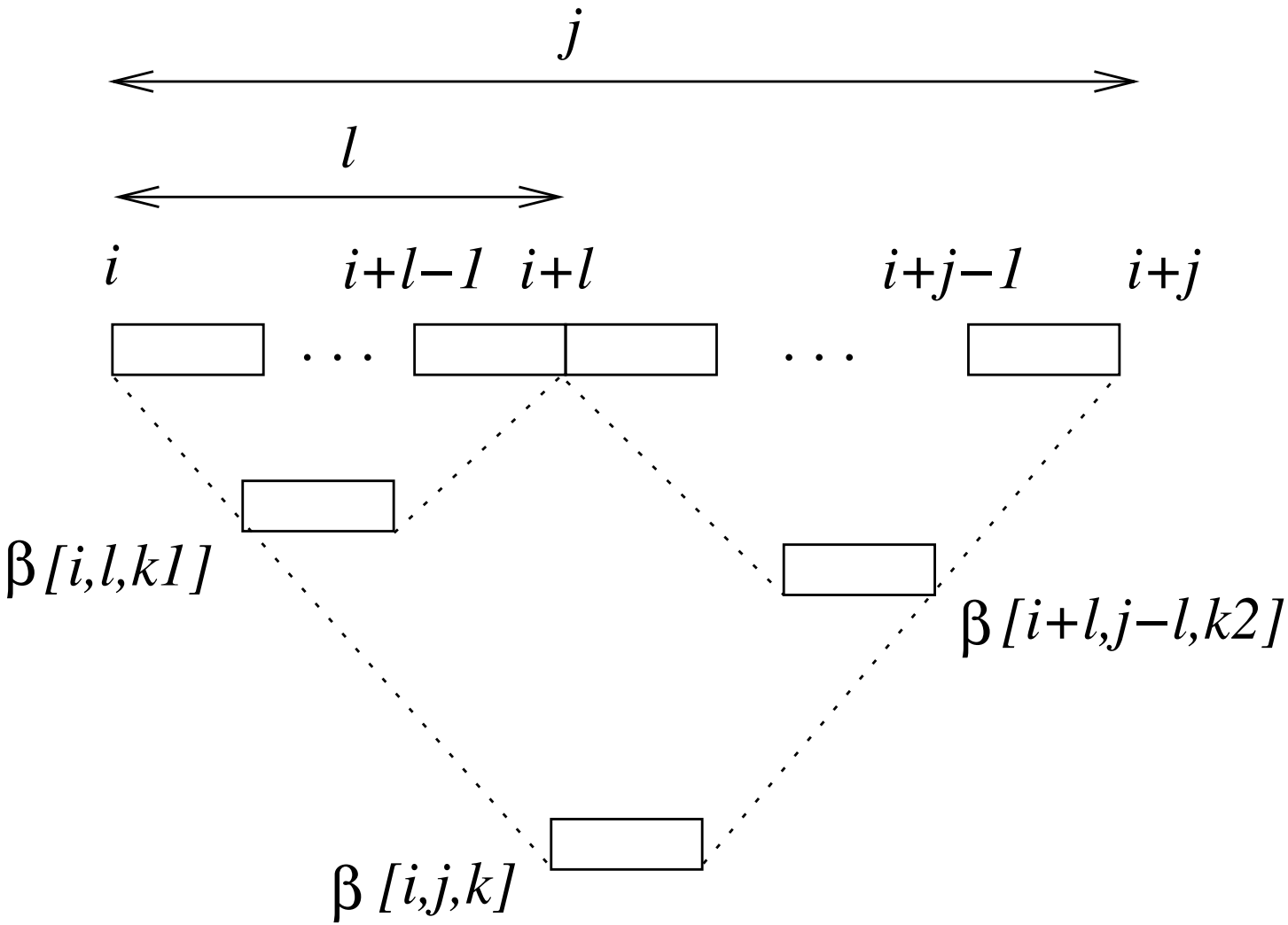
Natural Language Processing
CSCI 4152/6509 — Lecture 22
CYK Algorithm

Instructors: Vlado Keselj
Time and date: 13:05 – 14:25, 26-Nov-2026
Location: Goldberg Computer Science Building rm143

