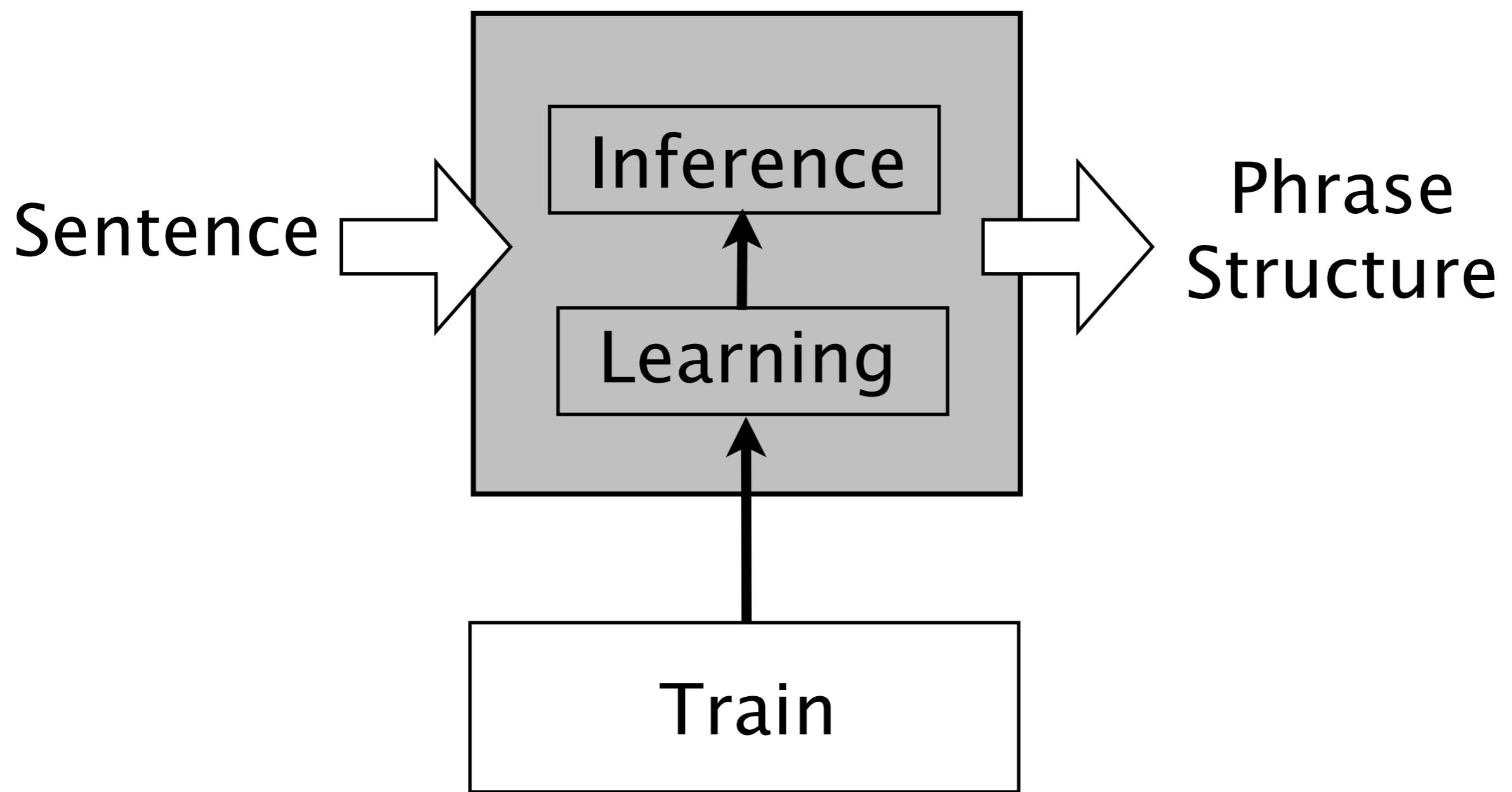
Introduction to Morphology



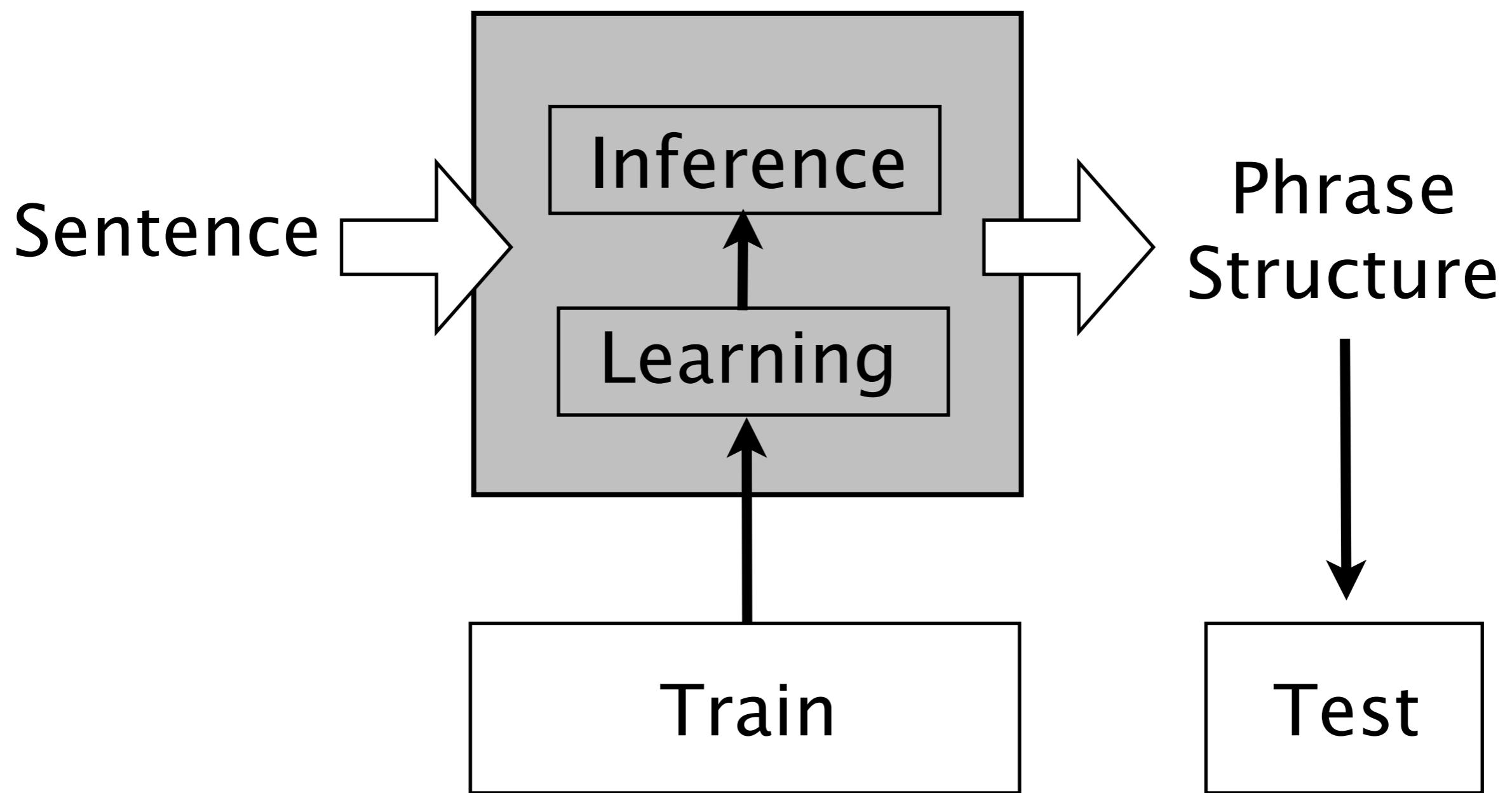
Day 2: Phrase-structure

- Day 3: Dependency-structure
- Day 4: Relational-Realizational
- Day 5: Evaluation and Multilinguality

The Task: PS for MRLs



The Task: PS for MRLs



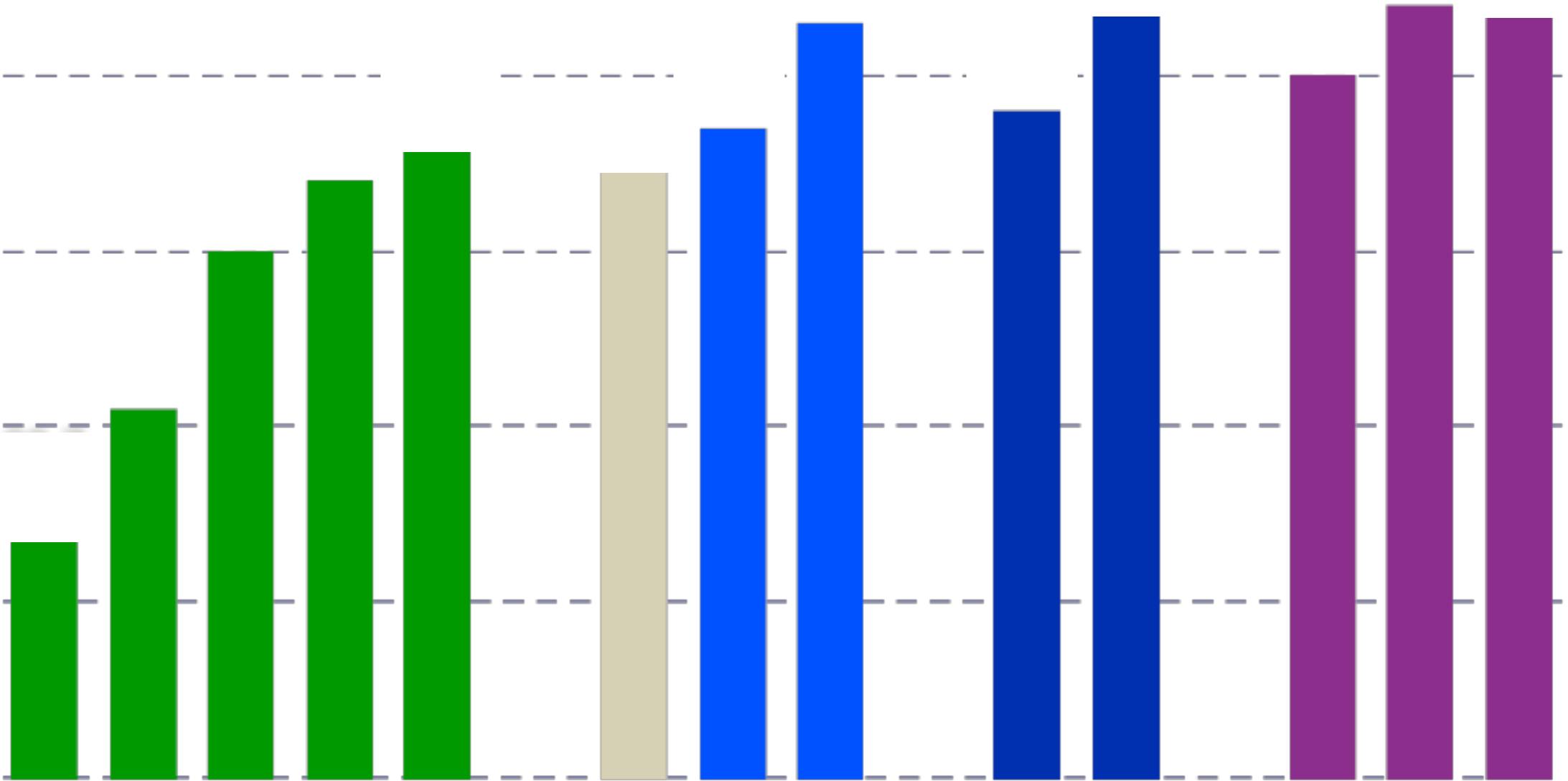
Motivation

(Google Research)

Thanks to Slav Petrov

Motivation

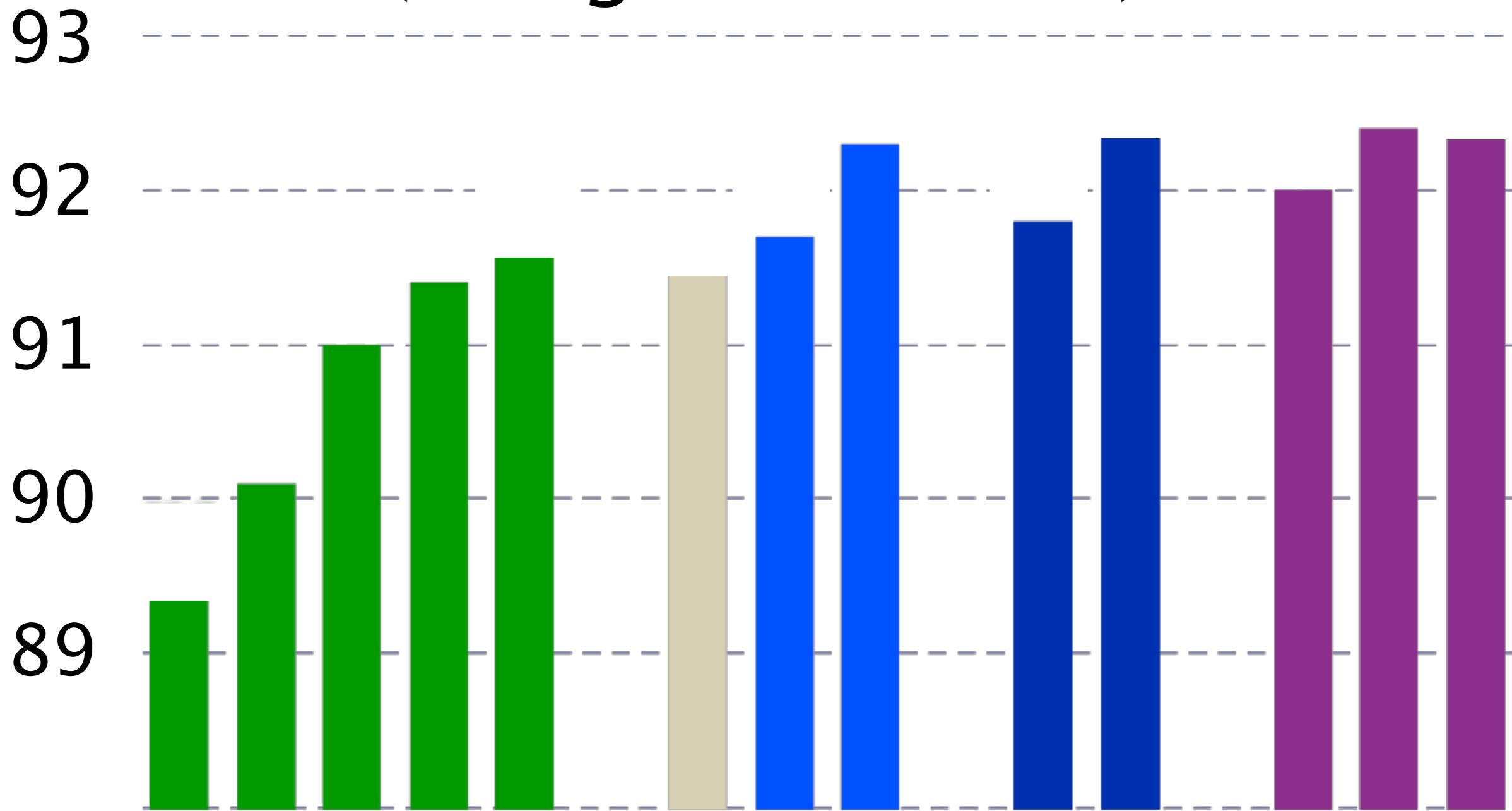
(Google Research)



English, By Slav Petrov

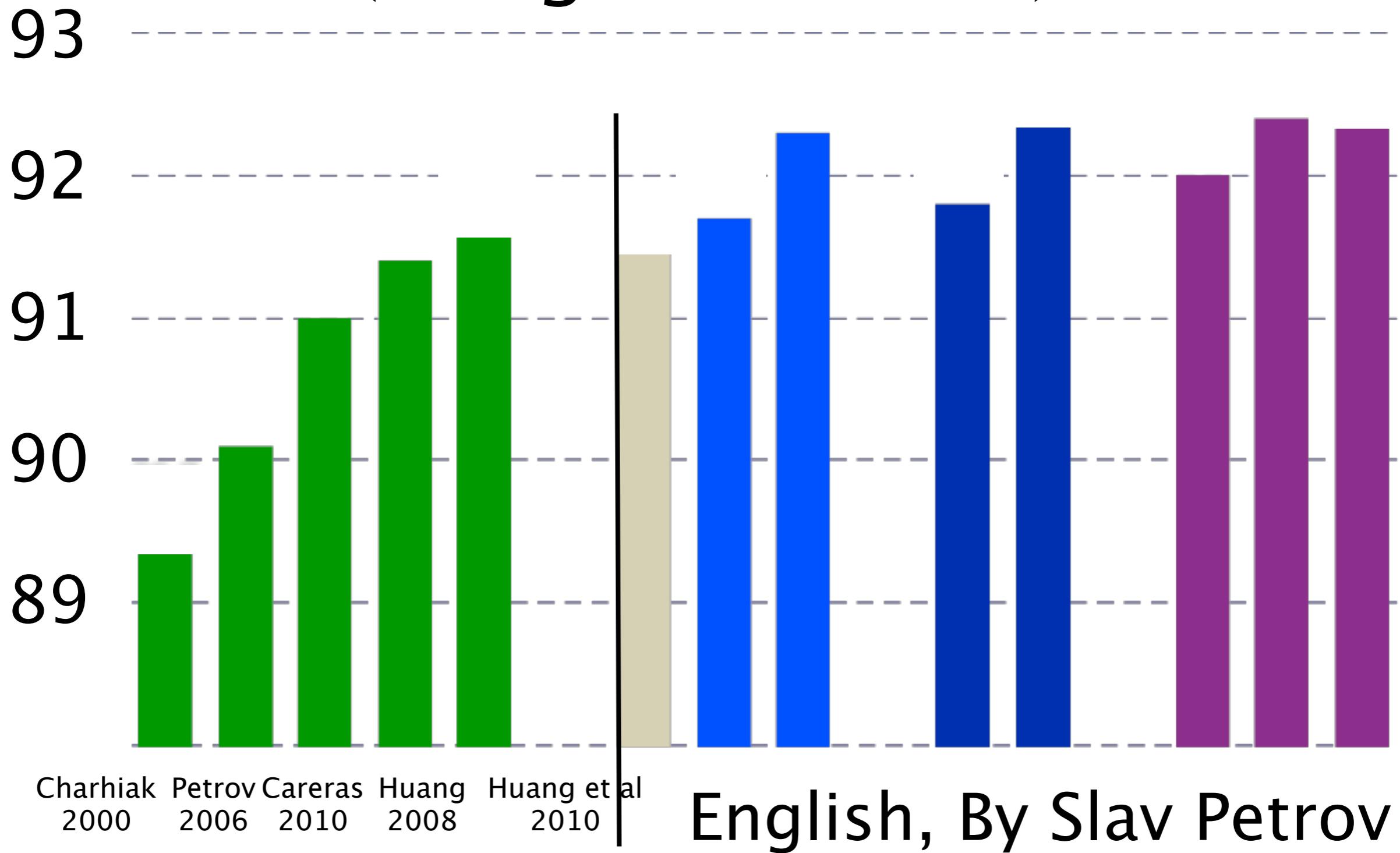
Motivation

(Google Research)

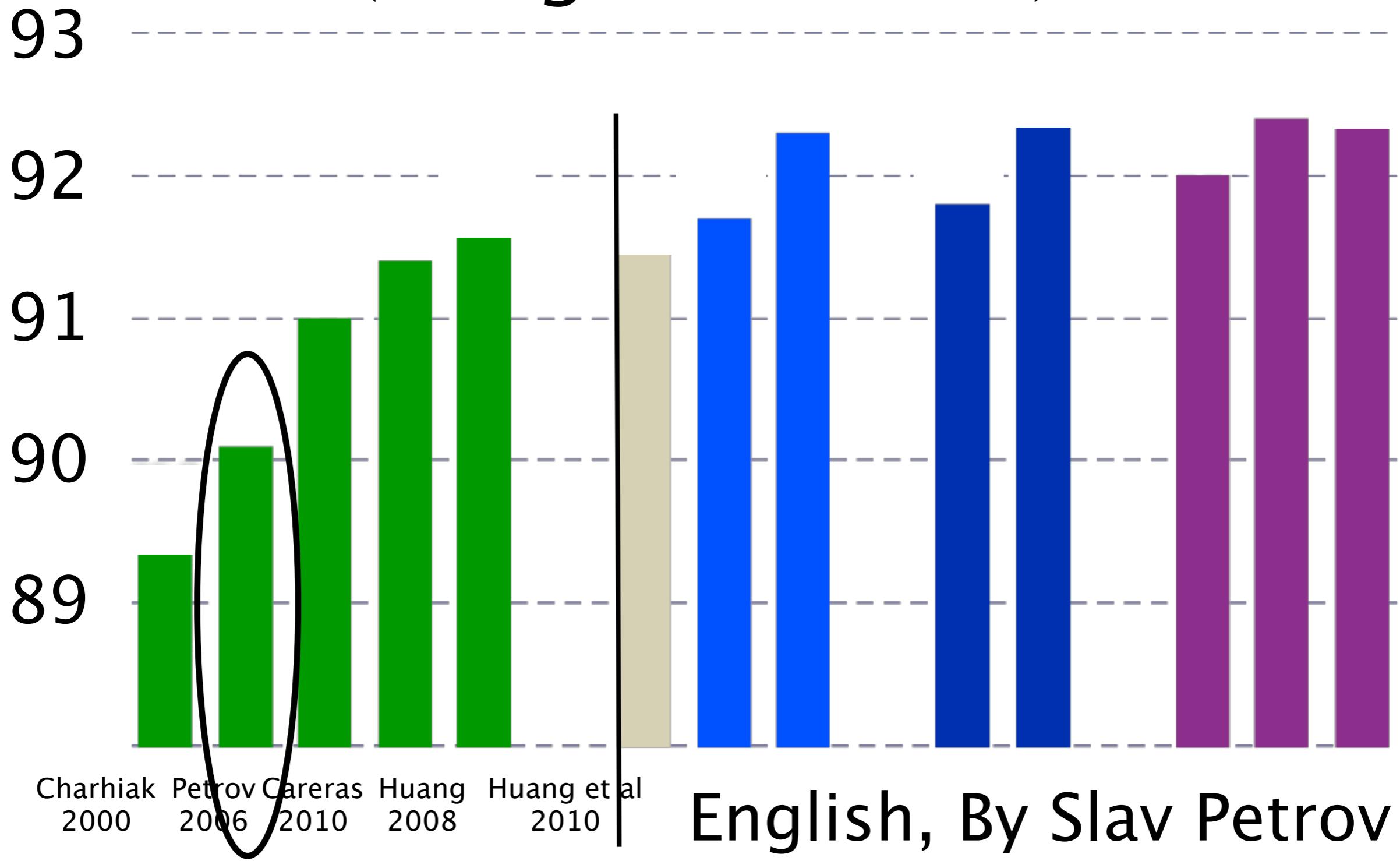


English, By Slav Petrov

Motivation (Google Research)

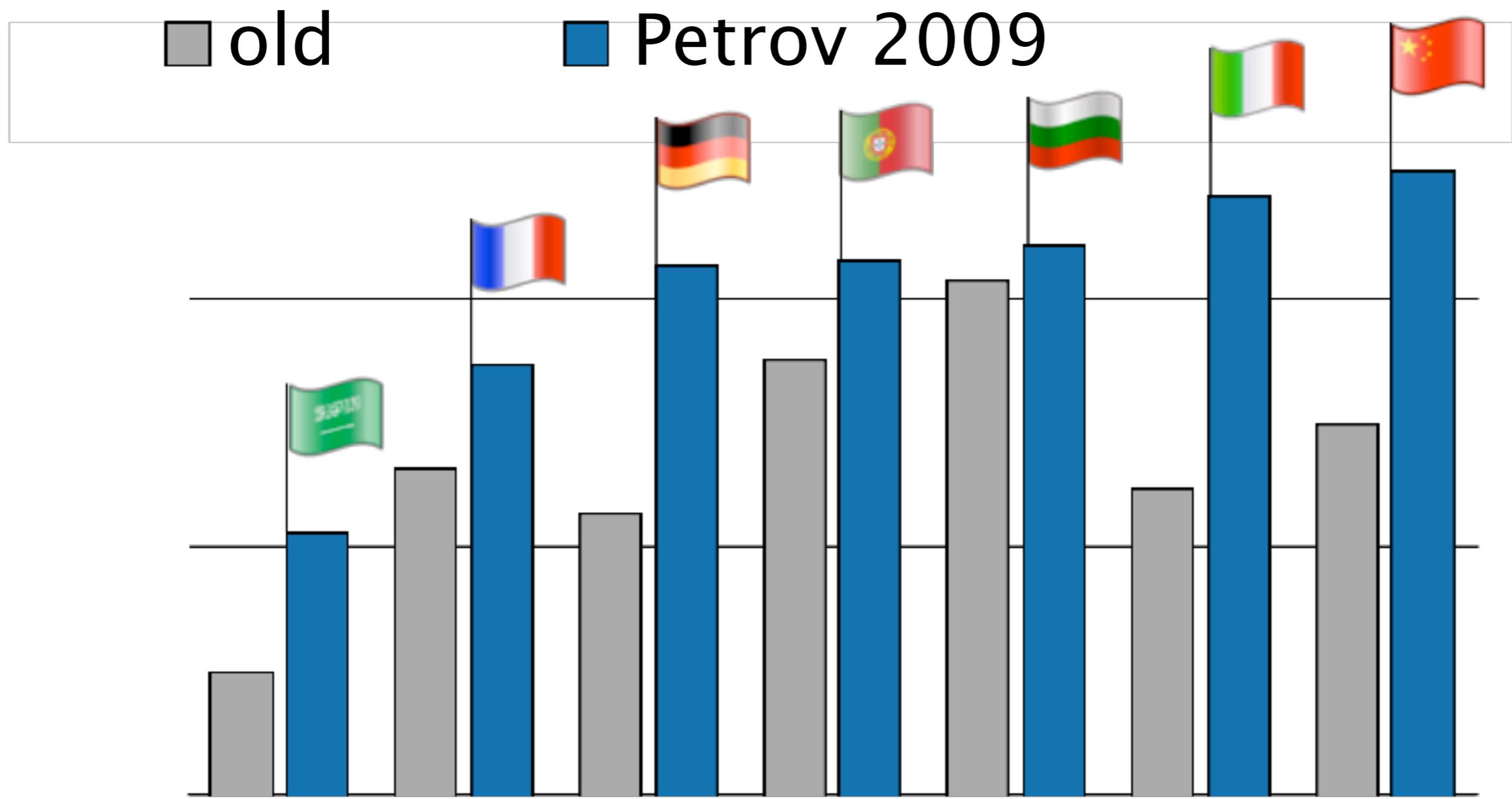


Motivation (Google Research)



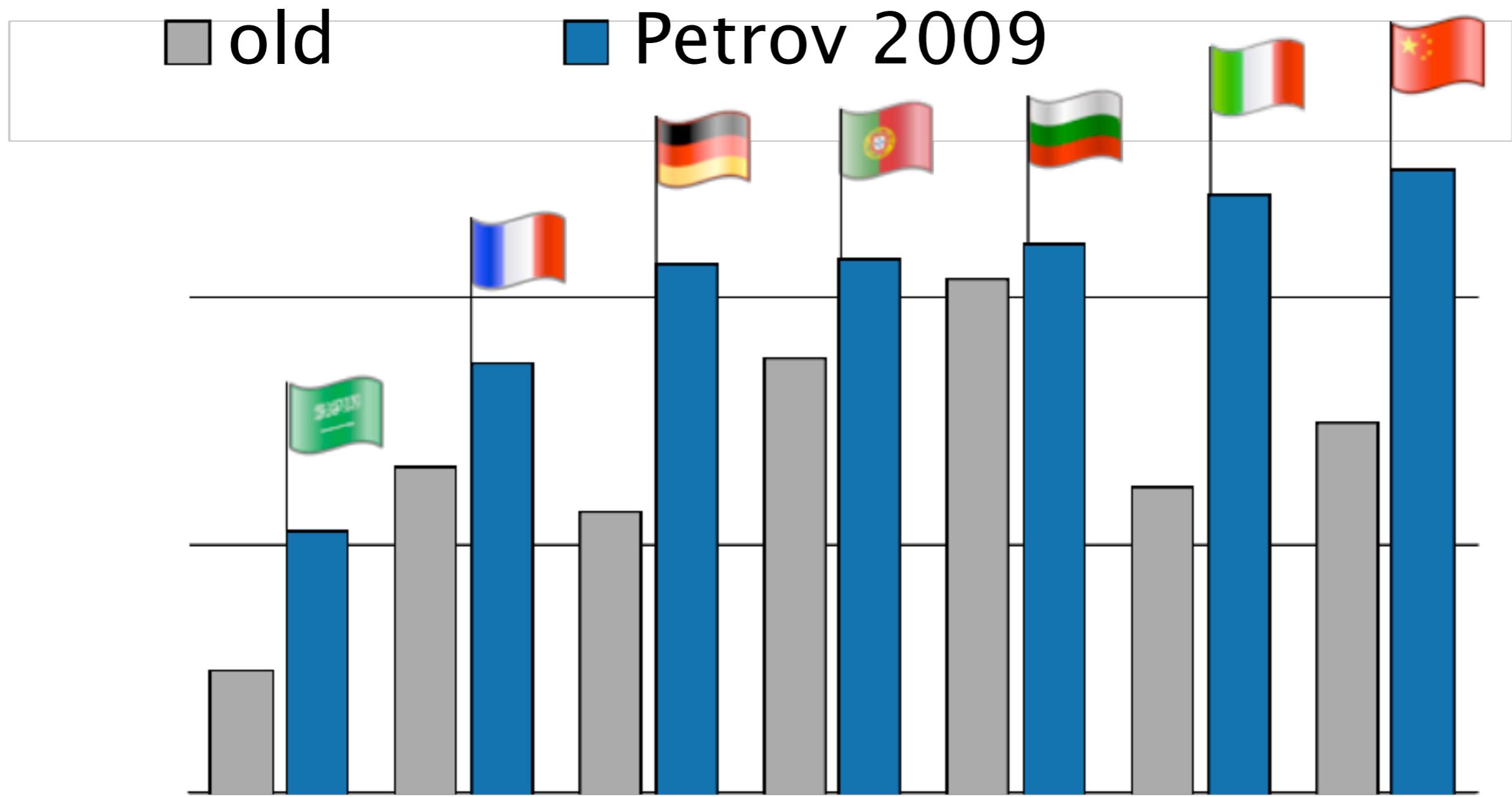
Motivation

(Google Research)



Motivation

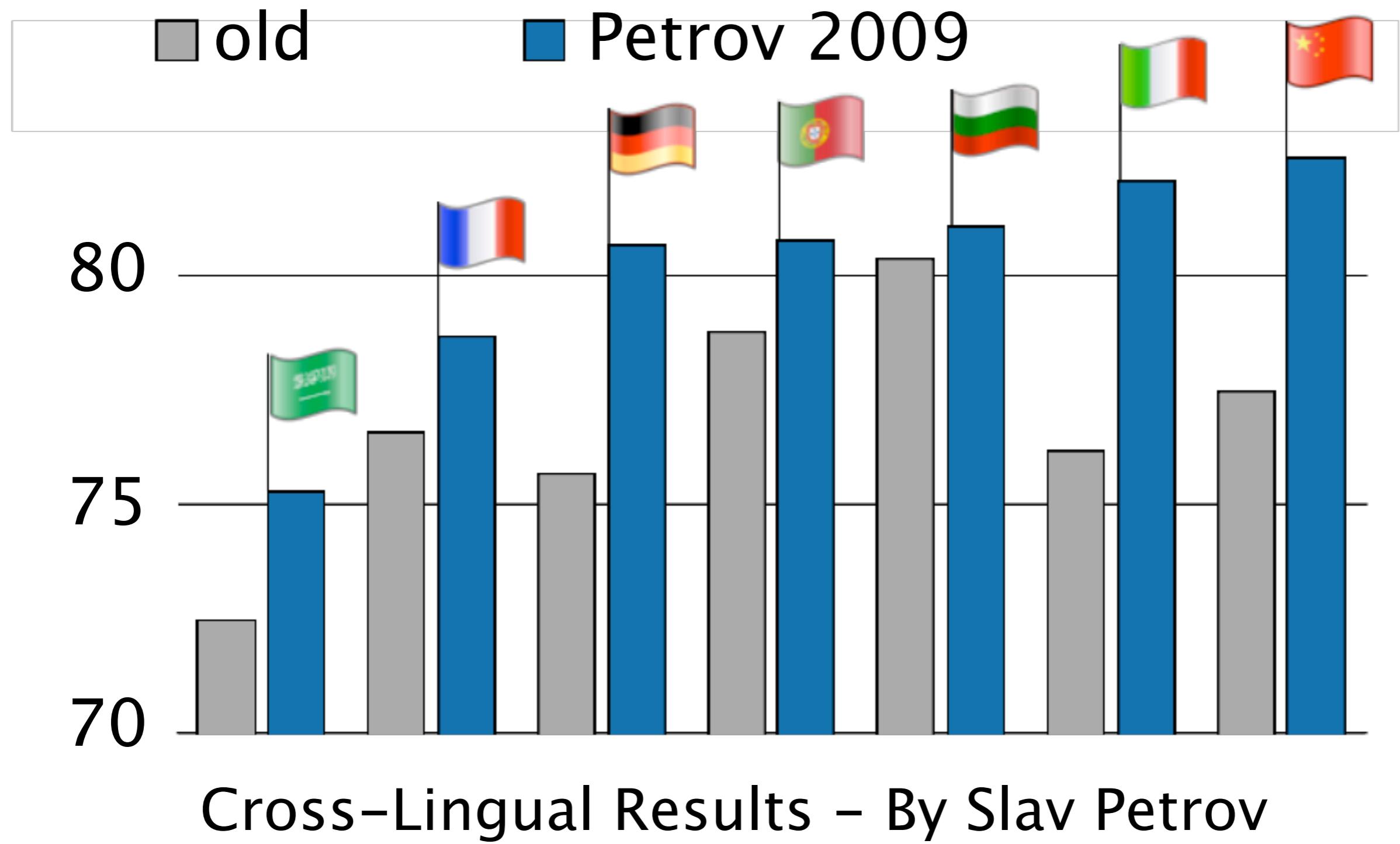
(Google Research)



Cross-Lingual Results – By Slav Petrov

Motivation

(Google Research)



Why is this so?

