Probabilistic Parsing: Issues & Improvement

LING 571 — Deep Processing Techniques for NLP

October 15th, 2018

UNIVERSITY OF WASHINGTON

PROFESSIONAL MASTER'S IN COMPUTATIONAL LINGUISTICS

Notes on HW #3

- Subtrees in cells
  - …cells *are* subtrees, just represented differently!
  - You could use NLTK's `Tree` objects as backpointers efficiently *iff*:
    - each left or right child is defined by reference to another subtree.
    - …this isn't more efficient than:

```
for tree_l in left_subtrees:
    for tree_r in right_subtrees:
        trees.append('({} {} {})'.format(nonterminal.symbol(),
                                          tree_l, tree_r))
```

A new type of vaccine? *(0 parses)*

```
  ------------------------------------------------------------------------------------------------
  | Det -> "A" |                | NP -> 0•Det•1 1•Nom•3   |               | NP -> 0•Det•1 1•Nom•5     |
  |            |                |                         |               | _X_7 -> 0•NP•3 3•PP•5     |
0 ------------------------------------------------------------------------------------------------
               | Adj -> "new"   | NP -> 1•ADJP•2 2•Nom•3  |               | NP -> 1•ADJP•2 2•Nom•5    |
               | ADJP -> "new"  | Nom -> 1•ADJP•2 2•Nom•3 |               | Nom -> 1•Nom•3 3•PP•5     |
               |                |                         |               | _X_7 -> 1•NP•3 3•PP•5     |
               |                |                         |               | NP -> 1•Nom•3 3•PP•5      |
               |                |                         |               | Nom -> 1•ADJP•2 2•Nom•5   |
             1 ----------------------------------------------------------------------------------
                                | Nom -> "type"           |               | Nom -> 2•Nom•3 3•PP•5     |
                                | NP -> "type"            |               | NP -> 2•Nom•3 3•PP•5      |
                                |                         |               | _X_7 -> 2•NP•3 3•PP•5     |
                              2 ------------------------------------------------------------------
                                                          | Prep -> "of"  | PP -> 3•Prep•4 4•NP•5     |
                                                        3 ------------------------------------------
                                                                          | NP -> "vaccine"           |
                                                                          | Nom -> "vaccine"          |
                                                                        4 ----------------------------
                                                                                                      5
```

Will this work in humans? (0 parses)

```
  ------------------------------------------------------------------------------------
  | MD -> "Will" |              |              |              |                       |
0 ------------------------------------------------------------------------------------
                 | NP -> "this" |              |              | S -> 1•NP•2 2•VP•5    |
                 |              |              |              | TOP -> 1•NP•2 2•VP•5  |
               1 --------------------------------------------------------------------
                                | VB -> "work" |              | VP -> 2•VB•3 3•PP•5   |
                              2 -----------------------------------------------------
                                               | Prep -> "in" | PP -> 3•Prep•4 4•NP•5 |
                                             3 --------------------------------------
                                                              | NP -> "humans"        |
                                                              | Nom -> "humans"       |
                                                            4 -----------------------
                                                                                      5
```

*Will this work in **apes**?* *(0 parses)*

```
  ------------------------------------------------------------------------------------
  | MD -> "Will" |                |                |                |   |
0 ------------------------------------------------------------------------------------
                 | NP -> "this" |                |                |   |
               1 ------------------------------------------------------------------
                                | VB -> "work" |                |   |
                              2 --------------------------------------------------
                                               | Prep -> "in" |   |
                                             3 ---------------------
                                                              |   |
                                                            4 ----
                                                                  5
```

They restored immunity in mice with a weak immune system. *(8 parses)*

```
 --------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
 | NP -> "They" |                  | TOP -> 0•NP•1 1•VP•3 |                  | TOP -> 0•NP•1 1•VP•5  |                |             |                |                      | TOP -> 0•NP•1 1•VP•10  |
 |              |                  | S -> 0•NP•1 1•VP•3   |                  | S -> 0•NP•1 1•VP•5    |                |             |                |                      | S -> 0•NP•1 1•VP•10    |
0 ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                | VBD -> "restored" | VP -> 1•VBD•2 2•NP•3 |                  | VP -> 1•VBD•2 2•_X_7•5 |               |             |                |                      | VP -> 1•VBD•2 2•_X_7•10 |
                |                   |                      |                  | VP -> 1•VBD•2 2•NP•5  |                |             |                |                      | VP -> 1•VBD•2 2•NP•10  |
               1 ------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                    | Nom -> "immunity"    |                  | Nom -> 2•Nom•3 3•PP•5 |                |             |                |                      | Nom -> 2•Nom•3 3•PP•10 |
                                    | NP -> "immunity"     |                  | NP -> 2•Nom•3 3•PP•5  |                |             |                |                      | _X_7 -> 2•NP•5 5•PP•10 |
                                    |                      |                  | _X_7 -> 2•NP•3 3•PP•5 |                |             |                |                      | NP -> 2•Nom•3 3•PP•10  |
                                    |                      |                  |                       |                |             |                |                      | _X_7 -> 2•NP•3 3•PP•10 |
                                    |                      |                  |                       |                |             |                |                      | Nom -> 2•Nom•5 5•PP•10 |
                                    |                      |                  |                       |                |             |                |                      | NP -> 2•Nom•5 5•PP•10  |
                                   2 ----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                                           | Prep -> "in"     | PP -> 3•Prep•4 4•NP•5 |                |             |                |                      | PP -> 3•Prep•4 4•NP•10 |
                                                          3 ---------------------------------------------------------------------------------------------------------------------------------------------------------------------------
                                                                              | Nom -> "mice"         |                |             |                |                      | _X_7 -> 4•NP•5 5•PP•10 |
                                                                              | NP -> "mice"          |                |             |                |                      | NP -> 4•Nom•5 5•PP•10  |
                                                                              |                       |                |             |                |                      | Nom -> 4•Nom•5 5•PP•10 |
                                                                             4 ------------------------------------------------------------------------------------------------------------------------------------------------
                                                                                                      | Prep -> "with" |             |                |                      | PP -> 5•Prep•6 6•NP•10 |
                                                                                                     5 ---------------------------------------------------------------------------------------------------------------
                                                                                                                       | Det -> "a"  |                |                      | NP -> 6•Det•7 7•Nom•10 |
                                                                                                                      6 -----------------------------------------------------------------------------------------------
                                                                                                                                     | Adj -> "weak"  | ADJP -> 7•Adj•8 8•ADJP•9 | Nom -> 7•ADJP•8 8•Nom•10 |
                                                                                                                                     | ADJP -> "weak" |                      | NP -> 7•ADJP•8 8•Nom•10  |
                                                                                                                                     |                |                      | Nom -> 7•ADJP•9 9•Nom•10 |
                                                                                                                                     |                |                      | NP -> 7•ADJP•9 9•Nom•10  |
                                                                                                                                    7 -----------------------------------------------------------------
                                                                                                                                                      | Adj -> "immune"      | Nom -> 8•ADJP•9 9•Nom•10 |
                                                                                                                                                      | ADJP -> "immune"     | NP -> 8•ADJP•9 9•Nom•10  |
                                                                                                                                                     8 ---------------------------------------------------
                                                                                                                                                                             | Nom -> "system"        |
                                                                                                                                                                             | NP -> "system"         |
                                                                                                                                                                            9 --------------------------
                                                                                                                                                                                                     10
```

Start Recording!

PCFG Induction

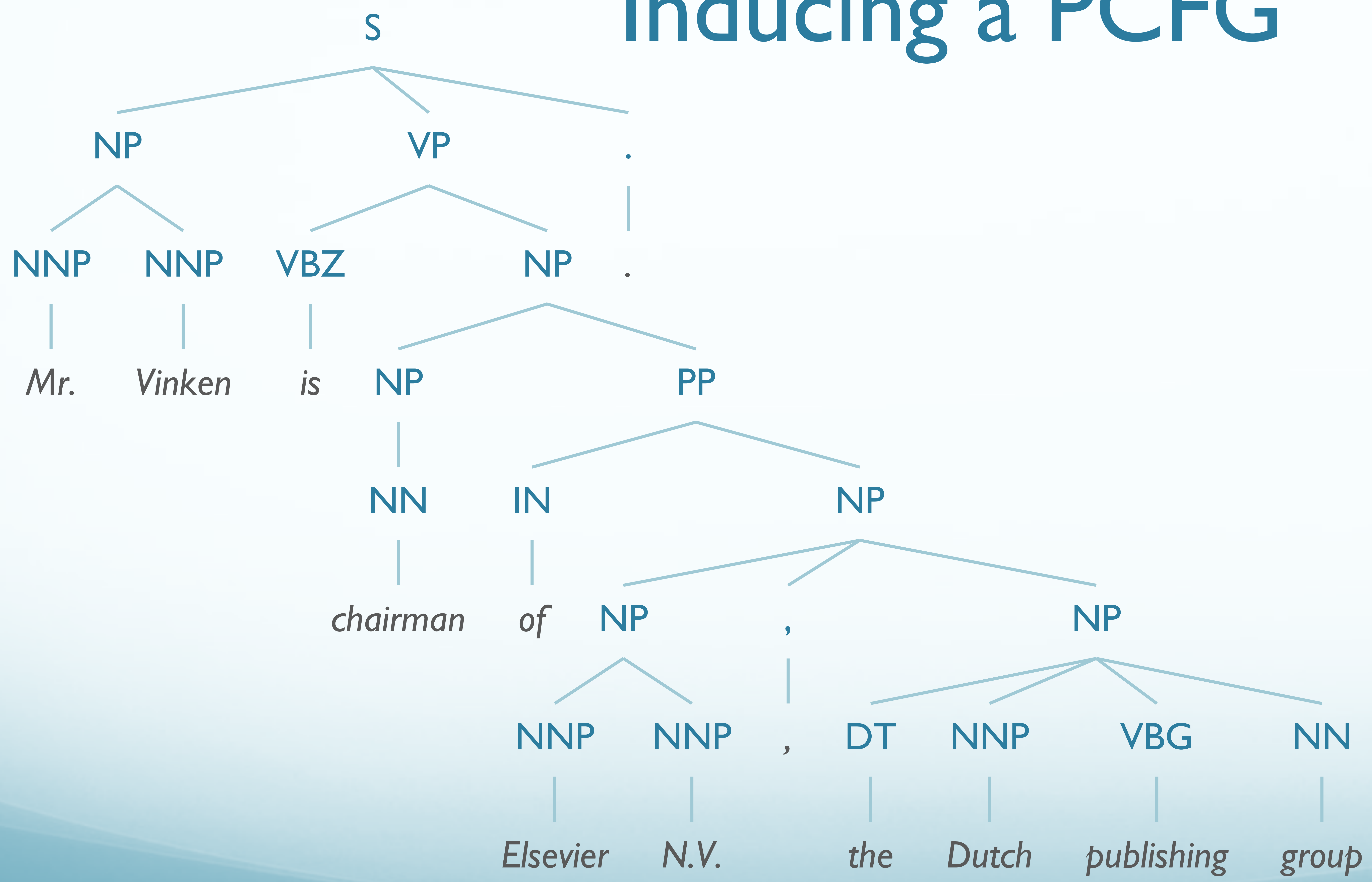
Learning Probabilities

- Simplest way:
  - Use treebank of parsed sentences
  - To compute probability of a rule, count:
    - Number of times a nonterminal is expanded: $\Sigma_\gamma\ Count(\alpha\rightarrow\gamma)$
    - Number of times a nonterminal is expanded by a given rule: $Count(\alpha\rightarrow\beta)$

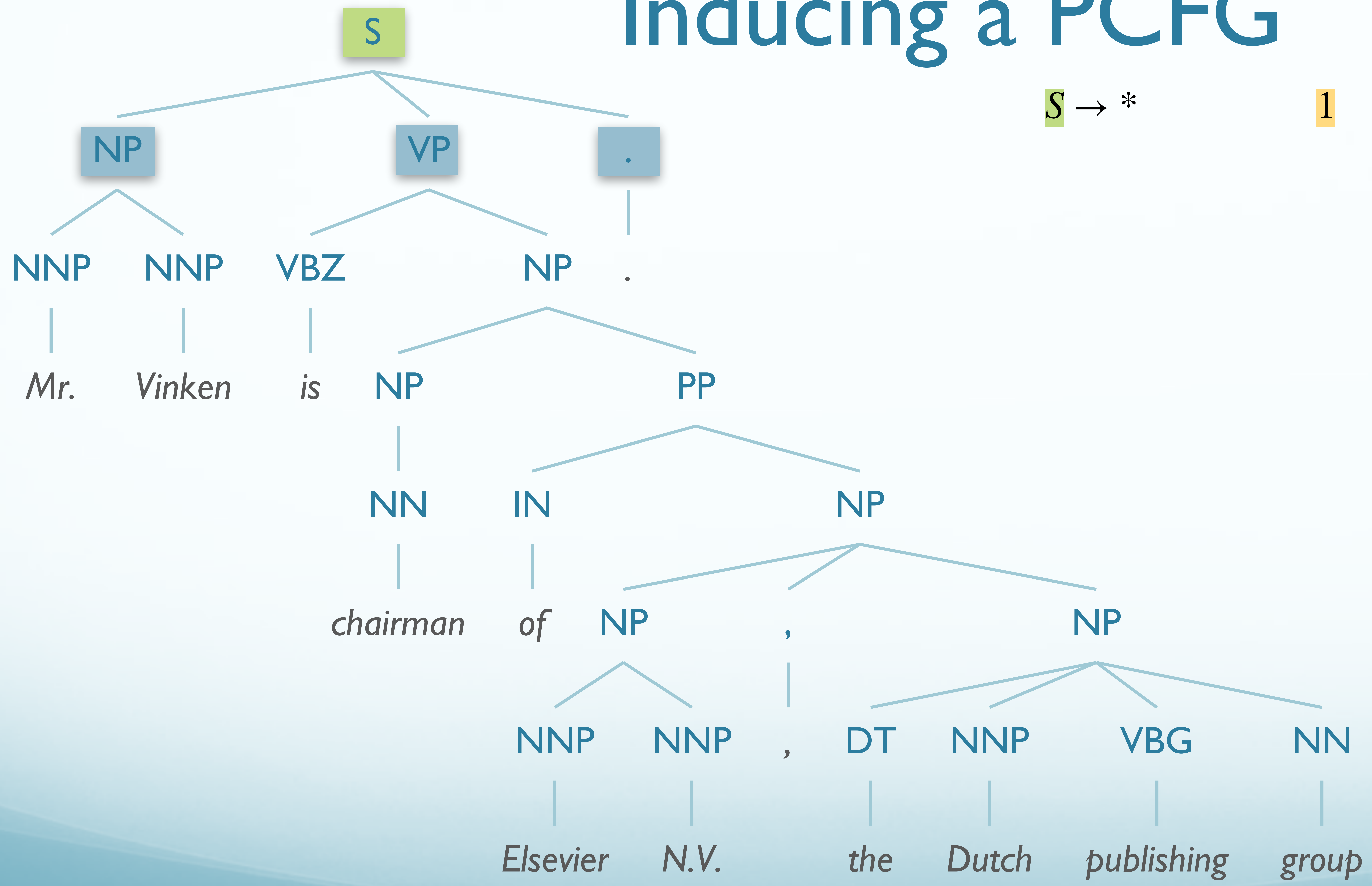
$$P(\alpha \rightarrow \beta \mid \alpha) = \frac{Count(\alpha \rightarrow \beta)}{\sum_\gamma Count(\alpha \rightarrow \gamma)} = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

- Alternative: Learn probabilities by re-estimating
  - (Later)