Previous Lecture

- Arguments and Adjuncts
- Some Parse Tree Examples
- Probabilistic Context-Free Grammar
  - PCFG as a probabilistic model
- **CYK Chart Parsing Algorithm**
- Chomsky Normal Form (CNF)
- CYK algorithm example

Explanation of Index Use in CYK



CYK Algorithm

Require: sentence $= w_1 \dots w_n$, and a CFG in CNF with nonterminals $N^1 \dots N^m$,
N^1 is the start symbol

Ensure: parsed sentence

allocate matrix $\beta \in \{0,1\}^{n \times n \times m}$ and initialize all entries to 0
for $i \leftarrow 1$ to n **do**
  for all rules $N^k \rightarrow w_i$ **do**
    $\beta[i,1,k] \leftarrow 1$
for $j \leftarrow 2$ to n **do**
  for $i \leftarrow 1$ to $n - j + 1$ **do**
    for $l \leftarrow 1$ to $j - 1$ **do**
      for all rules $N^k \rightarrow N^{k_1} N^{k_2}$ **do**
        $\beta[i,j,k] \leftarrow \beta[i,j,k]$ OR $(\beta[i,l,k_1]$ AND $\beta[i+l, j-l, k_2])$
return $\beta[1,n,1]$

Efficient Inference in PCFG Model

- consider marginalization task:

$\mathrm{P}(\text{sentence}) = ?$

- or: $\mathrm{P}(\text{sentence}) = \mathrm{P}(w_1 w_2 \ldots w_n | S)$
- One way to compute:

$$\mathrm{P}(\text{sentence}) = \sum_{t \in T} \mathrm{P}(t),$$

- Likely inefficient; need a parsing algorithm

Efficient PCFG Marginalization

- Idea: adapt CYK algorithm to store marginal probabilities
- Replace algorithm line:

$$\beta[i,j,k] \leftarrow \beta[i,j,k] \text{ OR } (\beta[i,l,k_1] \text{ AND } \beta[i+l,j-l,k_2])$$

with

$$\beta[i,j,k] \leftarrow \beta[i,j,k] + \mathrm{P}(N^k \rightarrow N^{k_1} N^{k_2}) \cdot \beta[i,l,k_1] \cdot \beta[i+l,j-l,k_2]$$

- and the first-chart-row line:

$$\beta[i,1,k] \leftarrow 1$$

with

$$\beta[i,1,k] \leftarrow \mathrm{P}(N^k \rightarrow w_i)$$

Probabilistic CYK for Marginalization

Require: sentence $= w_1 \dots w_n$, and a PCFG in CNF with nonterminals $N^1 \dots N^m$, N^1 is the start symbol