- Previously considered:



- More recently considered:



Why is this so?

- Previously considered:
 - Corpus Size
 - Annotation idiosyncracies
 - Evaluation Metrics
- More recently considered:



Why is this so?

- Previously considered:
 - Corpus Size
 - Annotation idiosyncracies
 - Evaluation Metrics
- More recently considered:
 - Language Type

Today@PMRL



Day 1: Introduction

→ Day 2: Phrase-structure (PS) Parsing

- PS Parsing in English
- PS Parsing in MRLs
- Day 3: Dependency-structure
- Day 4: Relational-Realizational
- Day 5: Evaluation and Multilinguality

Introducing English PS Parsing

Syntax (1)

We like Natural Language Processing

Syntax (1)

Noun

Verb

Adjec

Noun

Noun

We

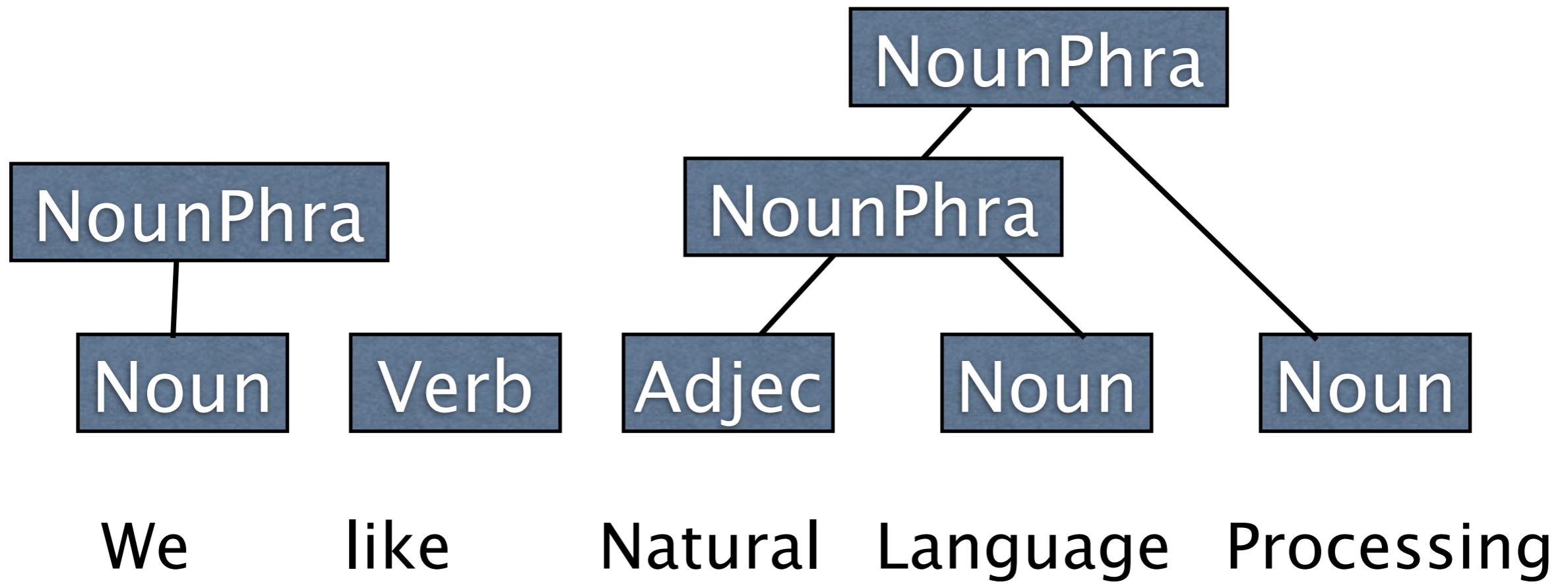
like

Natural

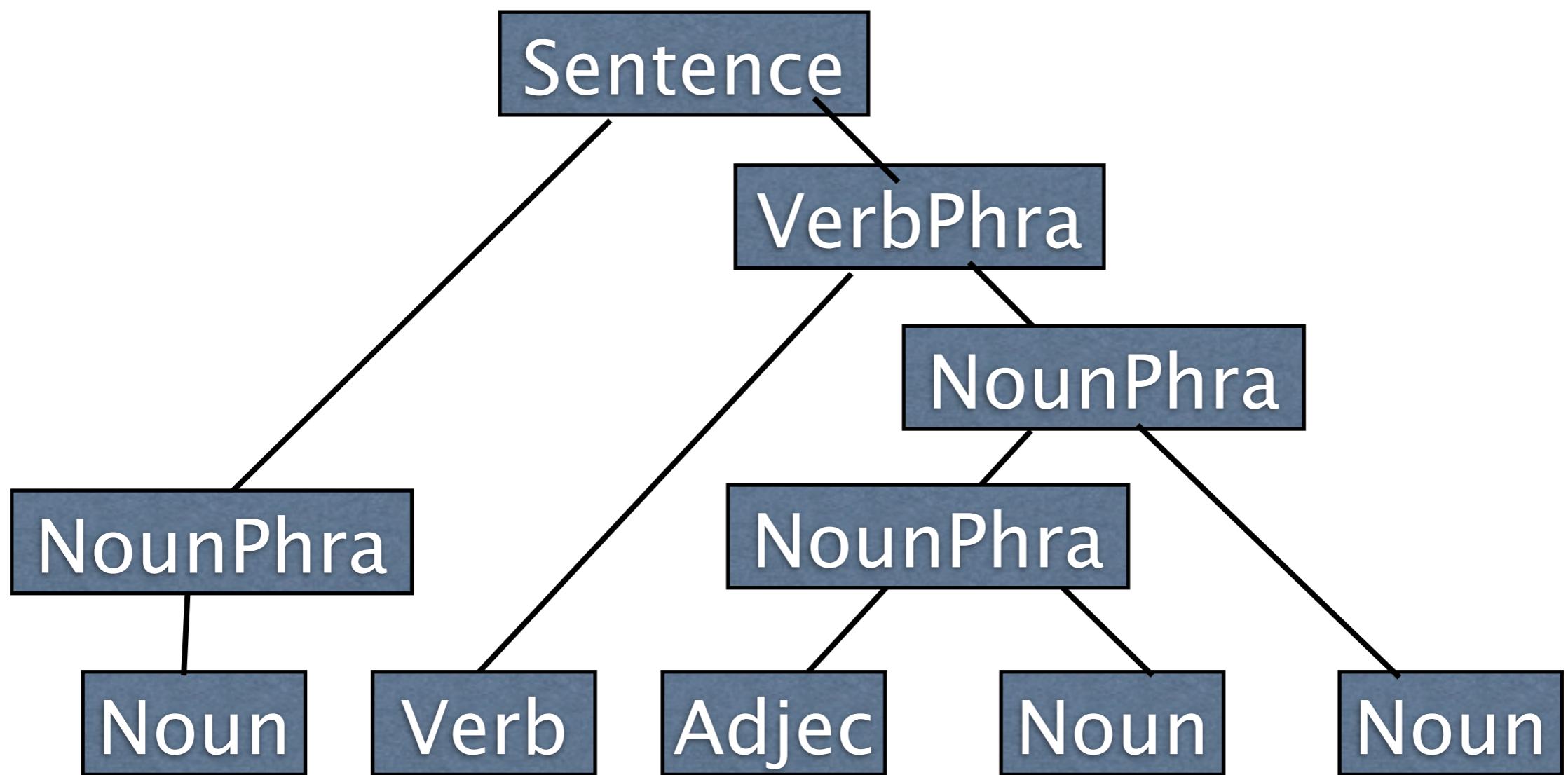
Language

Processing

Syntax (1)

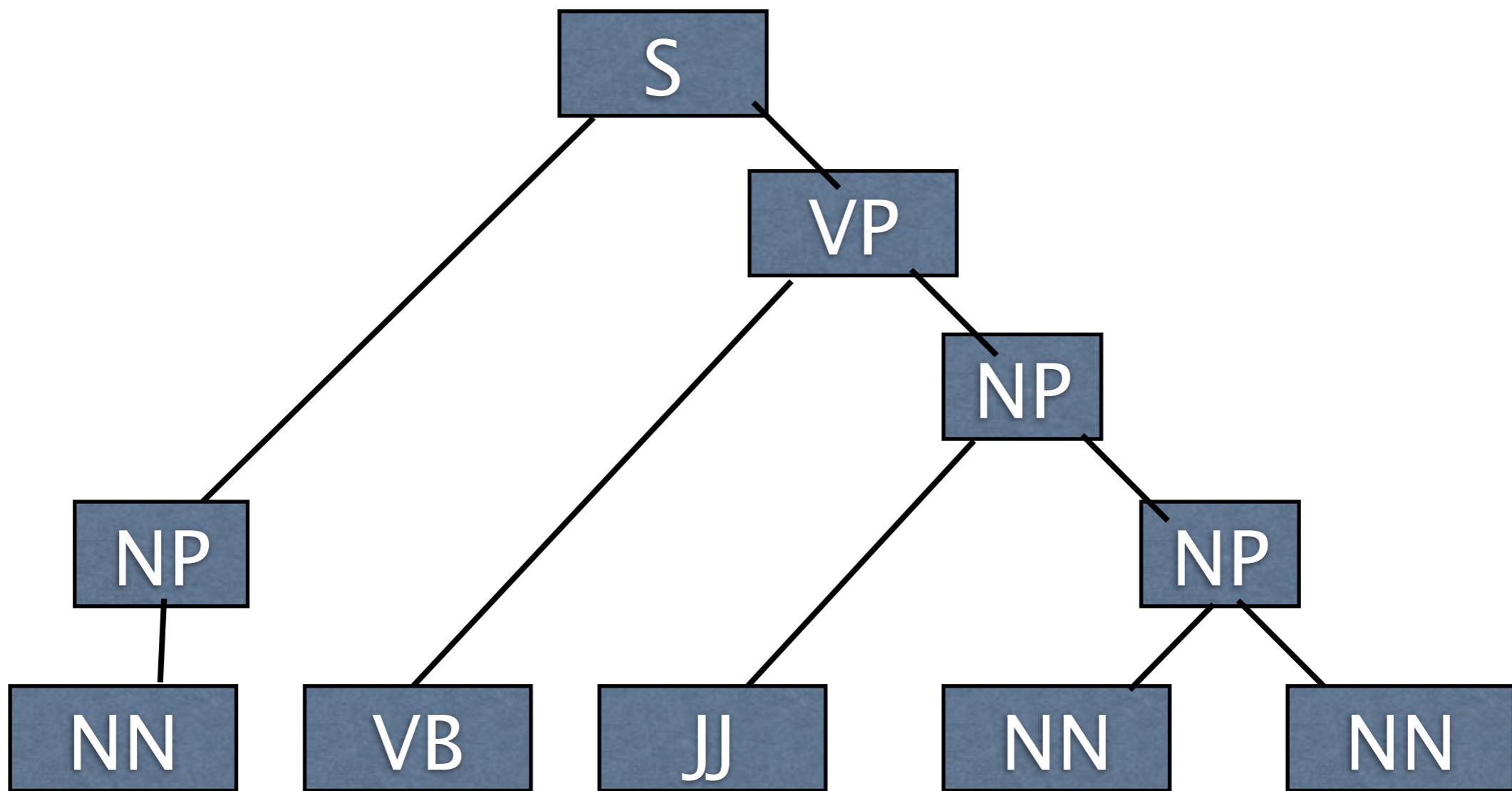


Syntax (1)



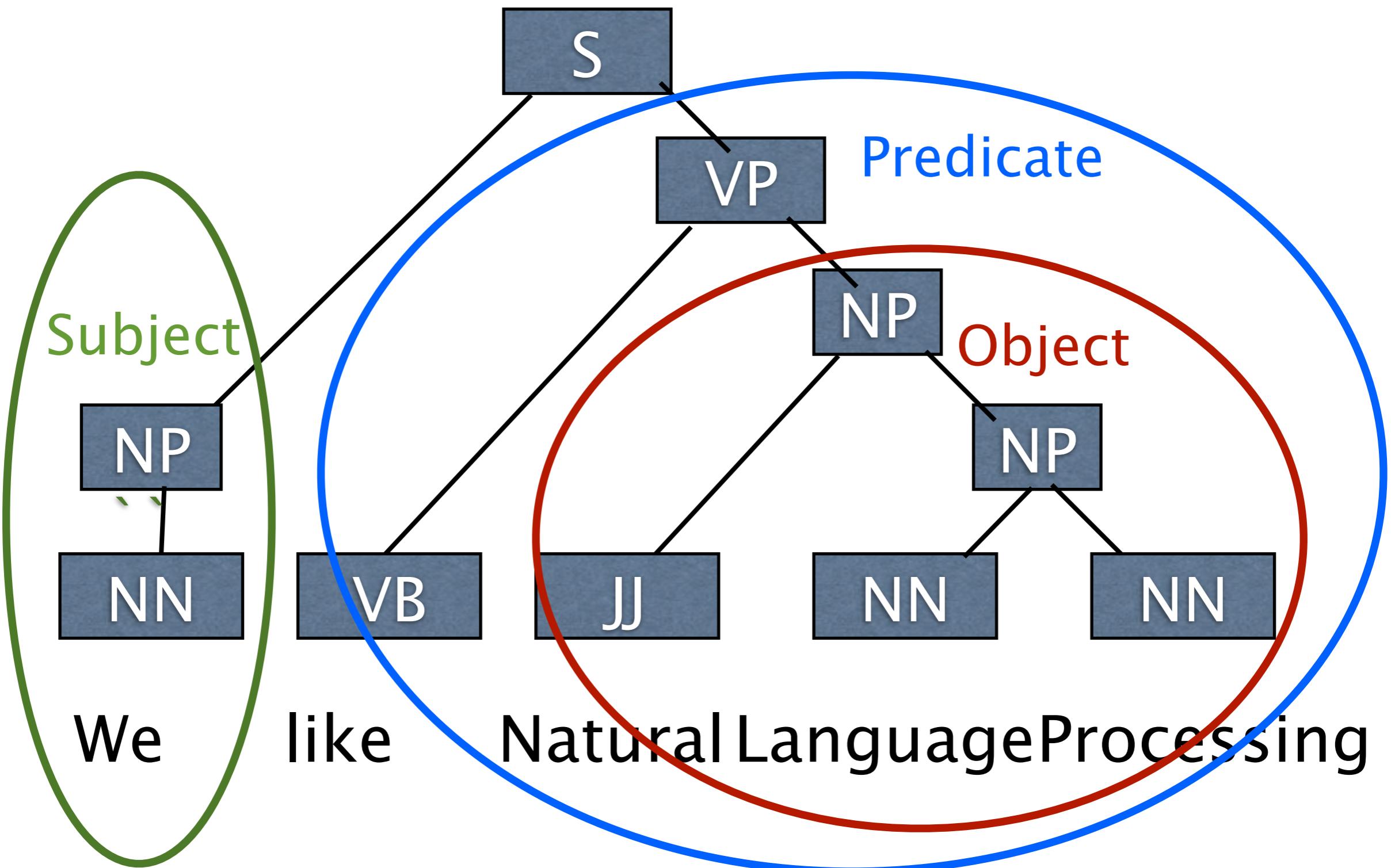
We like Natural Language Processing

Syntax (2)



We like Natural Language Processing

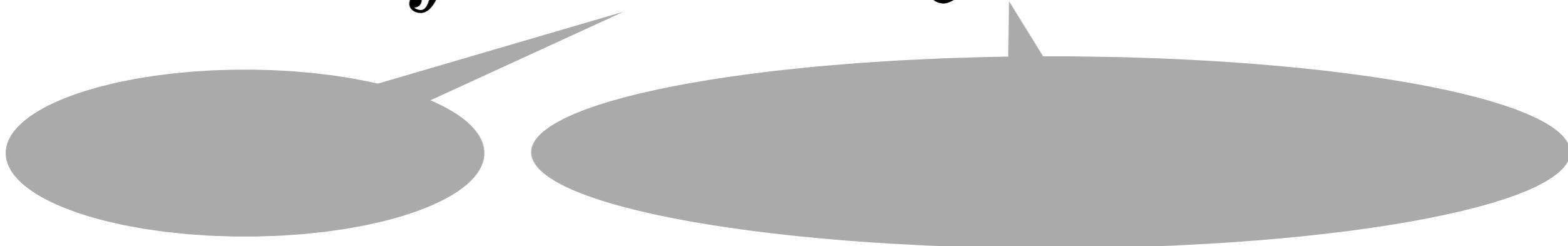
Interfacing Semantics



Representation

- Assume:
 - A finite vocabulary
 - Phrase-structure trees
- Then:

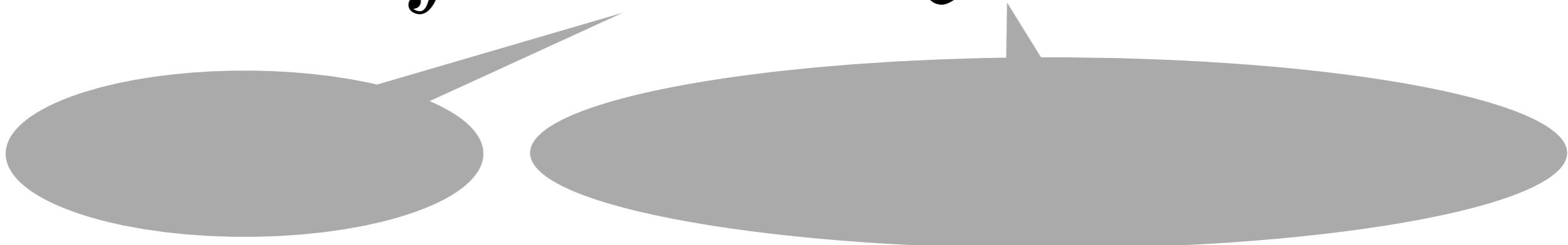
$$f : \mathcal{X} \rightarrow \mathcal{Y}$$



Representation

- Assume:
 - A finite vocabulary Σ
 - Phrase-structure trees
- Then:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$



Representation

- Assume:

- A finite vocabulary

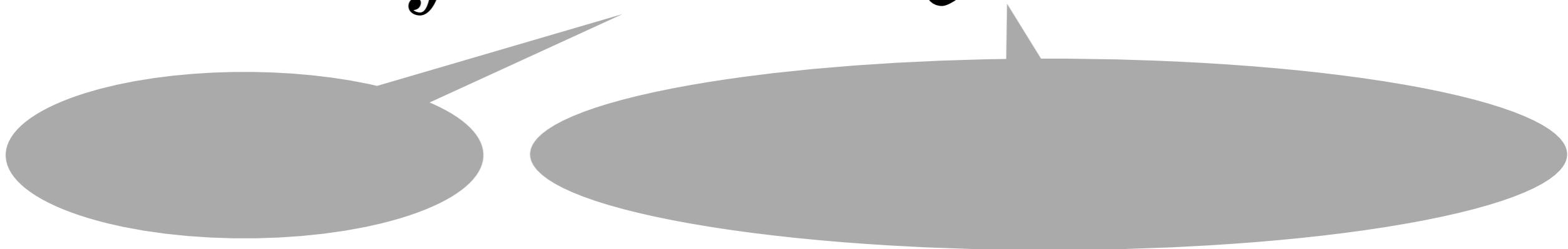
 Σ

- Phrase-structure trees

 \mathcal{G}

- Then:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$



Representation

- Assume:

- A finite vocabulary

$$\Sigma$$

- Phrase-structure trees

$$\mathcal{G}$$

- Then:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$


$$\mathcal{X} = \Sigma^*$$

Representation

- Assume:

- A finite vocabulary

 Σ

- Phrase-structure trees

 \mathcal{G}

- Then:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

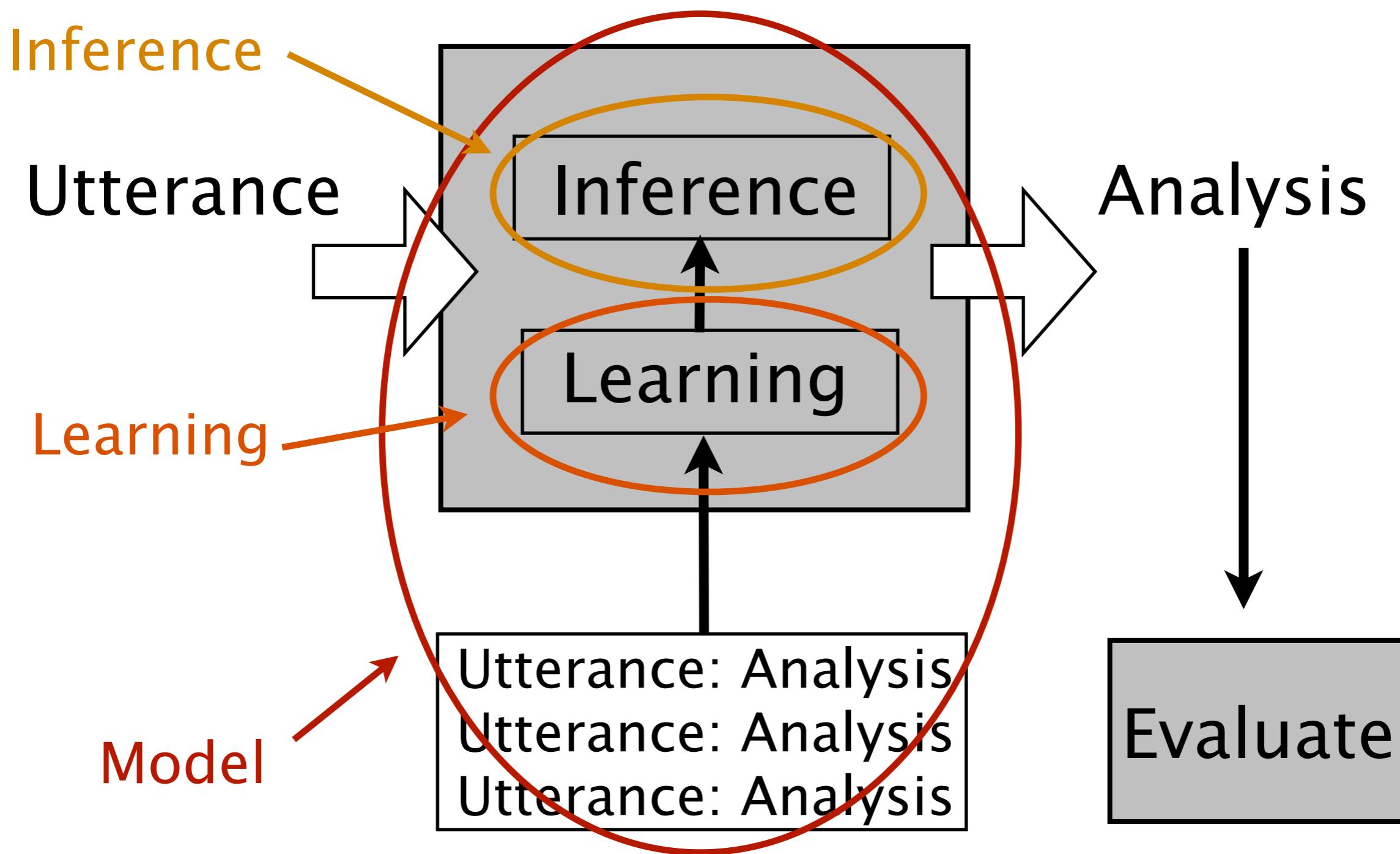
$$\mathcal{X} = \Sigma^*$$

$$\mathcal{Y} = \{y \mid \text{Root}(y) = S, \text{Yield}(y) \in \Sigma^*\}$$

Architectural Decisions

- Representation: Phrase-Structures
- Model: ?
- Learning: ?
- Inference: ?
- Evaluation: ?

Statistical Parsing



Modeling

- Probabilistic Modeling

$$f(x) = \arg \max_{y \in Y} P(y|x)$$

Modeling

- Probabilistic Modeling

$$\begin{aligned} f(x) &= \arg \max_{y \in Y} P(y|x) \\ &= \arg \max_{y \in Y} \frac{P(y, x)}{P(x)} \end{aligned}$$

Modeling

- Probabilistic Modeling

$$\begin{aligned} f(x) &= \arg \max_{y \in Y} P(y|x) \\ &= \arg \max_{y \in Y} \frac{P(y, x)}{P(x)} \\ &= \arg \max_{y \in Y} P(y, x) \end{aligned}$$

Modeling

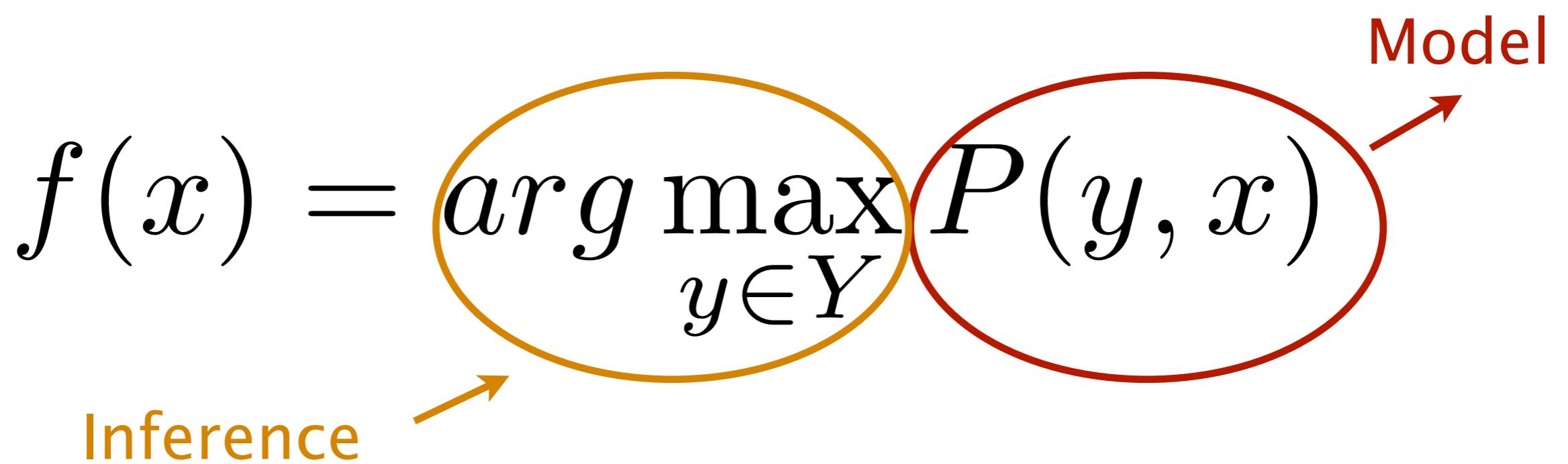
- Probabilistic Generative Modeling

$$f(x) = \arg \max_{y \in Y} P(y, x)$$

Model

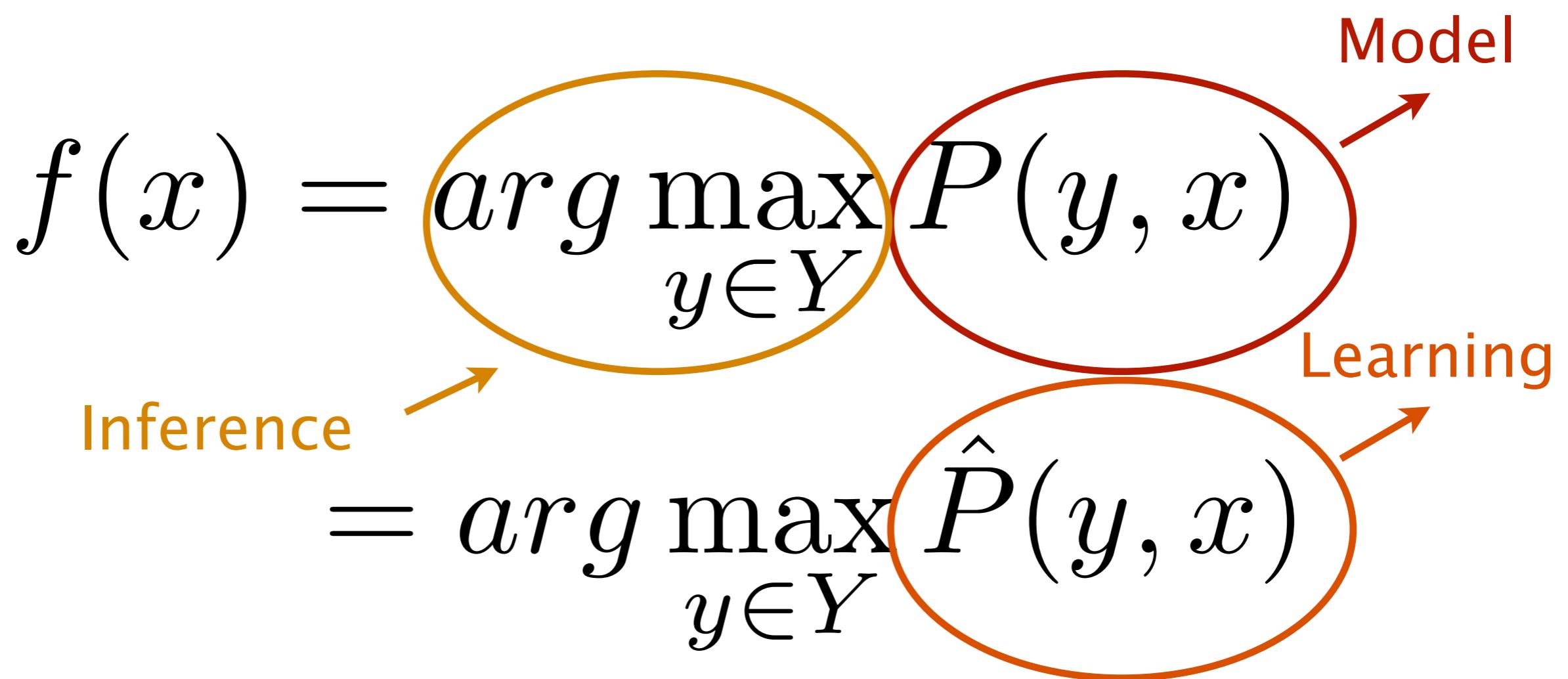
Modeling

- Probabilistic Generative Modeling



Modeling

- Probabilistic Generative Modeling



Generative Grammars

- "Generate": A Generative Grammar
 - Generates possible trees $\{y|x = \text{Yield}(y)\}$
 - "Assign - Assigns probabilities for $\{\langle x, y \rangle | x = \text{Yield}(y)\}$
- "Learn": A Treebank Grammar
 - Learns weights from $\{\langle x_i, y_i \rangle | x_i = \text{Yield}(y_i)\}$

Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$

Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$

Non-
Terminals

Context-Free Grammars

Terminals
 $G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$
Non-Terminals

Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$

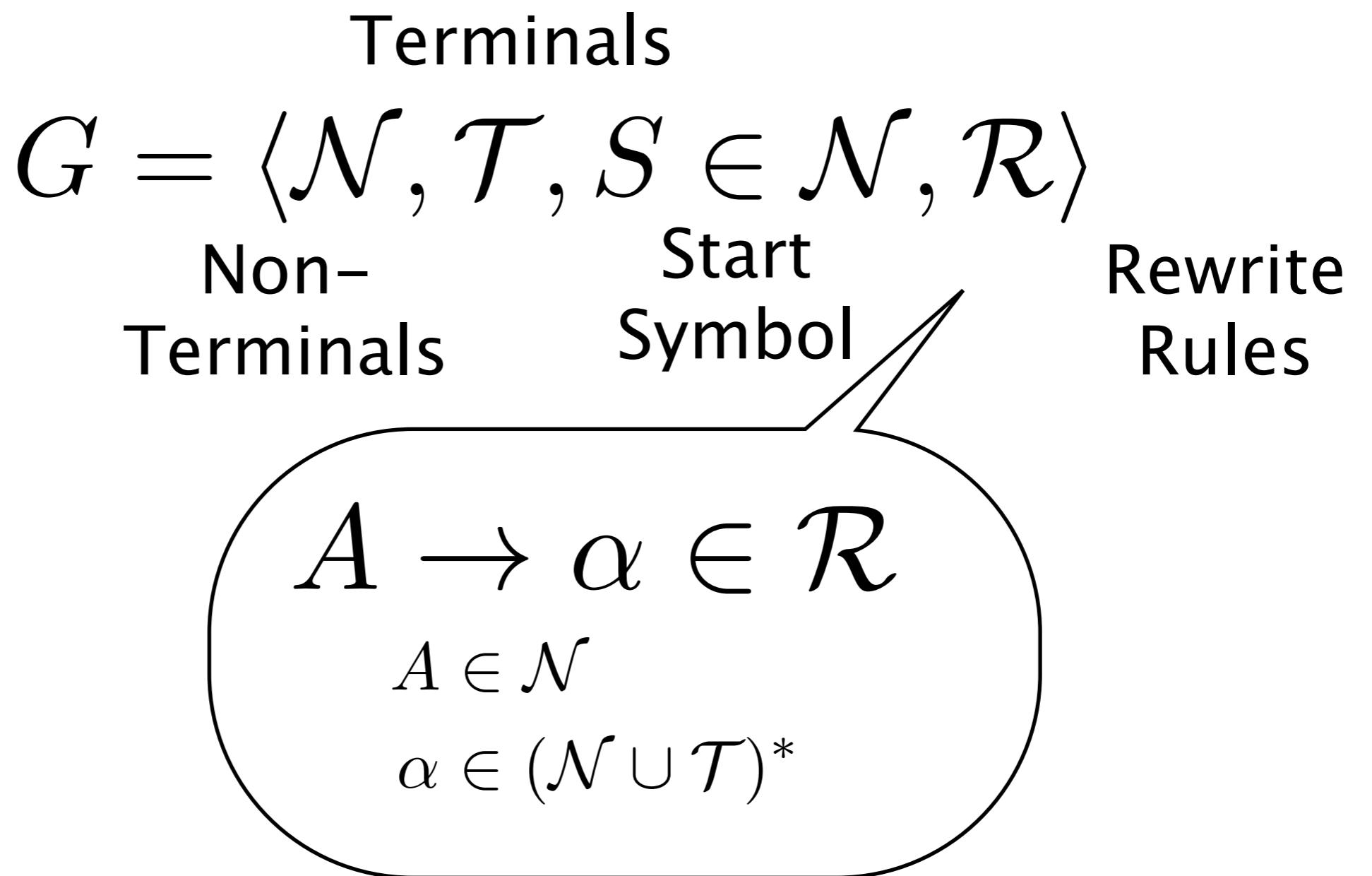
Terminals
Non-Terminals Start Symbol

Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$

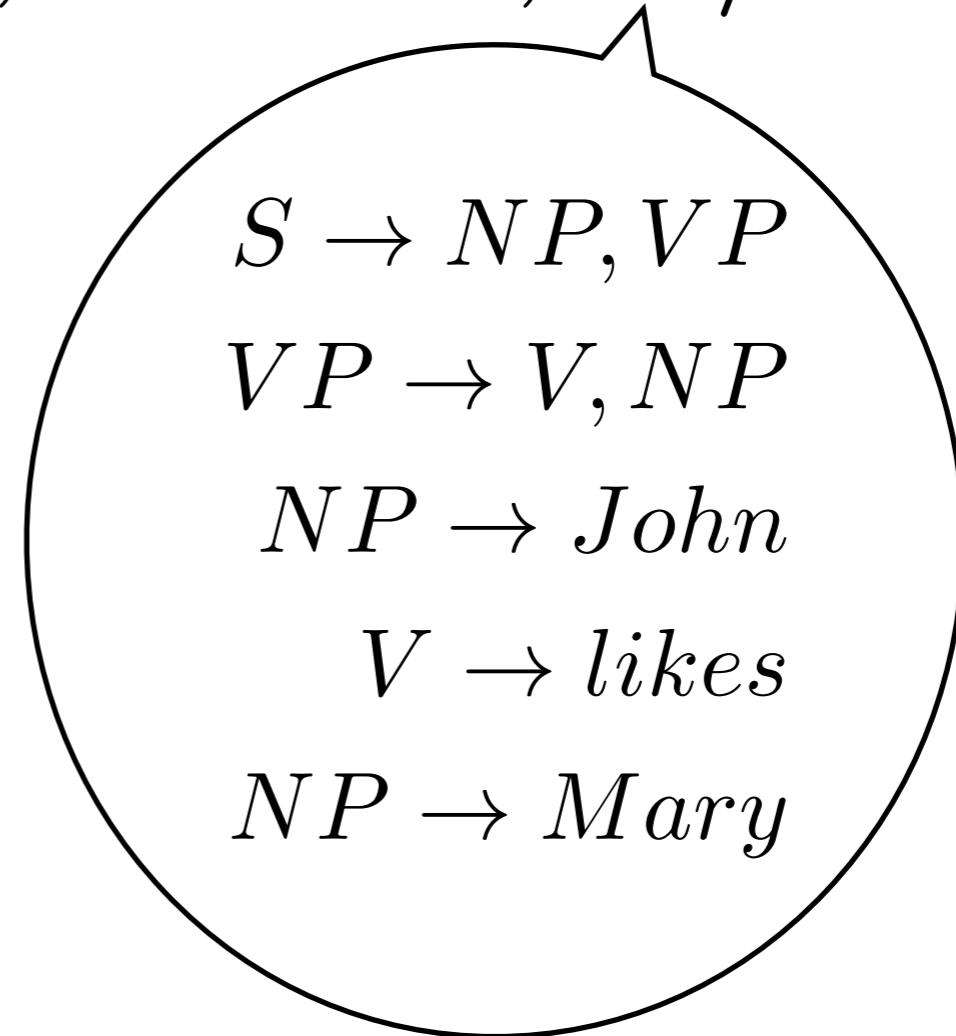
Terminals
Non-Terminals Start Symbol Rewrite Rules

Context-Free Grammars



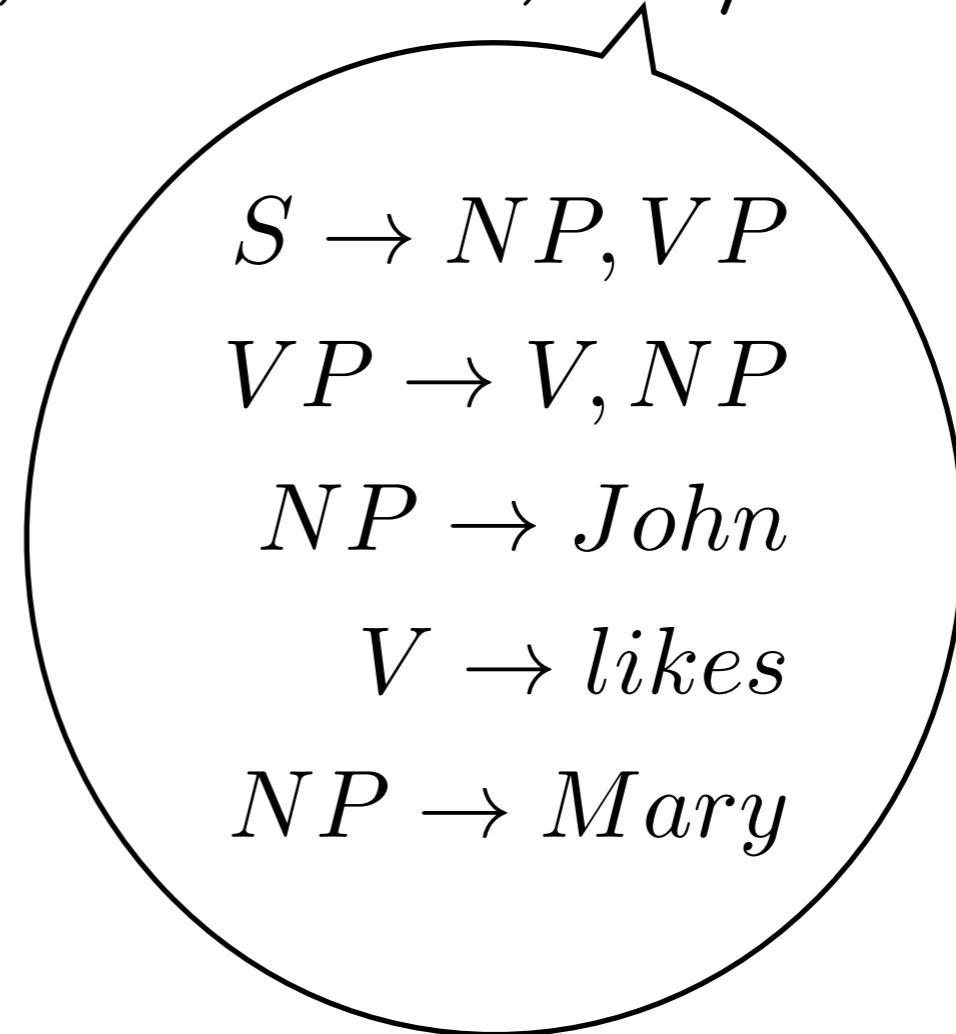
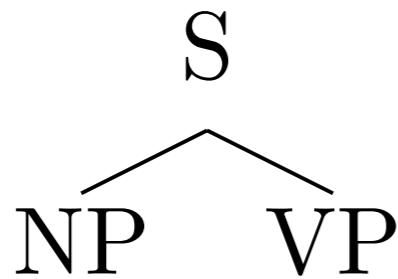
Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$

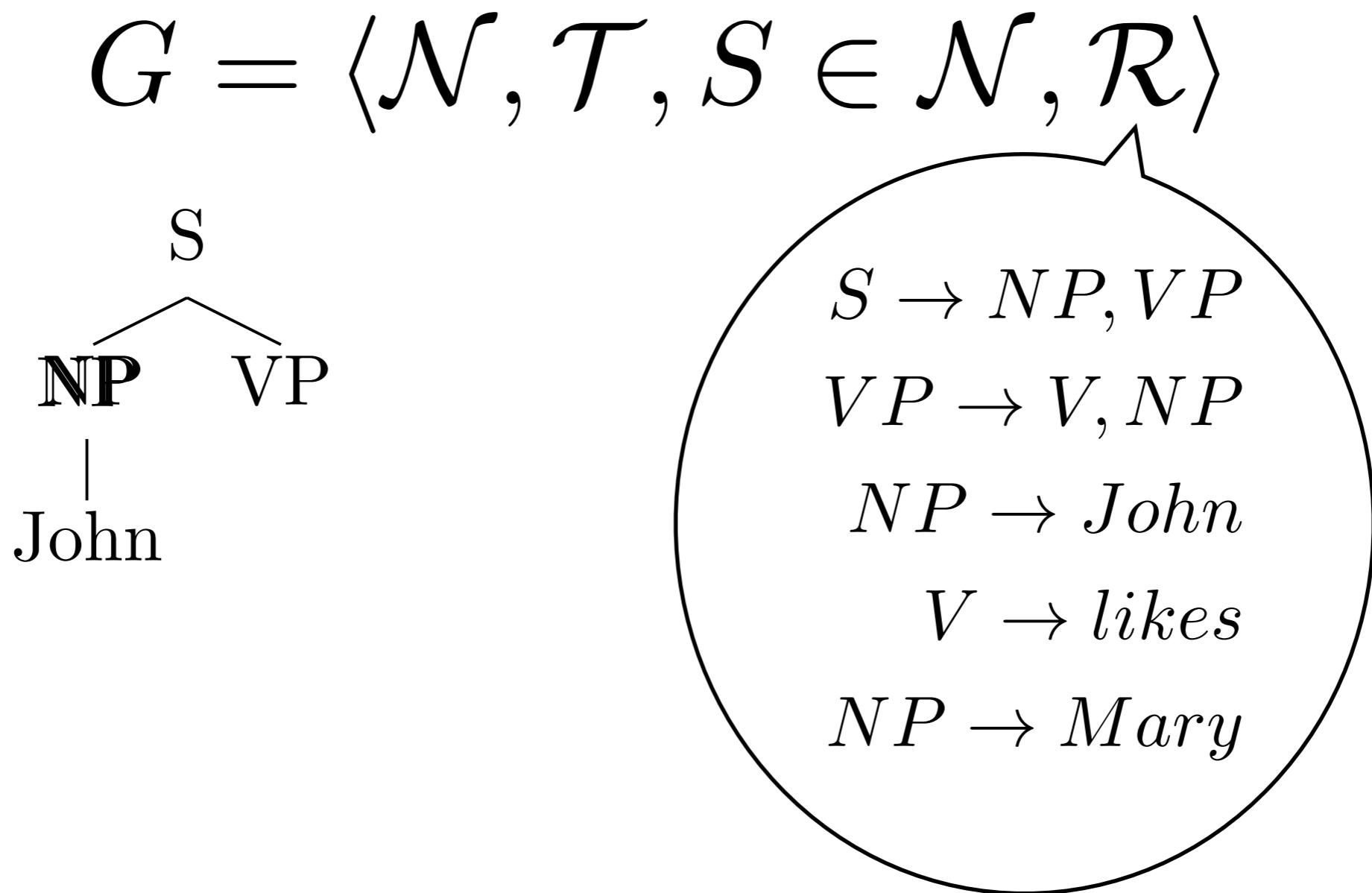


Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$

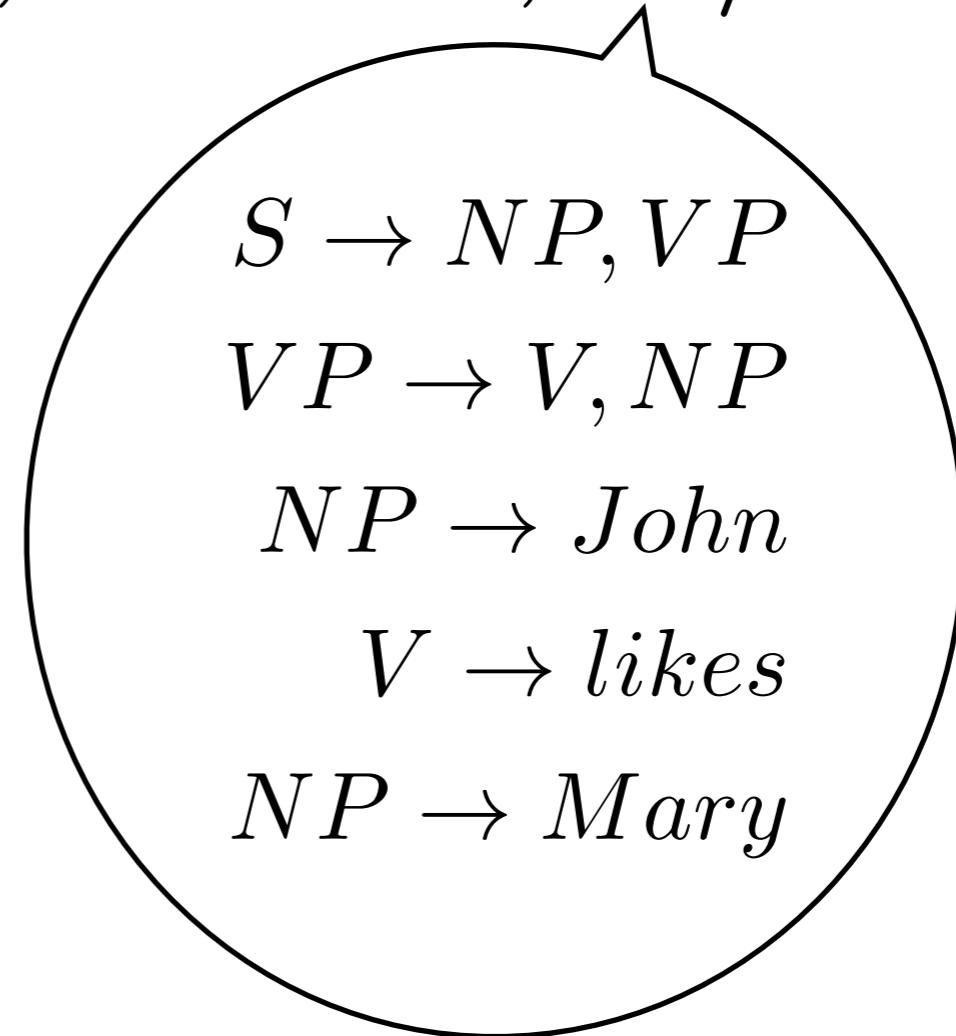
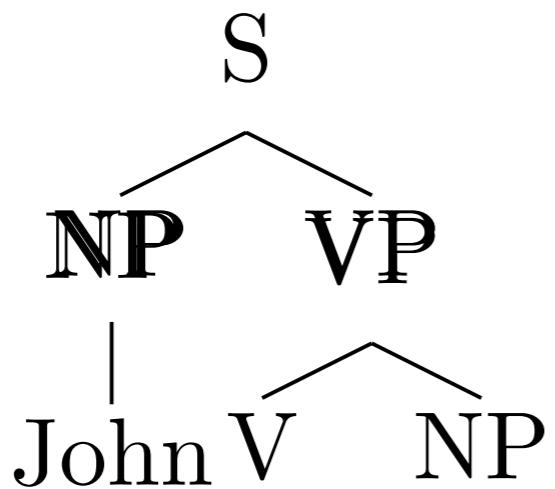


Context-Free Grammars



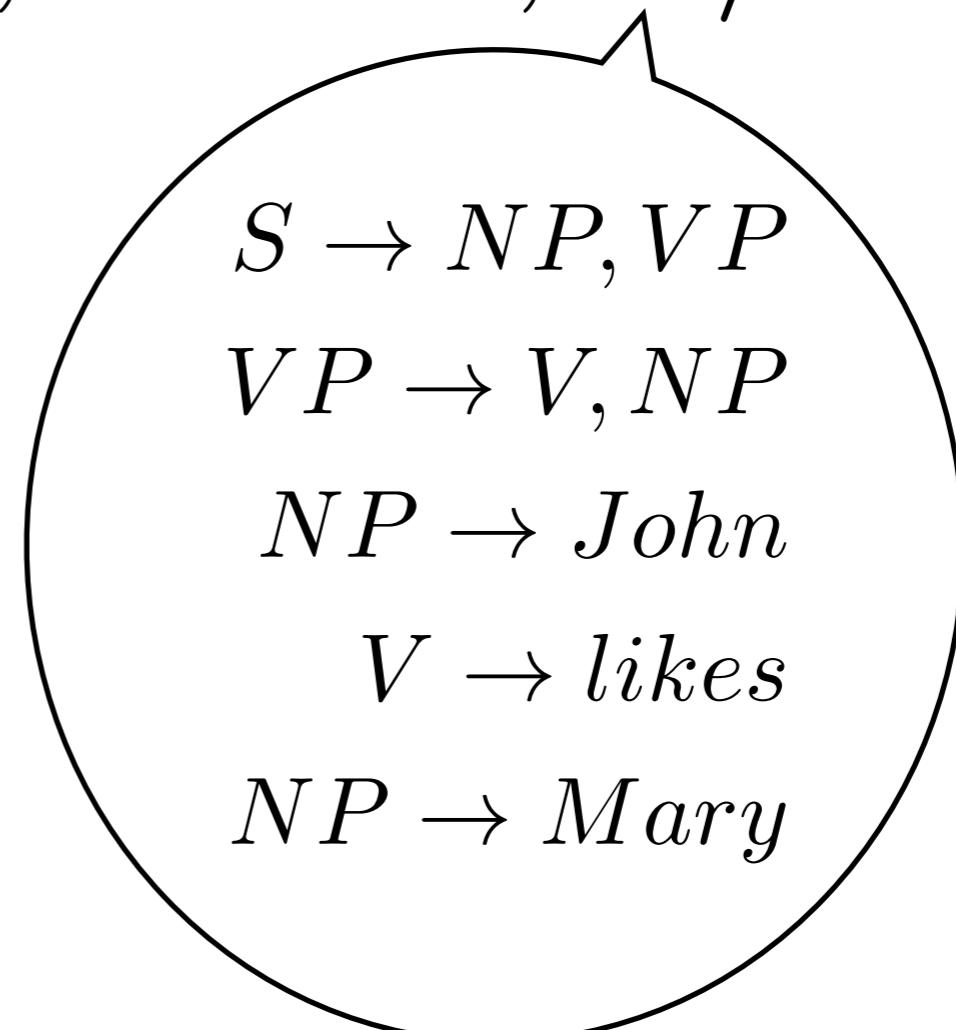
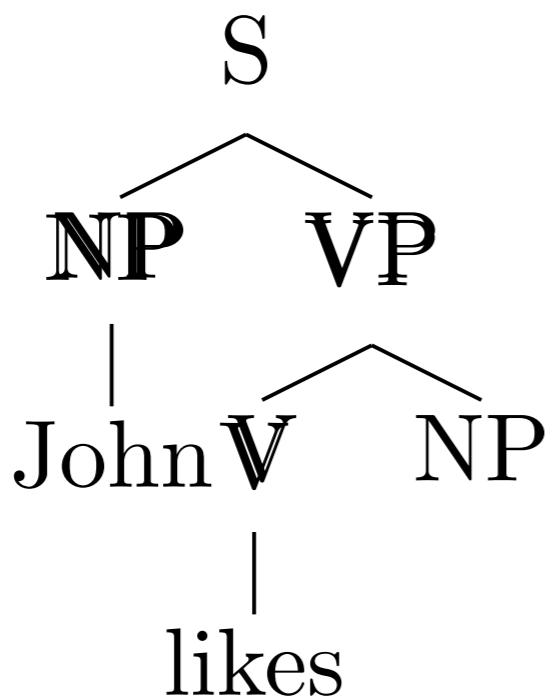
Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$



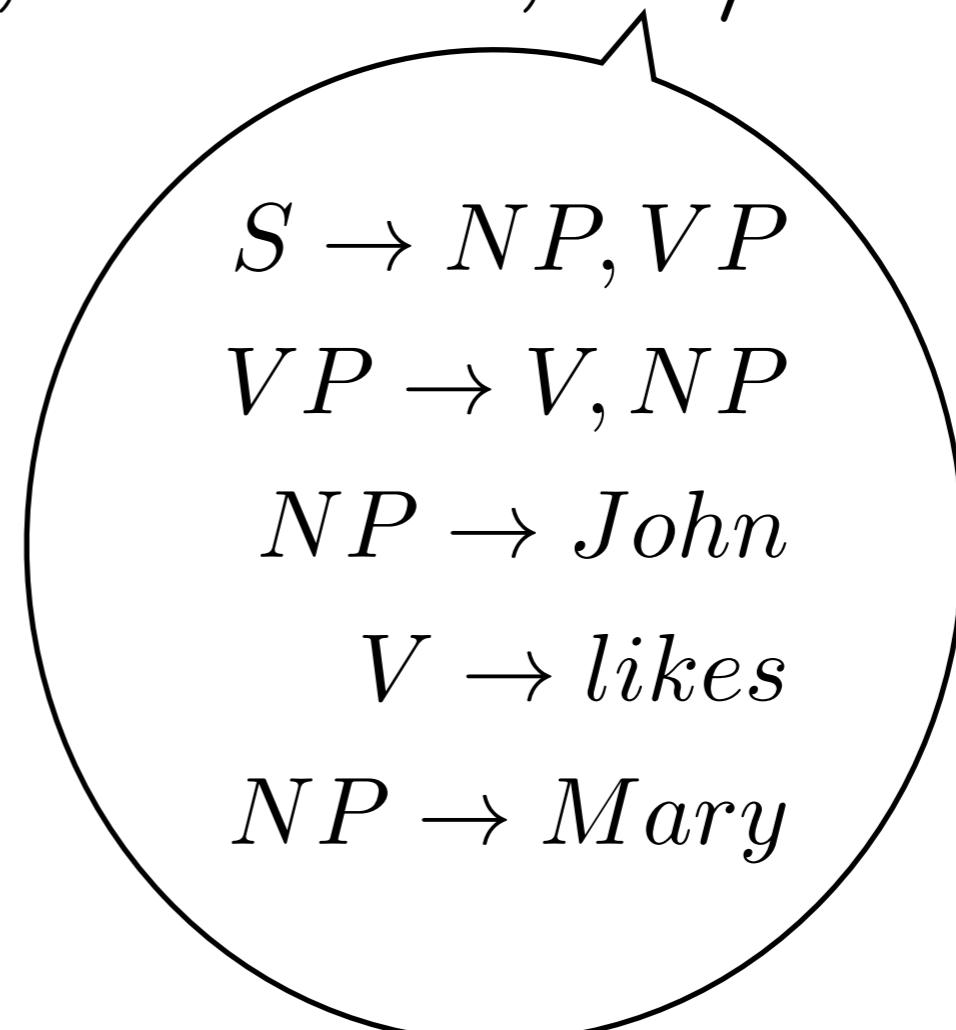
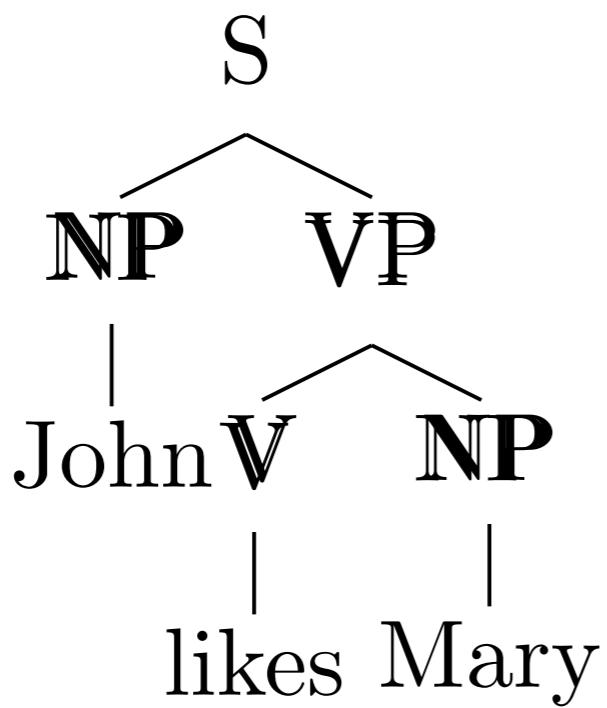
Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$



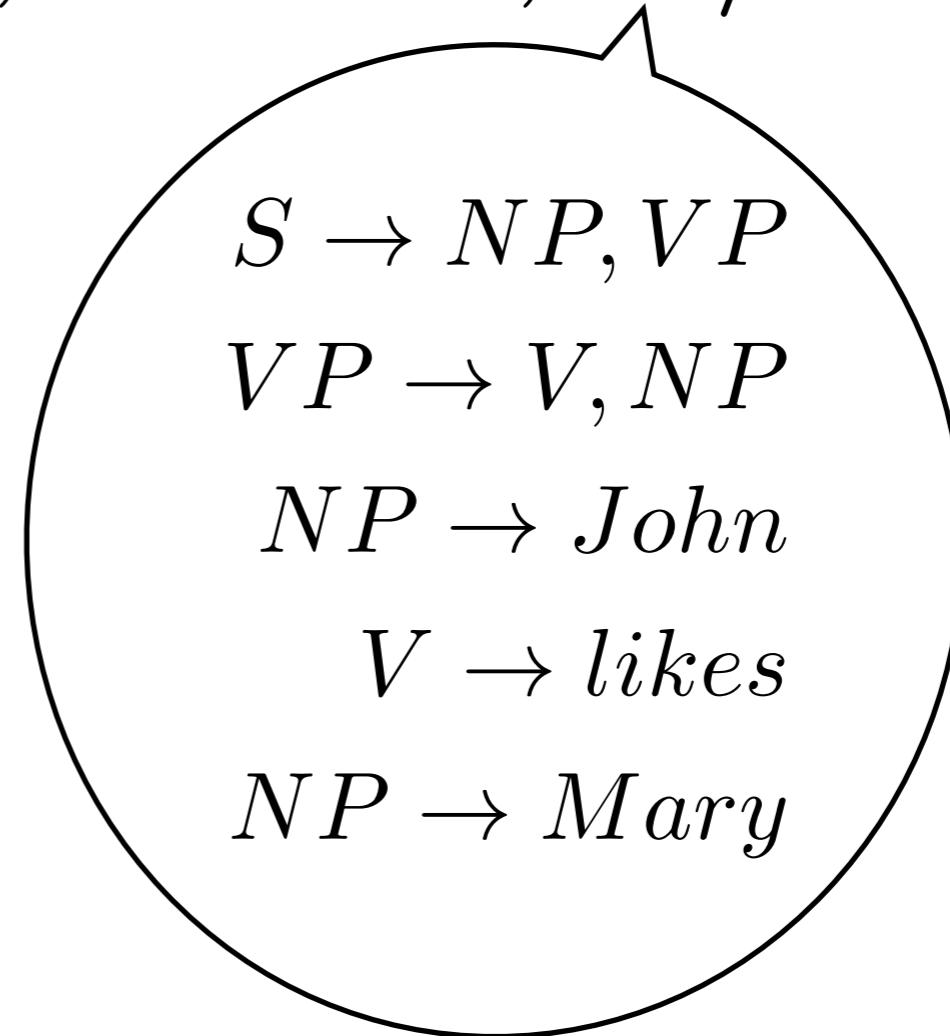
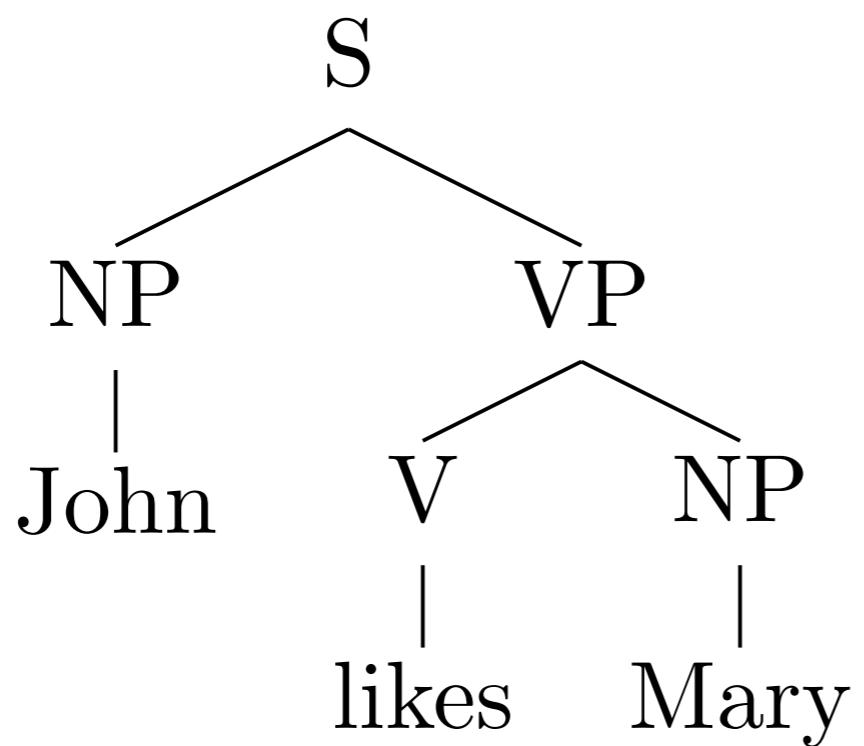
Context-Free Grammars