Inducing a PCFG

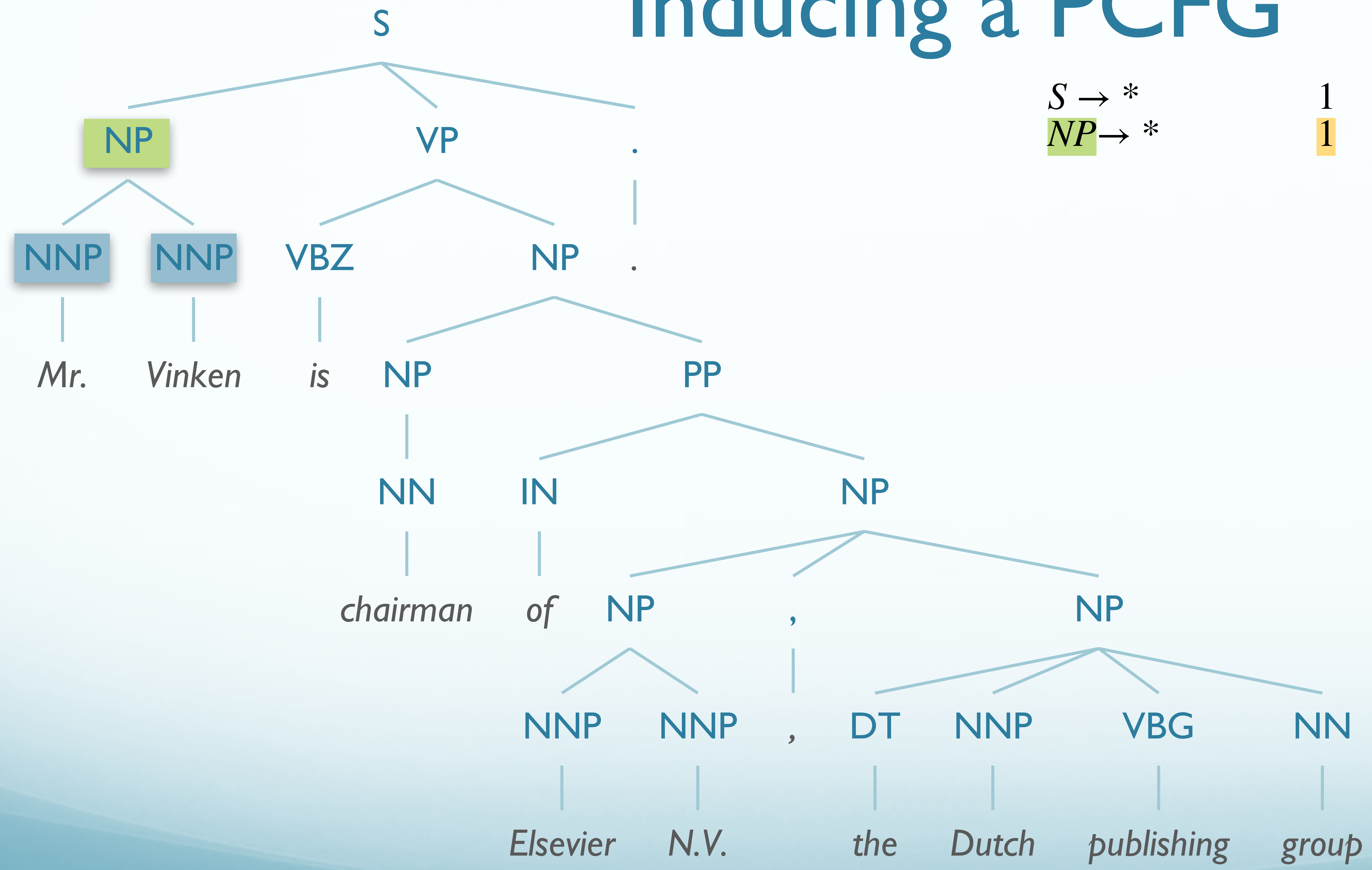


Inducing a PCFG



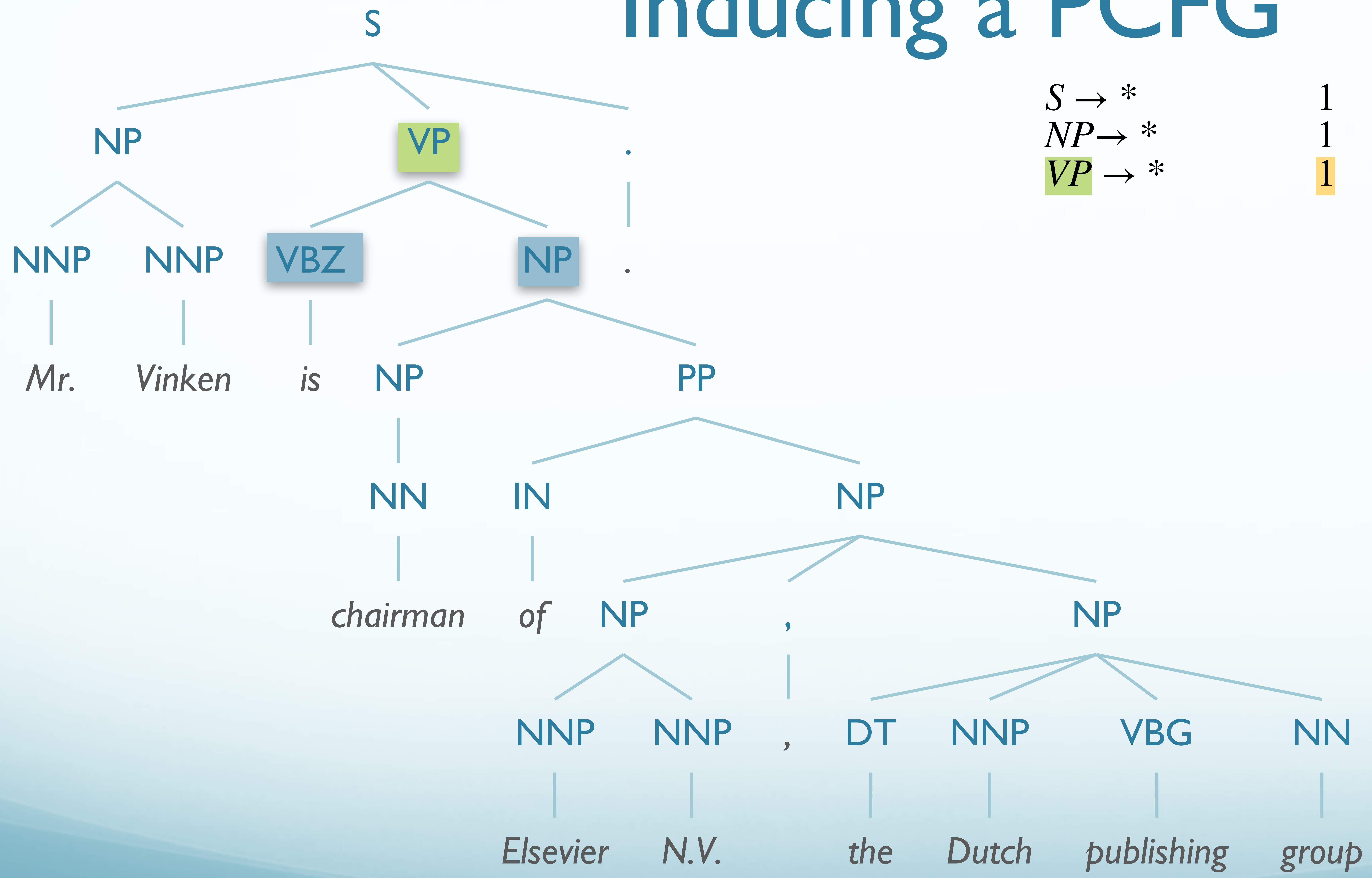
$S \rightarrow *$ 1 $S \rightarrow NP\,VP\ .$ 1

Inducing a PCFG



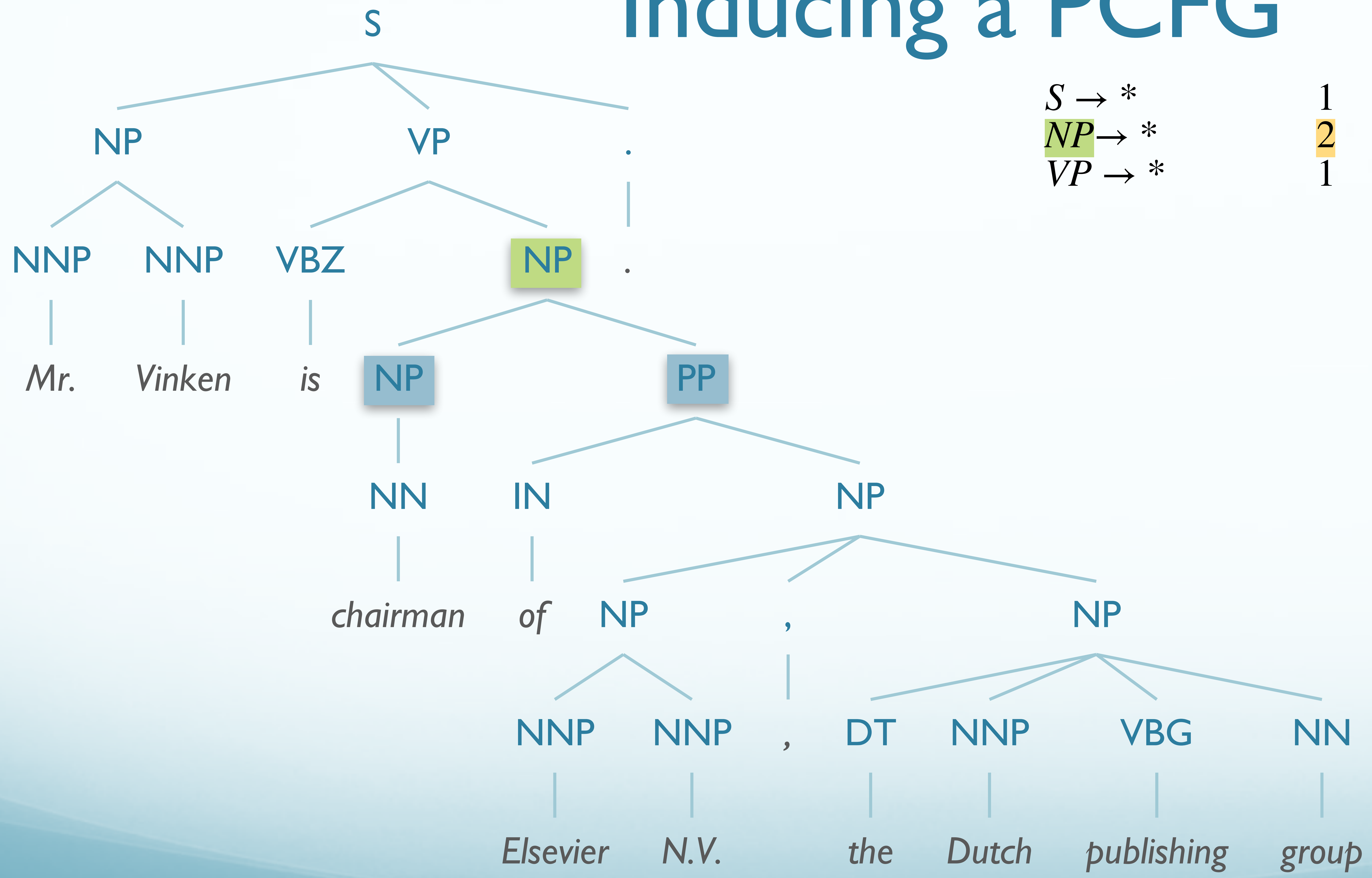
$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	1	$NP \rightarrow NNP\ NNP$	1

Inducing a PCFG



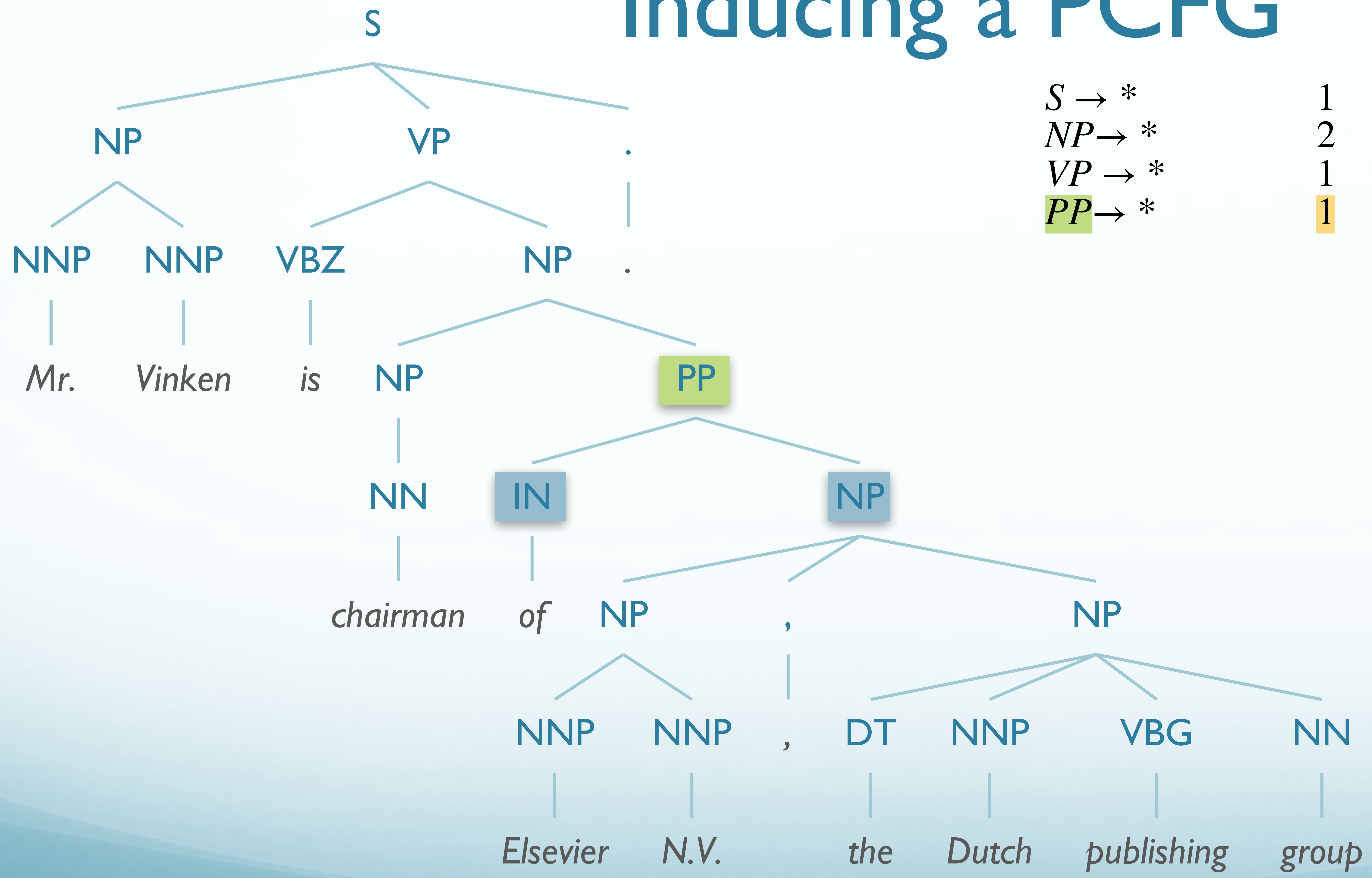
$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	1	$NP \rightarrow NNP\ NNP$	1
$VP \rightarrow *$	1	$VP \rightarrow VBZ\ NP$	1

Inducing a PCFG



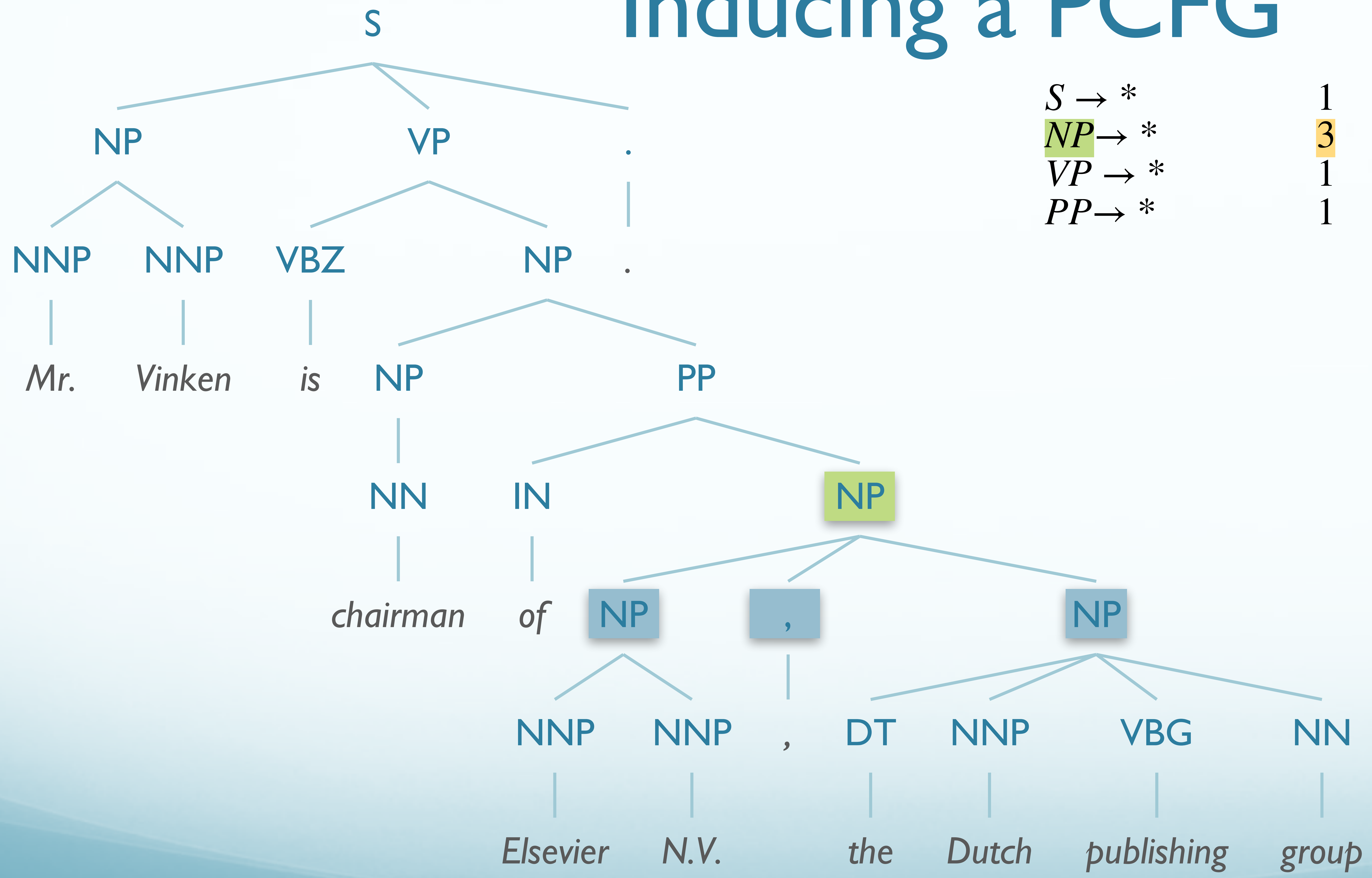
$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	2	$NP \rightarrow NNP\ NNP$	1
$VP \rightarrow *$	1	$VP \rightarrow VBZ\ NP$	1
		$NP \rightarrow NP\ PP$	1