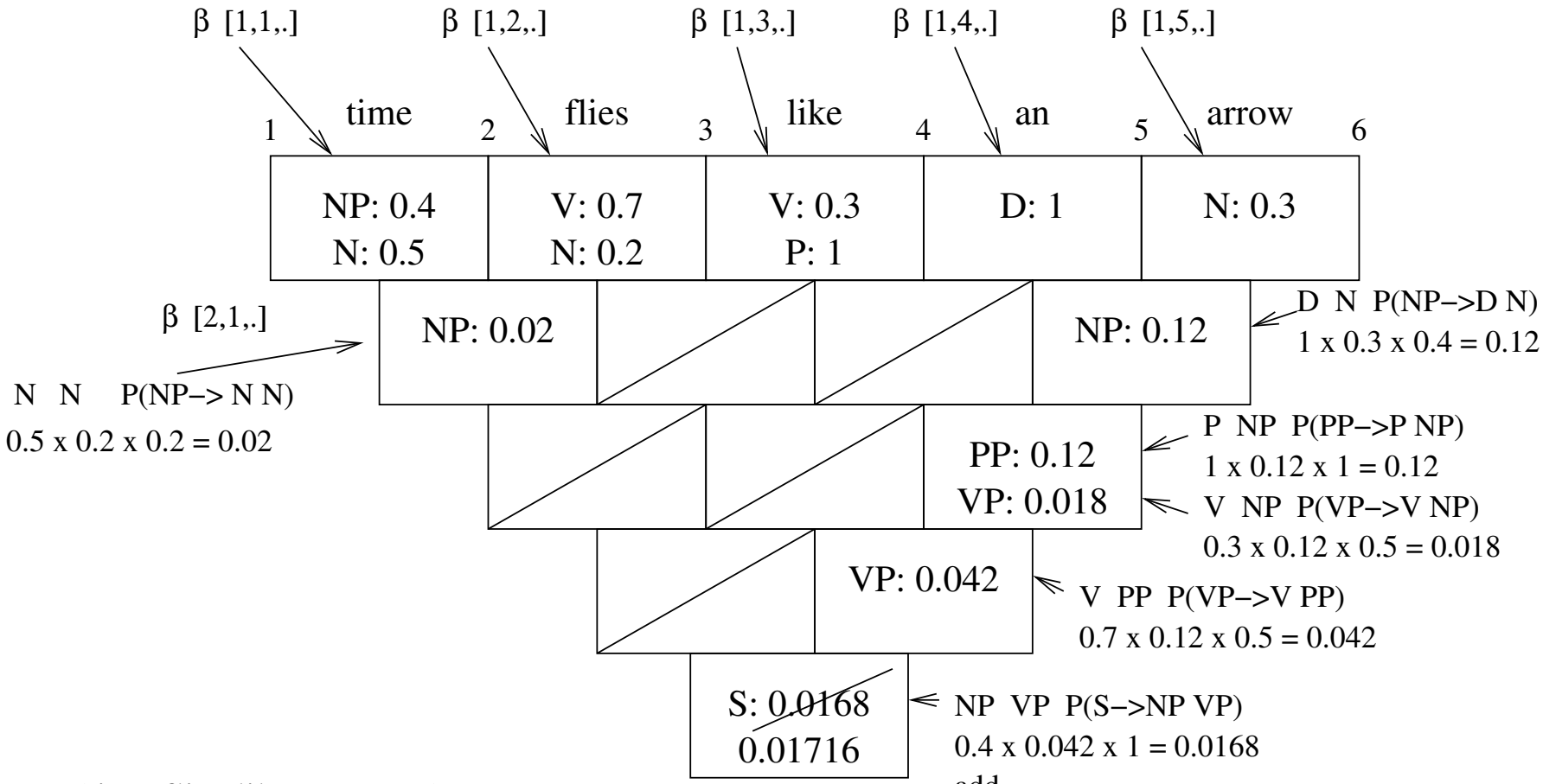
Ensure: P(sentence) is returned

allocate $\beta \in \mathbb{R}^{n \times n \times m}$ and initialize all entries to 0
for $i \leftarrow 1$ to n **do**
  for all rules $N^k \rightarrow w_i$ **do**
    $\beta[i, 1, k] \leftarrow \mathrm{P}(N^k \rightarrow w_i)$
for $j \leftarrow 2$ to n **do**
  for $i \leftarrow 1$ to $n - j + 1$ **do**
    for $l \leftarrow 1$ to $j - 1$ **do**
      for all rules $N^k \rightarrow N^{k_1} N^{k_2}$ **do**
        $\beta[i, j, k] \leftarrow \beta[i, j, k] +$
        $\mathrm{P}(N^k \rightarrow N^{k_1} N^{k_2}) \cdot \beta[i, l, k_1] \cdot \beta[i + l, j - l, k_2]$
return $\beta[1, n, 1]$

PCFG Marginalization Example (grammar)

S	$\rightarrow$	NP VP	/1	VP	$\rightarrow$	V NP	/.5	N	$\rightarrow$	time	/.5
NP	$\rightarrow$	time	/.4	VP	$\rightarrow$	V PP	/.5	N	$\rightarrow$	arrow	/.3
NP	$\rightarrow$	N N	/.2	PP	$\rightarrow$	P NP	/1	N	$\rightarrow$	flies	/.2
NP	$\rightarrow$	D N	/.4					D	$\rightarrow$	an	/1
V	$\rightarrow$	like	/.3								
V	$\rightarrow$	flies	/.7								
P	$\rightarrow$	like	/1								

PCFG Marginalization Example (chart)



P(time flies like an arrow) =

$= 0.01716$

Conditioning

- Conditioning in the PCFG model: $\mathrm{P}(\mathrm{tree}|\mathrm{sentence})$
- Use the formula:

$$\mathrm{P}(\mathrm{tree}|\mathrm{sentence}) = \frac{\mathrm{P}(\mathrm{tree}, \mathrm{sentence})}{\mathrm{P}(\mathrm{sentence})} = \frac{\mathrm{P}(\mathrm{tree})}{\mathrm{P}(\mathrm{sentence})}$$