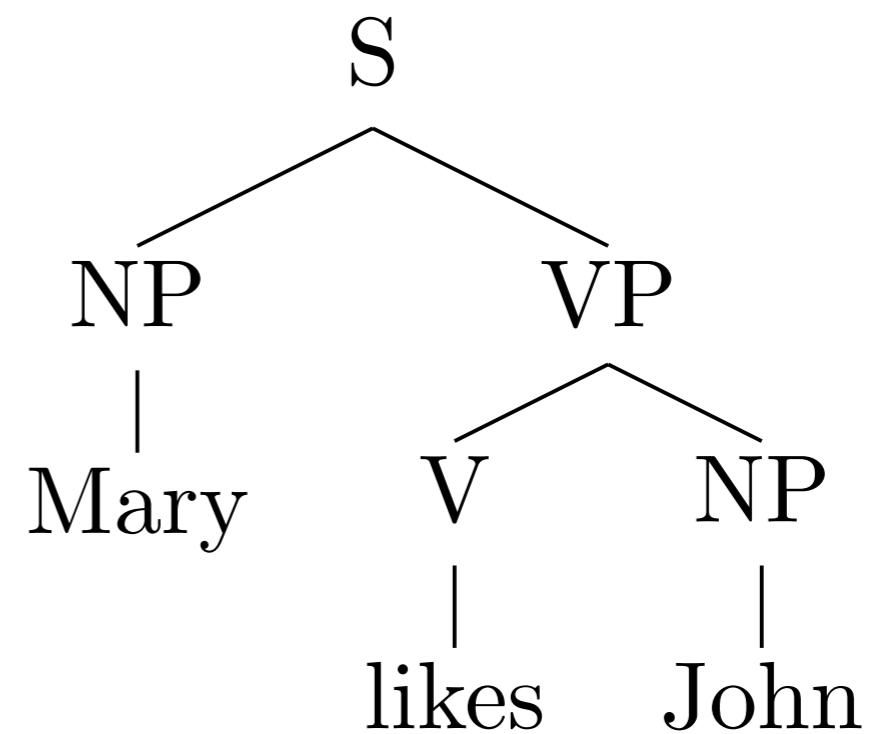
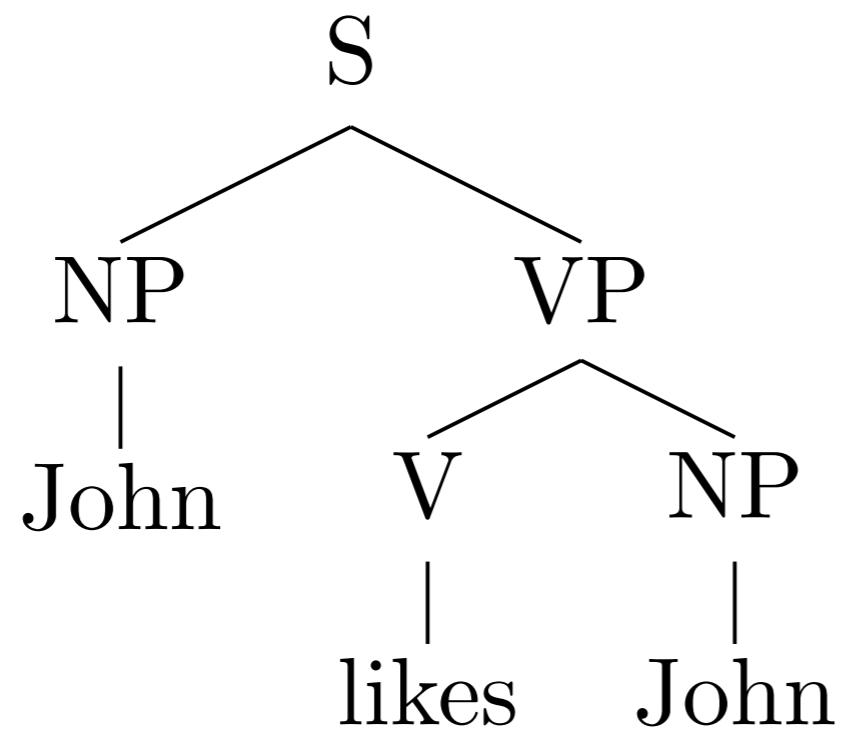
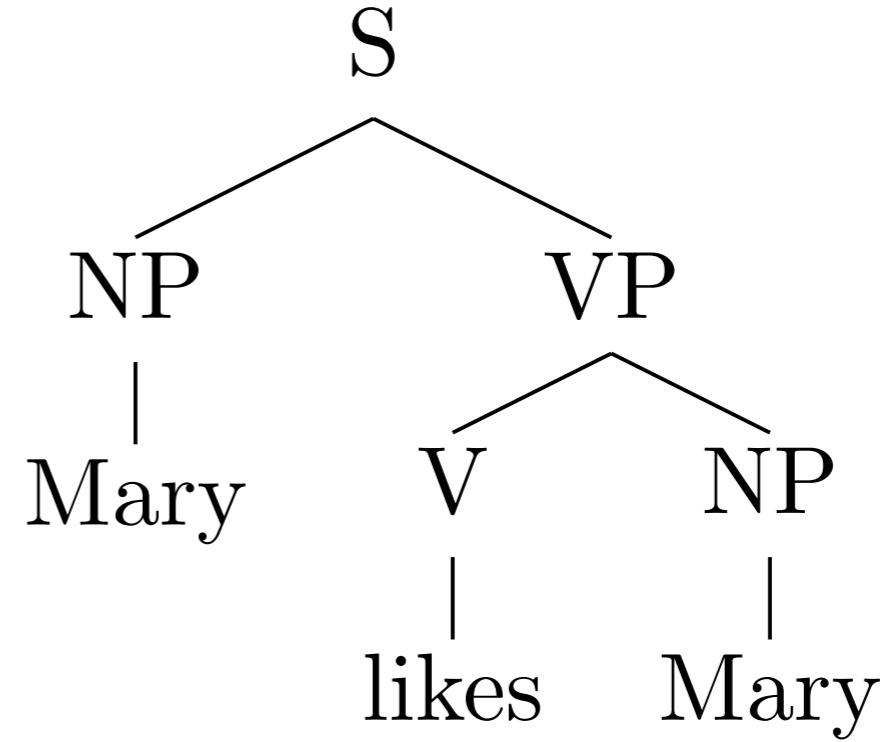
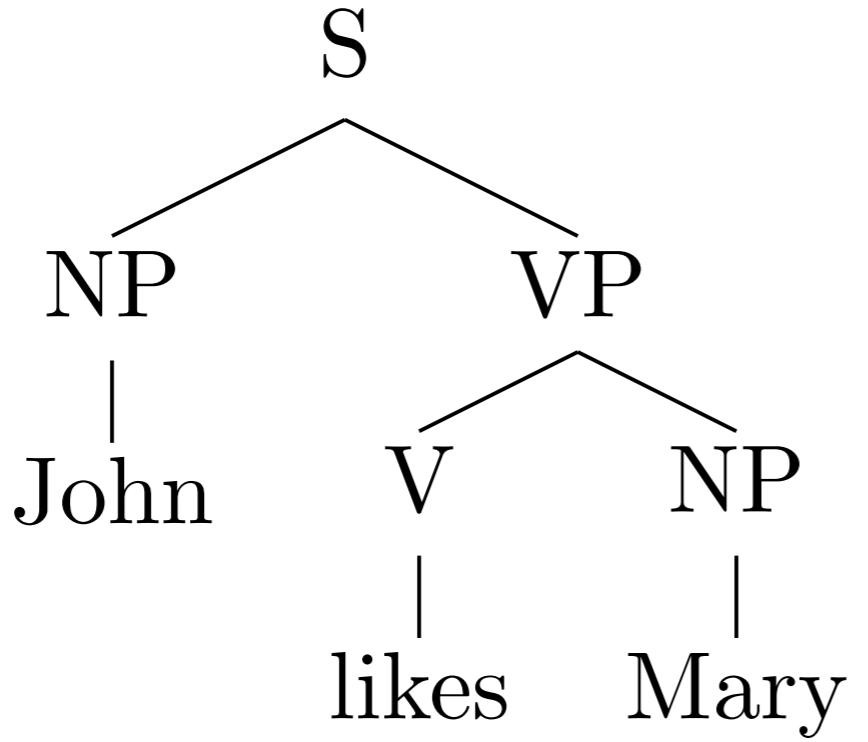
$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$



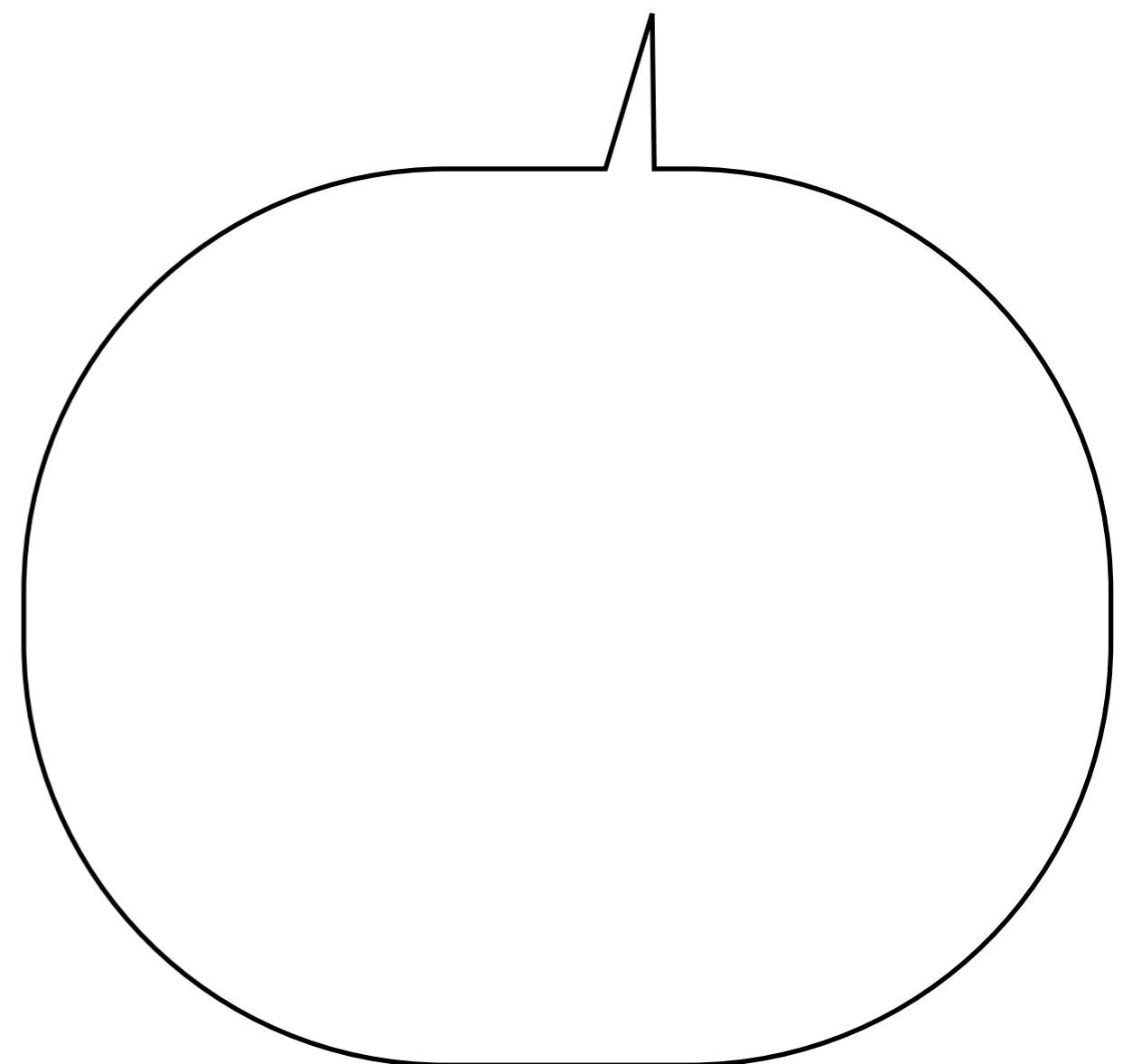
Context-Free Grammars

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R} \rangle$$



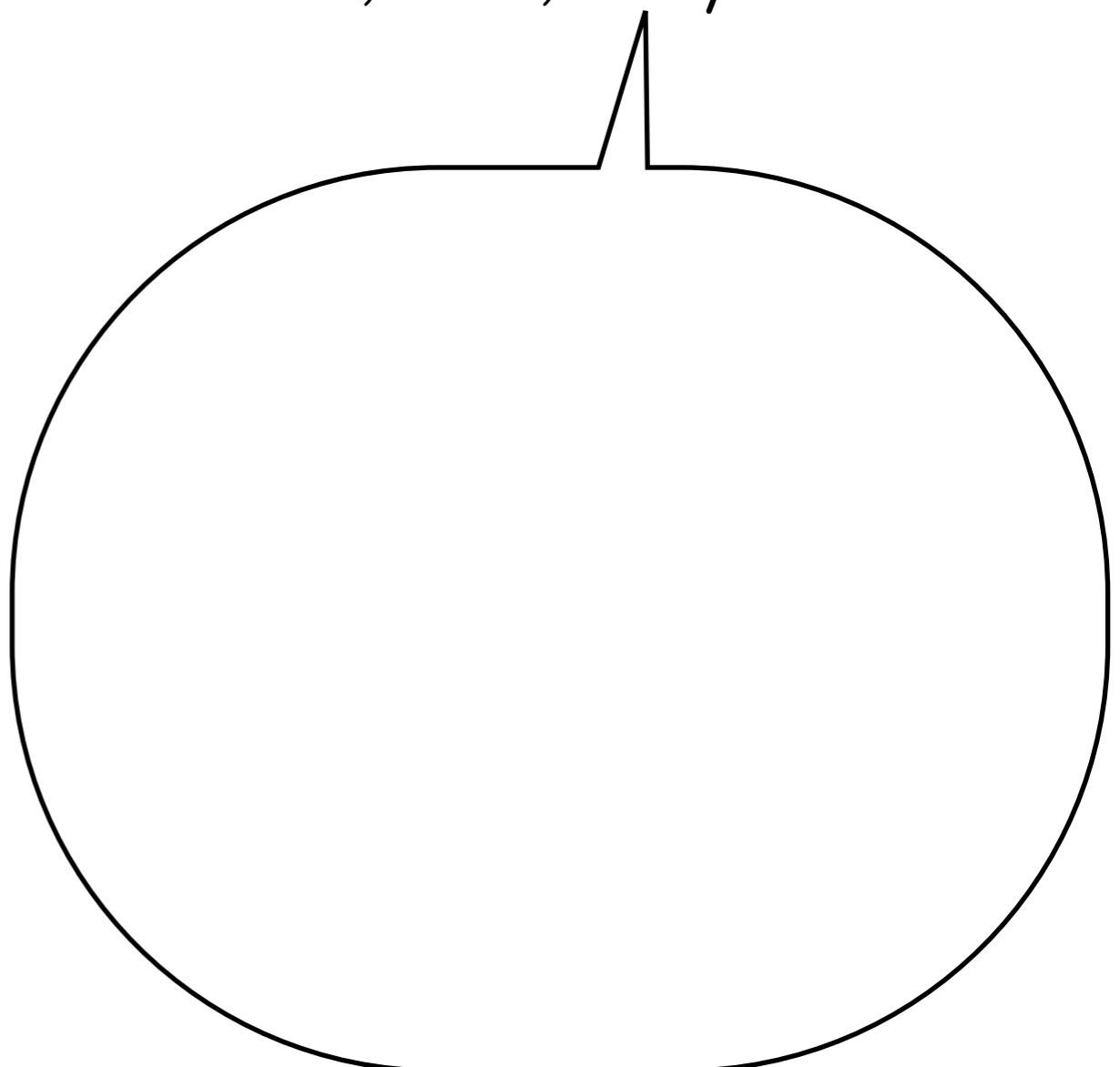


Probabilistic CFG



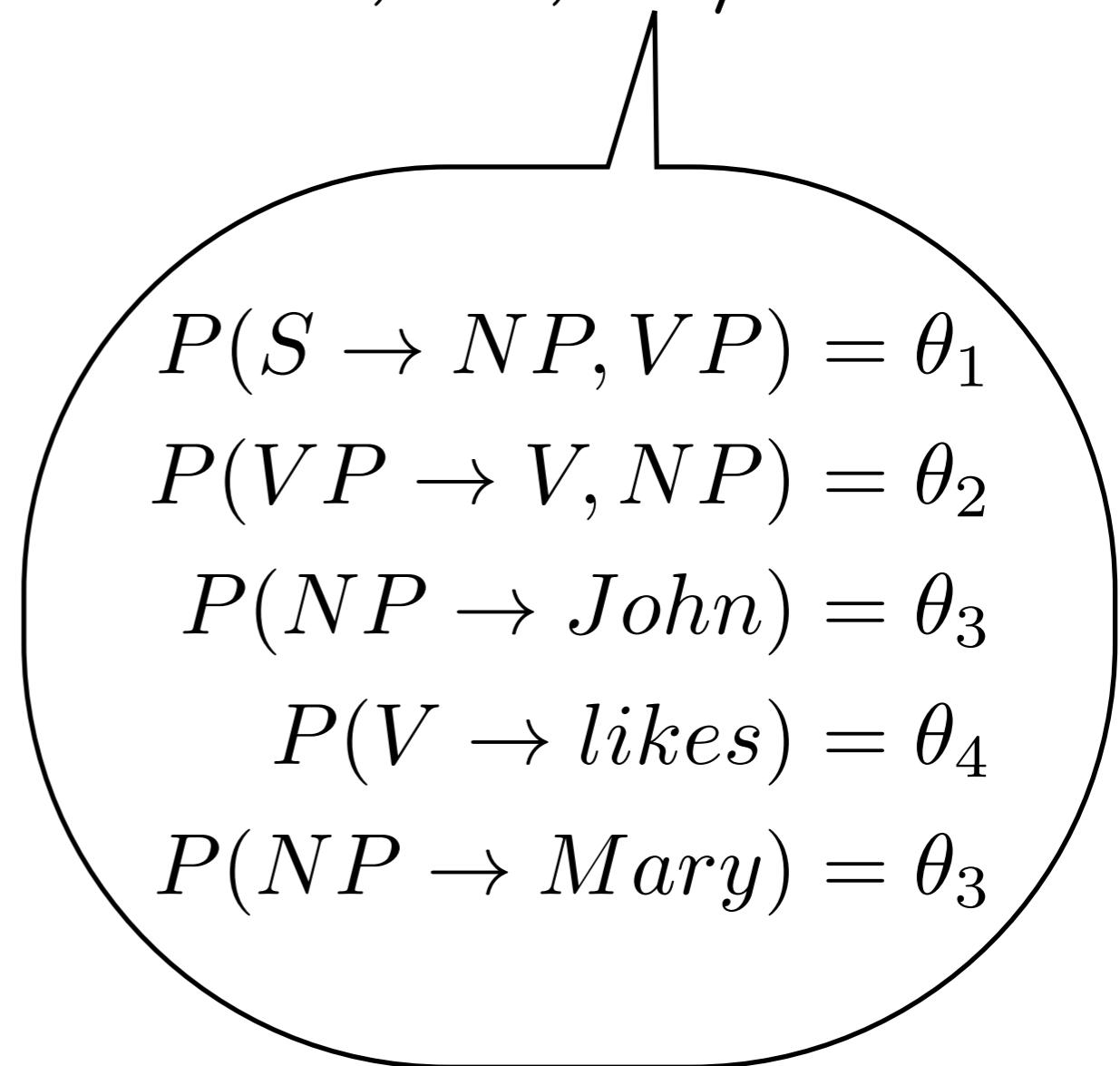
Probabilistic CFG

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$



Probabilistic CFG

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$



Probabilistic CFG

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$

parameters

$$P(S \rightarrow NP, VP) = \theta_1$$

$$P(VP \rightarrow V, NP) = \theta_2$$

$$P(NP \rightarrow John) = \theta_3$$

$$P(V \rightarrow likes) = \theta_4$$

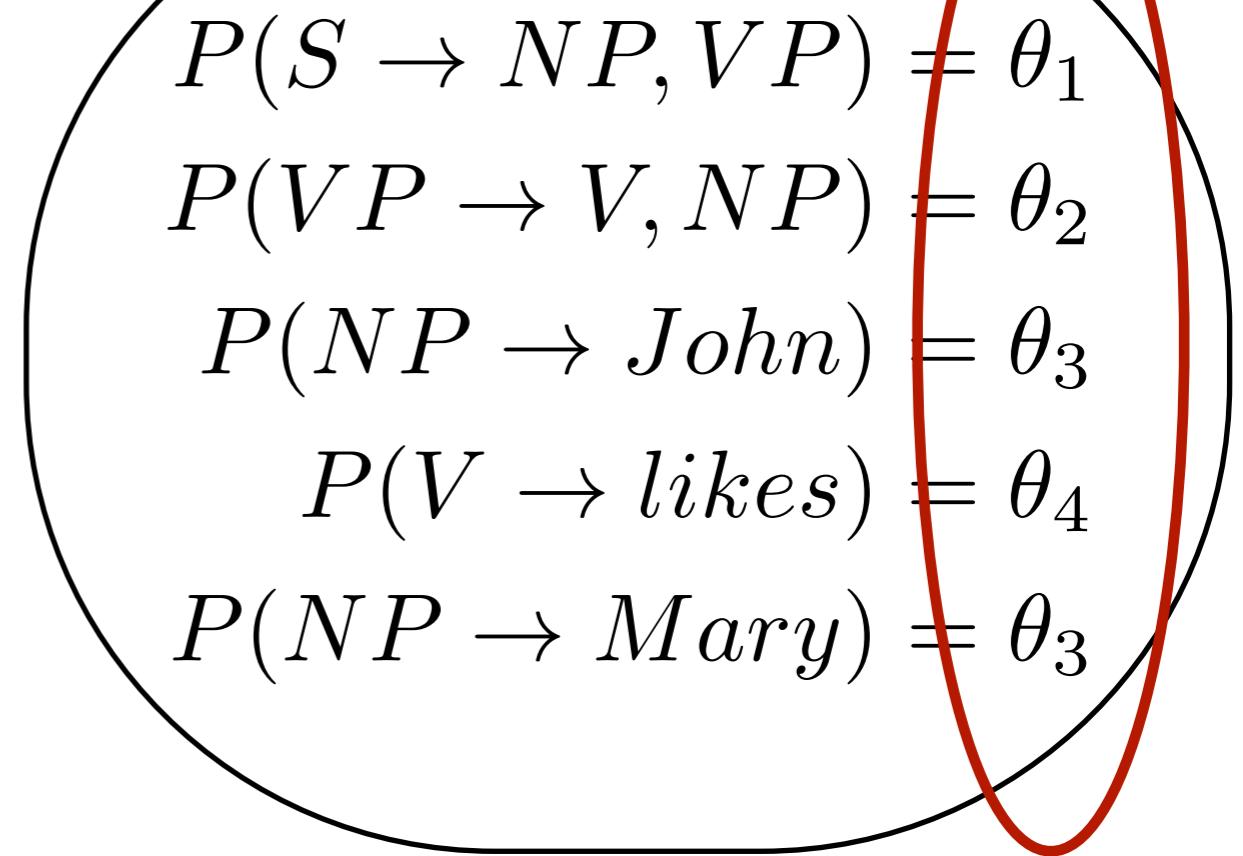
$$P(NP \rightarrow Mary) = \theta_3$$

Probabilistic CFG

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$

parameters

$$0 \leq P(A \rightarrow \alpha | A) \leq 1$$



Probabilistic CFG

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$

parameters

$$0 \leq P(A \rightarrow \alpha | A) \leq 1$$

$$\sum_{\alpha} P(A \rightarrow \alpha | A) = 1$$

$$P(S \rightarrow NP, VP) = \theta_1$$

$$P(VP \rightarrow V, NP) = \theta_2$$

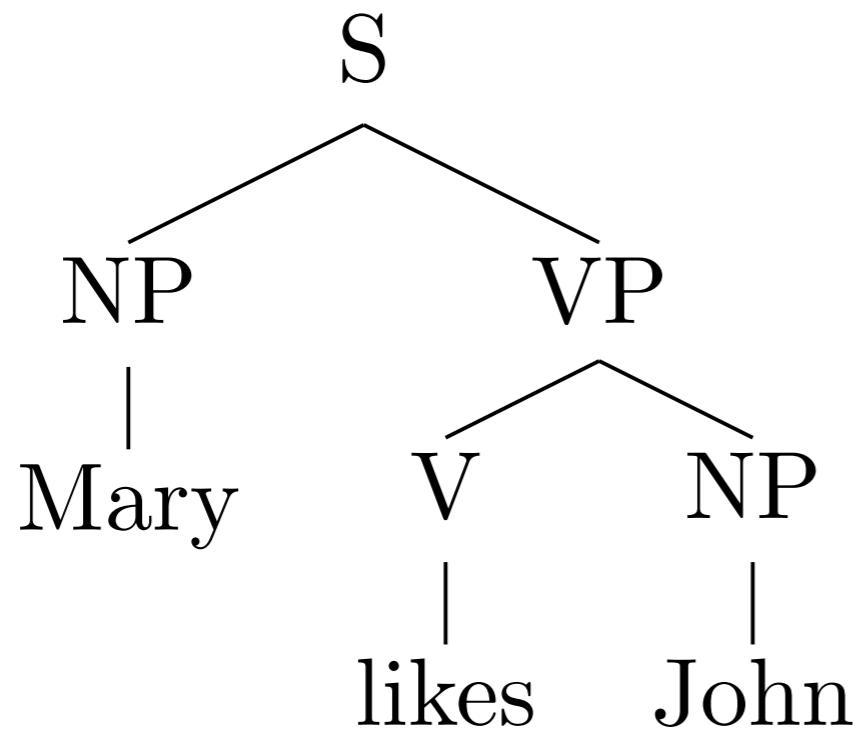
$$P(NP \rightarrow John) = \theta_3$$

$$P(V \rightarrow likes) = \theta_4$$

$$P(NP \rightarrow Mary) = \theta_3$$

Probability of a Tree

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$



$$P(S \rightarrow NP, VP) = 1.0$$

$$P(VP \rightarrow V, NP) = 1.0$$

$$P(NP \rightarrow John) = 0.5$$

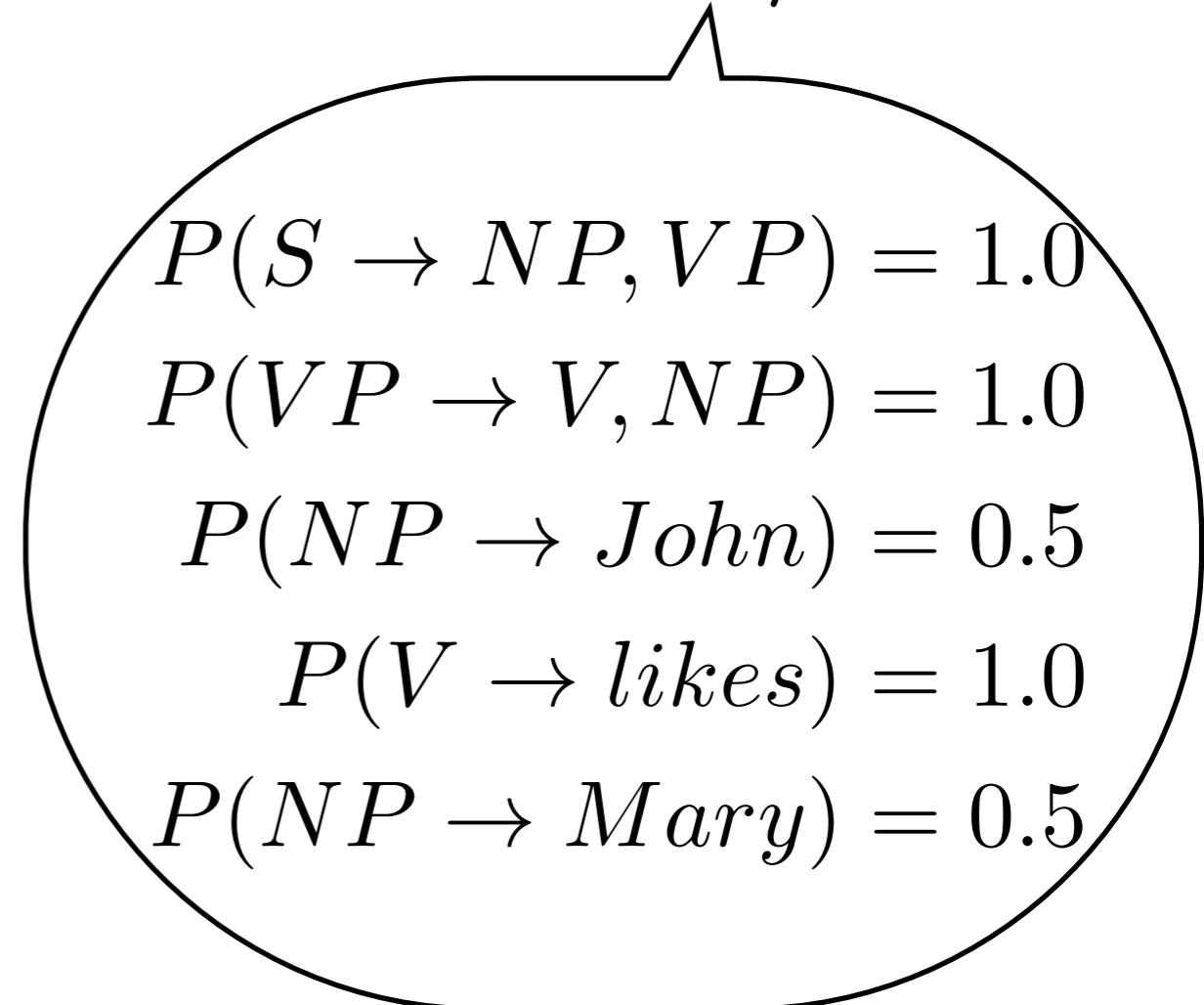
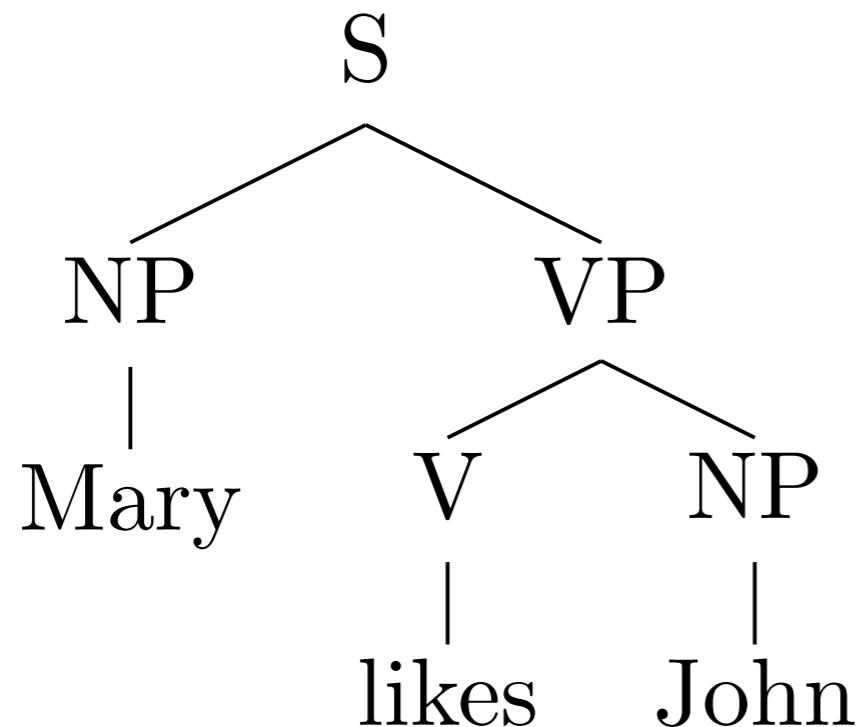
$$P(V \rightarrow likes) = 1.0$$