Inducing a PCFG



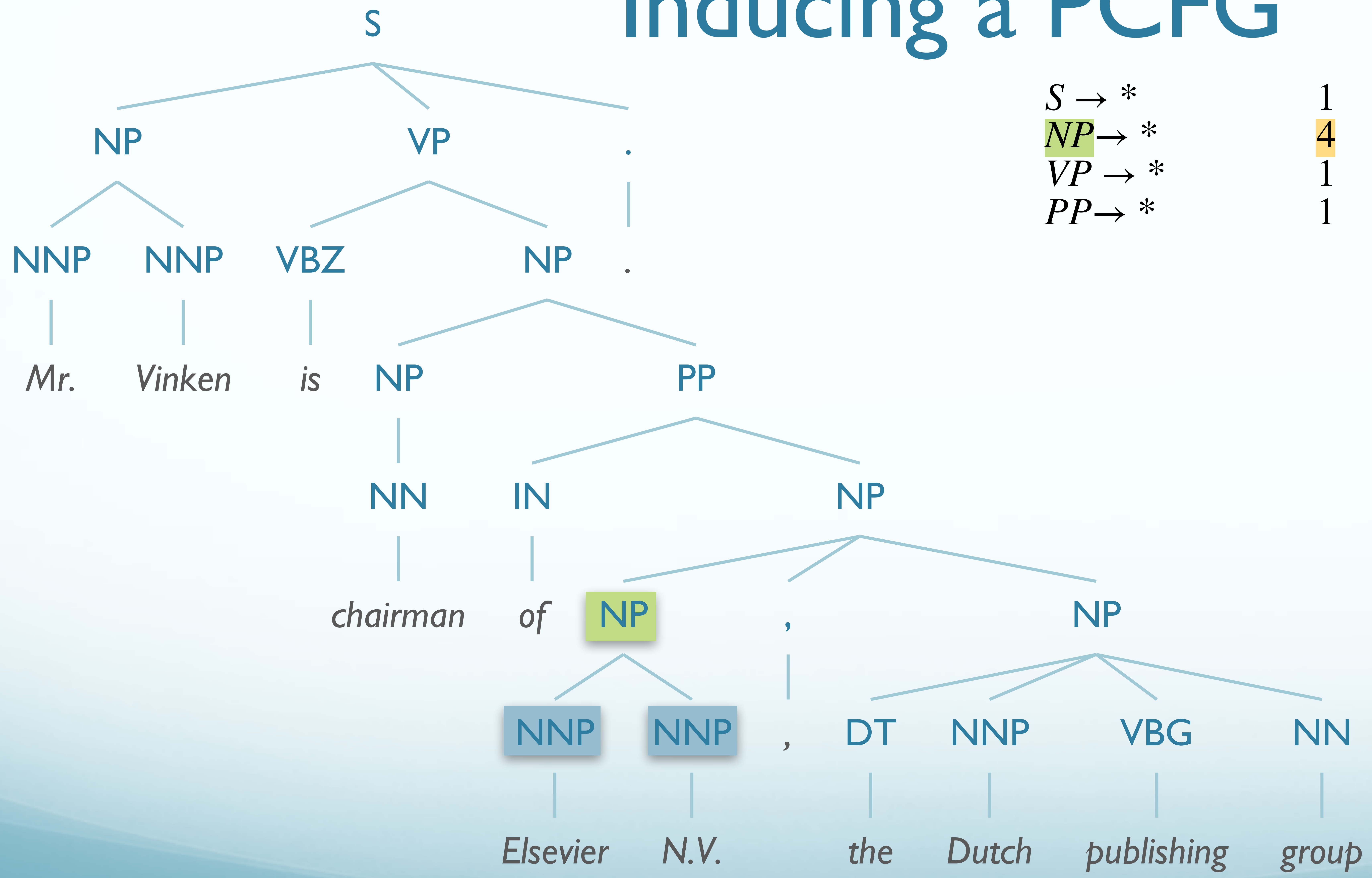
$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	2	$NP \rightarrow NNP\ NNP$	1
$VP \rightarrow *$	1	$VP \rightarrow VBZ\ NP$	1
$PP \rightarrow *$	1	$NP \rightarrow NP\ PP$	1
		$PP \rightarrow IN\ NP$	1

Inducing a PCFG



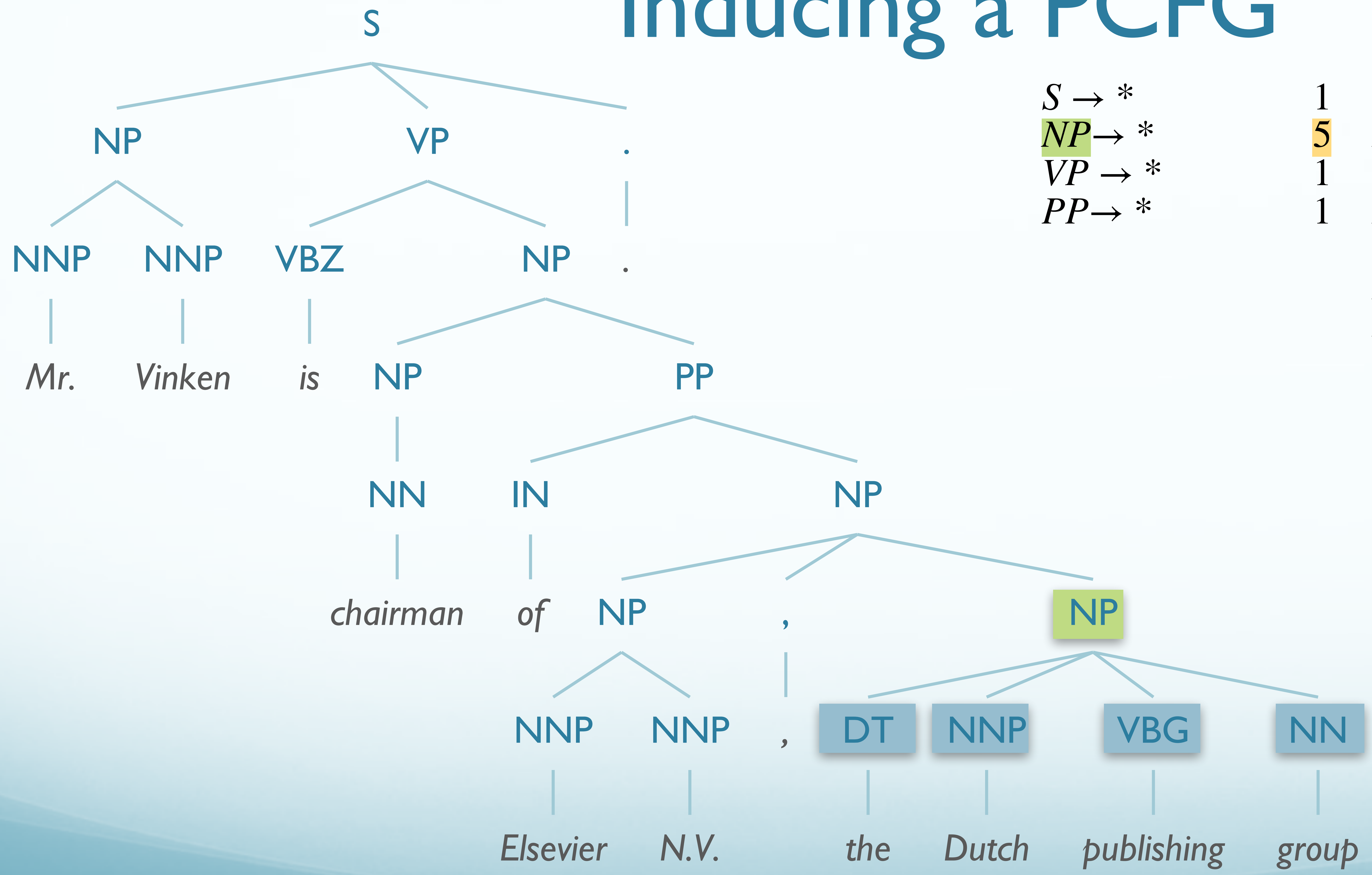
$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	3	$NP \rightarrow NNP\ NNP$	1
$VP \rightarrow *$	1	$VP \rightarrow VBZ\ NP$	1
$PP \rightarrow *$	1	$NP \rightarrow NP\ PP$	1
		$PP \rightarrow IN\ NP$	1
		$NP \rightarrow NP\ ,\ NP$	1

Inducing a PCFG



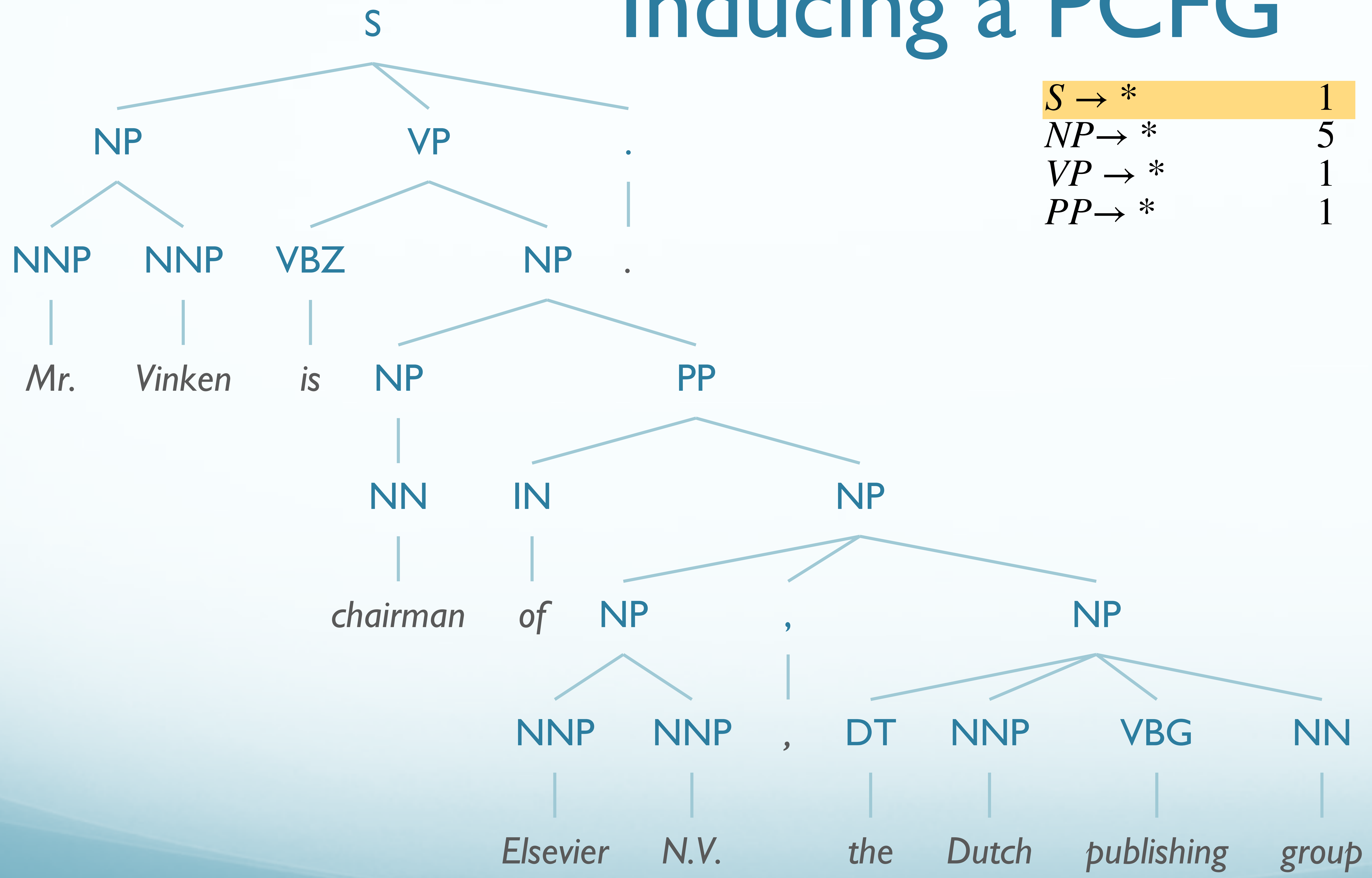
$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	4	$NP \rightarrow NNP\ NNP$	2
$VP \rightarrow *$	1	$VP \rightarrow VBZ\ NP$	1
$PP \rightarrow *$	1	$NP \rightarrow NP\ PP$	1
		$PP \rightarrow IN\ NP$	1
		$NP \rightarrow NP\ ,\ NP$	1

Inducing a PCFG



$S \rightarrow *$	1	$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow *$	5	$NP \rightarrow NNP\ NNP$	2
$VP \rightarrow *$	1	$VP \rightarrow VBZ\ NP$	1
$PP \rightarrow *$	1	$NP \rightarrow NP\ PP$	1
		$PP \rightarrow IN\ NP$	1
		$NP \rightarrow NP\ ,\ NP$	1
		$NP \rightarrow DT\ NNP\ VBG\ NN$	1

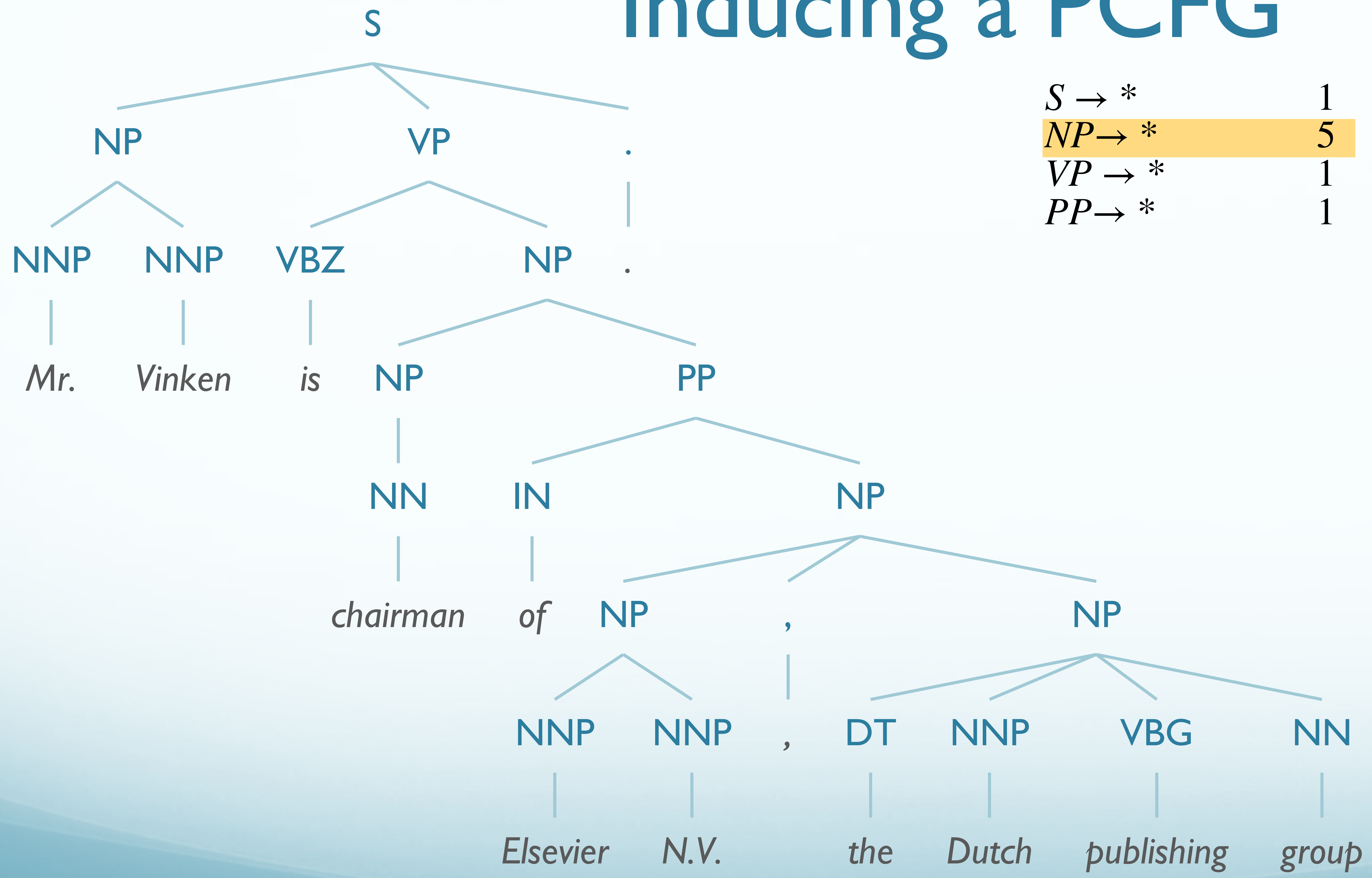
Inducing a PCFG



$S \rightarrow *$	1
$NP \rightarrow *$	5
$VP \rightarrow *$	1
$PP \rightarrow *$	1

$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow NNP\ NNP$	2
$VP \rightarrow VBZ\ NP$	1
$NP \rightarrow NP\ PP$	1
$PP \rightarrow IN\ NP$	1
$NP \rightarrow NP\ ,\ NP$	1
$NP \rightarrow DT\ NNP\ VBG\ NN$	1