- $\mathrm{P}(\mathrm{tree})$ — directly evaluated
- $\mathrm{P}(\mathrm{sentence})$ — marginalization

Completion

- Finding the most likely parse tree of a sentence:

$$\arg\max_{\text{tree}} \mathrm{P}(\text{tree}|\text{sentence})$$

- Use the CYK algorithm in which line 9 is replaced with:

9: $\beta[i,j,k] \leftarrow \max(\beta[i,j,k], \mathrm{P}(N^k \rightarrow N^{k_1} N^{k_2}) \cdot \beta[i,l,k_1] \cdot \beta[i+l, j-l, k_2])$

- Return the most likely tree

CYK-based Completion Algorithm

Require: sentence $= w_1 \dots w_n$, and a PCFG in CNF with nonterminals $N^1 \dots N^m$, N^1 is the start symbol

Ensure: The most likely parse tree is returned

allocate $\beta \in \mathbb{R}^{n \times n \times m}$ and initialize all entries to 0
for $i \leftarrow 1$ to n **do**
  for all rules $N^k \rightarrow w_i$ **do**
    $\beta[i, 1, k] \leftarrow \mathrm{P}(N^k \rightarrow w_i)$
for $j \leftarrow 2$ to n **do**
  for $i \leftarrow 1$ to $n - j + 1$ **do**
    for $l \leftarrow 1$ to $j - 1$ **do**
      for all rules $N^k \rightarrow N^{k_1} N^{k_2}$ **do**
        $\beta[i, j, k] \leftarrow \max(\beta[i, j, k], \mathrm{P}(N^k \rightarrow N^{k_1} N^{k_2}) \cdot \beta[i, l, k_1] \cdot \beta[i + l, j - l, k_2])$
return Reconstruct$(1, n, 1, \beta)$

Algorithm: Reconstruct(i, j, k, β)

Require: β — table from CYK, i — index of the first word, j — length of sub-string sentence, k — index of non-terminal

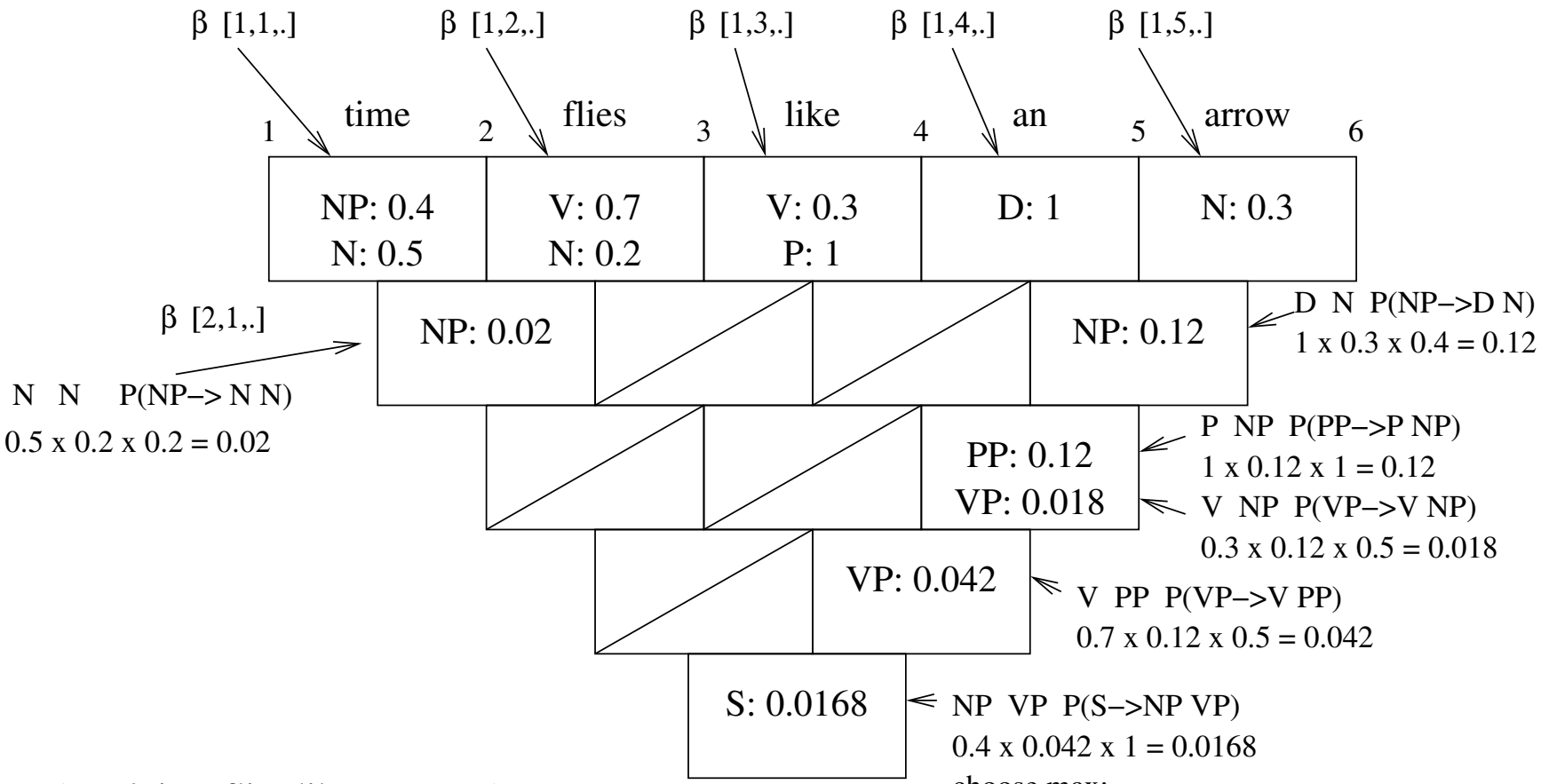
Ensure: a most probable tree with root N^k and leaves $w_i \ldots w_{i+j-1}$ is returned