$$P(NP \rightarrow Mary) = 0.5$$

$$P(\pi) = 1.0 \times 1.0 \times 0.5 \times 1.0 \times 0.5 = 0.25$$

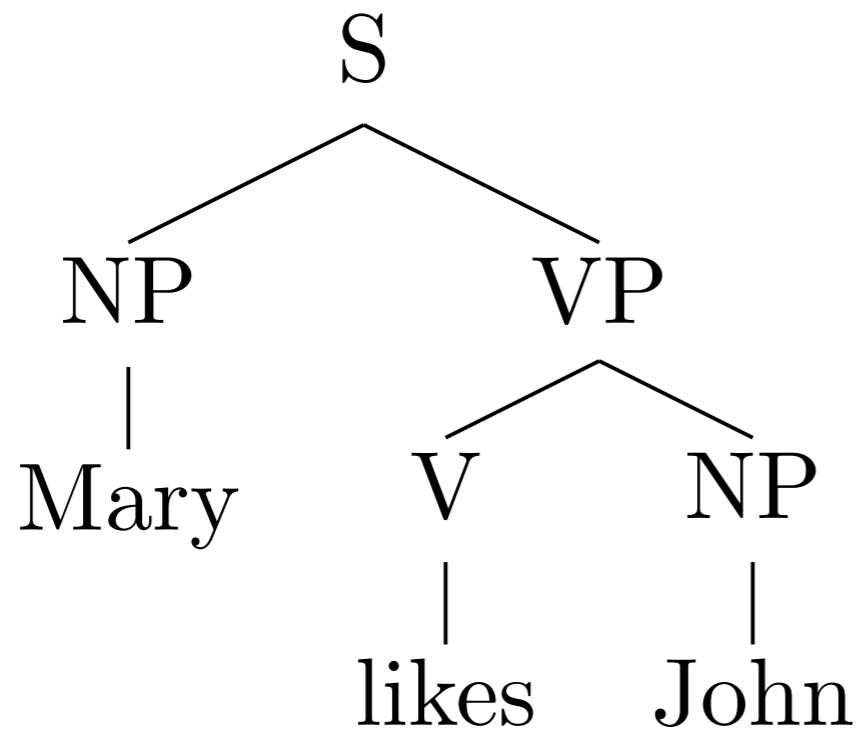
Probability of a Tree

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$



Probability of a Tree

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$



$$P(S \rightarrow NP, VP) = 1.0$$

$$P(VP \rightarrow V, NP) = 1.0$$

$$P(NP \rightarrow John) = 0.5$$

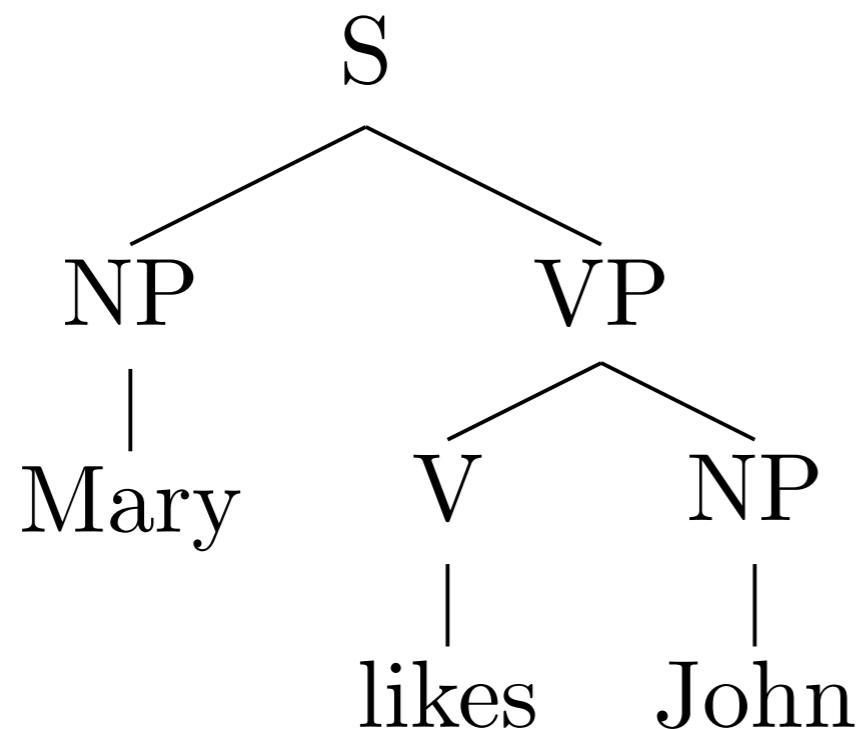
$$P(V \rightarrow likes) = 1.0$$

$$P(NP \rightarrow Mary) = 0.5$$

$$P(\pi) = \prod_{r \in \pi} P(r)$$

Probability of a Tree

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$

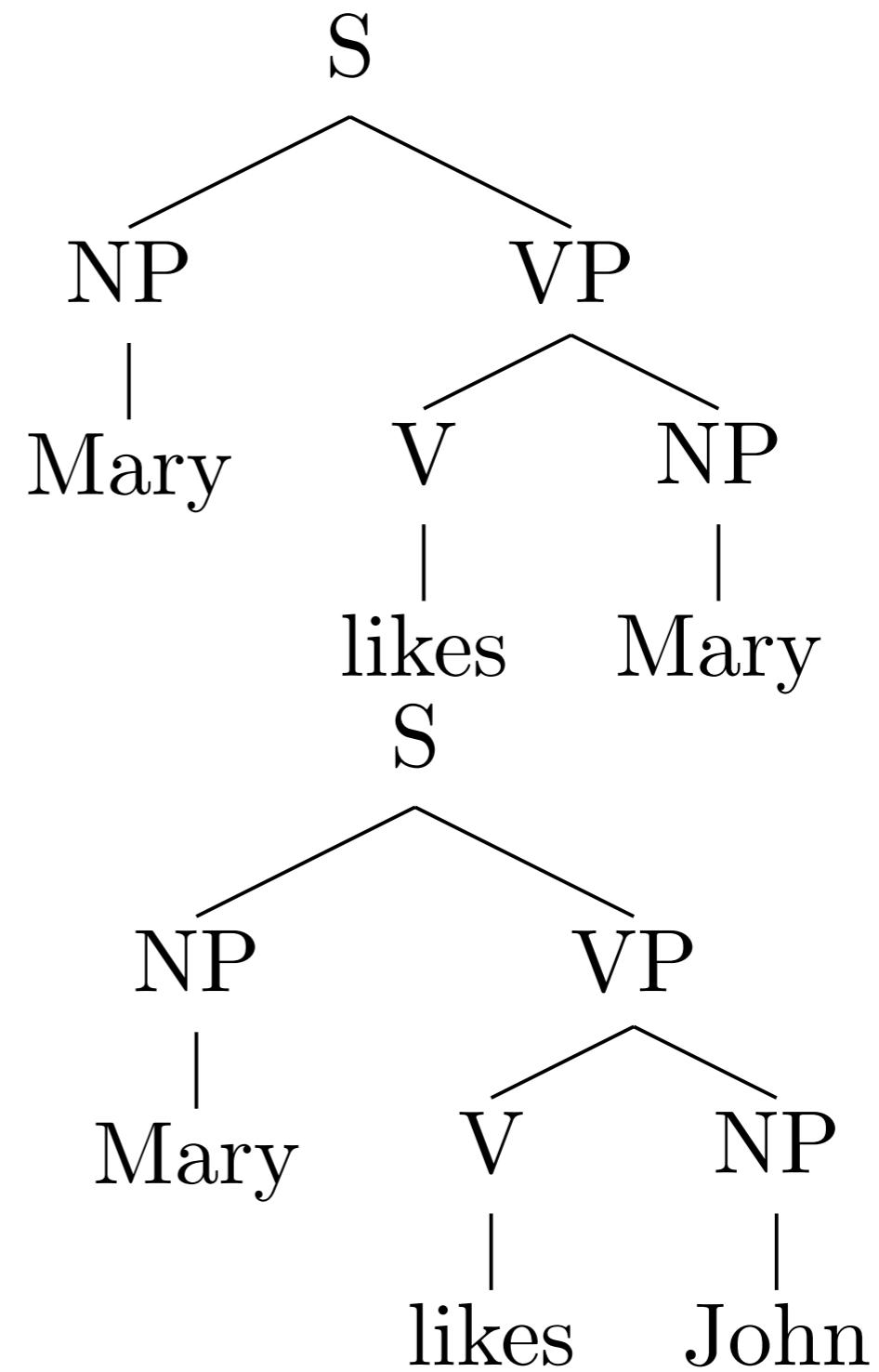
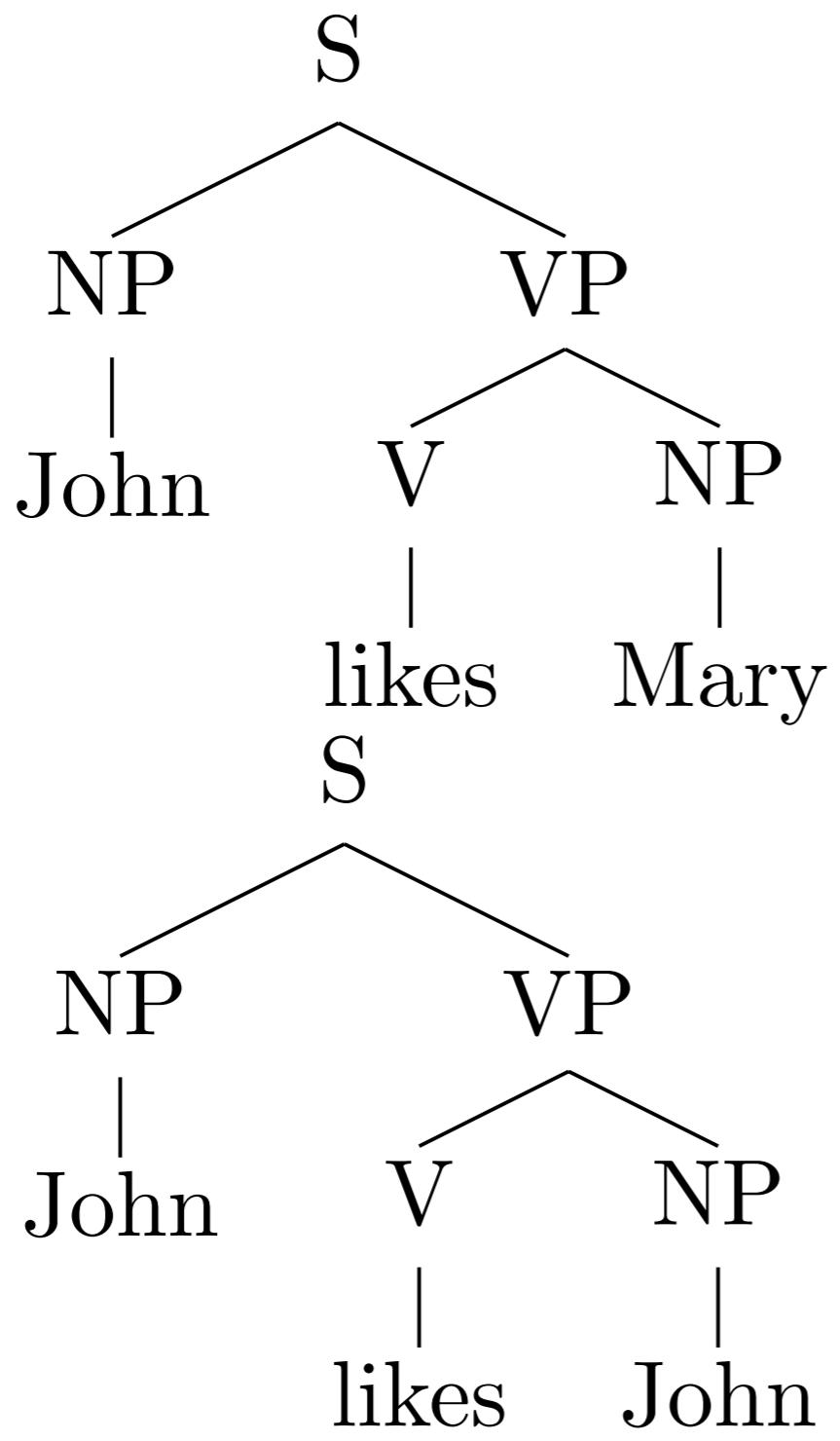


- $P(S \rightarrow NP, VP) = 1.0$
- $P(VP \rightarrow V, NP) = 1.0$
- $P(NP \rightarrow John) = 0.5$
- $P(V \rightarrow likes) = 1.0$
- $P(NP \rightarrow Mary) = 0.5$

$$P(\pi) = \prod_{r \in \pi} P(r)$$

Independence

Probability of a Tree

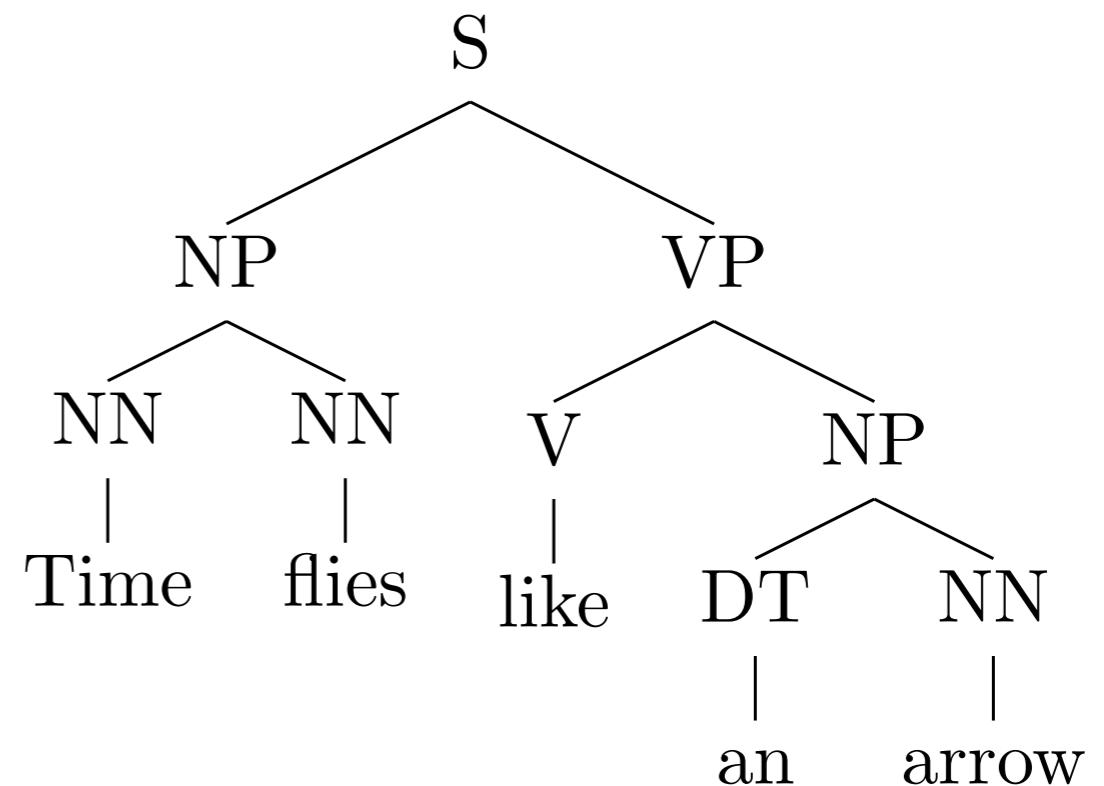
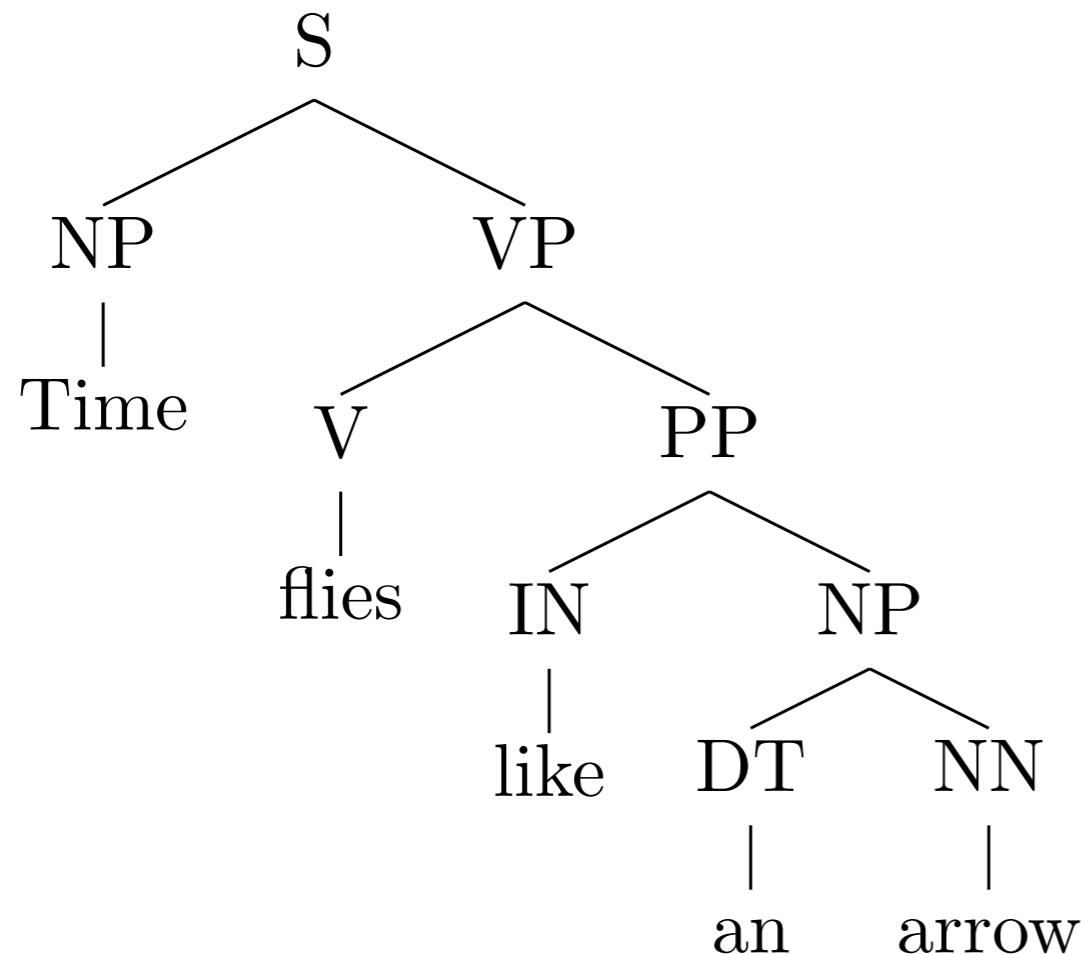


Probability of a String

Time flies like an arrow

http://en.wikipedia.org/wiki/Catalan_number

Probability of a String



Time flies like an arrow

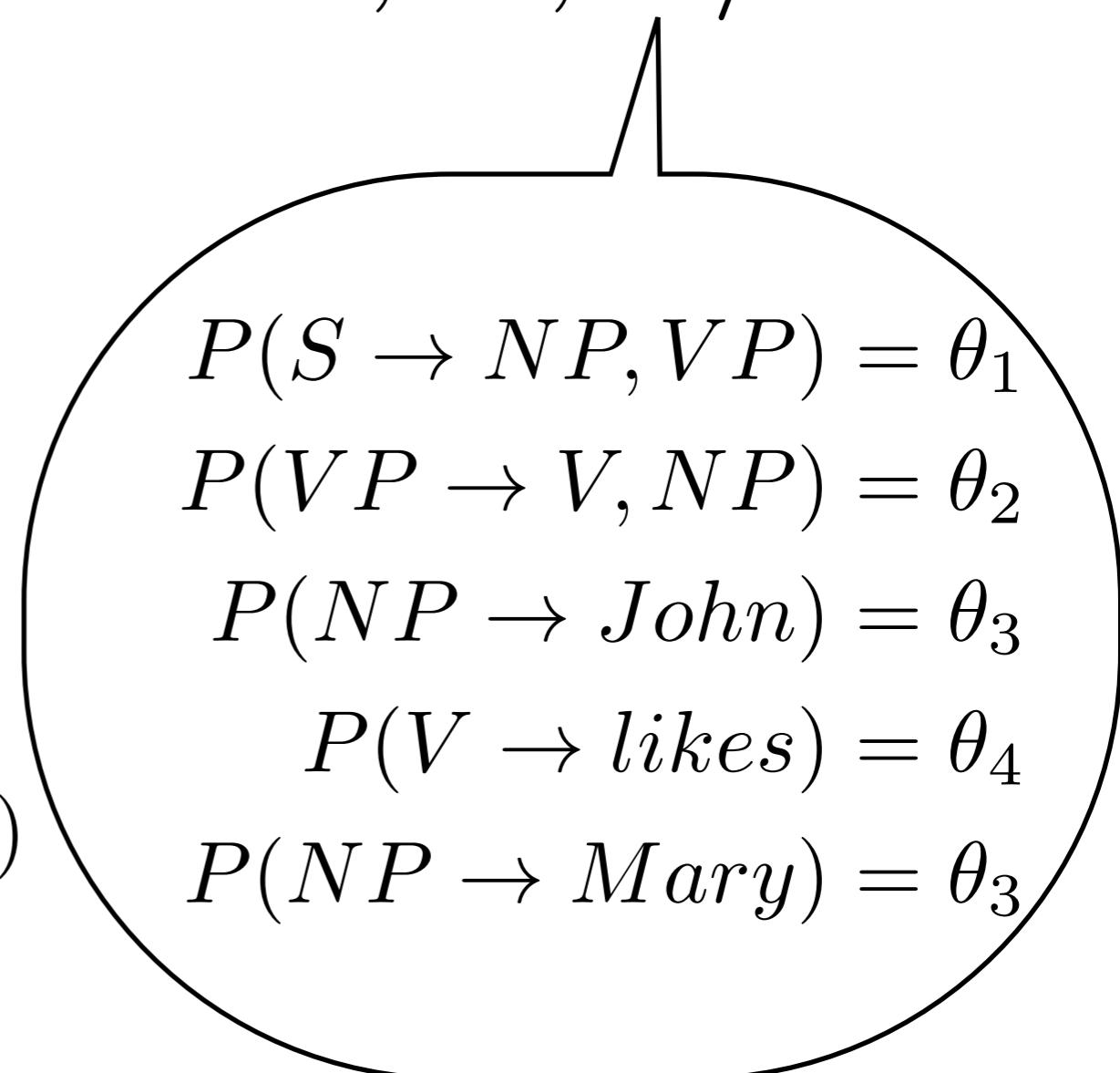
http://en.wikipedia.org/wiki/Catalan_number

Probability of a String

$$G = \langle \mathcal{N}, \mathcal{T}, S \in \mathcal{N}, \mathcal{R}, P \rangle$$

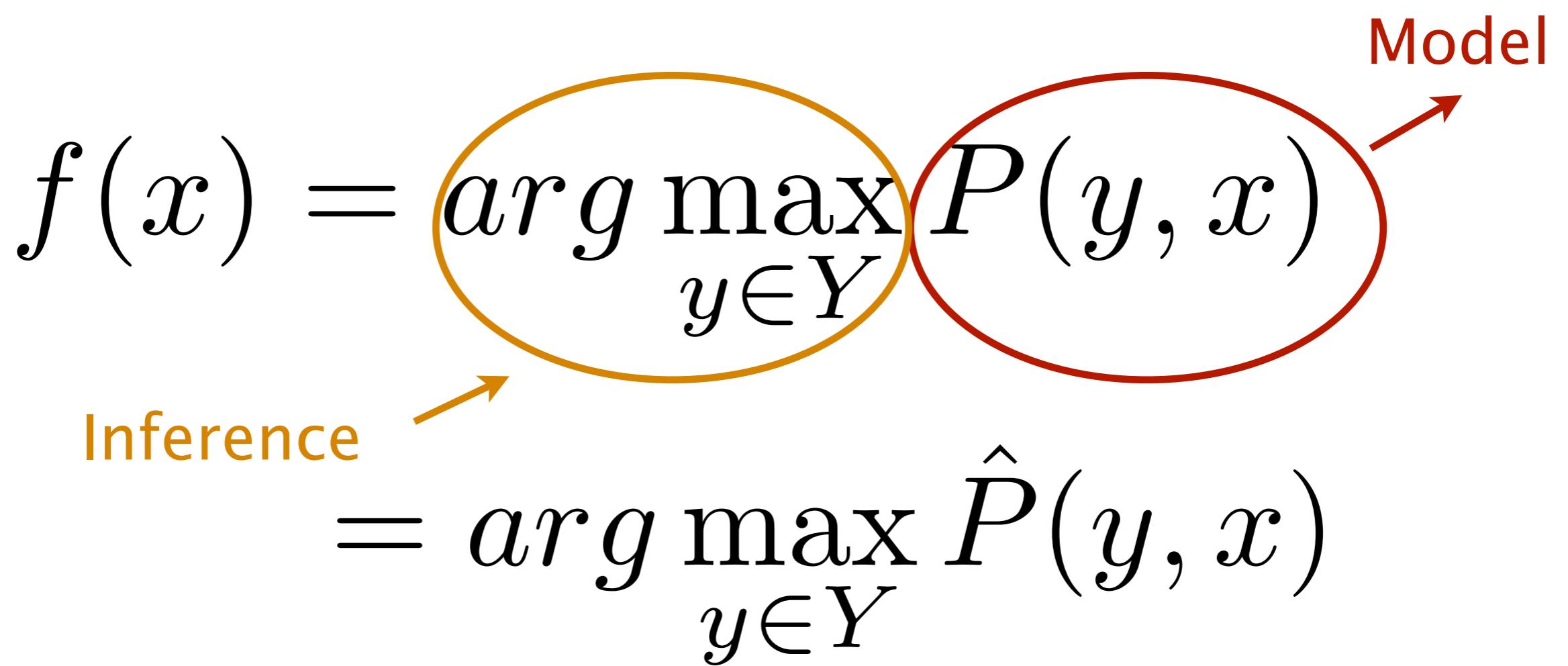
$$P(\pi) = \prod_{r \in \pi} P(r)$$

$$\begin{aligned} P(x) &= \sum_{\{\pi | \text{yield}(\pi)=x\}} P(\pi) \\ &= \sum_{\{\pi | \text{yield}(\pi)=x\}} \prod_{r \in \pi} P(r) \end{aligned}$$

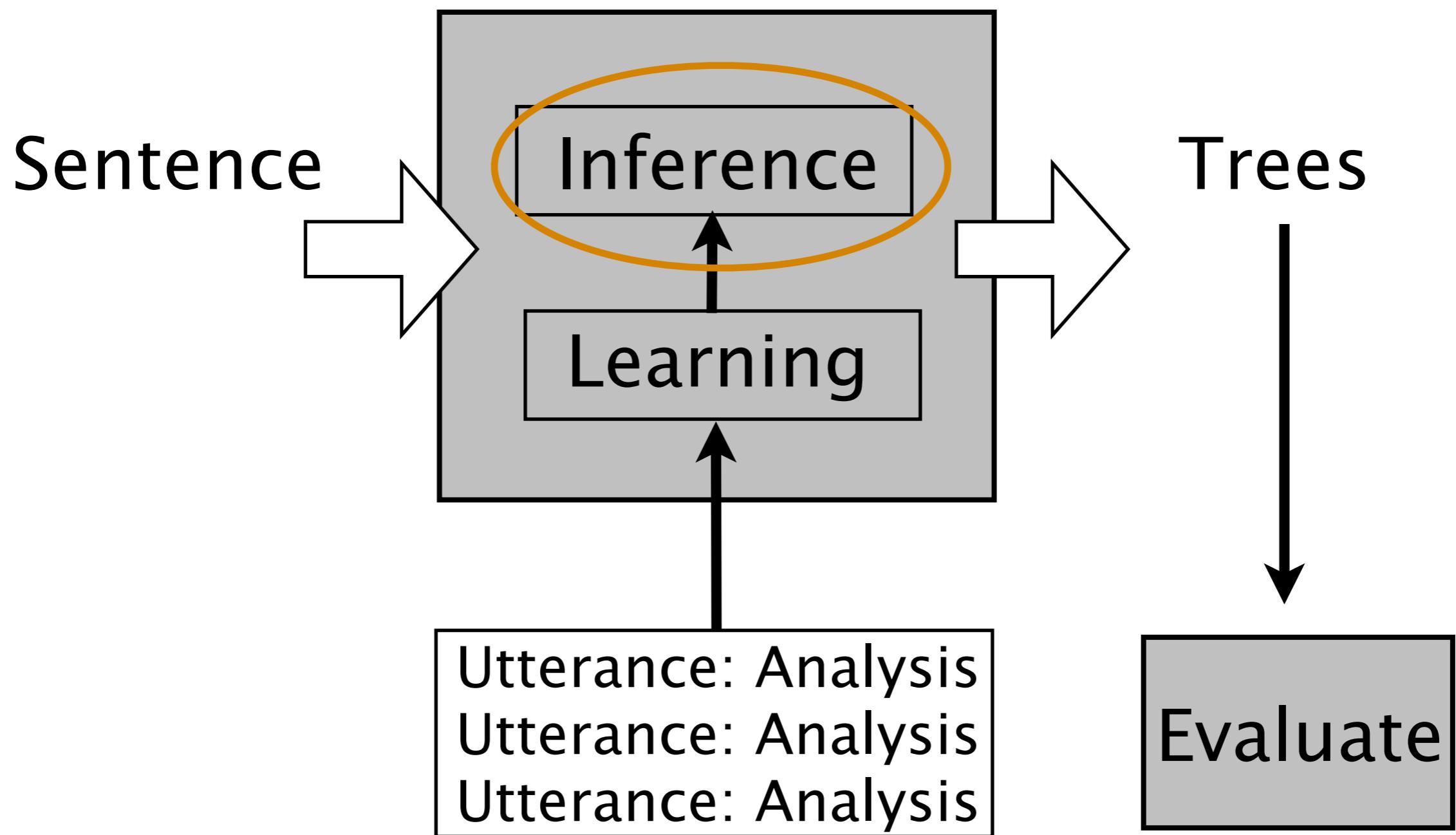


Modeling

- Probabilistic Generative Modeling



Inference



Inference

- Input:
 - A PCFG
 - A sentence
- We need to solve **argmax**

$$f(x) = \arg \max_{y \in Y} P(y, x)$$

The Challenge

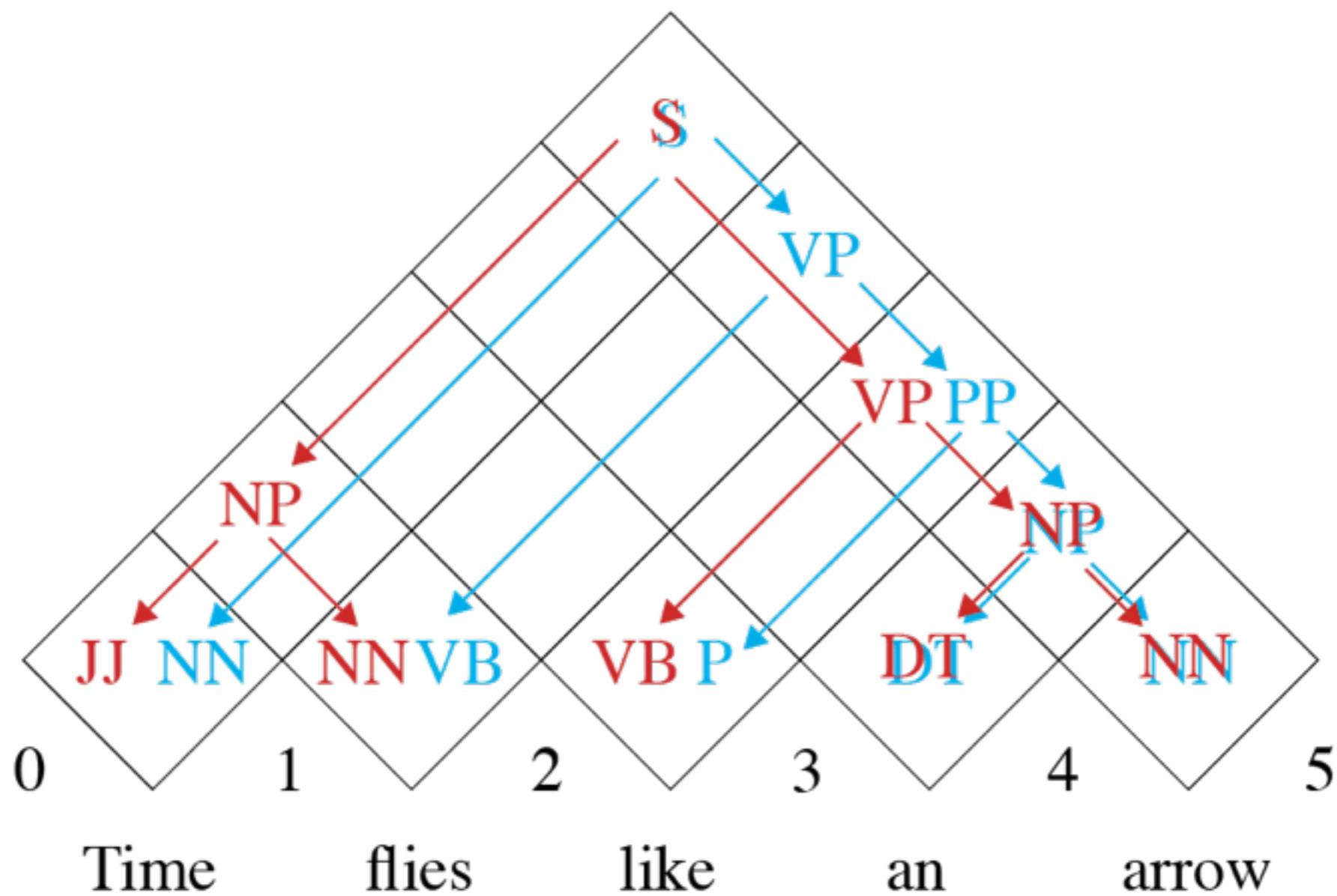
- The problem:
 - Number of trees exponential in $|x|$
- An Efficient Solution:
 - Pack all trees (using overlaps)
 - Calculate (and reuse) scores
 - Get maximum scoring tree

Key idea

- Data Structure:
 - A 2D array $[n+1][n+1]$
- Input:
 - A PCFG in Binary Form
 - A Sentence x in English
- Tasks:
 - Recognition (finding all trees)
 - Parsing (finding the best tree)