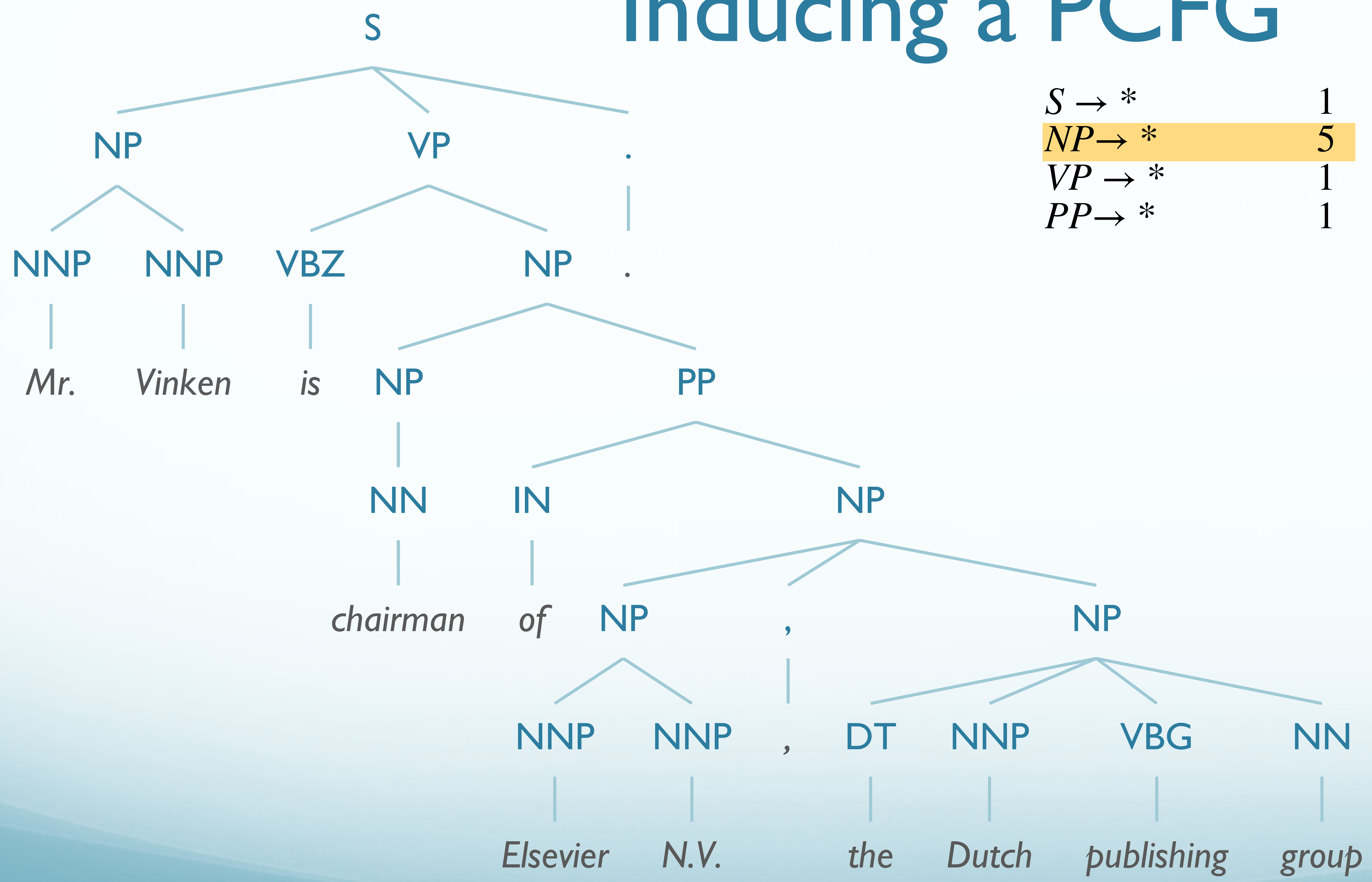
Inducing a PCFG



$S \rightarrow *$	1
$NP \rightarrow *$	5
$VP \rightarrow *$	1
$PP \rightarrow *$	1

$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow NNP\ NNP$	2/5
$VP \rightarrow VBZ\ NP$	1
$NP \rightarrow NP\ PP$	1/5
$PP \rightarrow IN\ NP$	1
$NP \rightarrow NP\ ,\ NP$	1/5
$NP \rightarrow DT\ NNP\ VBG\ NN$	1/5

Inducing a PCFG



$S \rightarrow *$	1
$NP \rightarrow *$	5
$VP \rightarrow *$	1
$PP \rightarrow *$	1

$S \rightarrow NP\ VP\ .$	1
$NP \rightarrow NNP\ NNP$	0.4
$VP \rightarrow VBZ\ NP$	1
$NP \rightarrow NP\ PP$	0.2
$PP \rightarrow IN\ NP$	1
$NP \rightarrow NP\ ,\ NP$	0.2
$NP \rightarrow DT\ NNP\ VBG\ NN$	0.2

Problems with PCFGs

Problems with PCFGs

- Independence Assumption
  - Assume that rule probabilities are independent
- Lack of Lexical Conditioning
  - Lexical items should influence the choice of analysis

Issues with PCFGs: Independence Assumption

- *Context Free ⇒ Independence Assumption*
  - Rule expansion is context-independent
  - Allows us to multiply probabilities
- If we have two rules:
  - $NP \rightarrow DT\ NN \quad [0.28]$
  - $NP \rightarrow PRP \quad [0.25]$

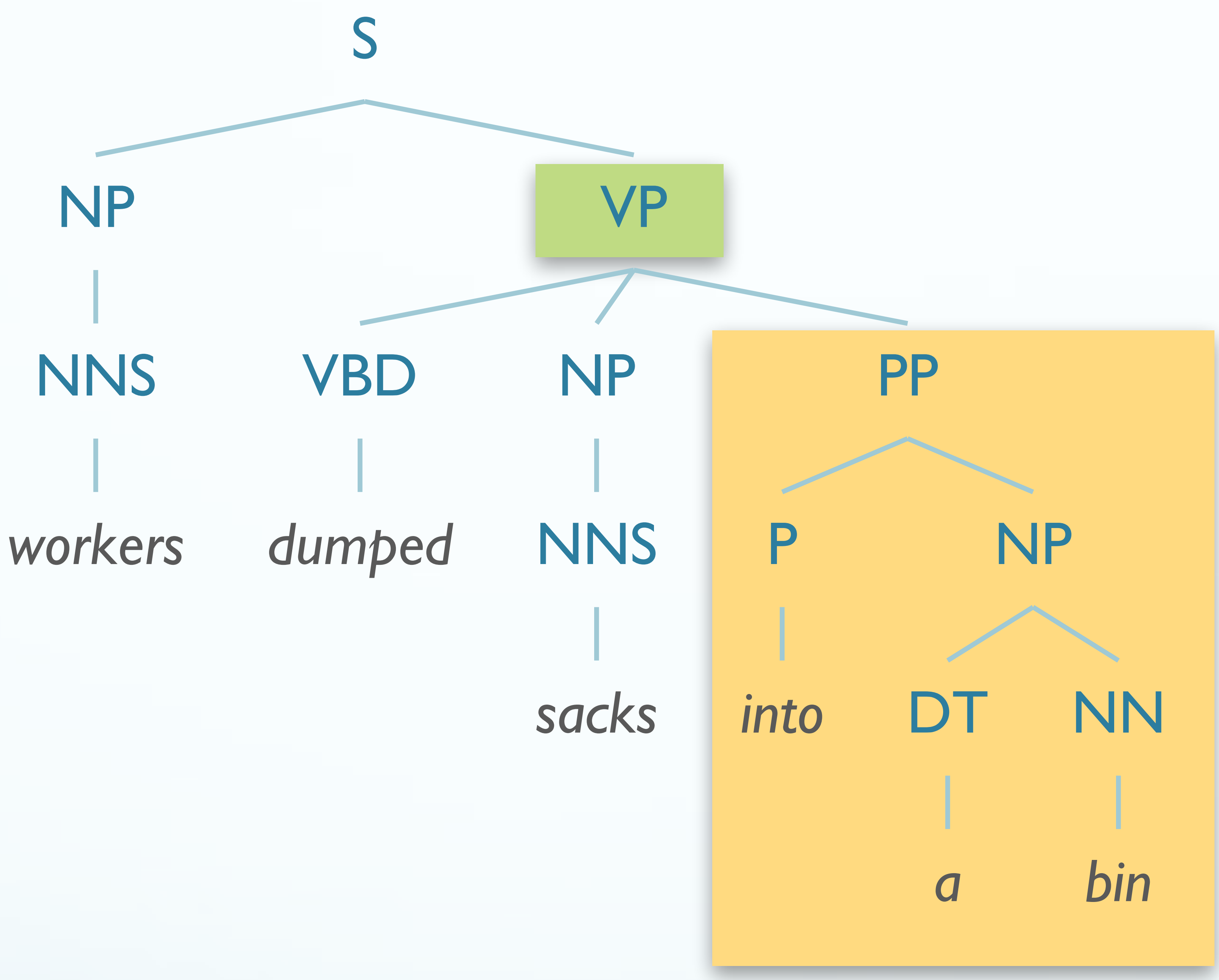
Semantic Role of **NPs** in Switchboard Corpus

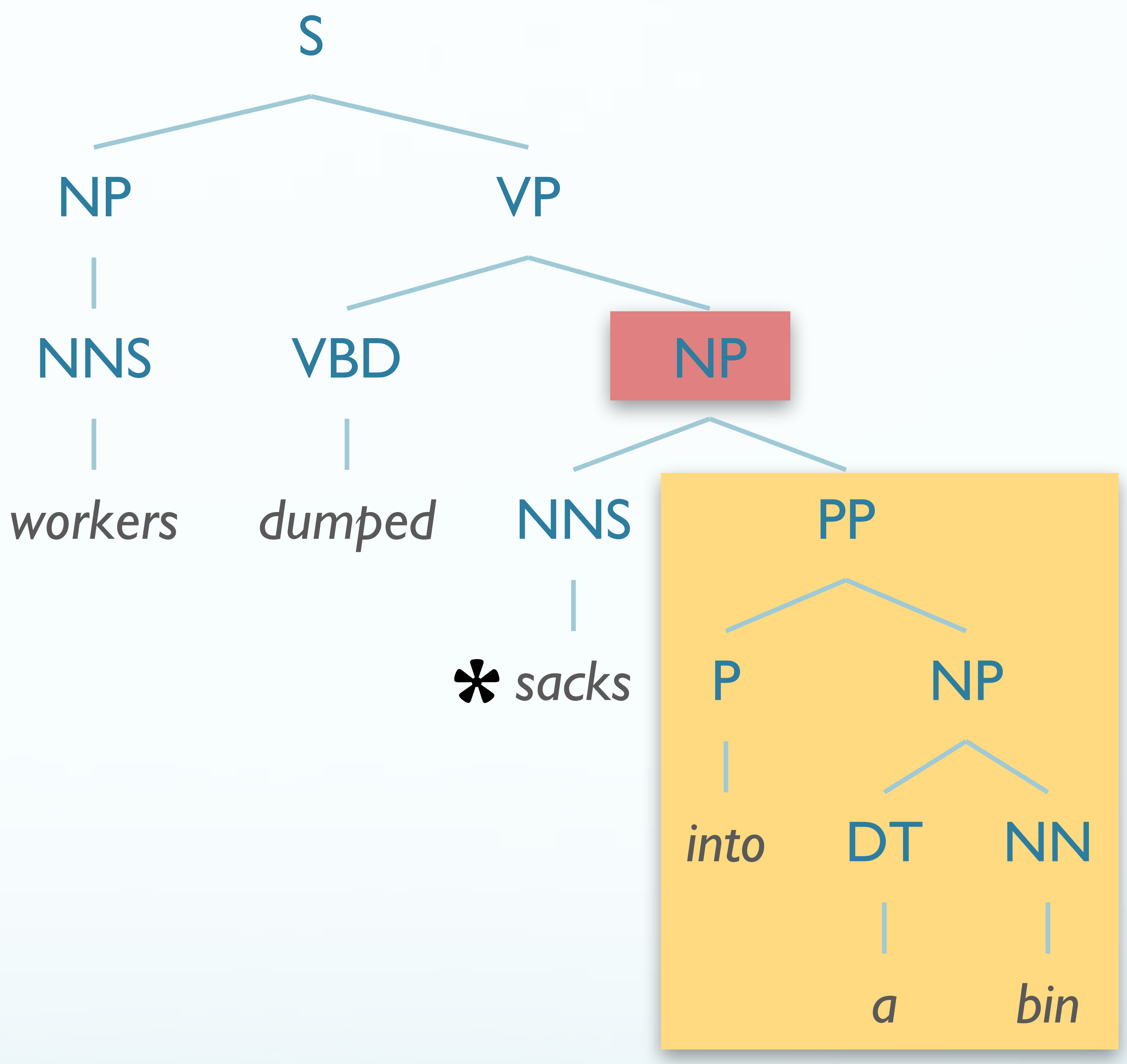
	Pronomial	Non-Pronomial
Subject	91%	9%
Object	34%	66%

- What does this new data tell us?
  - $NP \rightarrow DT\ NN \quad [0.09 \textbf{ if } NP_{\Theta=subject} \textbf{ else } 0.66]$
  - $NP \rightarrow PRP \quad [0.91 \textbf{ if } NP_{\Theta=subject} \textbf{ else } 0.34]$