1: **if** $j = 1$ **then**
2: **return** tree with root N^k and child w_i
3: **for** $l \leftarrow 1$ to $j - 1$ **do**
4: **for all** rules $N^k \rightarrow N^{k_1} N^{k_2}$ **do**
5: **if** $\beta[i, j, k] = \mathrm{P}(N^k \rightarrow N^{k_1} N^{k_2}) \cdot \beta[i, l, k_1] \cdot \beta[i + l, j - l, k_2]$ **then**
6: create a tree t with root N^k
7: $t.\mathit{left_child} \leftarrow \mathrm{Reconstruct}(i, l, k_1, \beta)$
8: $t.\mathit{right_child} \leftarrow \mathrm{Reconstruct}(i + l, j - l, k_2, \beta)$
9: **return** t

PCFG Completion Example (grammar)

S	$\rightarrow$	NP VP	/1	VP	$\rightarrow$	V NP	/.5	N	$\rightarrow$	time	/.5
NP	$\rightarrow$	time	/.4	VP	$\rightarrow$	V PP	/.5	N	$\rightarrow$	arrow	/.3
NP	$\rightarrow$	N N	/.2	PP	$\rightarrow$	P NP	/1	N	$\rightarrow$	flies	/.2
NP	$\rightarrow$	D N	/.4					D	$\rightarrow$	an	/1
V	$\rightarrow$	like	/.3								
V	$\rightarrow$	flies	/.7								
P	$\rightarrow$	like	/1								