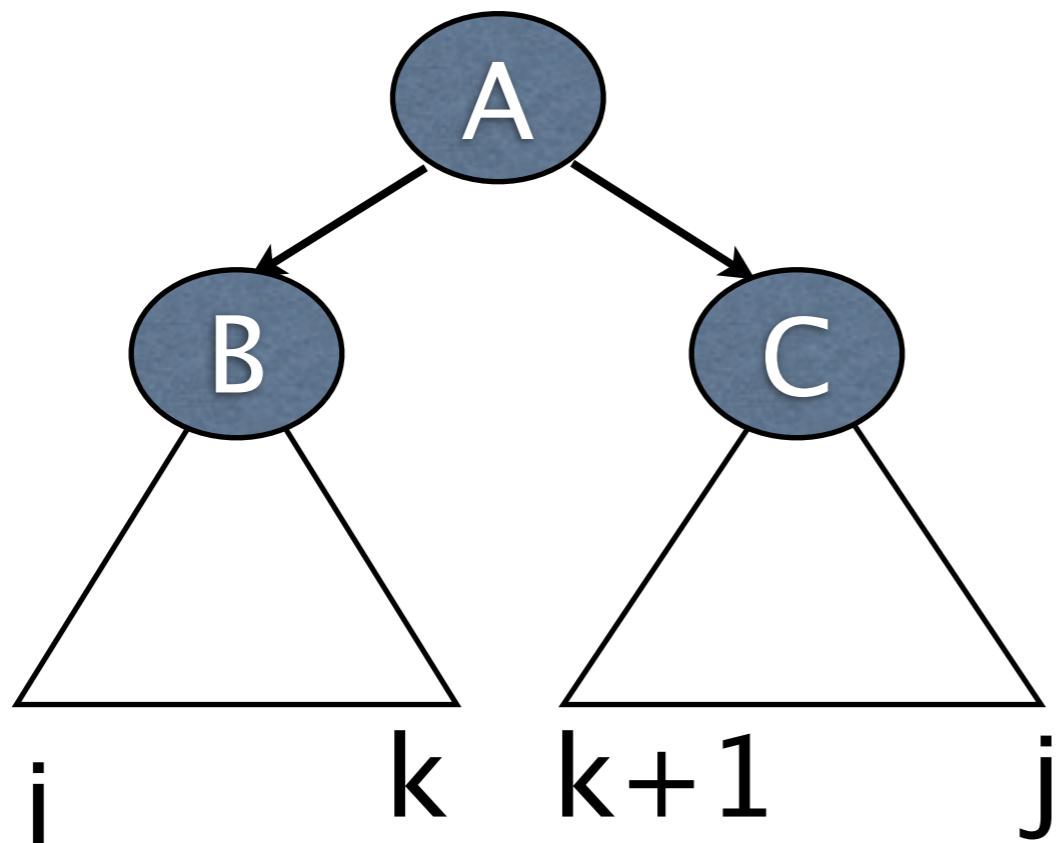
The Data Structure

The Data Structure



Parsing as Search (1)

- Search for all possible subtrees:

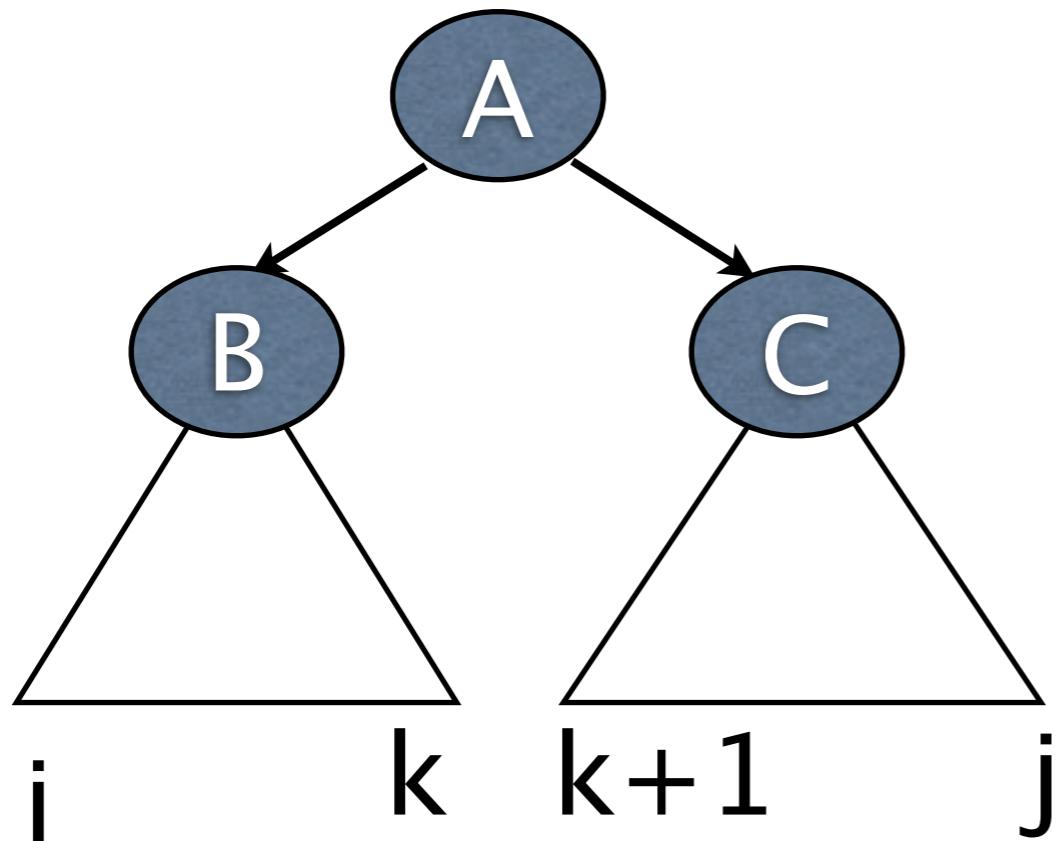


- For every span
- For every split
- Apply CFG rules

Parsing as Search (2)

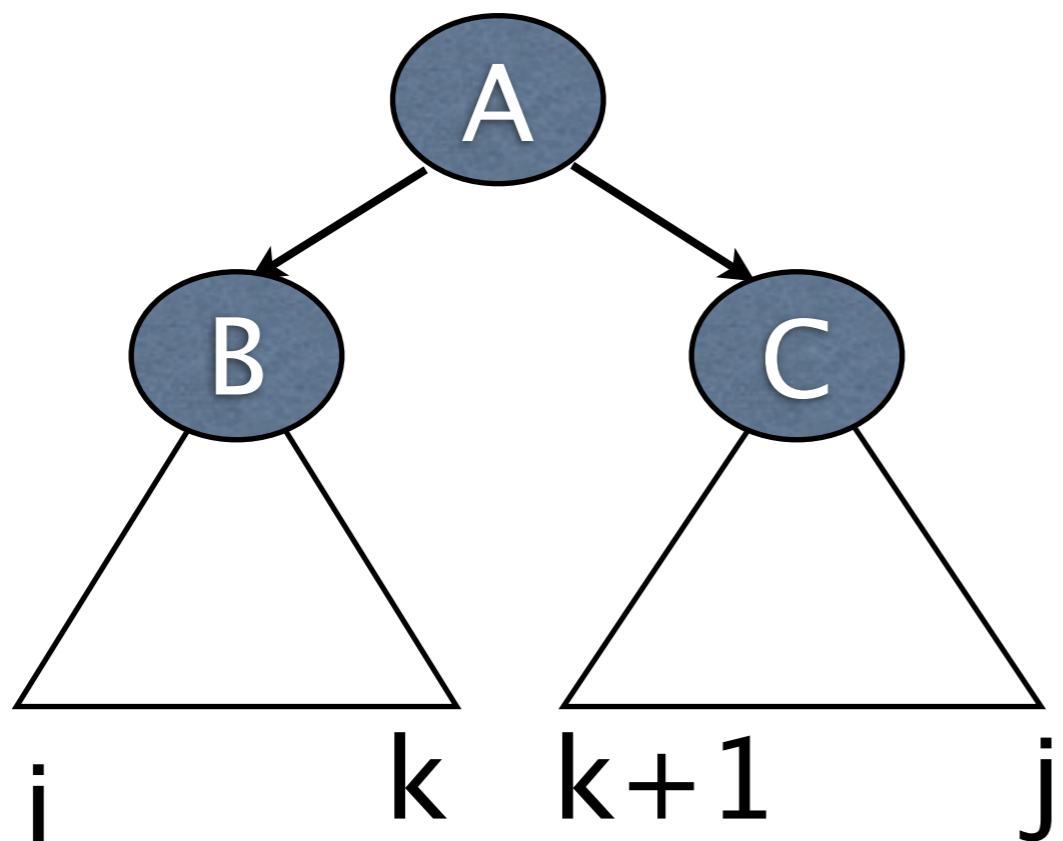
- Re-use the probability of subtrees:

$$\begin{aligned} P_t(A, i, j) = & P(A \rightarrow BC) \\ & \times P_t(B, i, k) \\ & \times P_t(C, k + 1, j) \end{aligned}$$



Parsing as Search (3)

- Re-use the probability of subtrees:



$$\delta(A, i, j) = \arg \max_{k, B, C} P(A \rightarrow B, C) \times \delta(B, i, k) \times \delta(C, k + 1, j)$$

CKY parsing

Algorithm 1 The CKY Algorithm for Chart Parsing (with Back-Pointers)

```
1: for  $i = 1 \rightarrow n$  do
2:   for  $L = 1 \rightarrow |\mathcal{N}|$  do
3:      $\delta_L[i - 1, i] \leftarrow p(A_L \rightarrow w_i)$             $\triangleright$  Initiate pre-terminal probs
4:      $\beta_L[i] \leftarrow \langle w_i \rangle$                     $\triangleright$  Store words
5:   for  $span = 2 \rightarrow n$  do                       $\triangleright$  Fill in the chart
6:     for  $end = span \rightarrow n$  do
7:        $begin \leftarrow end - span$ 
8:       for  $L = 1 \rightarrow |\mathcal{N}|$  do
9:          $\delta_L(begin, end) \leftarrow \max_{\langle m, J, K \rangle} p(A_L \rightarrow A_J A_K) \times \delta_J(begin, m) \times \delta_K(m, end)$ 
10:         $\beta_L(begin, end) \leftarrow \operatorname{argmax}_{\langle m, J, K \rangle} p(A_L \rightarrow A_J A_K) \times \delta_J(begin, m) \times \delta_K(m, end)$ 
11:   return BUILD-TREE( $\delta_S[0, n]$ ,  $\beta_S[0, n]$ )            $\triangleright$  Follow back-pointers
```