…Can try **parent annotation**

Issues with PCFGs: Lexical Conditioning

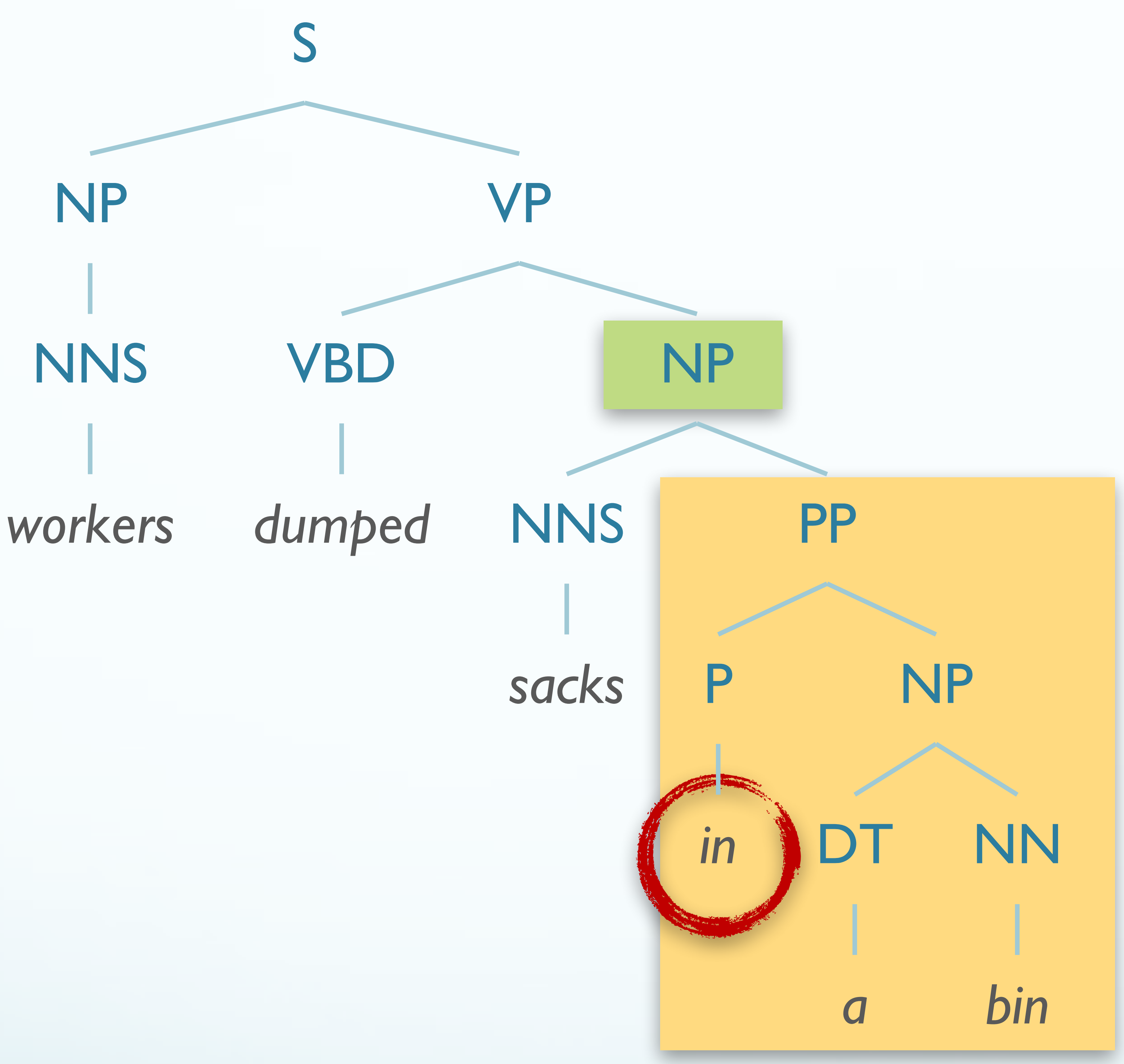


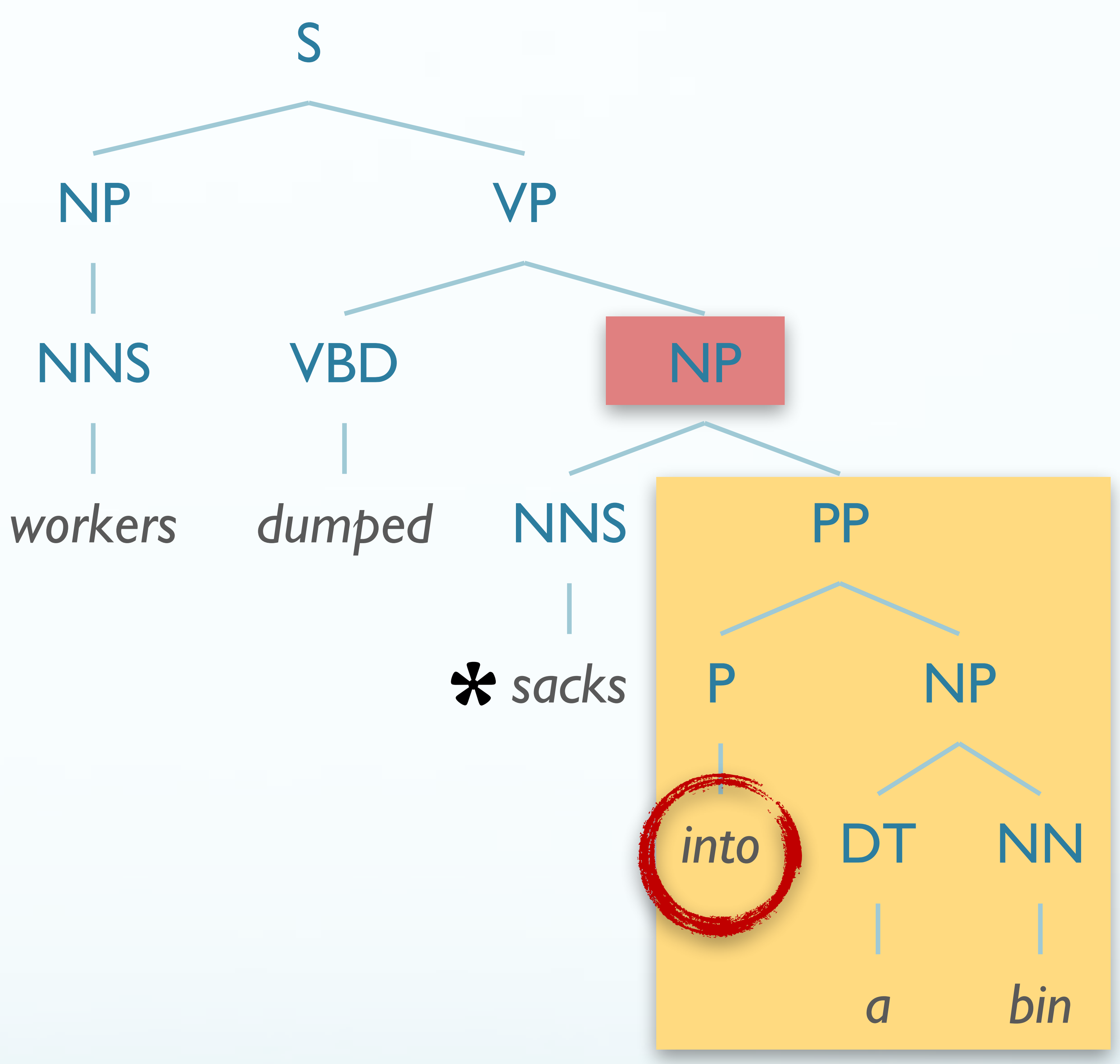


(“into a bin” = location of sacks after dumping)
OK!

(“into a bin” = *the sacks which were located ***in PP***)
not OK

Issues with PCFGs: Lexical Conditioning





("***in*** a bin" = location of sacks ***before*** dumping)
OK!

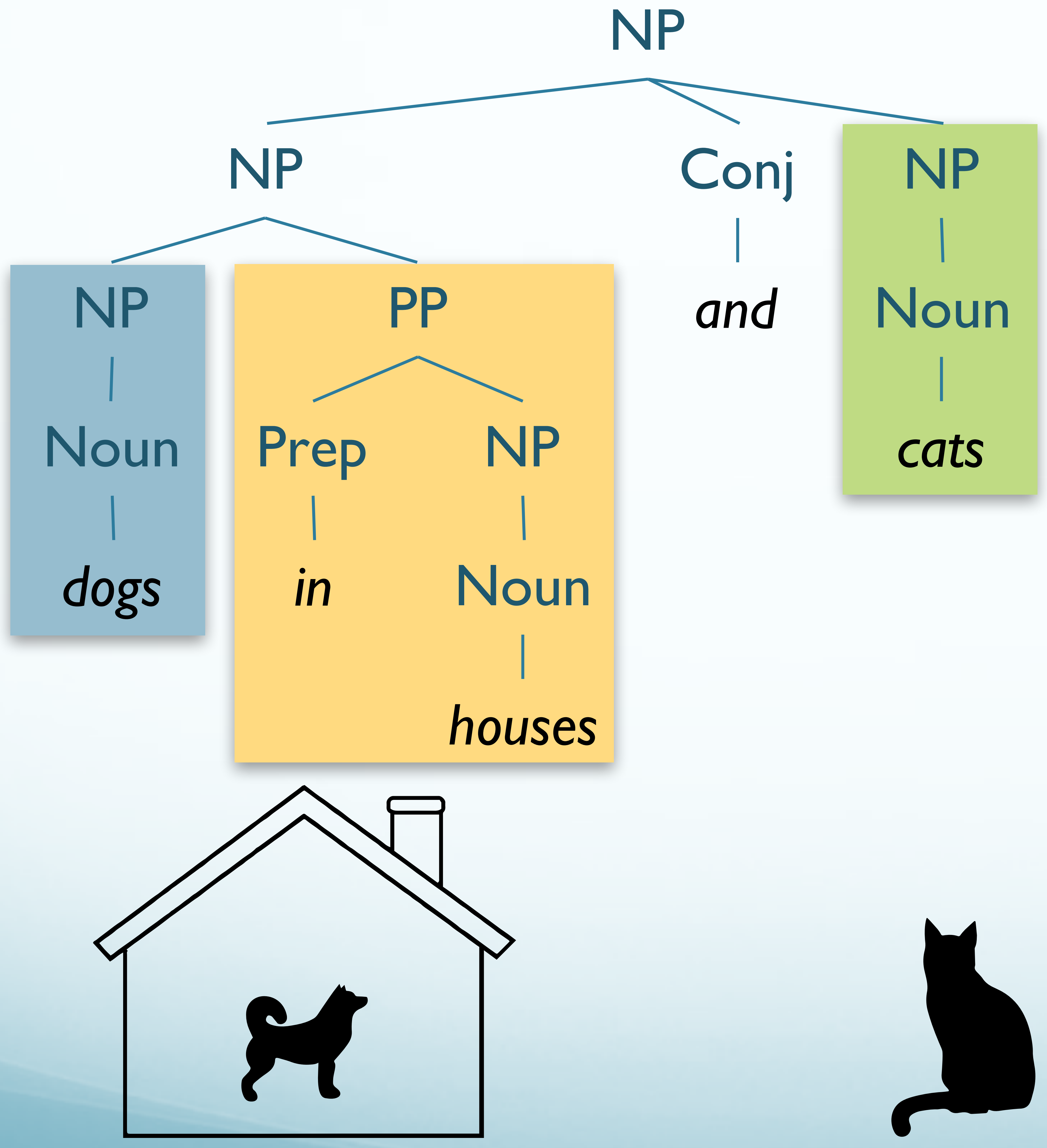
("***into*** a bin" = *the sacks which were located ***in PP***)
not OK

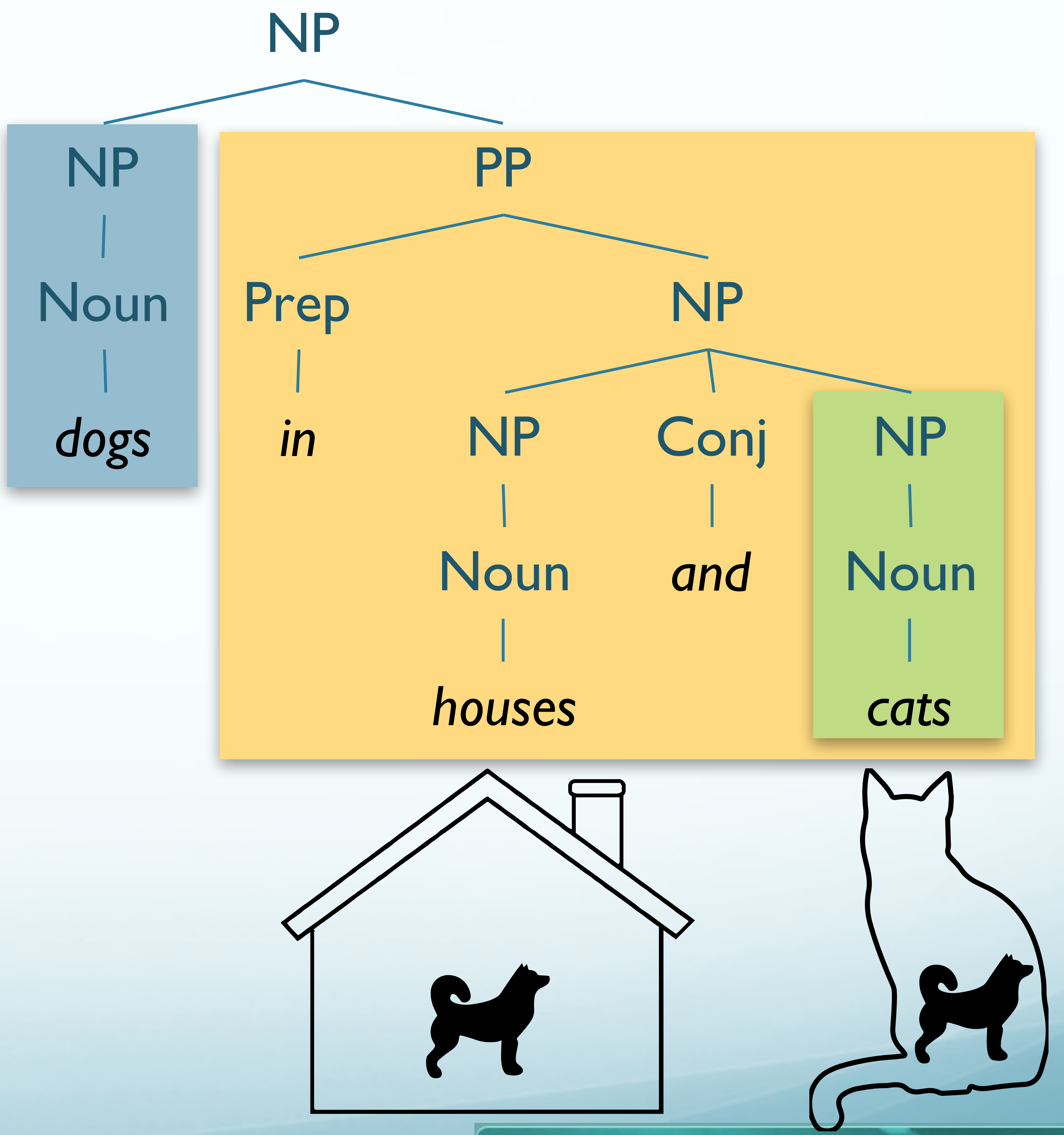
Issues with PCFGs: Lexical Conditioning

- *workers dumped sacks into a bin*
  - ***into*** should **prefer** modifying ***dumped***
  - ***into*** should **disprefer** modifying ***sacks***

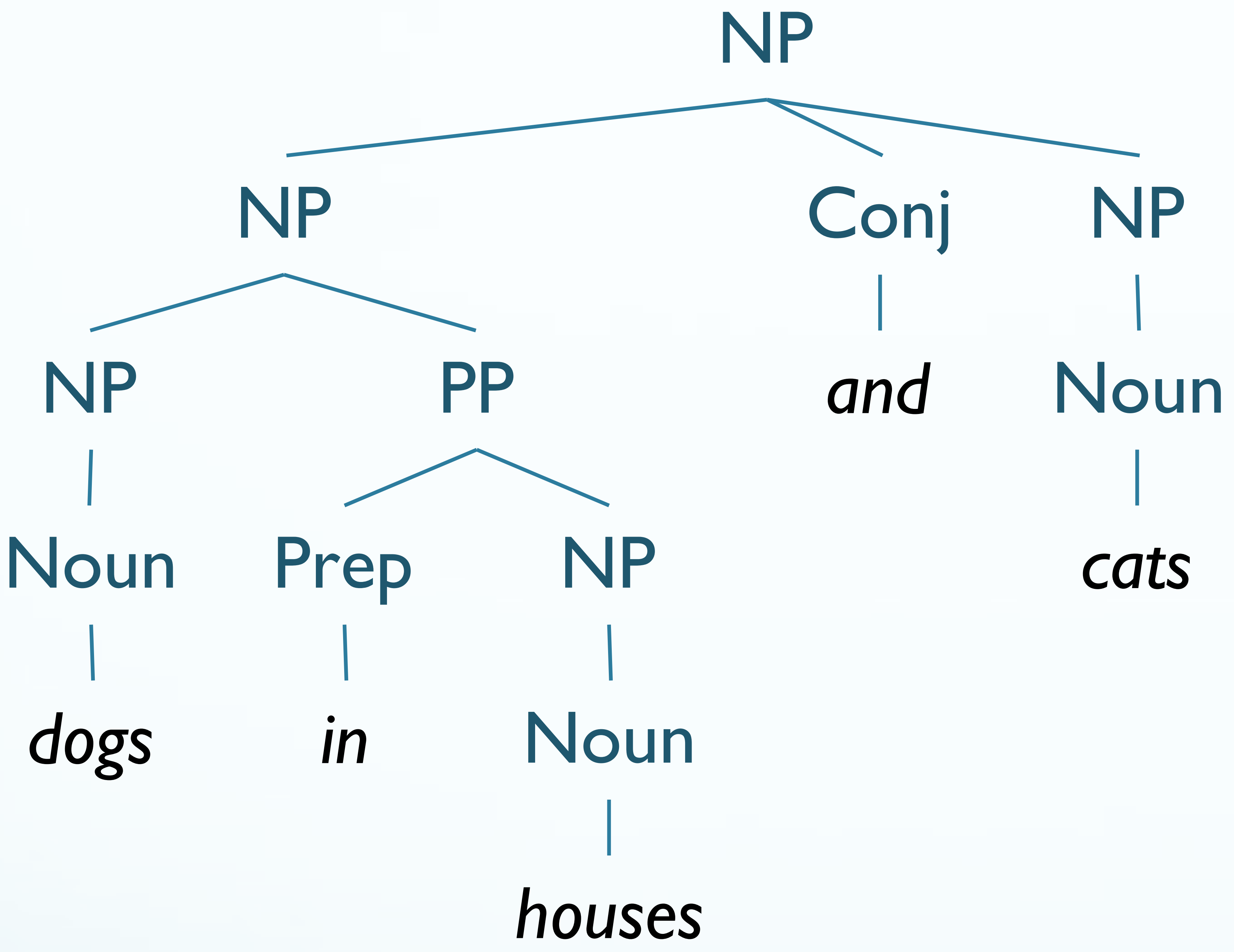
- *fishermen caught tons of herring*
  - ***of*** should **prefer** modifying ***tons***
  - ***of*** should **disprefer** modifying ***caught***

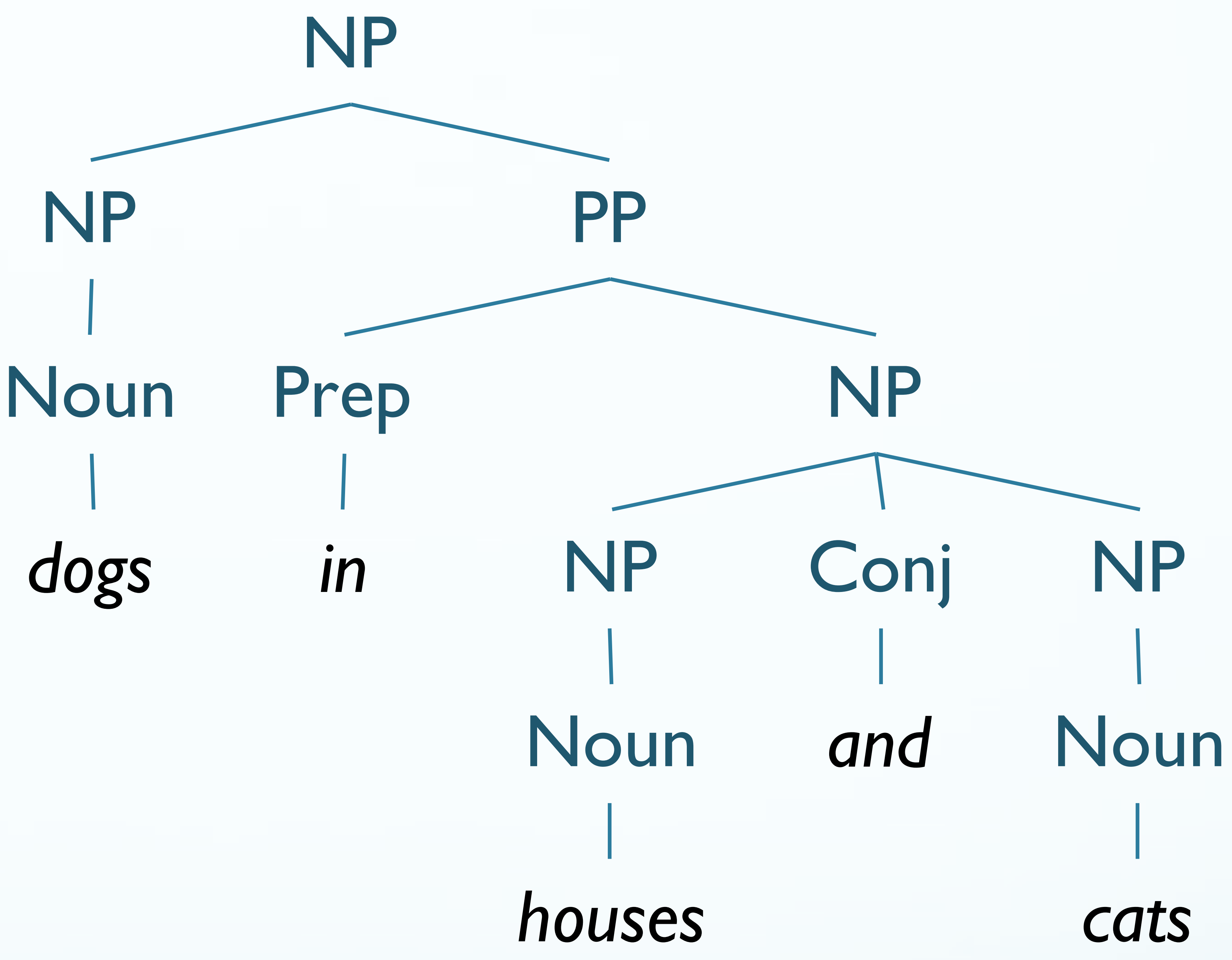
Issues with PCFGs: Coordination Ambiguity