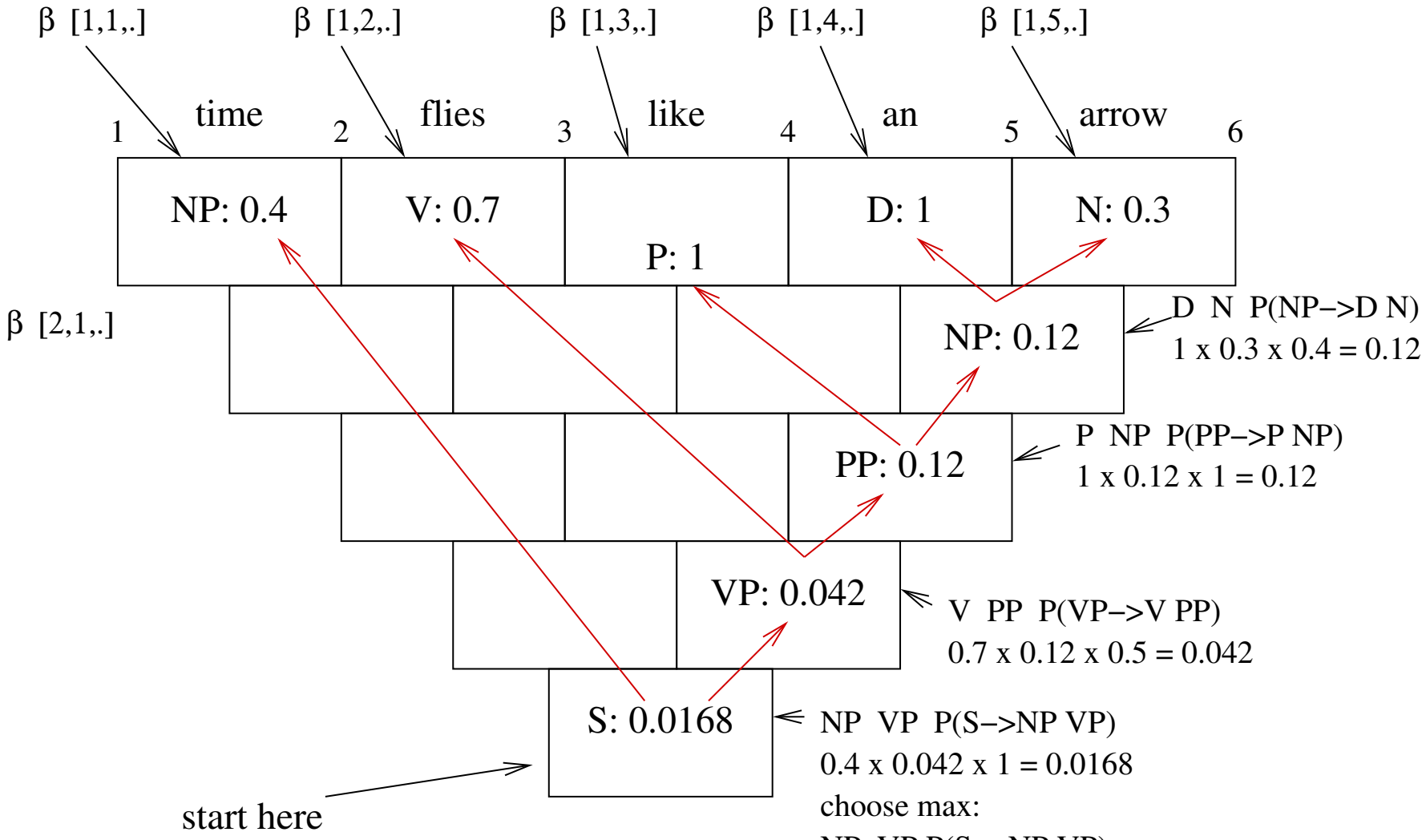
PCFG Completion Example (chart)