Chart-based parsing

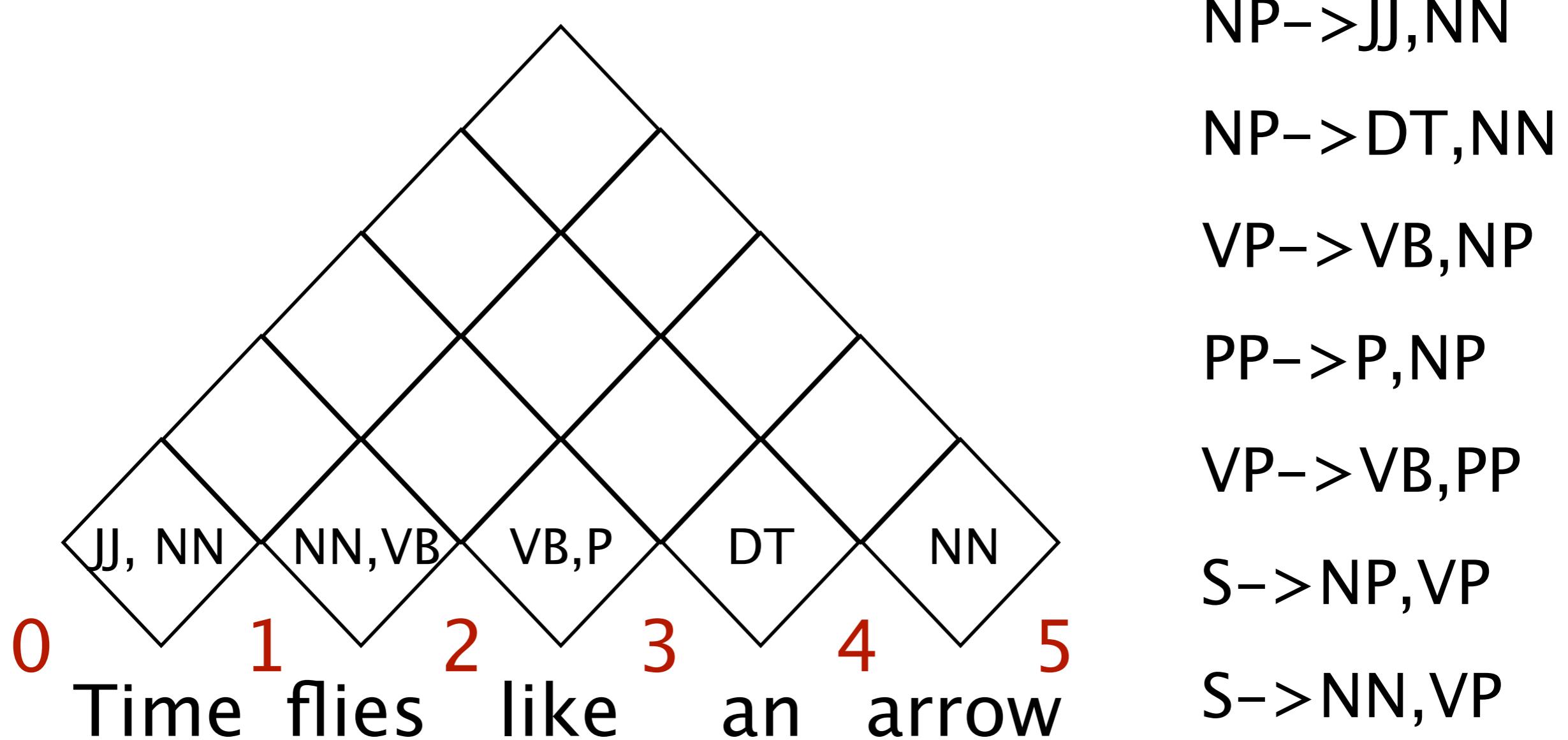


Chart-based parsing

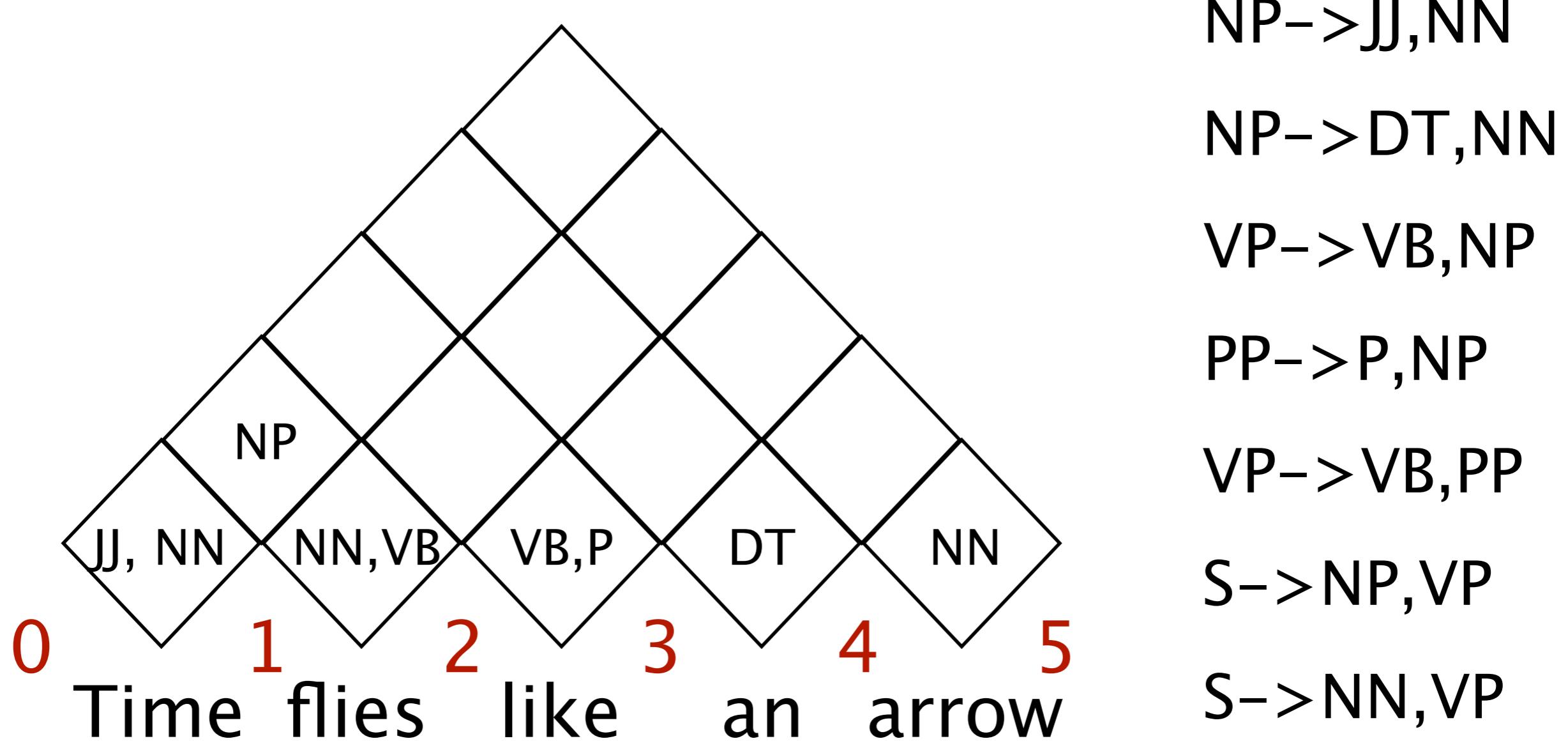


Chart-based parsing

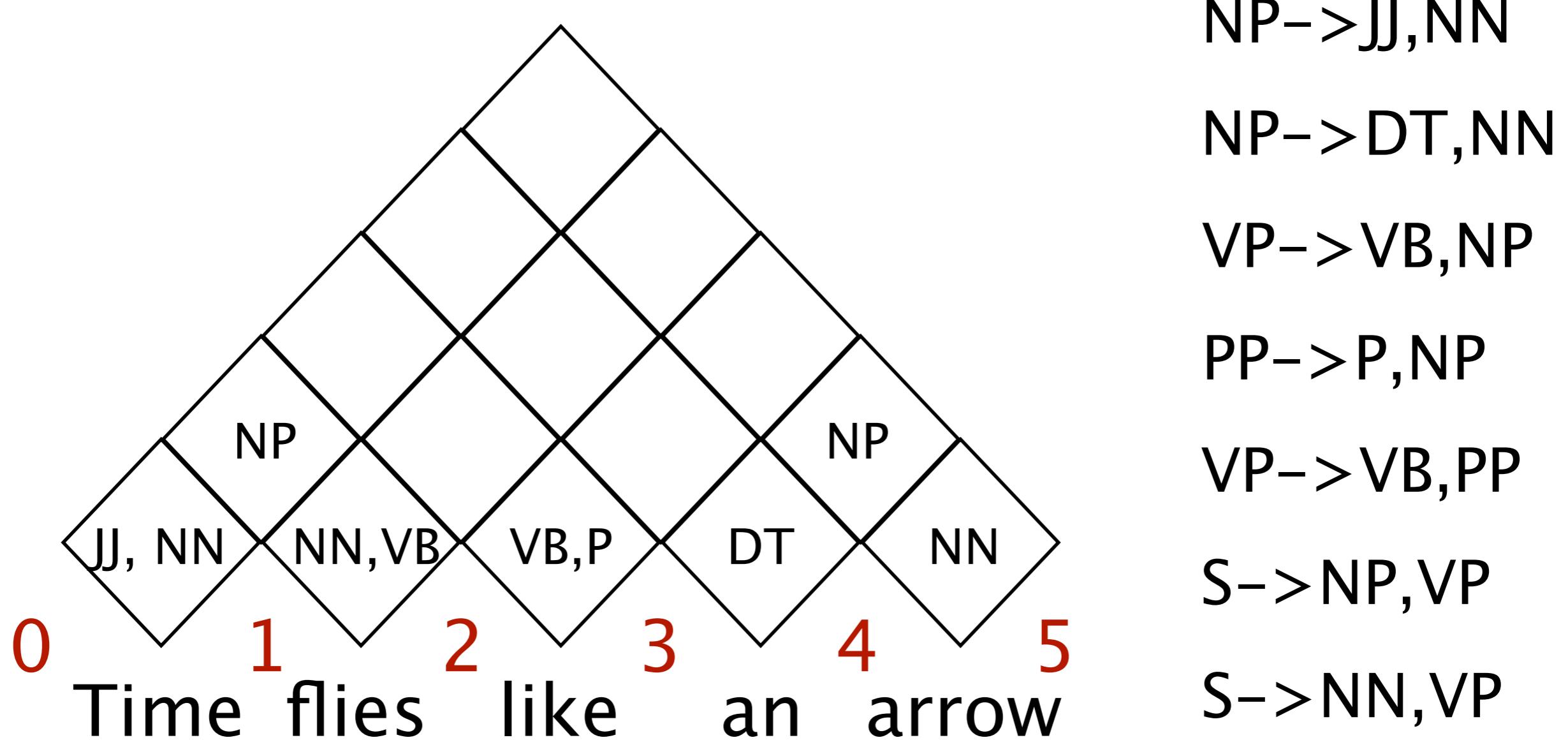


Chart-based parsing

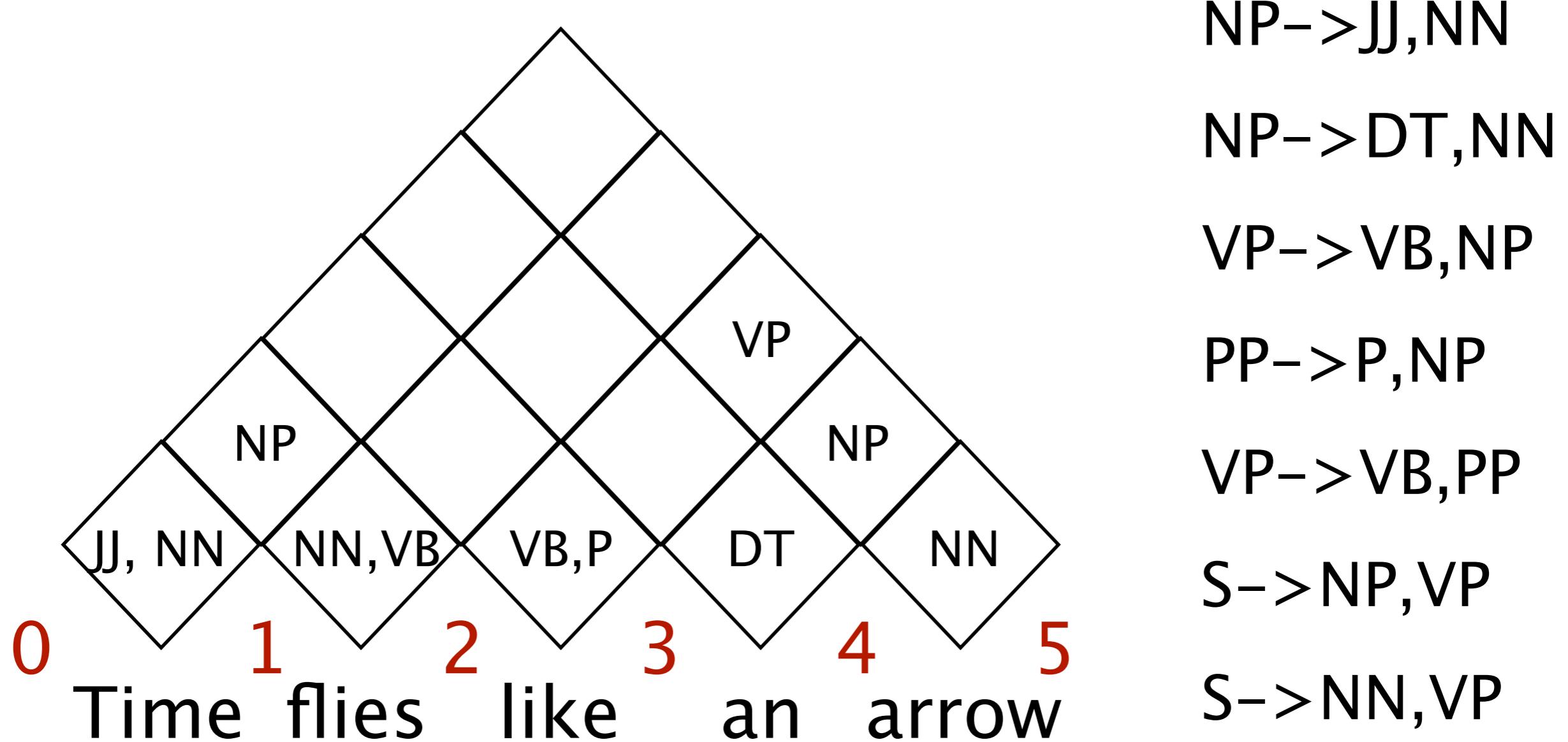


Chart-based parsing

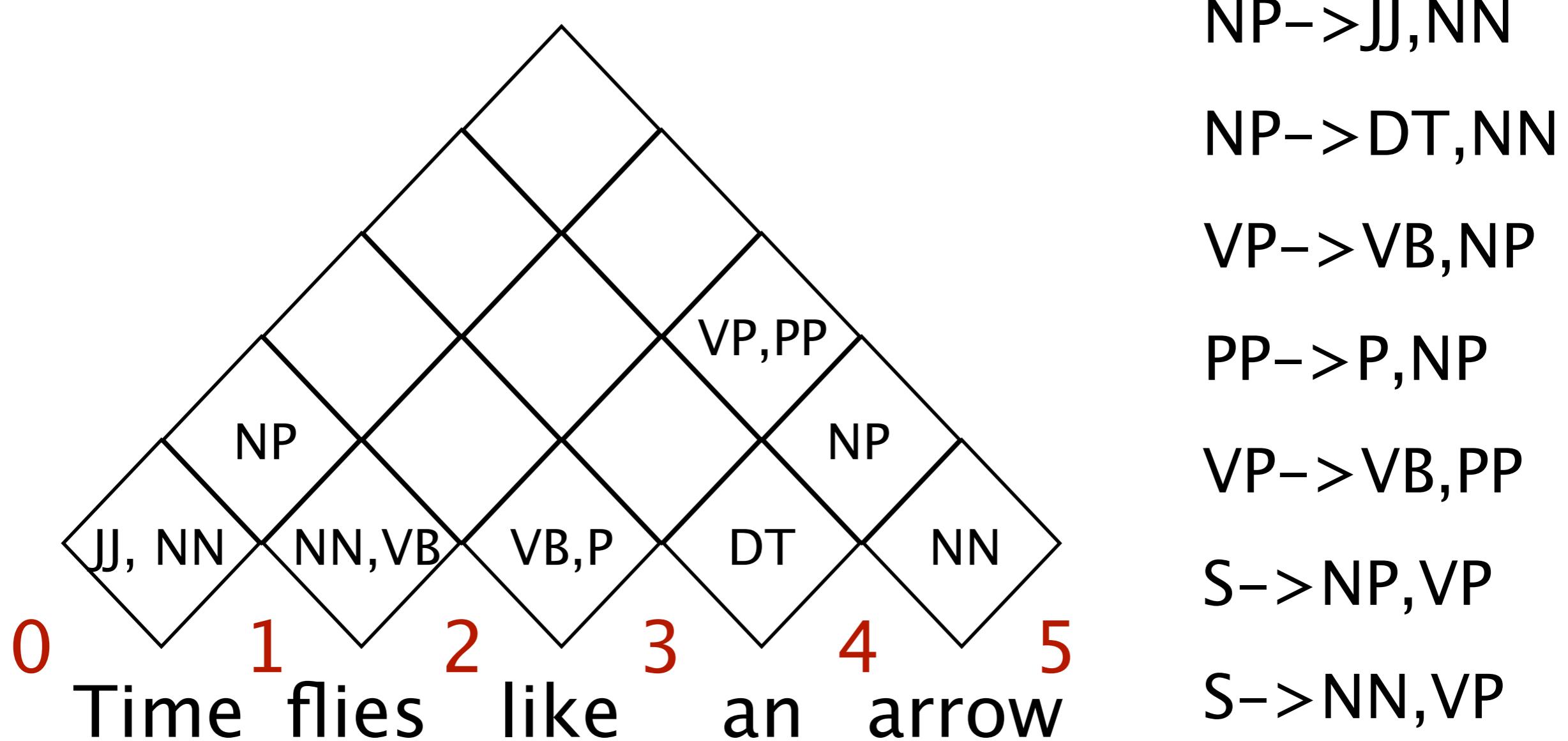


Chart-based parsing

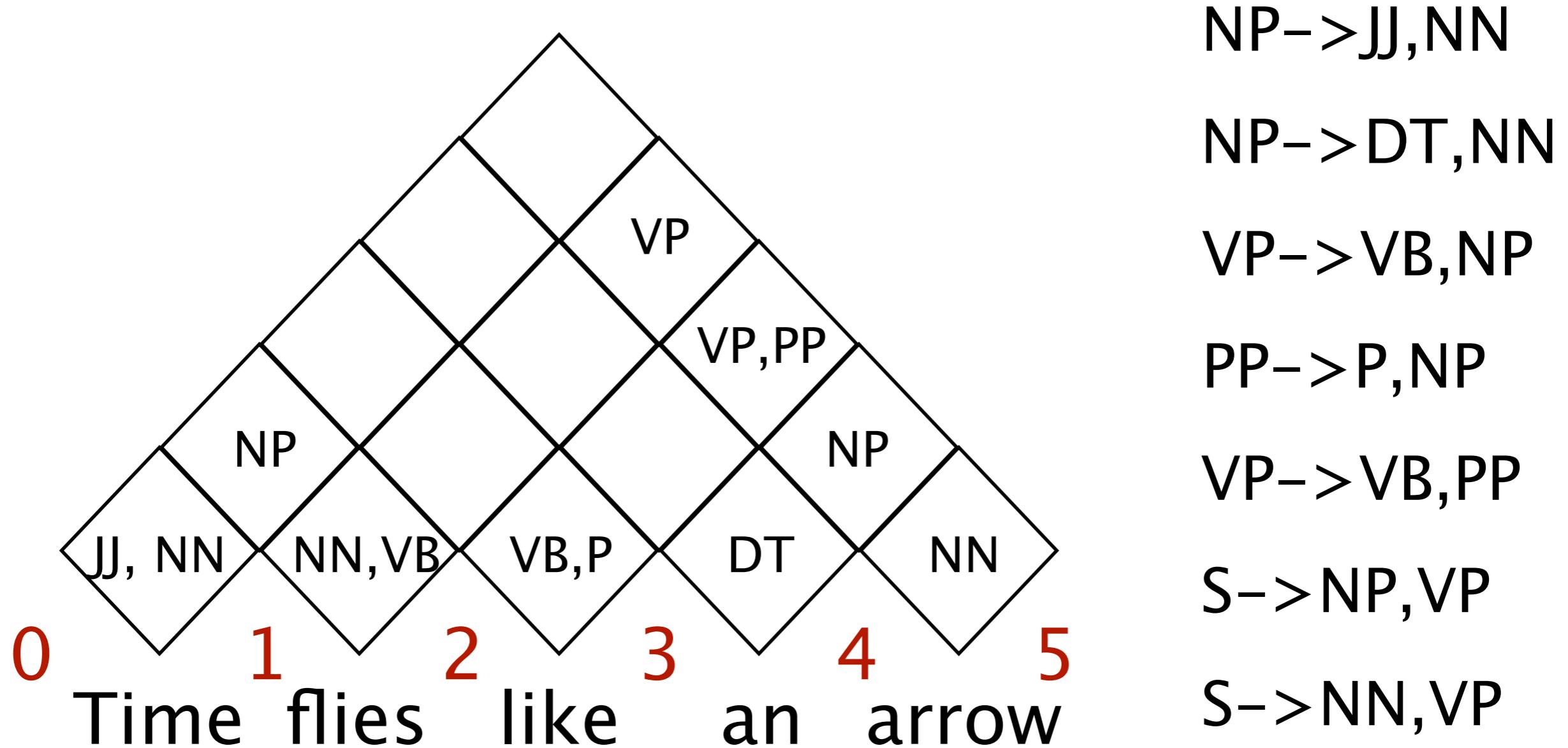


Chart-based parsing

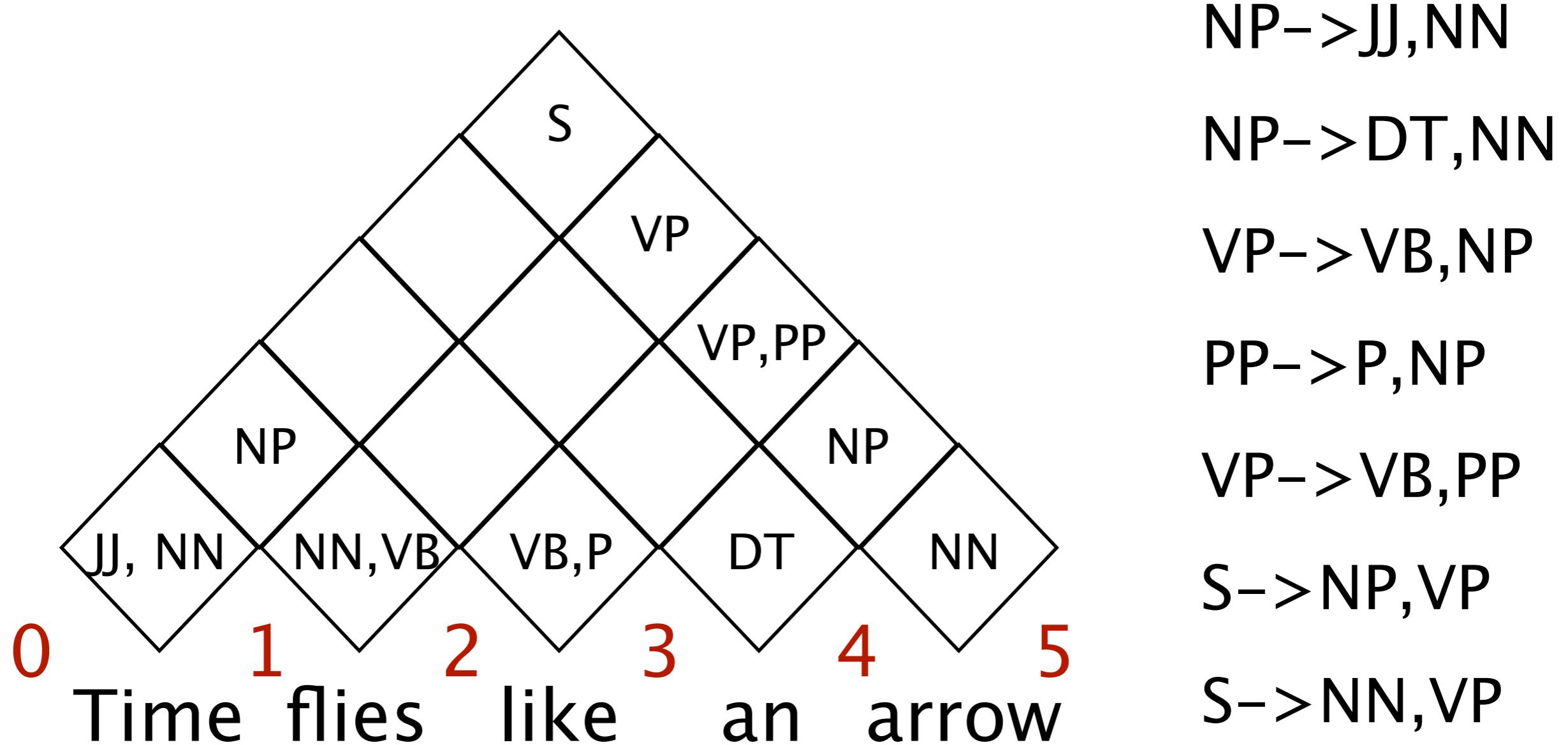
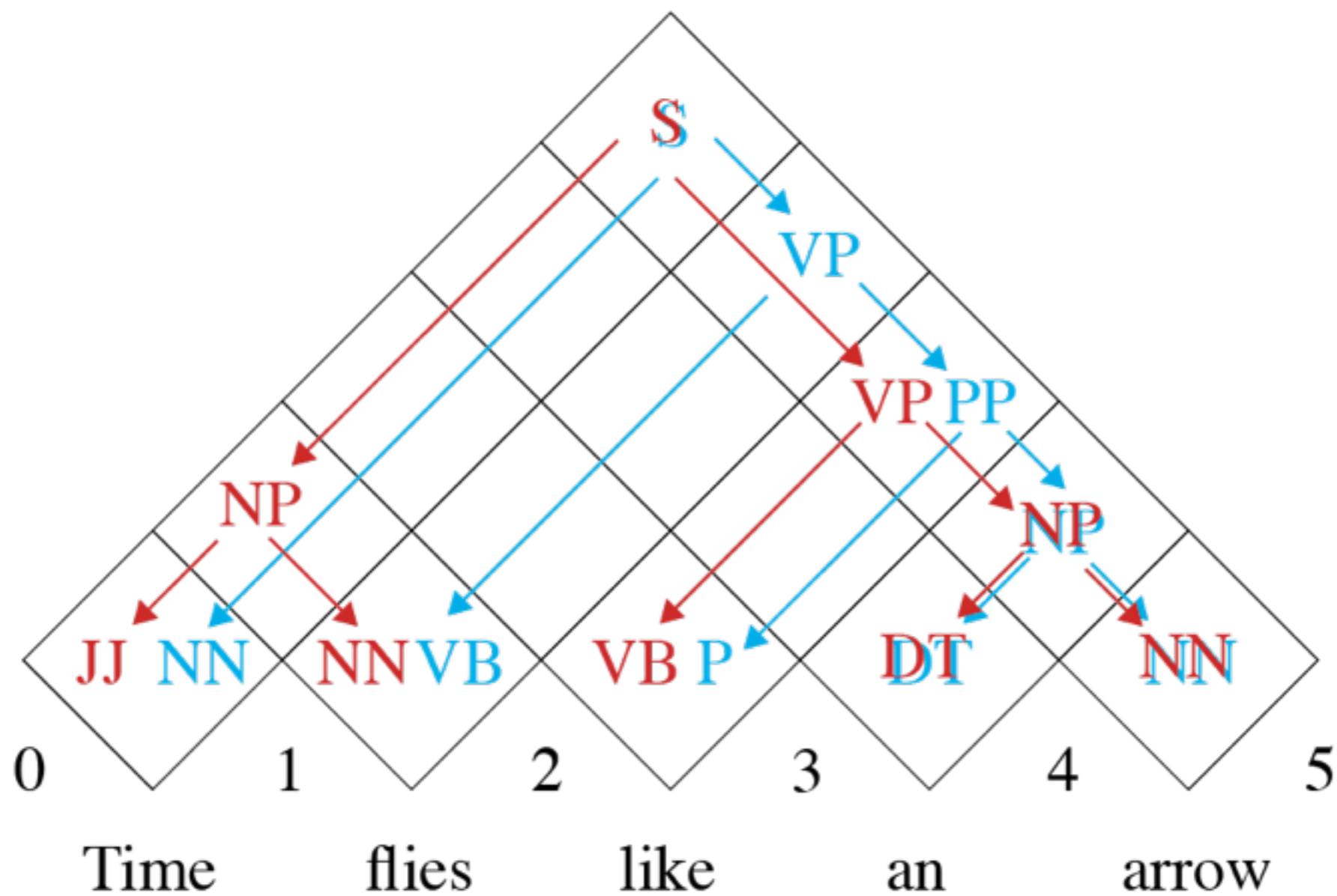
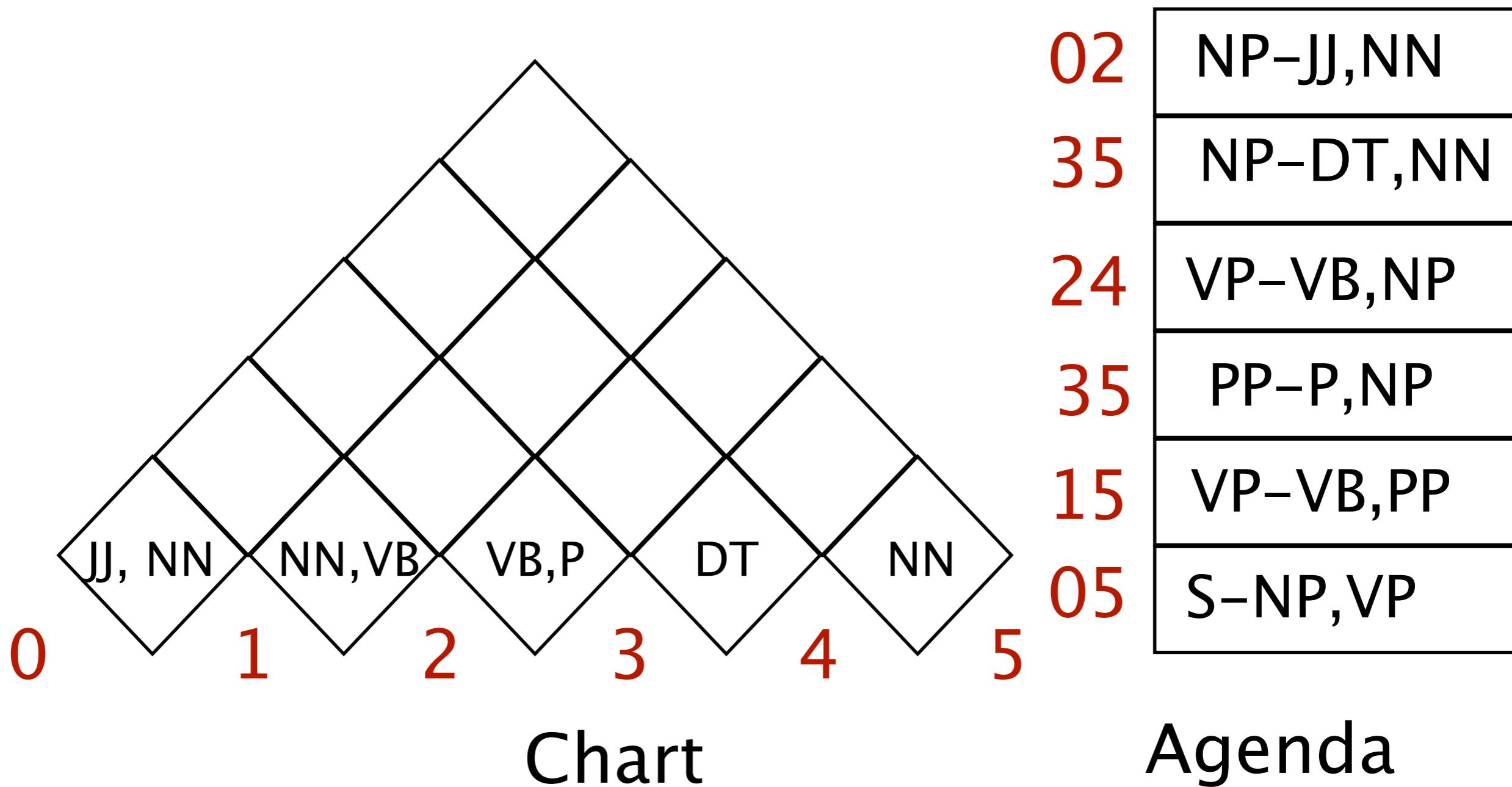


Chart-based parsing



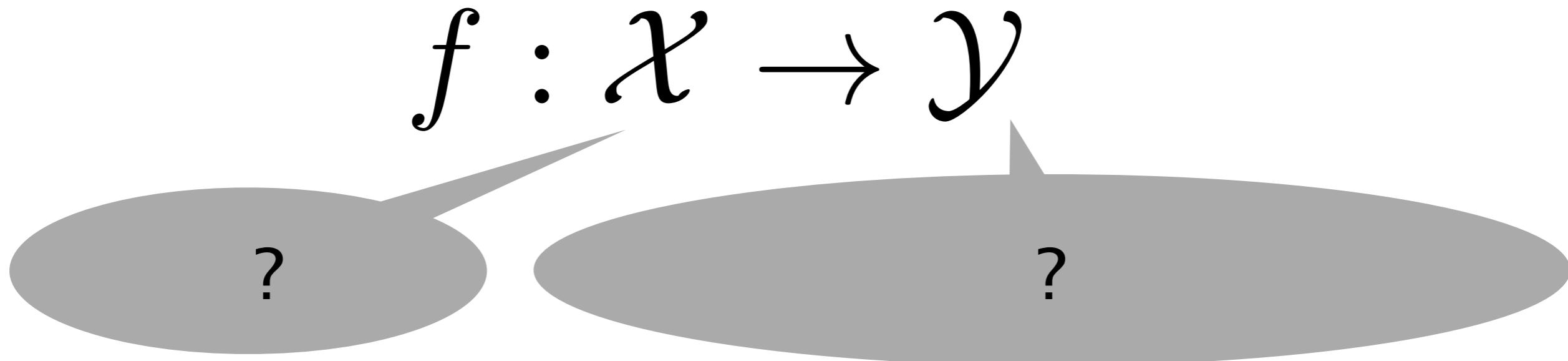
Agenda-Based Parsing



Introducing PS Parsing for MRLs

Representation

- Assume:
 - A morphologically rich lexicon
 - Phrase-structures on MSRs
- Then:



The Challenge

- Turkish
 - tan-ıs²-tır-ıl-a-ma-dık-lar-ın-dan-dır.
 - ‘It is because they cannot be introduced to each other.’
- Yu’pik (Central Alaska)
 - qaya:liy'u:l'u:n'i
 - ‘He was excellent at making kayaks’
- German
 - lebensversicherungsgesellschaftsangestellter

Defining the MRLon

- Assume:
 - Let **L** set of lemmas (a.k.a. senses)
 - Let **C** be a set of categories (a.k.a. POS)
 - Let **A** be a set of attributes
 - Let **V** be a set of values
- Then:
 - We define a MRLon entry **e=(l,c,{a:v})**

Defining the MRLon

"Lucy"	Lucy NN {gender:feminine, num:singular}
"Lucy"	Lucy NN {gender:mASCULINE, num:singular}
"snowing"	snow VB {tense:present}
"apples"	apple NN {num:plural, countable:+}
"sings"	sing VB {pers:3, num:singular, tense:present}
"the"	_ DET {denitness:+}
"a"	_ DET {denitness:-}
"will"	_ AUX {tense:future}
"is"	_ COP {pers:3, num:singular, tense:present}
"his"	_ PRP\$ {gender:mASCULINE, num:singular}
"yesterday"	yesterday RB {}
"knowingly"	knowingly RB {}

Table 1: An illustrative morphologically-rich lexicon for English

Defining the MRLon

Defining the MRLon

"John's" = "John"

"won't" = "will" + "not"

"she's" = "she" + "is"

Defining the MRLon

"John's" = "John"

"won't" = "will" + "not"

"she's" = "she" + "is"

"FMNH" = "FMNH"

"FMNH" = "F" + "MNH"

"FMNH" = "F" + "MN" + "H"

Defining the MRLon

"John's" = "John"

"won't" = "will" + "not"

"she's" = "she" + "is"

"FMNH" = "FMNH"

"FMNH" = "F" + "MNH"

"FMNH" = "F" + "MN" + "H"

"fat"

"that" + "counted"

"that" + "bird" + "her"

Defining the MRLon

"John's" = "John"

"won't" = "will" + "not"

"she's" = "she" + "is"

Each
Segment has
an MSR

"FMNH" = "FMNH"

"FMNH" = "F" + "MNH"

"FMNH" = "F" + "MN" + "H"

"fat"

"that" + "counted"

"that" + "bird" + "her"

Morphological Analysis (1)

$$\mathcal{MA} : \Sigma \rightarrow \mathcal{P}(\mathcal{L}^*)$$

Morphological Analyzer

space delimited tokens

Sequences of MRLon entries

Morphological Analysis (2)

$$\mathcal{MA} : \Sigma^* \rightarrow \mathcal{P}(\mathcal{L}^*)$$

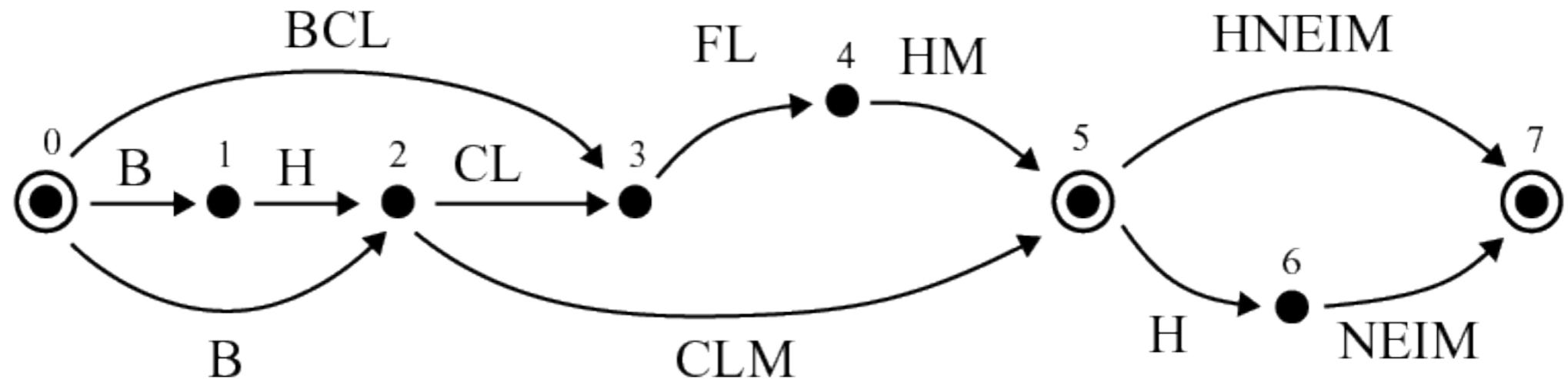
Morphological Analyzer

Sequences of tokens

Sequences of entries

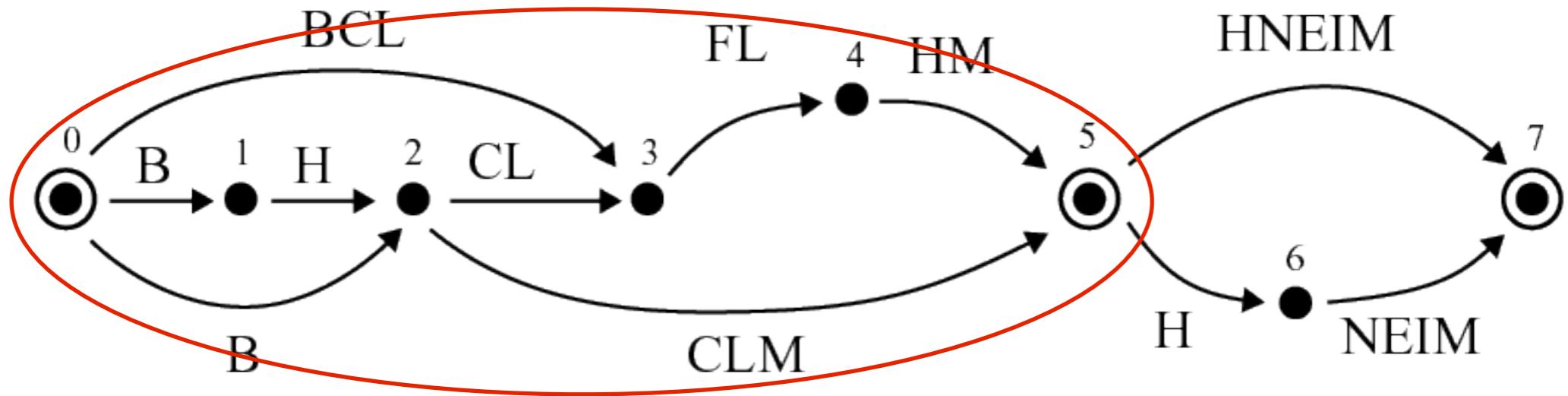
Morphological Analysis (3)

$\mathcal{MA}("BCLM\ HNEIM") =$



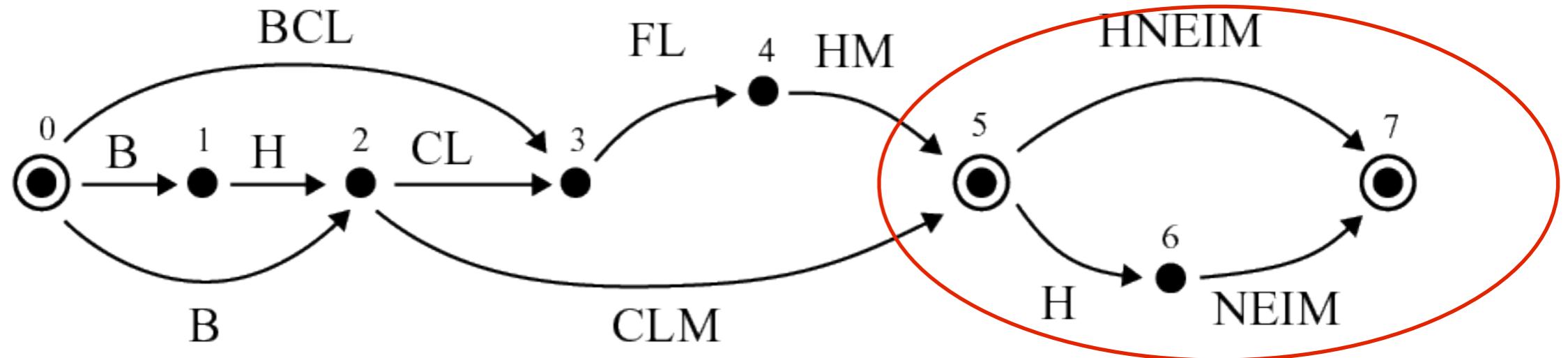
Morphological Analysis (3)

$\mathcal{MA}(\text{"BCLM HNEIM"}) =$



Morphological Analysis (3)

$\mathcal{MA}("BCLM \text{ HNEIM}") =$



Representation

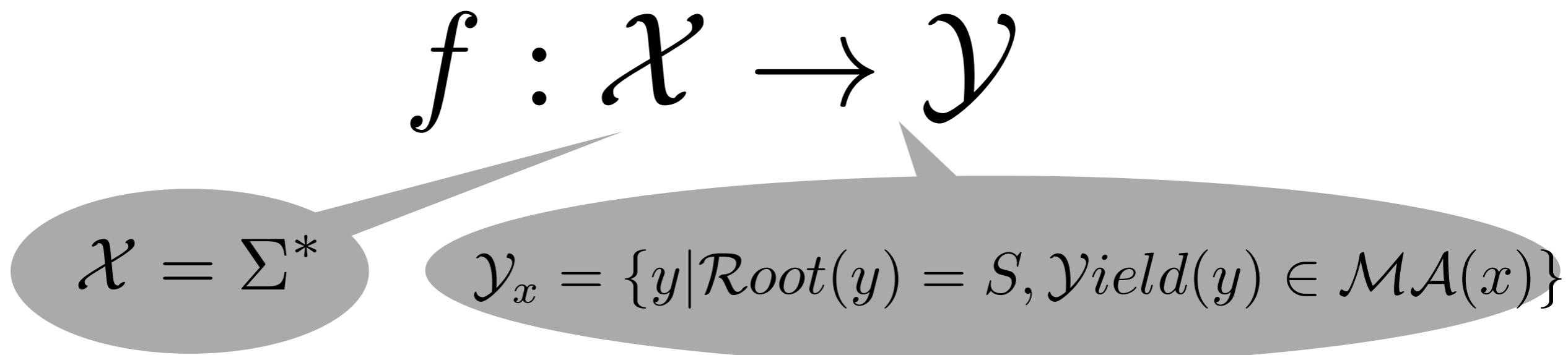
- Assume:
 - A morphologically rich lexicon
 - Phrase-structures on MSRs
- Then:

$$f : \mathcal{X} \rightarrow \mathcal{Y}$$

$\mathcal{Y}_x = \{y \mid \text{Root}(y) = S, \text{Yield}(y) \in \mathcal{MA}(x)\}$

Representation

- Assume:
 - A morphologically rich lexicon
 - Phrase-structures on MSRs
- Then:

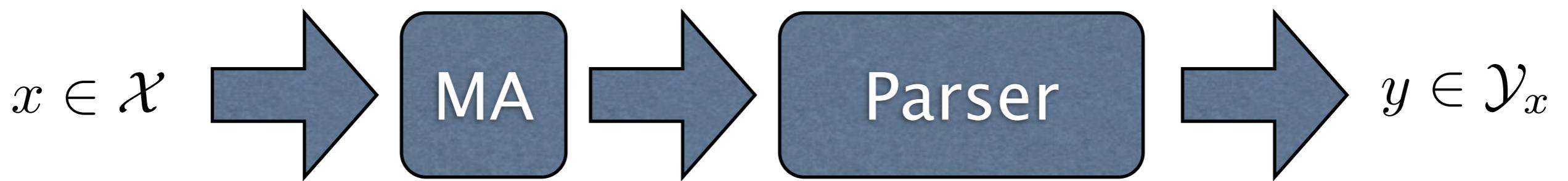


Two Possible Architectures

- Pipeline Architecture
- Joint Architecture

Two Possible Architectures

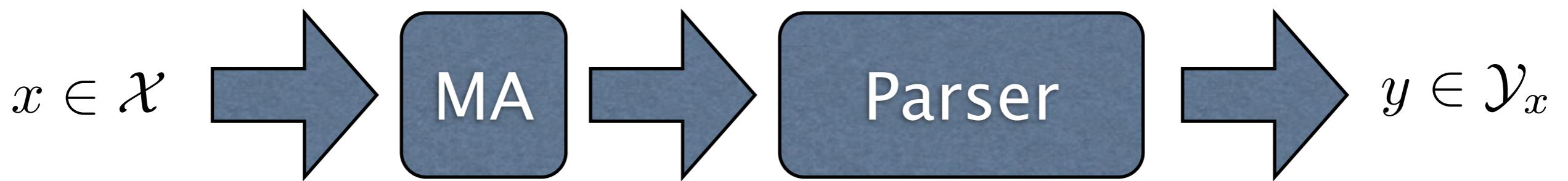
- Pipeline Architecture



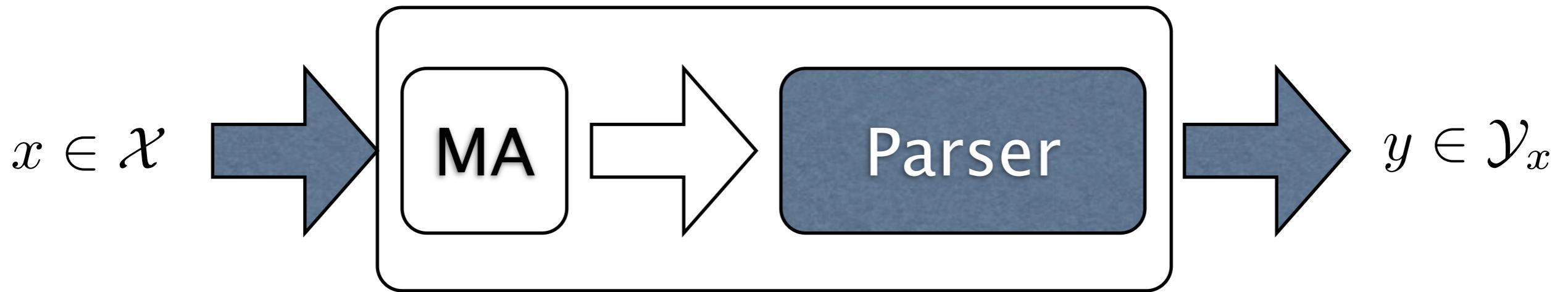
- Joint Architecture

Two Possible Architectures

- Pipeline Architecture



- Joint Architecture



Modeling

- Probabilistic Modeling

$$f(x) = \arg \max_{\{y | \text{Yield}(y) \in \mathcal{MA}(x)\}} P(y|x)$$

$$= \arg \max_{\{y | \text{Yield}(y) \in \mathcal{MA}(x)\}} \frac{P(y, x)}{P(x)}$$

$$= \arg \max_{\{y | \text{Yield}(y) \in \mathcal{MA}(x)\}} P(y, x)$$

Modeling

- Probabilistic Modeling

$$= \arg \max_{\{y | Yield(y) \in \mathcal{MA}(x)\}} P(y, x)$$

$$= \arg \max_{\{y | Yield(y) \in \mathcal{MA}(x)\}} P(y)$$

There is only one way to
compose the MSRs

Modeling

There is one way to compose the MSRs

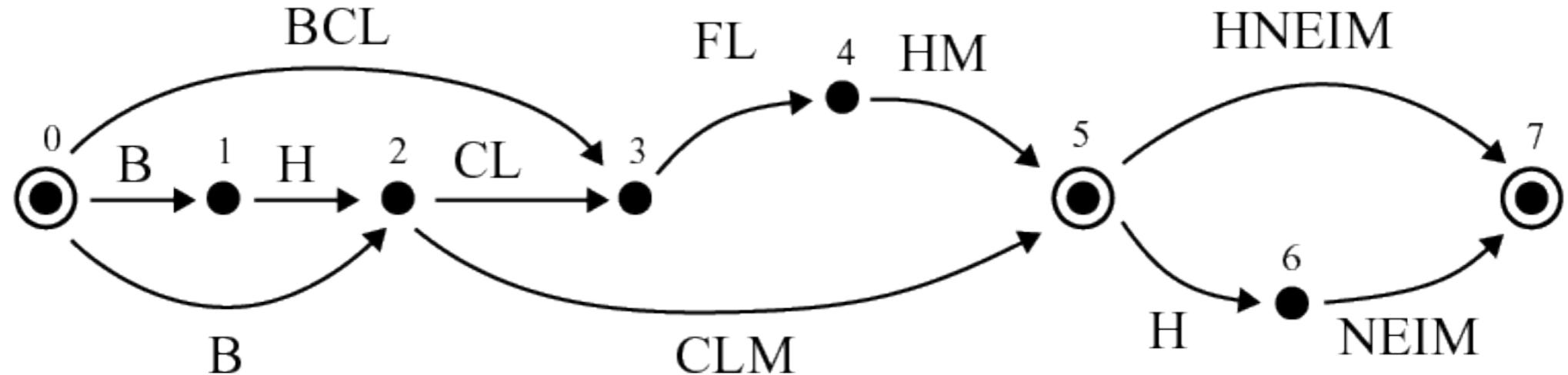
$$= \arg \max_{\{y | \text{Yield}(y) \in \mathcal{MA}(x)\}} P(y, x)$$

$$= \arg \max_{\{y | \text{Yield}(y) \in \mathcal{MA}(x)\}} P(y)$$

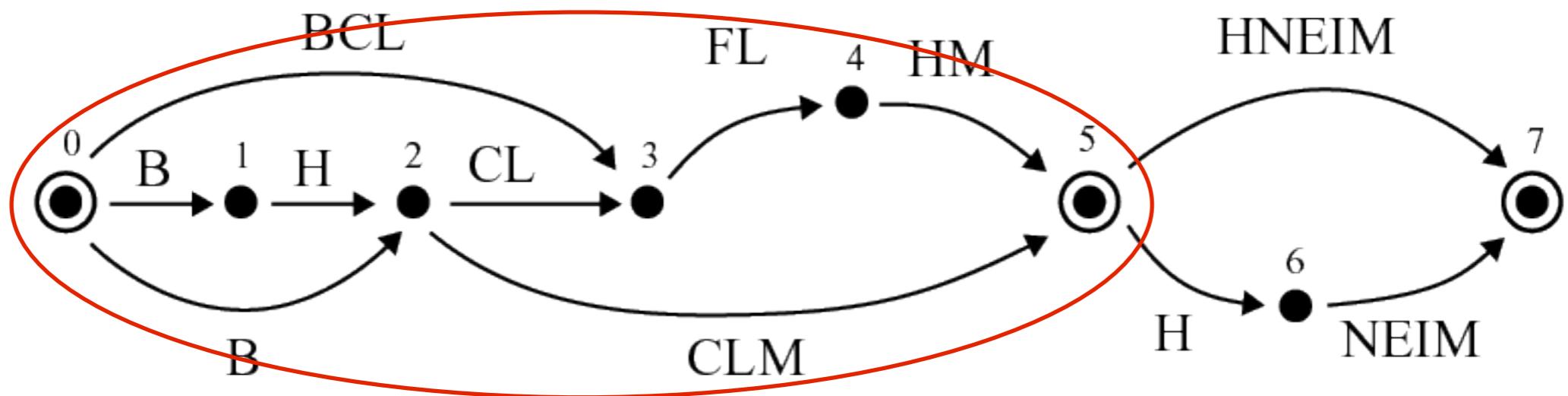
How do we do inference?