Issues with PCFGs: Coordination Ambiguity



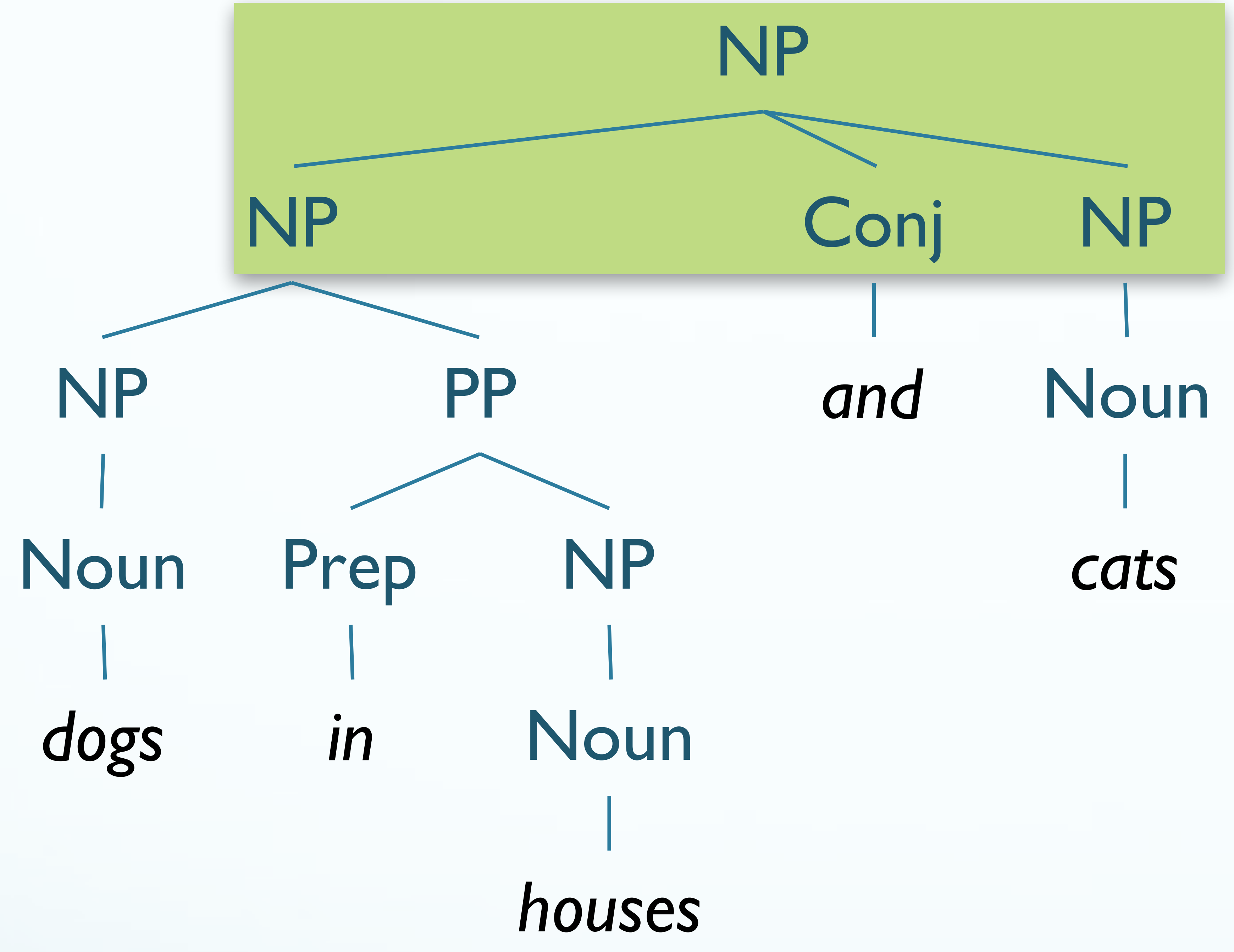


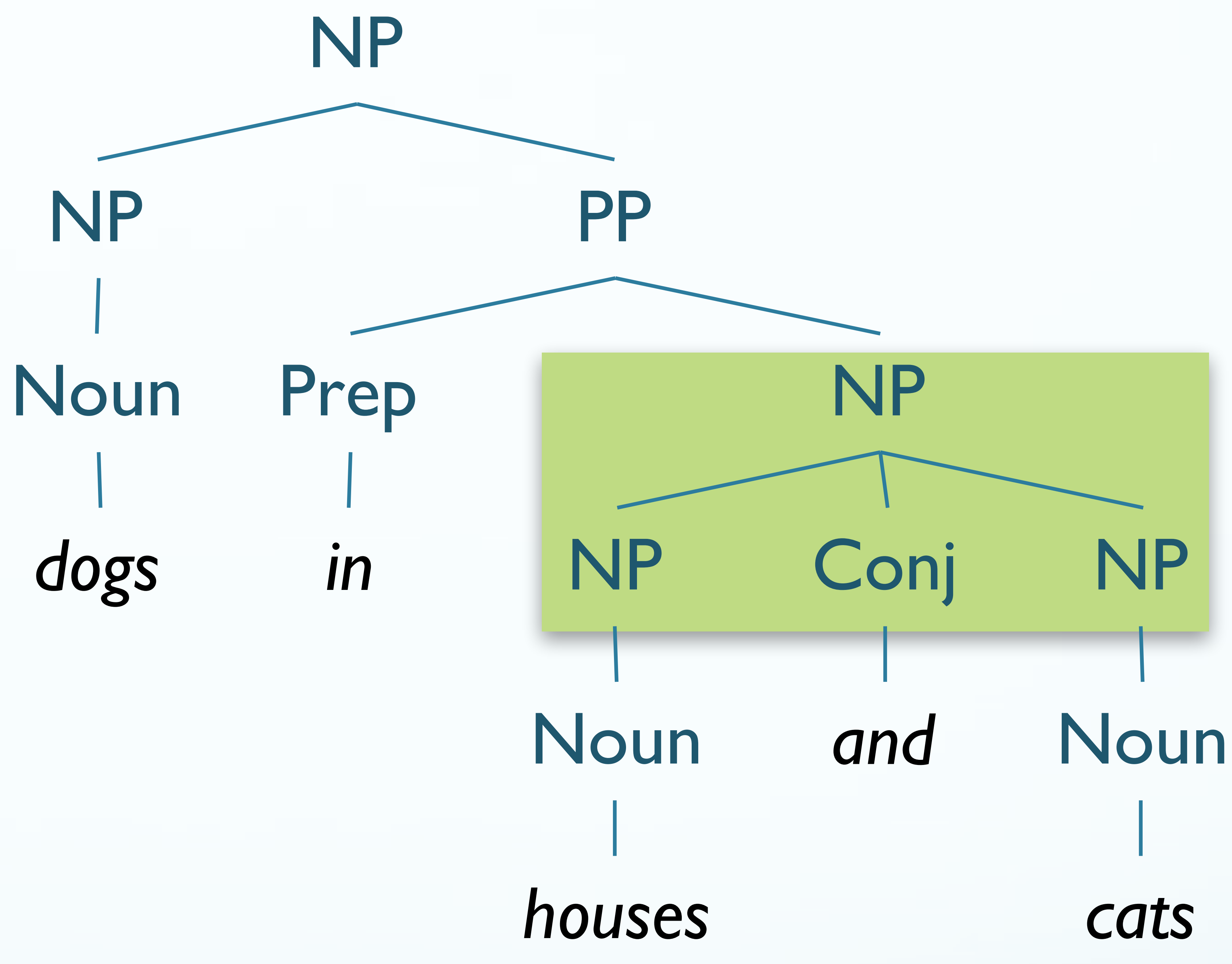
NP → NP Conj NP
NP → NP PP
Noun → "dogs"
PP → Prep NP
Prep → "in"
NP → Noun
Noun → "houses"
Conj → "and"
NP → Noun
Noun → "cats"

Same Rules!

NP → NP PP
Noun → "dogs"
PP → Prep NP
Prep → "in"
NP → NP Conj NP
NP → Noun
Noun → "houses"
Conj → "and"
NP → Noun
Noun → "cats"

Issues with PCFGs: Coordination Ambiguity



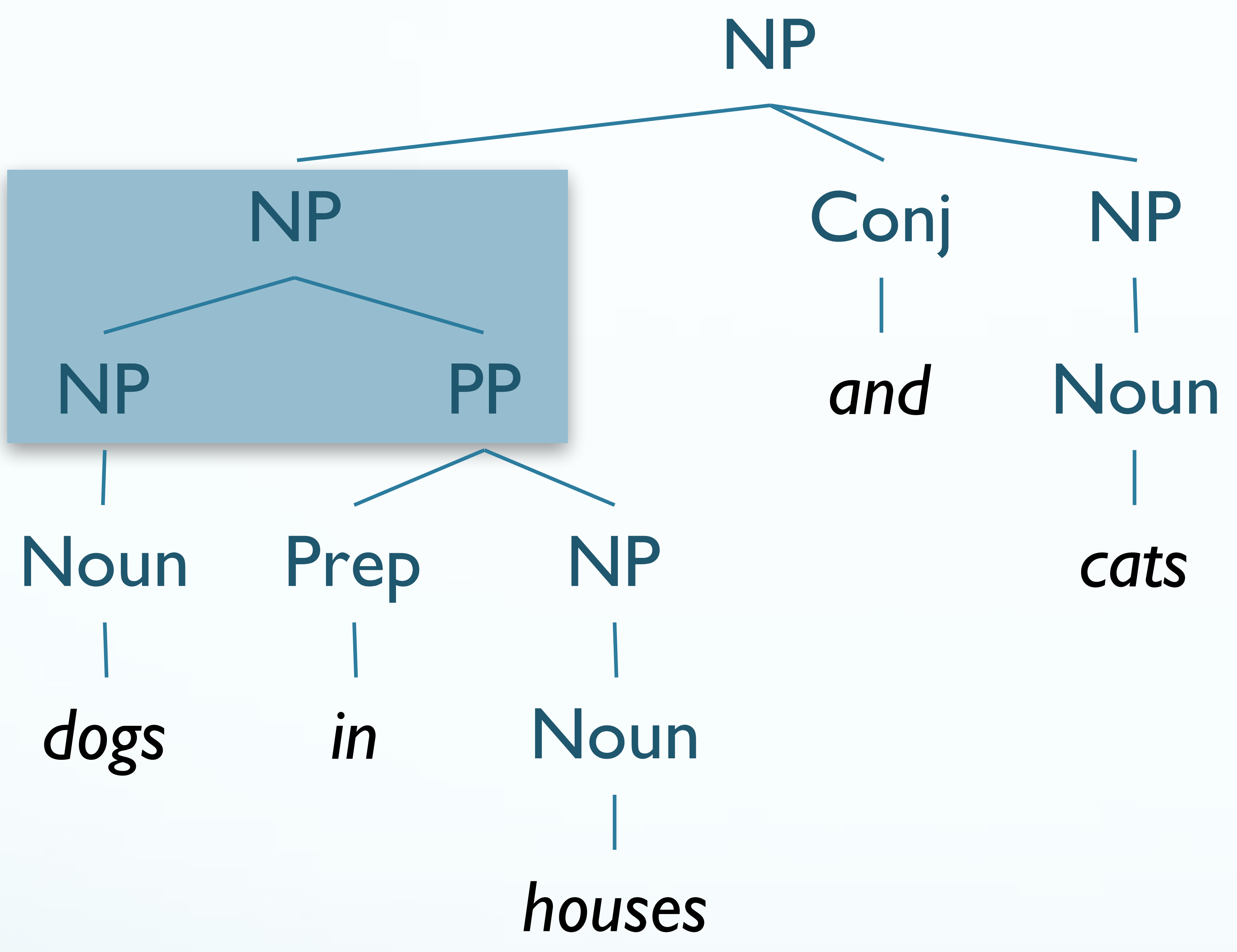


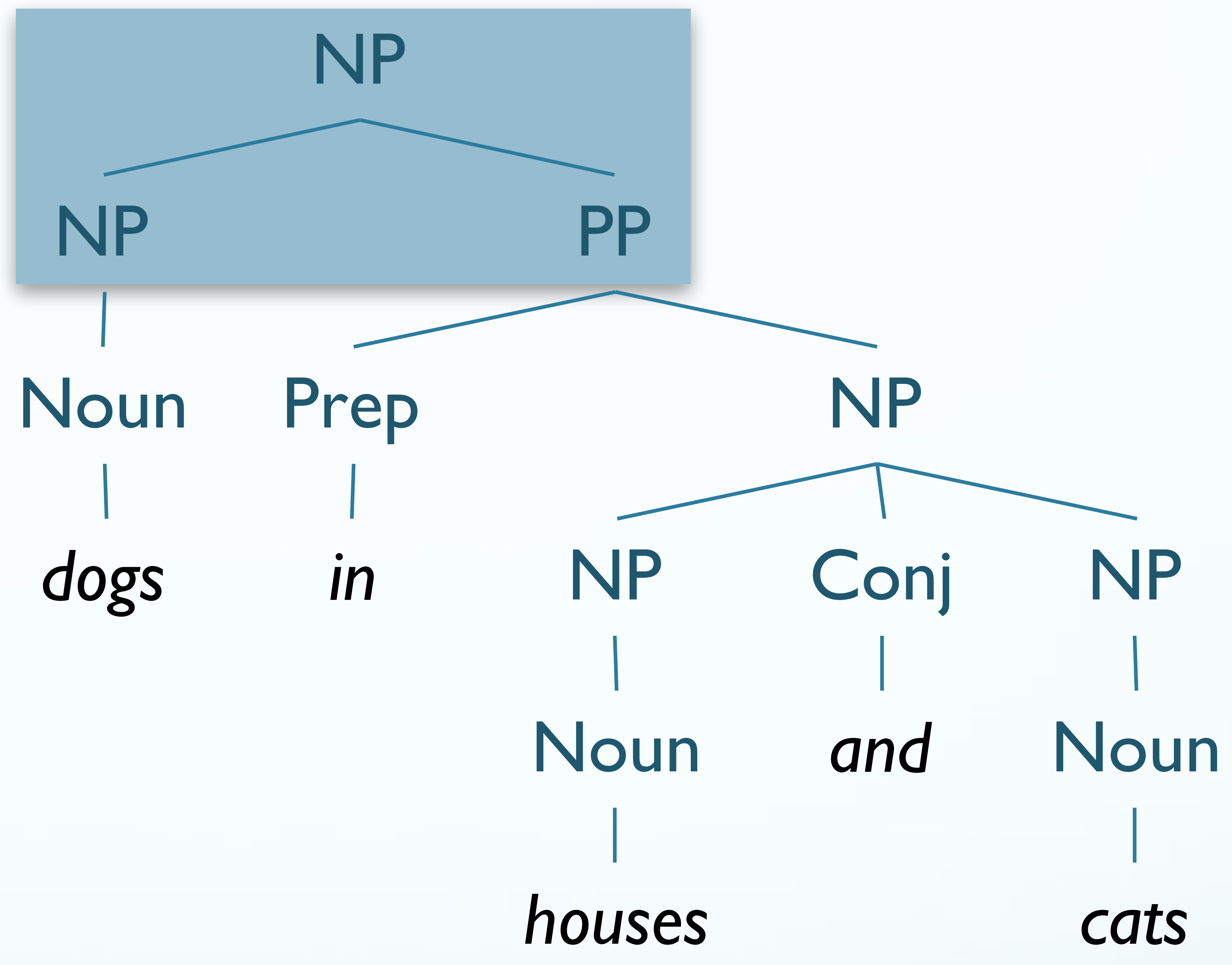
NP → NP Conj NP
NP → NP PP
Noun → "dogs"
PP → Prep NP
Prep → "in"
NP → Noun
Noun → "houses"
Conj → "and"
NP → Noun
Noun → "cats"

Same Rules!

NP → NP PP
Noun → "dogs"
PP → Prep NP
Prep → "in"
NP → NP Conj NP
NP → Noun
Noun → "houses"
Conj → "and"
NP → Noun
Noun → "cats"

Issues with PCFGs: Coordination Ambiguity





NP → NP Conj NP
NP → NP PP
Noun → "dogs"
PP → Prep NP
Prep → "in"
NP → Noun
Noun → "houses"
Conj → "and"
NP → Noun
Noun → "cats"

Same Rules!