max P(tree | time flies like an arrow) =

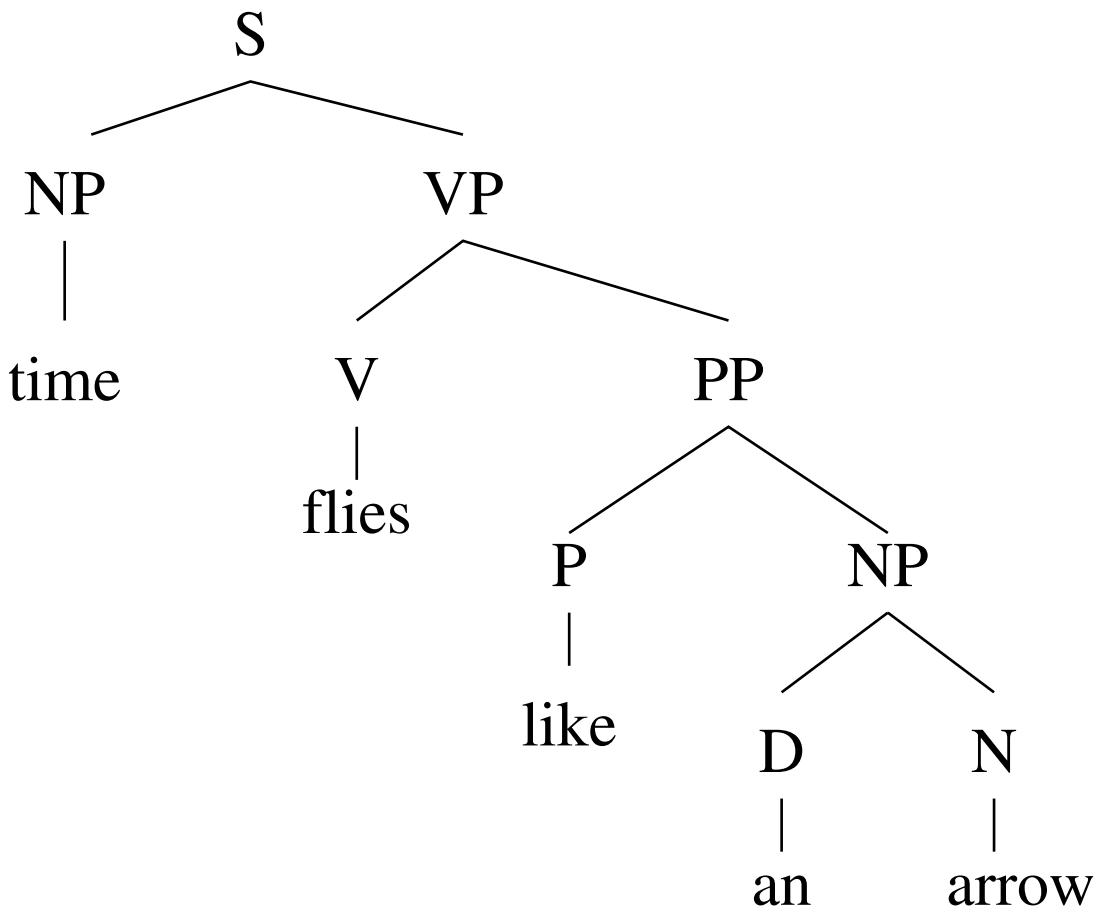
= 0.0168

PCFG Completion Example (tree reconstruction)



PCFG Completion Example (final tree)

The most probable three:



Issues with PCFGs

1. Structural dependencies
   - Dependency on position in a tree
   - Example: consider rules NP $\rightarrow$ PRP and NP $\rightarrow$ DT NN
   - PRP is more likely as a subject than an object
   - NL parse trees are usually deeper on their right side
2. Lexical dependencies
   - Example: PP-attachment problem
   - In a PCFG, decided using probabilities for higher level rules; e.g., $\mathrm{NP} \rightarrow \mathrm{NP\ PP}$, $\mathrm{VP} \rightarrow \mathrm{VBD\ NP}$, and $\mathrm{VP} \rightarrow \mathrm{VBD\ NP\ PP}$
   - Actually, they frequently depend on the actual words

PP-Attachment Example

- Consider sentences:
  - "Workers dumped sacks into a bin." and
  - "Workers dumped sacks of fish."
- and rules:
  - $\text{NP} \rightarrow \text{NP PP}$
  - $\text{VP} \rightarrow \text{VBD NP}$
  - $\text{VP} \rightarrow \text{VBD NP PP}$

A Solution: Probabilistic Lexicalized CFGs

- use heads of phrases
- expanded set of rules, e.g.:
  VP(dumped) $\rightarrow$ VBD(dumped) NP(sacks) PP(into)
- large number of new rules
- sparse data problem
- solution: new independence assumptions

Are Natural Languages Context-Free?

- Can we use CFG directly to model the syntax?
- Surprisingly effective in many cases
- However, not considered sufficient
- Some NL are provably not context-free due to $ww = w^2$ forms
- Additionally, NL Phenomena

Natural Language Phenomena

Three well-known phenomena:

- Agreement
- Movement
- Subcategorization

Agreement

- Phenomenon which requires that constituents must agree on some features before being combined to larger constituents
- Example: "This book" vs. "These book"*, or "He works" vs. "He work"*
- The relevant features are propagated from child nodes to parent nodes; e.g., consider examples:
  These problems usually persist.
  This problem usually persists.

Agreement Examples

- subject-verb agreement
  For example, "I work." and "He works." vs. *"I works." and *"He work."
- specifier-head agreement
  For example, "This book." and "These books." vs. *"This books." and "These book."