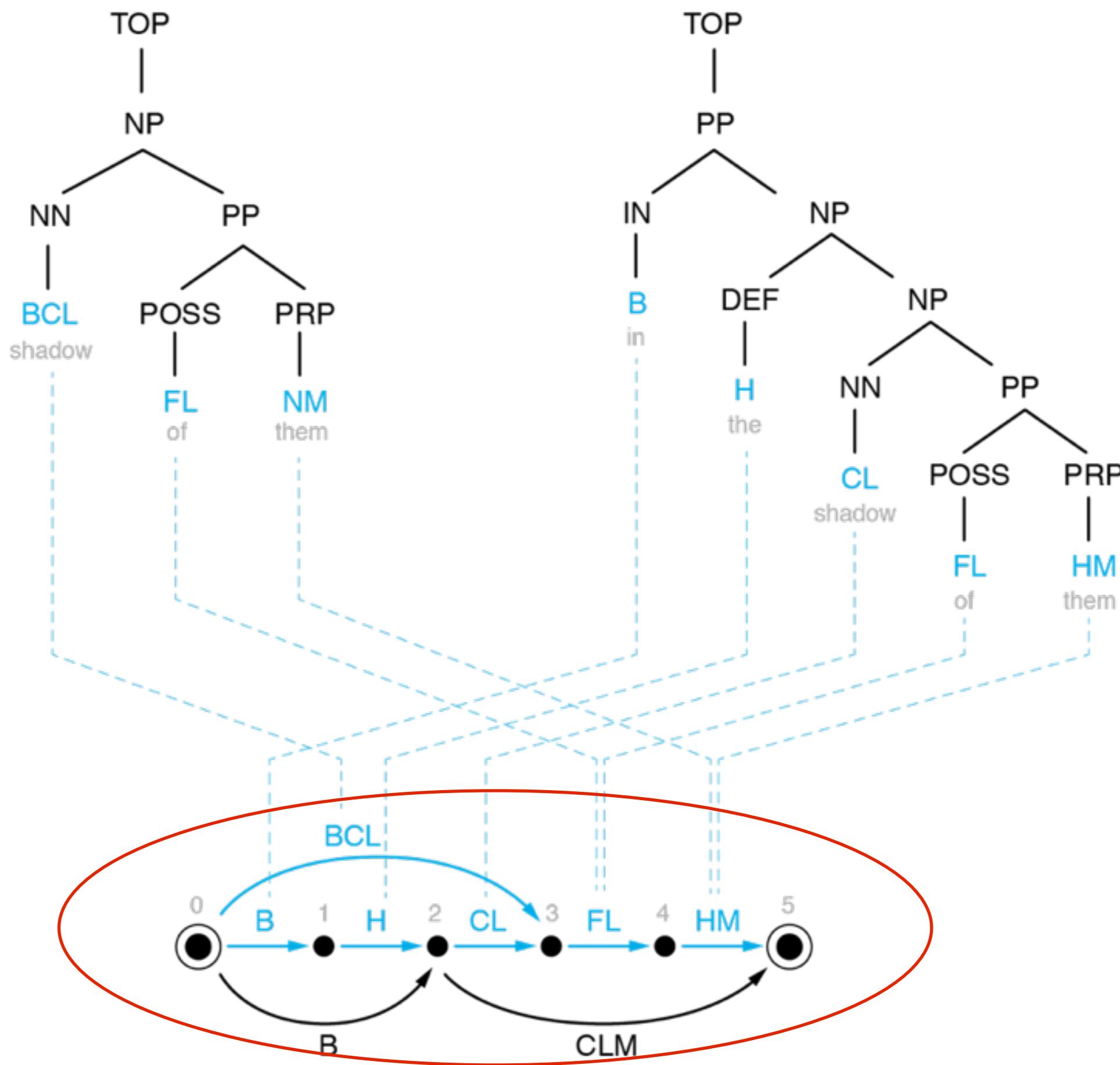
How do we learn a good model?

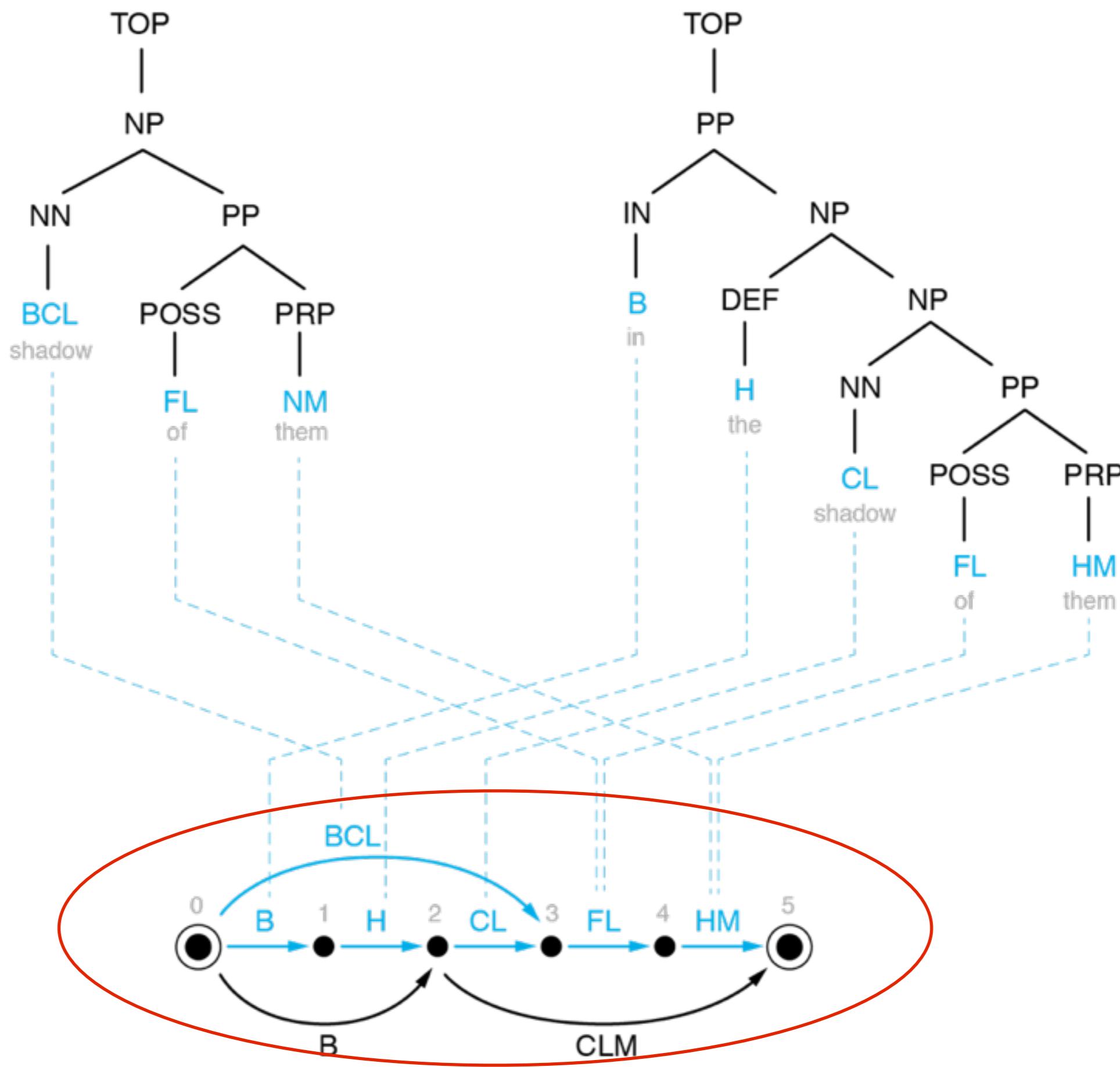
The Key Idea: Lattice Parsing



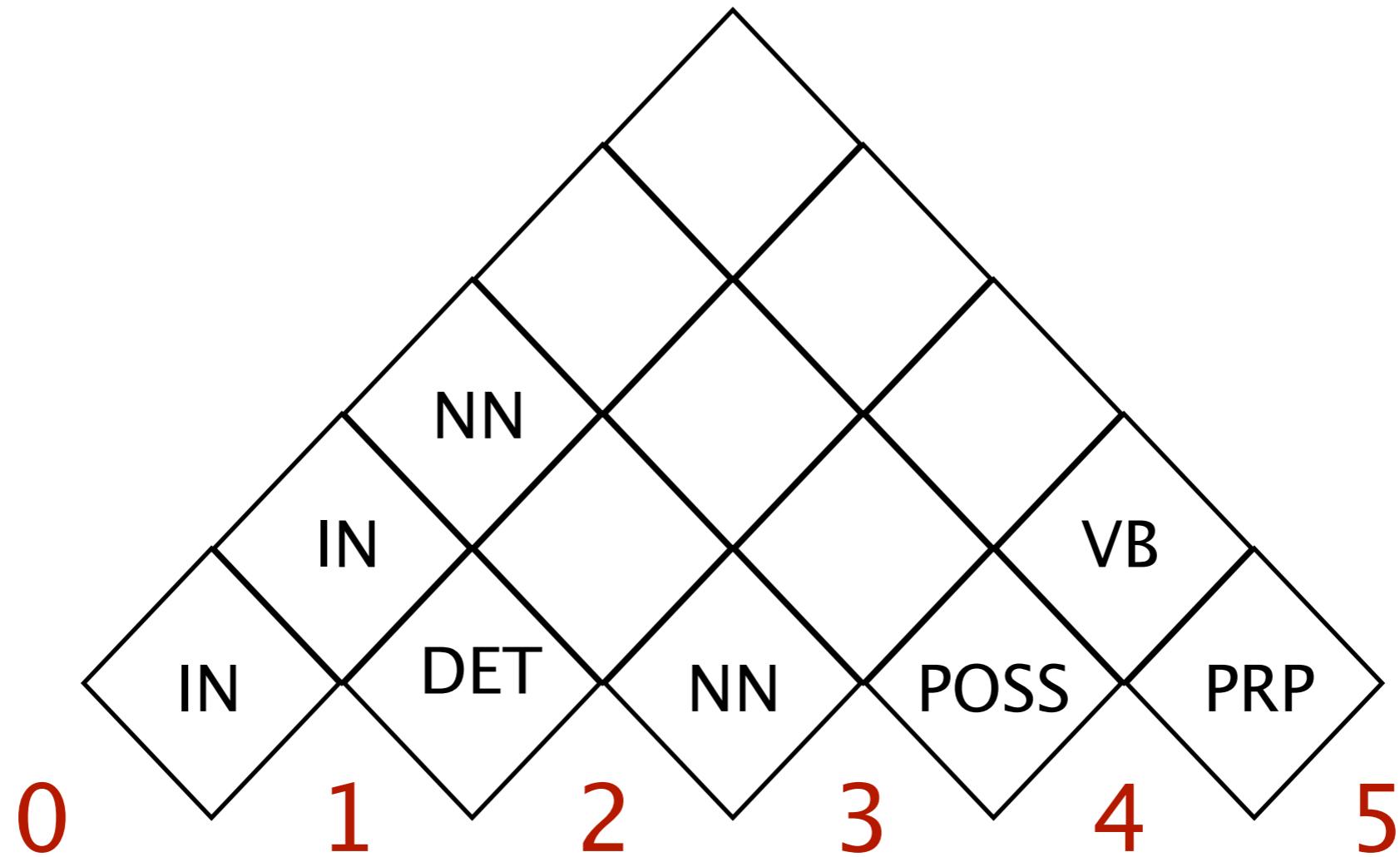
The Key Idea: Lattice Parsing



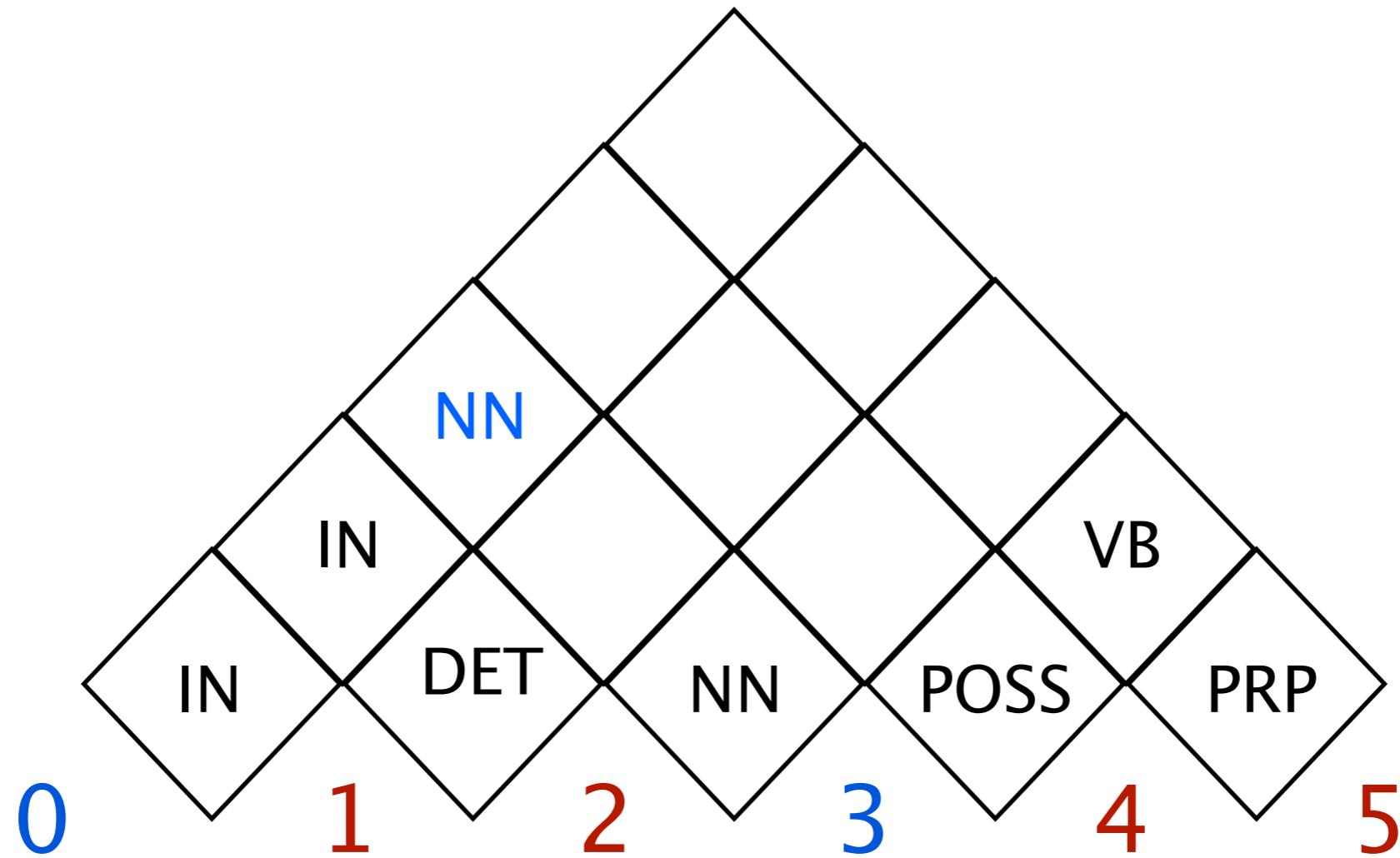




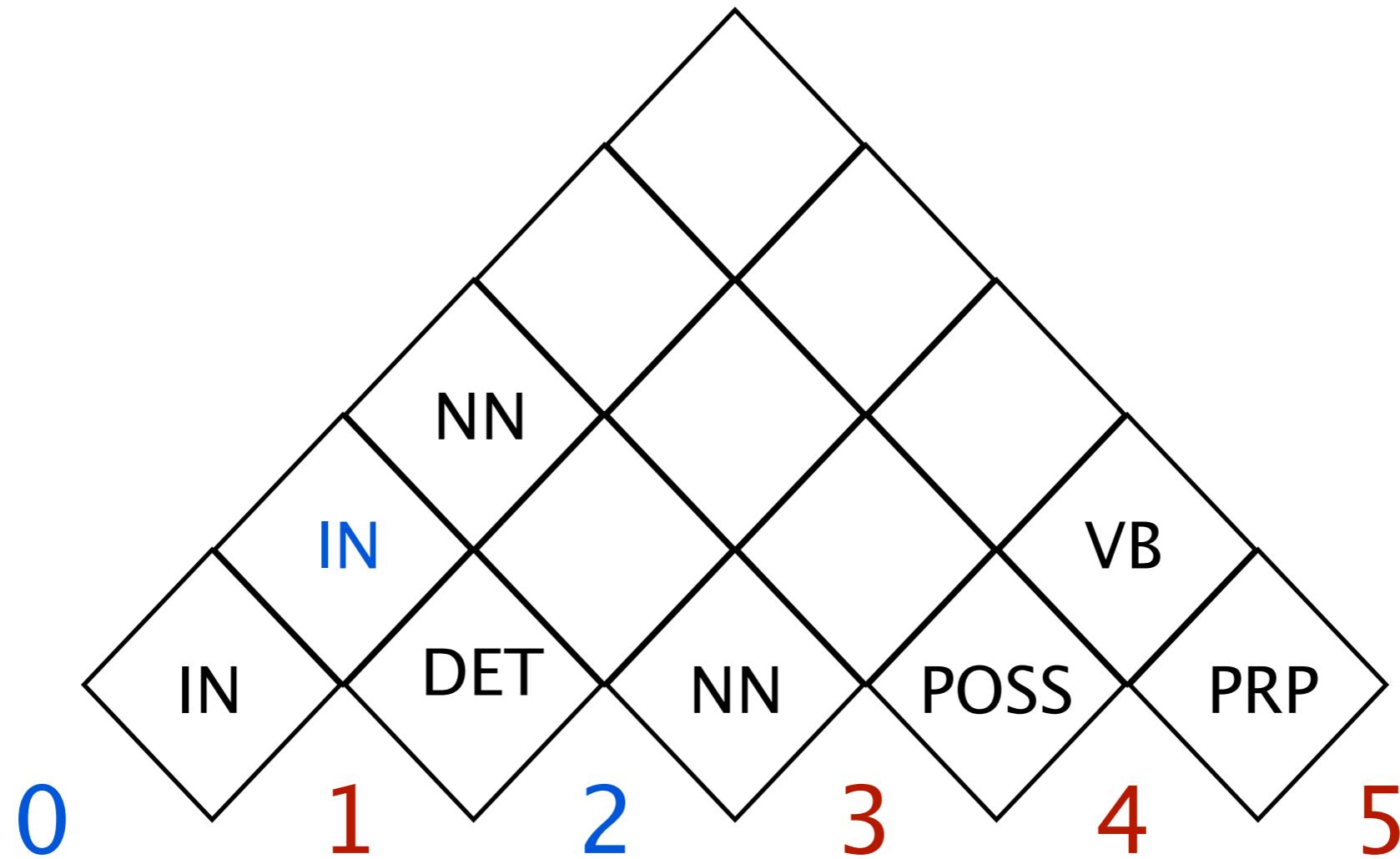
Agenda-Based (A*) Parsing



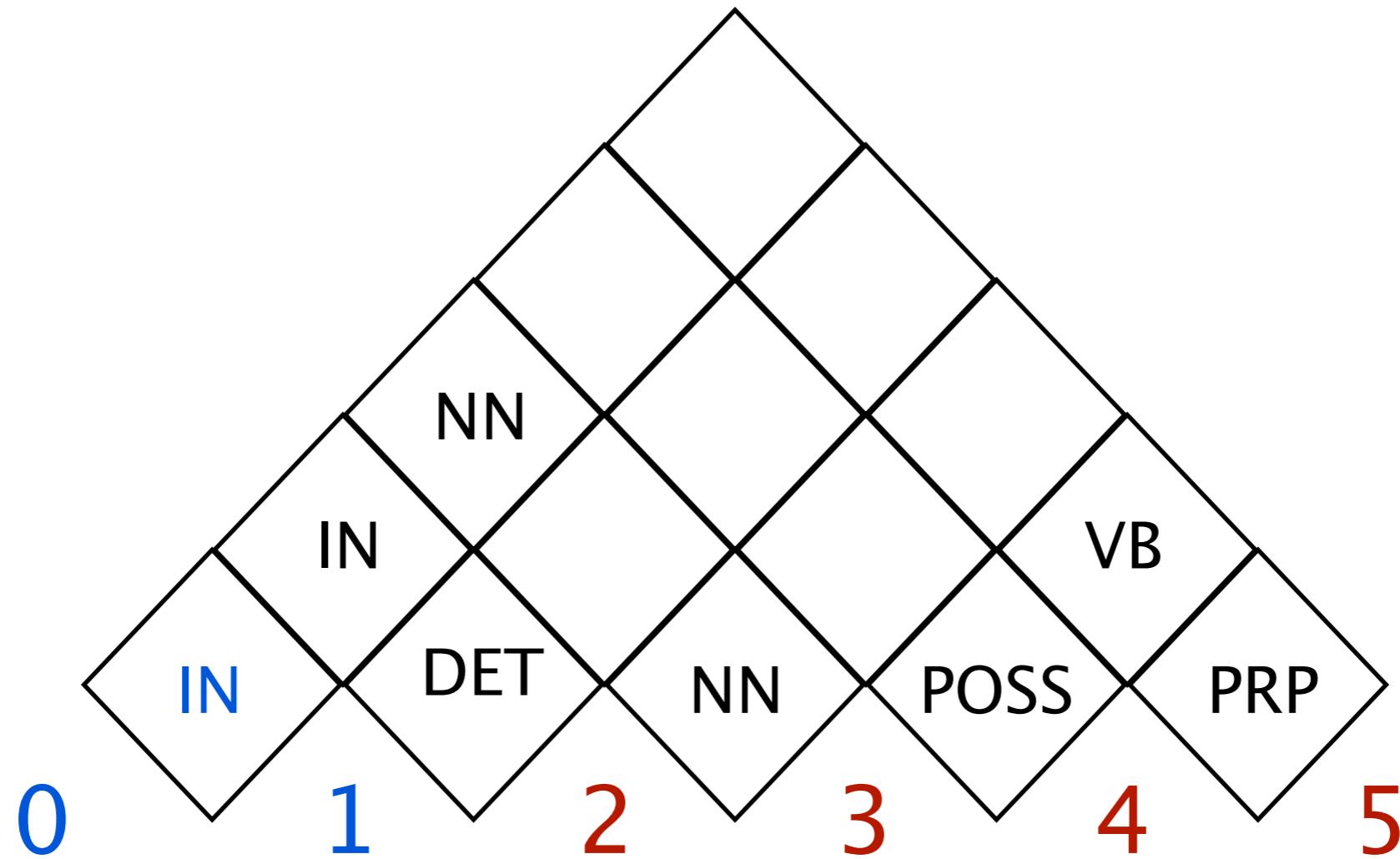
Agenda-Based (A*) Parsing



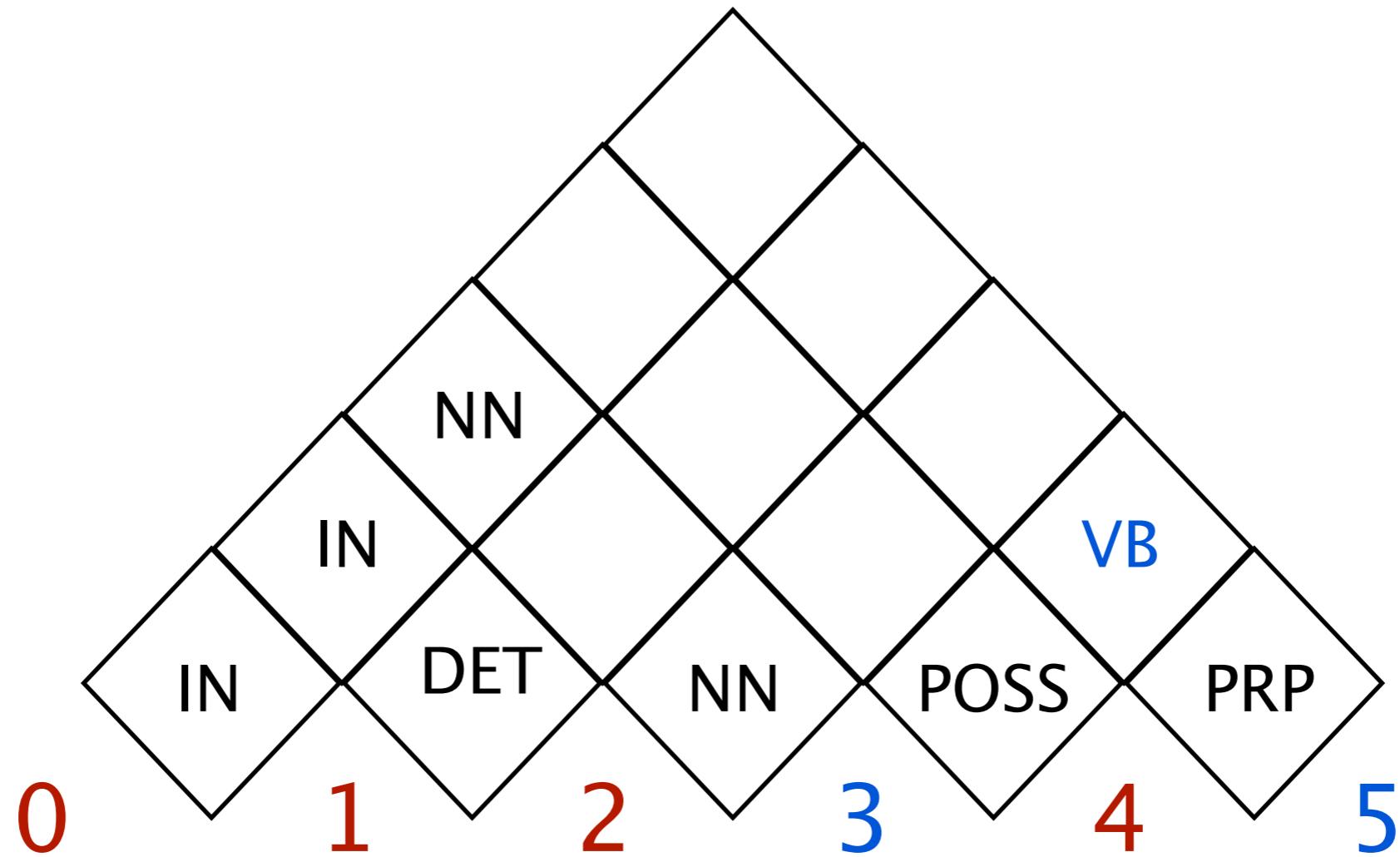
Agenda-Based (A*) Parsing



Agenda-Based (A*) Parsing



Agenda-Based (A*) Parsing



Lattice-Based Parsing

Algorithm 2 The CKY Algorithm for Lattice Parsing (with Back-Pointers)

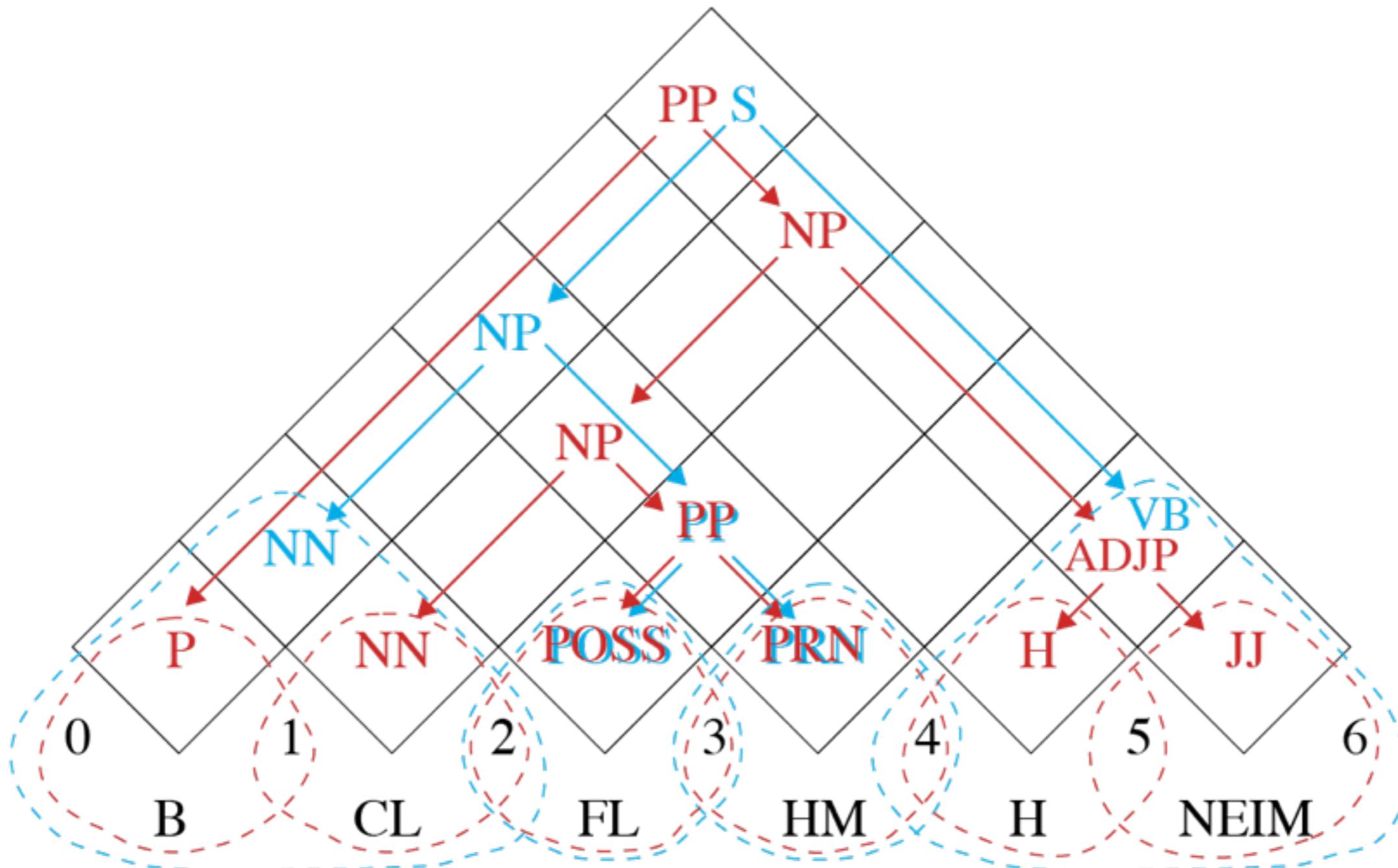
```
1: for  $\langle i, L, j \rangle \in \text{EDGES}(MA(x))$  do           ▷ traverse the morphological analysis lattice
2:    $\delta_L[i, j] \leftarrow p(A_L \rightarrow s_{i,j})$           ▷ Initiate segments probs
3:    $\beta_L[i, j] \leftarrow \langle s_{i,j} \rangle$             ▷ Store segments
4: for  $span = 2 \rightarrow n$  do                      ▷ Fill in the chart
5:   for  $end = span \rightarrow n$  do
6:      $begin \leftarrow end - span$ 
7:     for  $L = 1 \rightarrow |\mathcal{N}|$  do
8:        $\delta_L(begin, end) \leftarrow \max_{\langle m, J, K \rangle} p(A_L \rightarrow A_J A_K) \times \delta_J(begin, m) \times \delta_K(m, end)$ 
9:        $\beta_L(begin, end) \leftarrow \text{argmax}_{\langle m, J, K \rangle} p(A_L \rightarrow A_J A_K) \times \delta_J(begin, m) \times \delta_K(m, end)$ 
10:  return BUILD-TREE  $\delta_S[0, n]), \beta_S[0, n]$            ▷ Follow back-pointers
```

Lattice-Based Parsing

Algorithm 2 The CKY Algorithm for Lattice Parsing (with Back-Pointers)

```
1: for  $\langle i, L, j \rangle \in \text{EDGES}(MA(x))$  do                                ▷ traverse the morphological analysis lattice
2:    $\delta_L[i, j] \leftarrow p(A_L \rightarrow s_{i,j})$                                          ▷ Initiate segments probs
3:    $\beta_L[i, j] \leftarrow \langle s_{i,j} \rangle$                                             ▷ Store segments
4: for  $span = 2 \rightarrow n$  do                                                 ▷ Fill in the chart
5:   for  $end = span \rightarrow n$  do
6:      $begin \leftarrow end - span$ 
7:     for  $L = 1 \rightarrow |\mathcal{N}|$  do
8:        $\delta_L(begin, end) \leftarrow \max_{\langle m, J, K \rangle} p(A_L \rightarrow A_J A_K) \times \delta_J(begin, m) \times \delta_K(m, end)$ 
9:        $\beta_L(begin, end) \leftarrow \text{argmax}_{\langle m, J, K \rangle} p(A_L \rightarrow A_J A_K) \times \delta_J(begin, m) \times \delta_K(m, end)$ 
10:  return BUILD-TREE  $\delta_S[0, n]), \beta_S[0, n]$                                          ▷ Follow back-pointers
```

Chart-Based Decoding



Homework

- Extract a Treebank PCFG from the MRL trees on the previous slide
- Fill in the chart with node probabilities according to the lattice-based CKY
- Which sentence received the highest probability?
- Think of how you could use lattice-based decoding for parsing MWEs. (Provide pseudo-code, or, better: implement!)

Architectural Decisions

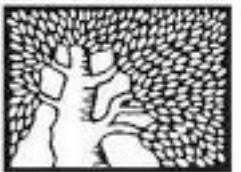
- Representation: Phrase-Structures
- Model: Probabilistic Grammars
- Inference: Chart-Based Algorithms
- Learning: ?
- Evaluation: ?

Tomorrow@PMRL

- ✓ Day 1: Introduction
- ✓ Day 2: Phrase-structure (PS) Parsing
- ✓ PS Inference in English and MRLs
- PS Learning in English and MRLs
- Day 3: Relational–Realizational
- Day 4: Dependency–structure
- Day 5: Evaluation and Multilinguality



UPPSALA
UNIVERSITET



WEIZMANN
INSTITUTE

Thanks

reut.tsarfaty@weizmann.ac.il