NP → NP PP
Noun → "dogs"
PP → Prep NP
Prep → "in"
NP → NP Conj NP
NP → Noun
Noun → "houses"
Conj → "and"
NP → Noun
Noun → "cats"

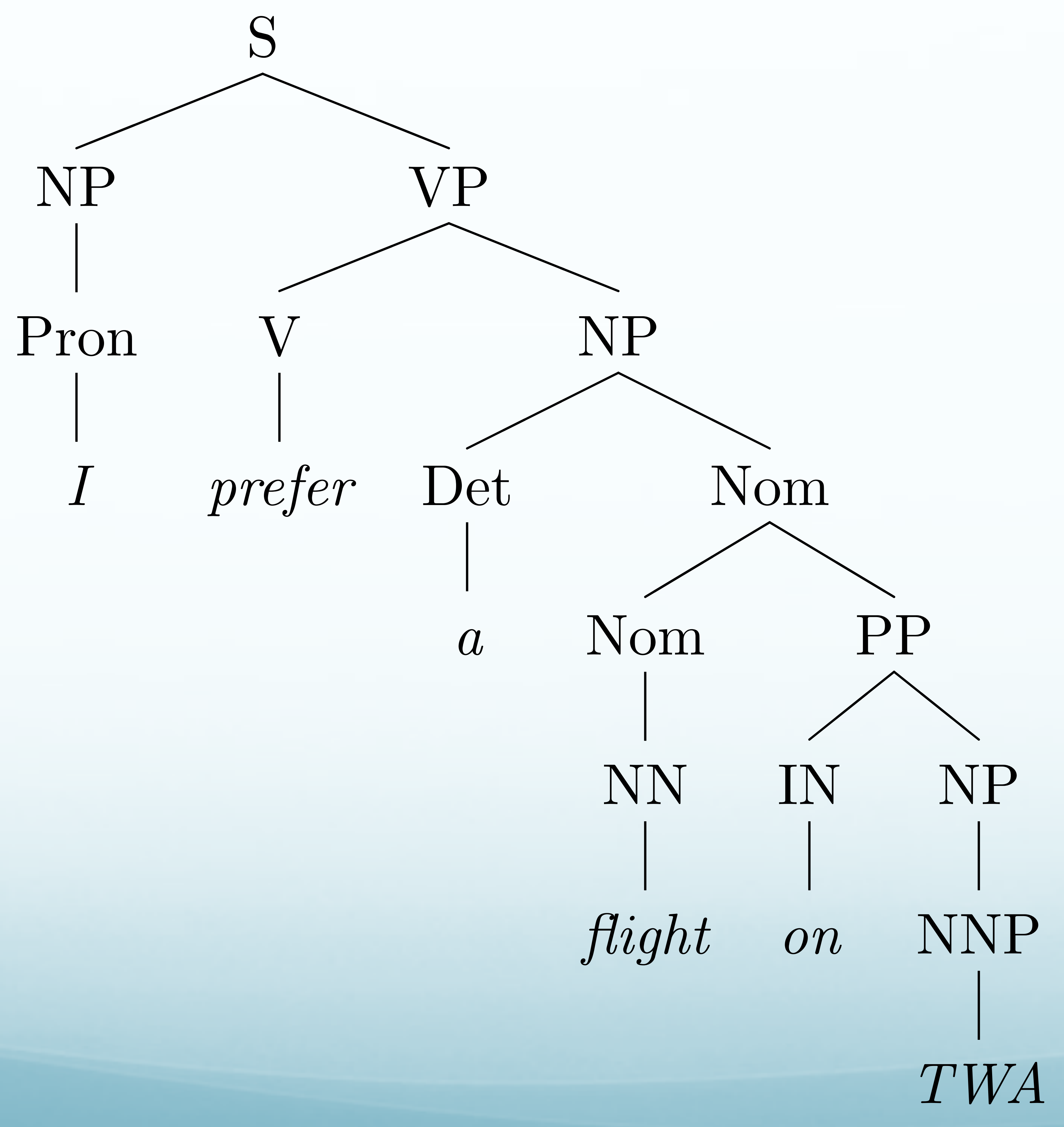
Improving PCFGs

Improving PCFGs

- **Parent Annotation**
- Lexicalization
- Markovization
- Reranking

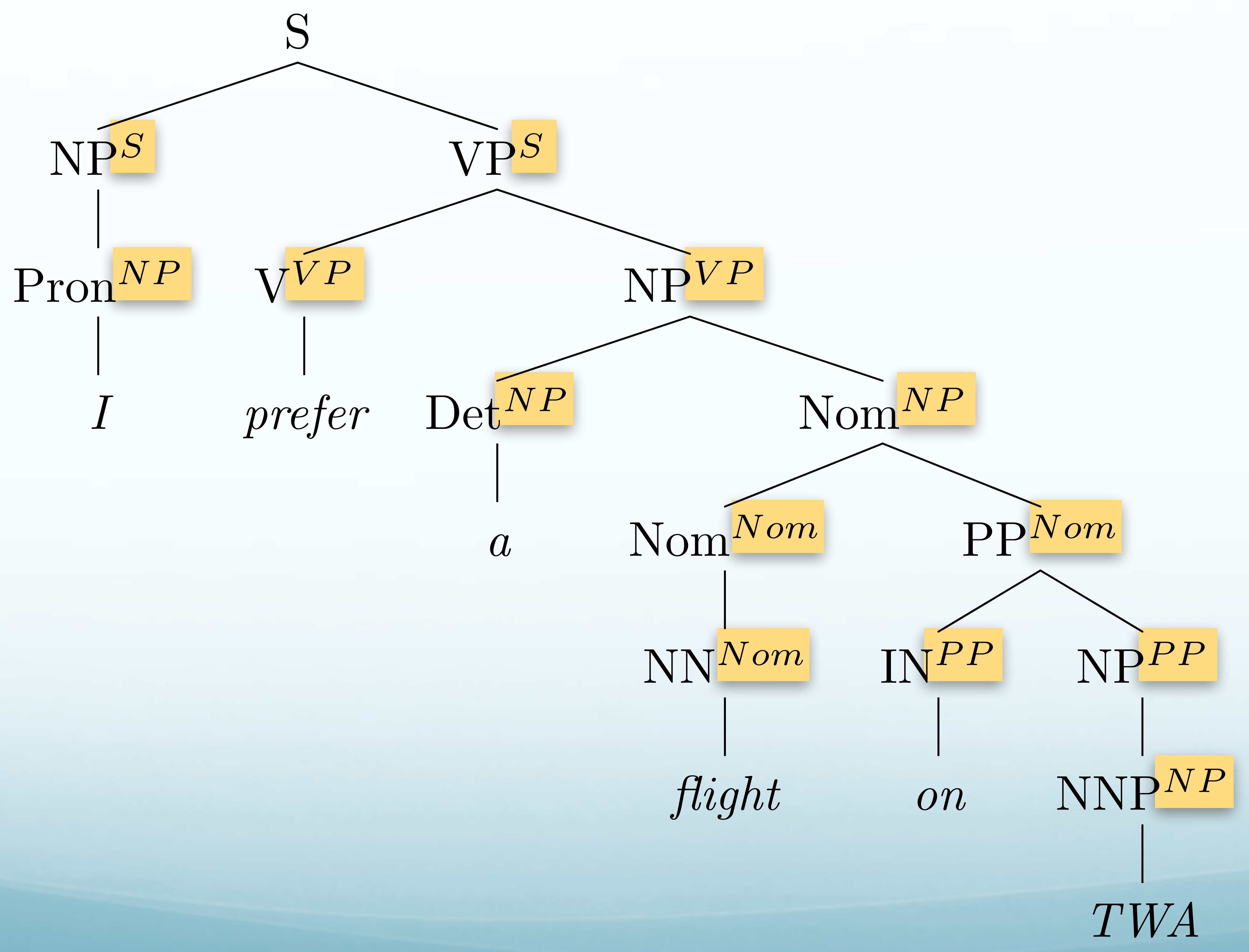
Improving PCFGs: Parent Annotation

- To handle the $NP \rightarrow PRP$ $[0.91 \text{ if } NP_{\Theta=subject} \text{ else } 0.34]$
  - Can annotate each node with its parent



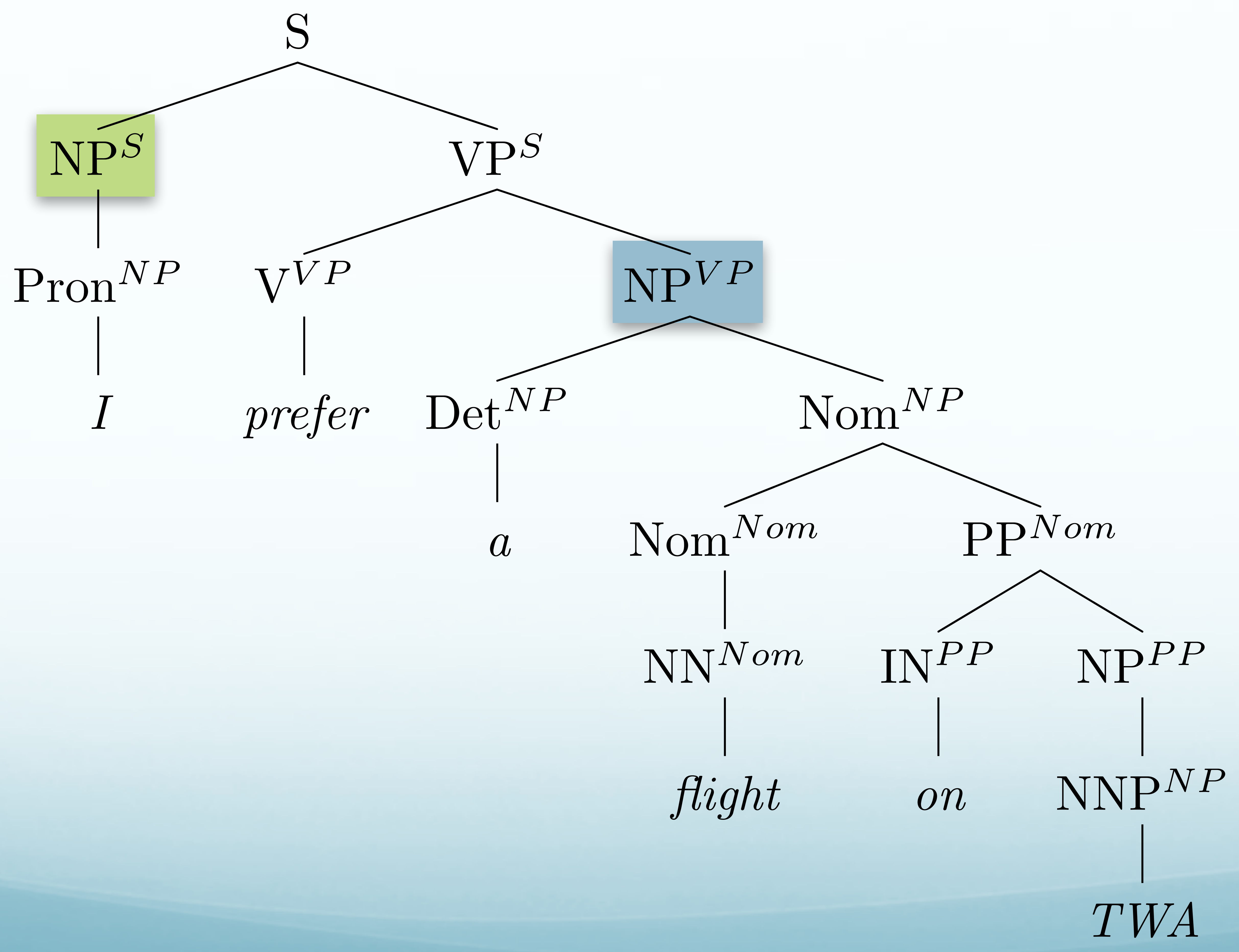
Improving PCFGs: Parent Annotation

- To handle the $NP \rightarrow PRP$ $[0.91 \text{ if } NP_{\Theta=subject} \text{ else } 0.34]$
  - Can annotate each node with its parent



Improving PCFGs: Parent Annotation

- To handle the $NP \rightarrow PRP$ $[0.91 \text{ if } NP_{\Theta=subject} \text{ else } 0.34]$
  - Can annotate each node with its parent



Improving PCFGs: Parent Annotation

- Advantages:
  - Captures structural dependencies in grammar
- Disadvantages:
  - Explodes number of rules in grammar
    - Same problem with subcategorization
  - Results in sparsity problems
- Strategies to find an optimal number of splits
  - Petrov et al (2006)

Improving PCFGs

- Parent Annotation
- **Lexicalization**
- Markovization
- Reranking

Improving PCFGs: Lexical "Heads"

- Remember back to syntax intro (Lecture #1)
  - Phrases are "headed" by key words
    - **VP** are headed by **V**
    - **NP** by **NN, NNS, PRON**
    - **PP** by **PREP**

- We can take advantage of this in our grammar!

Improving PCFGs: Lexical Dependencies

- As we've seen, some rules should be conditioned on certain words
- **Proposal**: annotate nonterminals with lexical head

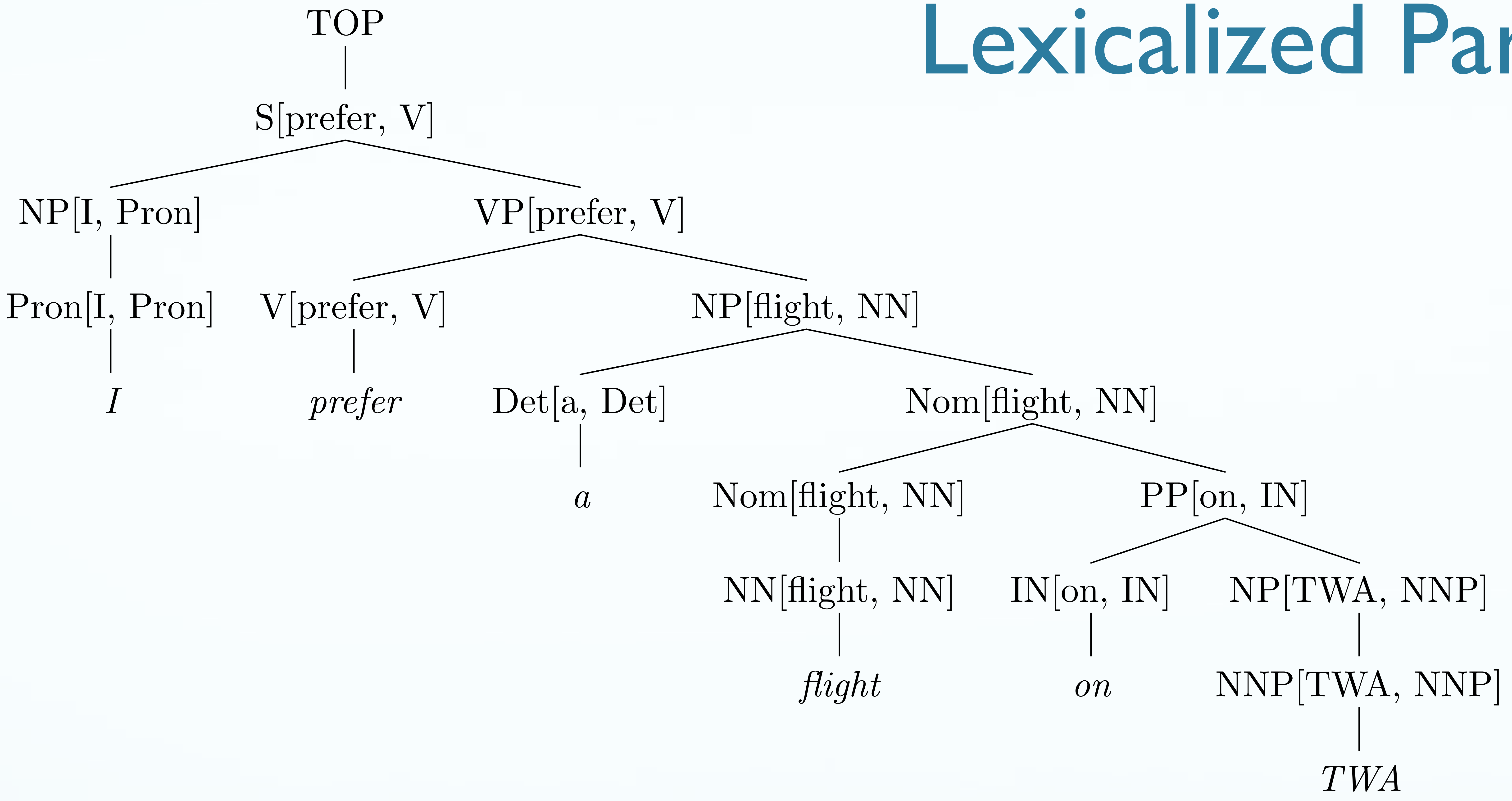
$$VP \rightarrow VBD\ NP\ PP$$

$$VP(\boldsymbol{dumped}) \rightarrow VBD(\boldsymbol{dumped})\ NP(\boldsymbol{sacks})\ PP(\boldsymbol{into})$$

- **Additionally**: annotate with lexical head + POS

$$VP(dumped,\ \boldsymbol{VBD}) \rightarrow VBD(dumped,\ \boldsymbol{VBD})\ NP(sacks,\ \boldsymbol{NNS})\ PP(into,\ \boldsymbol{IN})$$