Agreement can be a non-local dependency, e.g:
The **women** who found the wallet **were** given a reward.

Movement

- *movement:* an natural language phenomenon, in which a constituent in a grammatically valid sentence, can sometimes be moved to another position and the new sentence remains grammatically valid
- example: "Are you well?" from "You are well."

Movement Examples

E.g, wh-movement

Which book should Peter buy ?
filler *gap*

Another example:
```
(S (NP (NP Air Canada) ,
       (NP (NP-*filler* one of many airline companies)
           (SBAR that (S (NP-*gap*)
                         (VP flies from Halifax
                                    to Toronto))
           )) ,
   (VP canceled the flights yesterday) ) . )
```

Subcategorization

- **Subcategorization** phenomenon: tendency of verbs to prefer or require certain types of arguments
- Example, correct sentences:
  The defendant **disappeared**.
  The defendant **denied** *the accusation.*
- but the following sentences are not correct:
  The defendant **denied**.
  The defendant **disappeared** *the accusation.*
- The verbs 'deny' and 'disappear' belong to different subcategories.
- For example, some verbs do not take a noun-phrase object, and some do (direct and indirect objects)

Parser Evaluation

- PARSEVAL measures are used to evaluate context-free parsing performance
- Precision and recall of labeled and unlabeled constituents

$$\text{labelled recall} = \frac{\text{number of correct LC in PT}}{\text{number of LC in GT}}$$

$$\text{labelled precision} = \frac{\text{number of correct LC in PT}}{\text{number of LC in PT}}$$

$$\text{F-measure} = \frac{2 \cdot (\text{labelled precision}) \cdot (\text{labelled recall})}{(\text{labelled precision}) + (\text{labelled recall})}$$

- Labeled constituent: (span-start, span-end, non-terminal)
- Example: (0, 2, NP)

Example: PARSEVAL Measures

Let us consider the following two sentences:

Time flies like an arrow.

and

He ate the cake with a spoon.

Gold standard

```
(S (NP (NN time) (NN flies))
   (VP (VB like)
       (NP (DT an) (NN arrow))))

(S (NP (PRP he))
   (VP (VBD ate) (NP (DT the)
                     (NN cake))
       (PP (IN with)
           (NP (DT a) (NN spoon)))))
```

Parser result

```
(S (NP (NN time))
   (VP (VB flies)
       (PP (IN like)
           (NP (DT an) (NN arrow)))))

(S (NP (PRP he))
   (VP (VBD ate)
       (NP (DT the) (NN cake)
           (PP (IN with)
               (NP (DT a) (NN spoon))))))
```

```
  time flies like an arrow
0     1     2    3  4     5
```

Gold standard:

```
S  0,5 time flies like an arrow
NP 0,2 time flies
NN 0,1 time
NN 1,2 flies
VP 2,5 like an arrow
VB 2,3 like
NP 3,5 an arrow
DT 3,4 an
NN 4,5 arrow
```

Parser result:

```
S  0,5 time flies like an arrow
NP 0,1 time
NN 0,1 time
VP 1,5 flies like an arrow
VB 1,2 flies
PP 2,5 like an arrow
IN 2,3 like
NP 3,5 an arrow
DT 3,4 an
NN 4,5 arrow
```

```
 he ate the cake with a spoon
0  1   2   3    4    5 6     7
```

Gold standard:

```
S   0,7 he ate ...ke with a spoon
NP  0,1 he
PRP 0,1 he
VP  1,7 ate the cake with a spoon
VBD 1,2 ate
NP  2,4 the cake
DT  2,3 the
NN  3,4 cake
PP  4,7 with a spoon
IN  4,5 with
NP  5,7 a spoon
DT  5,6 a
NN  6,7 spoon
```

Parser result:

```
S   0,7 he ate the cake with a spoon
NP  0,1 he
PRP 0,1 he
VP  1,7 ate the cake with a spoon
VBD 1,2 ate
NP  2,7 the cake with a spoon
DT  2,3 the
NN  3,4 cake
PP  4,7 with a spoon
IN  4,5 with
NP  5,7 a spoon
DT  5,6 a
NN  6,7 spoon
```

Precision$= \frac{17}{23} \approx 0.739130434782609$
Recall $= \frac{17}{22} \approx 0.772727272727273.$