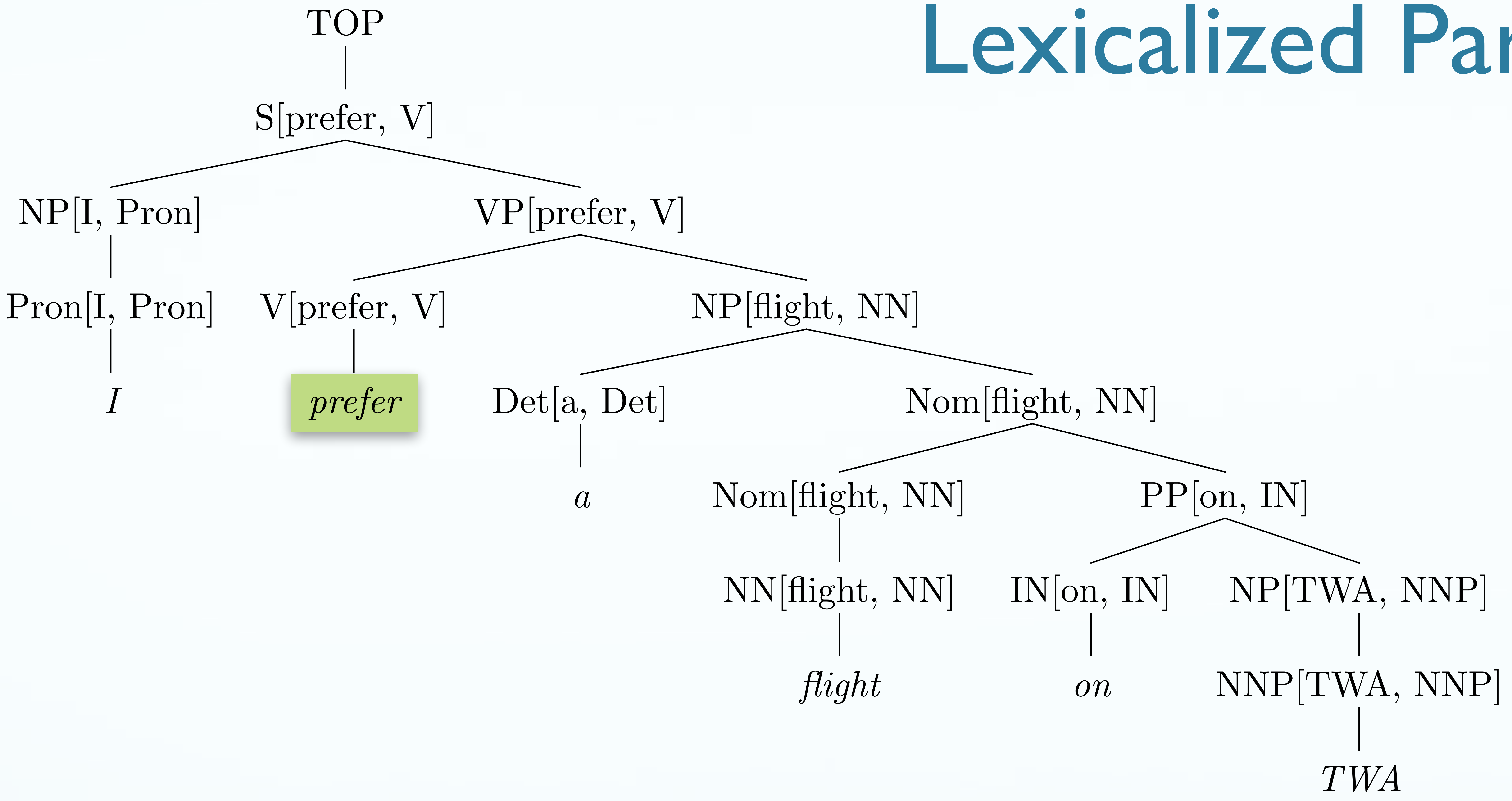
Lexicalized Parse Tree



Internal Rules			
TOP	→	*S(prefer, V)*	
S(prefer, V)	→	*NP(I, Pron)*	*VP(prefer, V)*
NP(I, Pron)	→	*Pron(I, Pron)*	
VP(prefer, V)	→	*V(prefer, V)*	*NP(flight, NN)*
NP(flight, NN)	→	*Det(a, Det)*	*Nom(flight, NN)*
PP(on, IN)	→	*IN(on, IN)*	*NP(TWA, NNP)*

Lexical Rules		
Pron(I, Pron)	→	I
V(prefer, V)	→	prefer
Det(a, Det)	→	a
NN(flight, NN)	→	flight
IN(on, IN)	→	on
NNP(NWA, NNP)	→	TWA

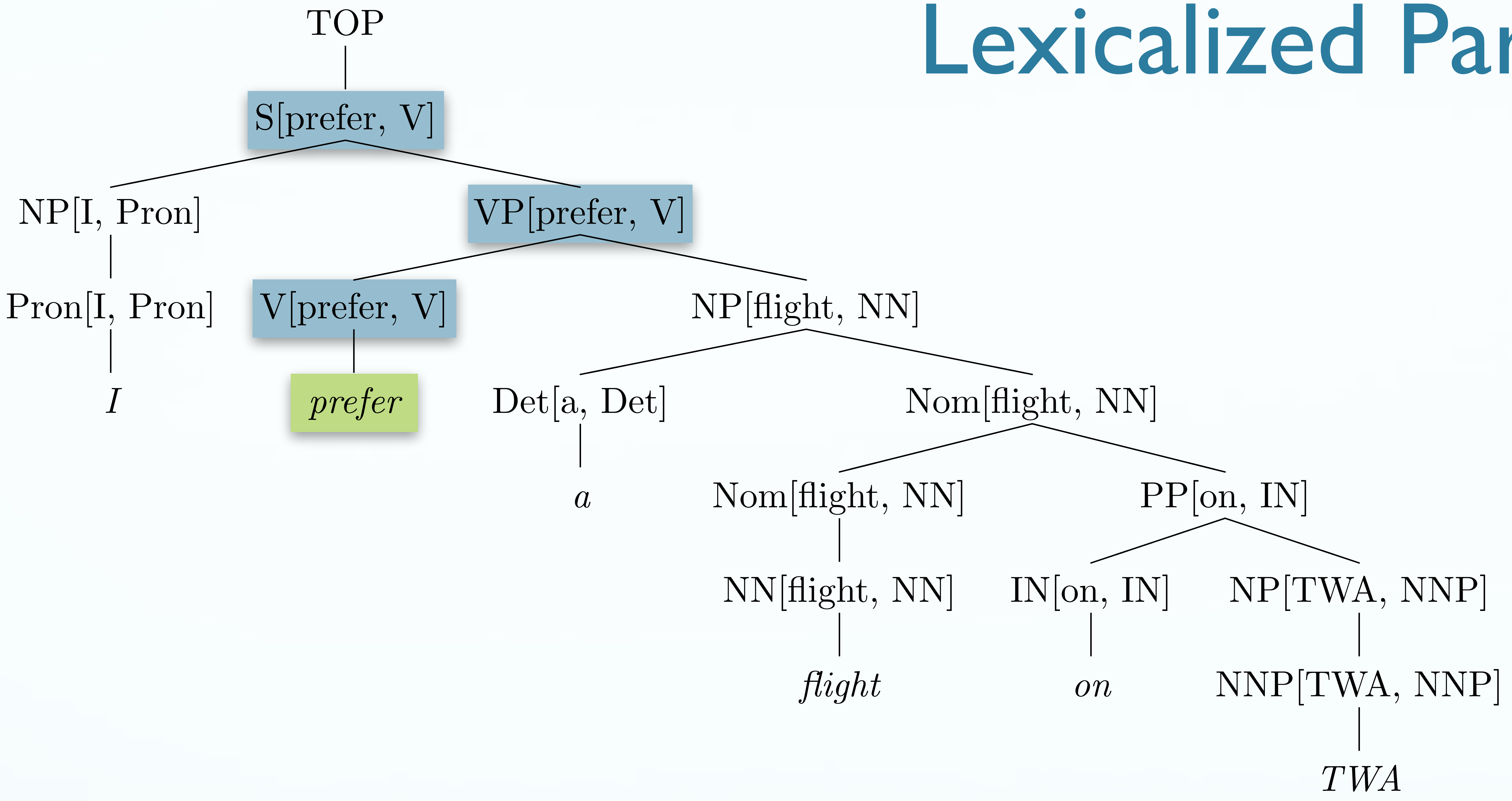
Lexicalized Parse Tree



Internal Rules			
TOP	$\rightarrow$	$S(prefer,\ V)$	
$S(prefer,\ V)$	$\rightarrow$	$NP(I,\ Pron)$	$VP(prefer,\ V)$
$NP(I,\ Pron)$	$\rightarrow$	$Pron(I,\ Pron)$	
$VP(prefer,\ V)$	$\rightarrow$	$V(prefer,\ V)$	$NP(flight,\ NN)$
$NP(flight,\ NN)$	$\rightarrow$	$Det(a,\ Det)$	$Nom(flight,\ NN)$
$PP(on,\ IN)$	$\rightarrow$	$IN(on,\ IN)$	$NP(TWA,\ NNP)$

Lexical Rules		
$Pron(I,\ Pron)$	$\rightarrow$	I
$V(prefer,\ V)$	$\rightarrow$	prefer
$Det(a,\ Det)$	$\rightarrow$	a
$NN(flight,\ NN)$	$\rightarrow$	flight
$IN(on,\ IN)$	$\rightarrow$	on
$NNP(NWA,\ NNP)$	$\rightarrow$	TWA

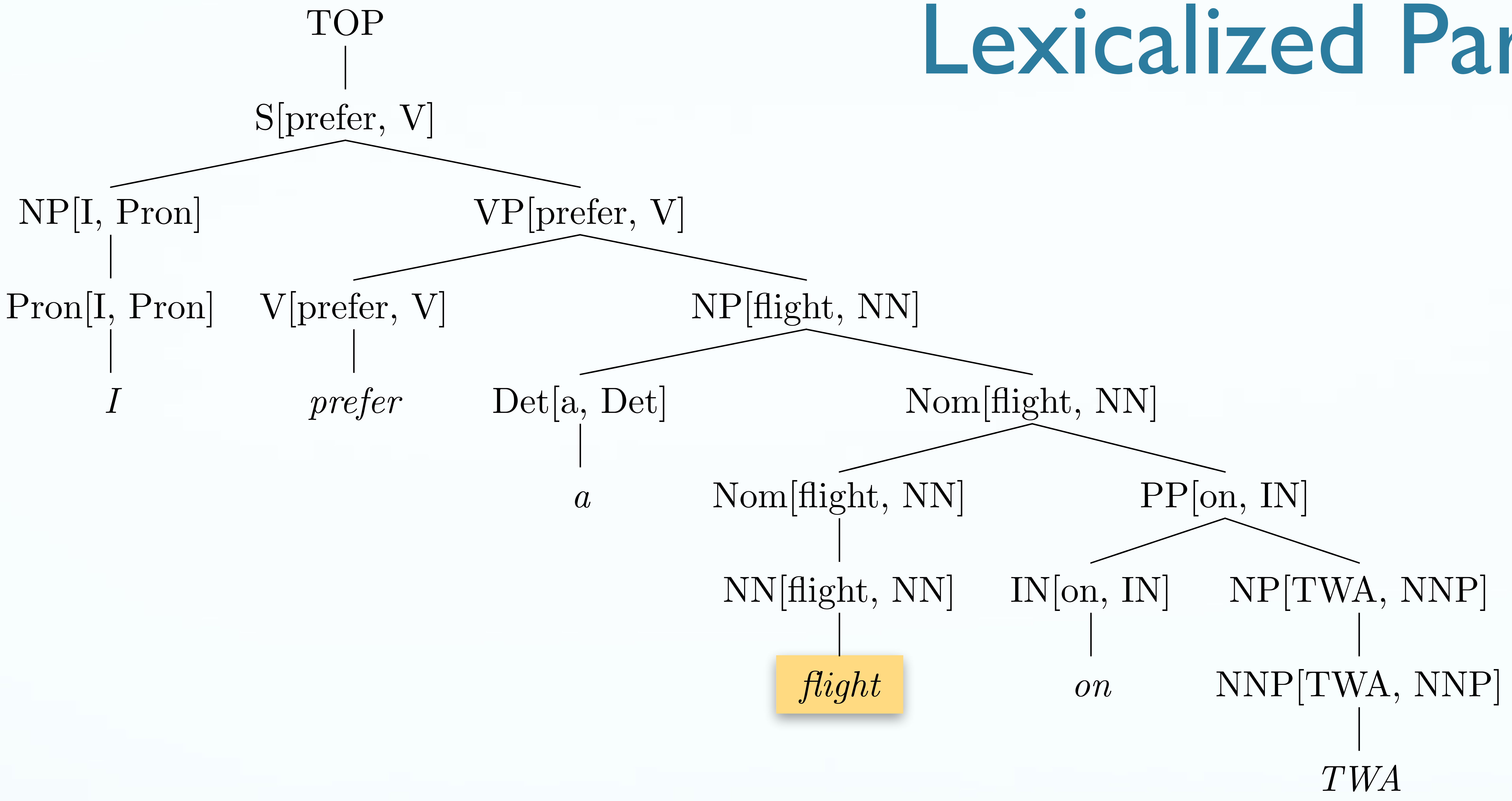
Lexicalized Parse Tree



Internal Rules			
TOP	→	*S(prefer, V)*	
S(prefer, V)	→	*NP(I, Pron)*	*VP(prefer, V)*
NP(I, Pron)	→	*Pron(I, Pron)*	
VP(prefer, V)	→	*V(prefer, V)*	*NP(flight, NN)*
NP(flight, NN)	→	*Det(a, Det)*	*Nom(flight, NN)*
PP(on, IN)	→	*IN(on, IN)*	*NP(TWA, NNP)*

Lexical Rules		
Pron(I, Pron)	→	I
V(prefer, V)	→	prefer
Det(a, Det)	→	a
NN(flight, NN)	→	flight
IN(on, IN)	→	on
NNP(NWA, NNP)	→	TWA

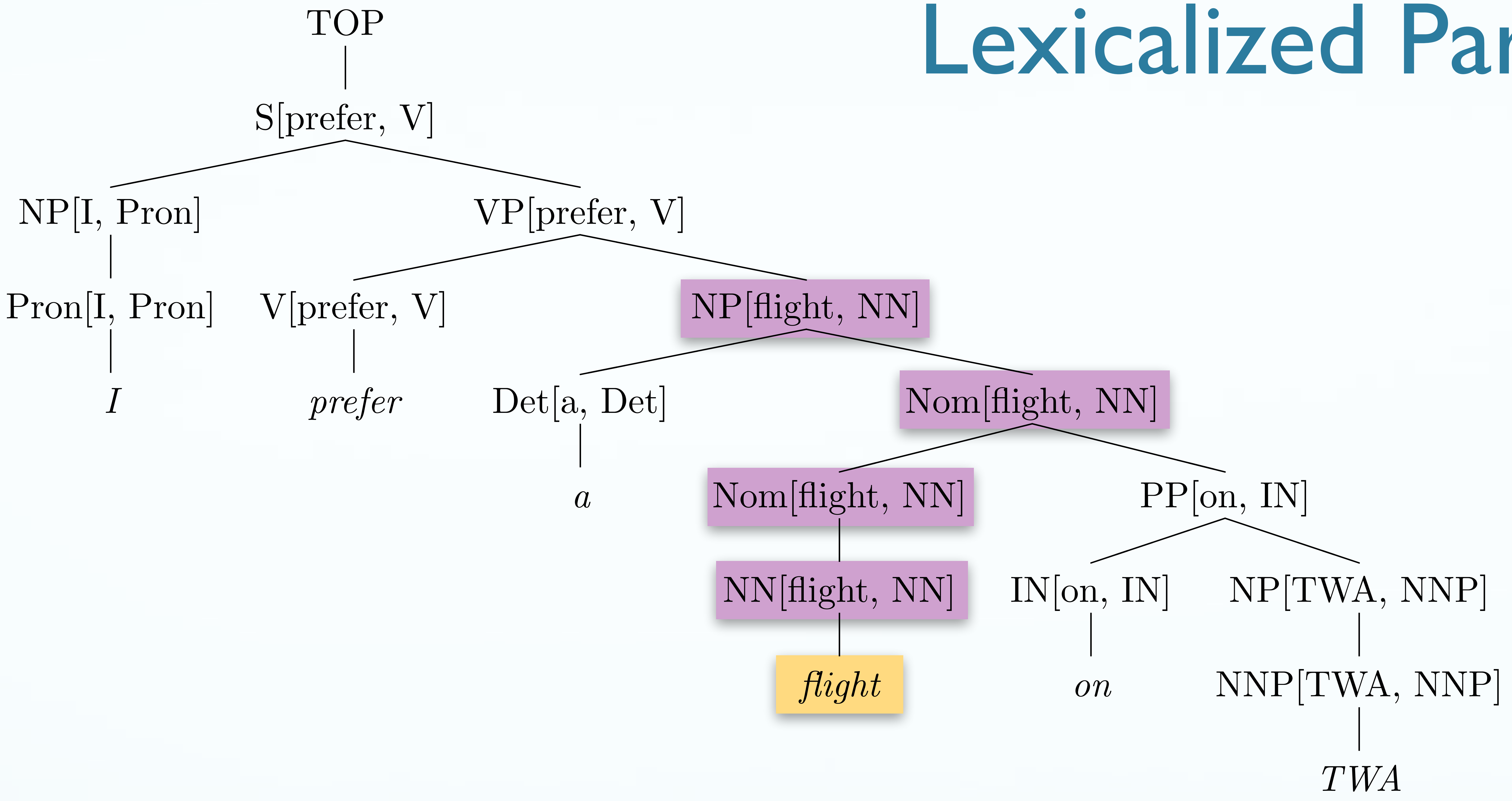
Lexicalized Parse Tree



Internal Rules			
TOP	→	*S(prefer, V)*	
S(prefer, V)	→	*NP(I, Pron)*	*VP(prefer, V)*
NP(I, Pron)	→	*Pron(I, Pron)*	
VP(prefer, V)	→	*V(prefer, V)*	*NP(flight, NN)*
NP(flight, NN)	→	*Det(a, Det)*	*Nom(flight, NN)*
PP(on, IN)	→	*IN(on, IN)*	*NP(TWA, NNP)*

Lexical Rules		
Pron(I, Pron)	→	I
V(prefer, V)	→	prefer
Det(a, Det)	→	a
NN(flight, NN)	→	flight
IN(on, IN)	→	on
NNP(NWA, NNP)	→	TWA

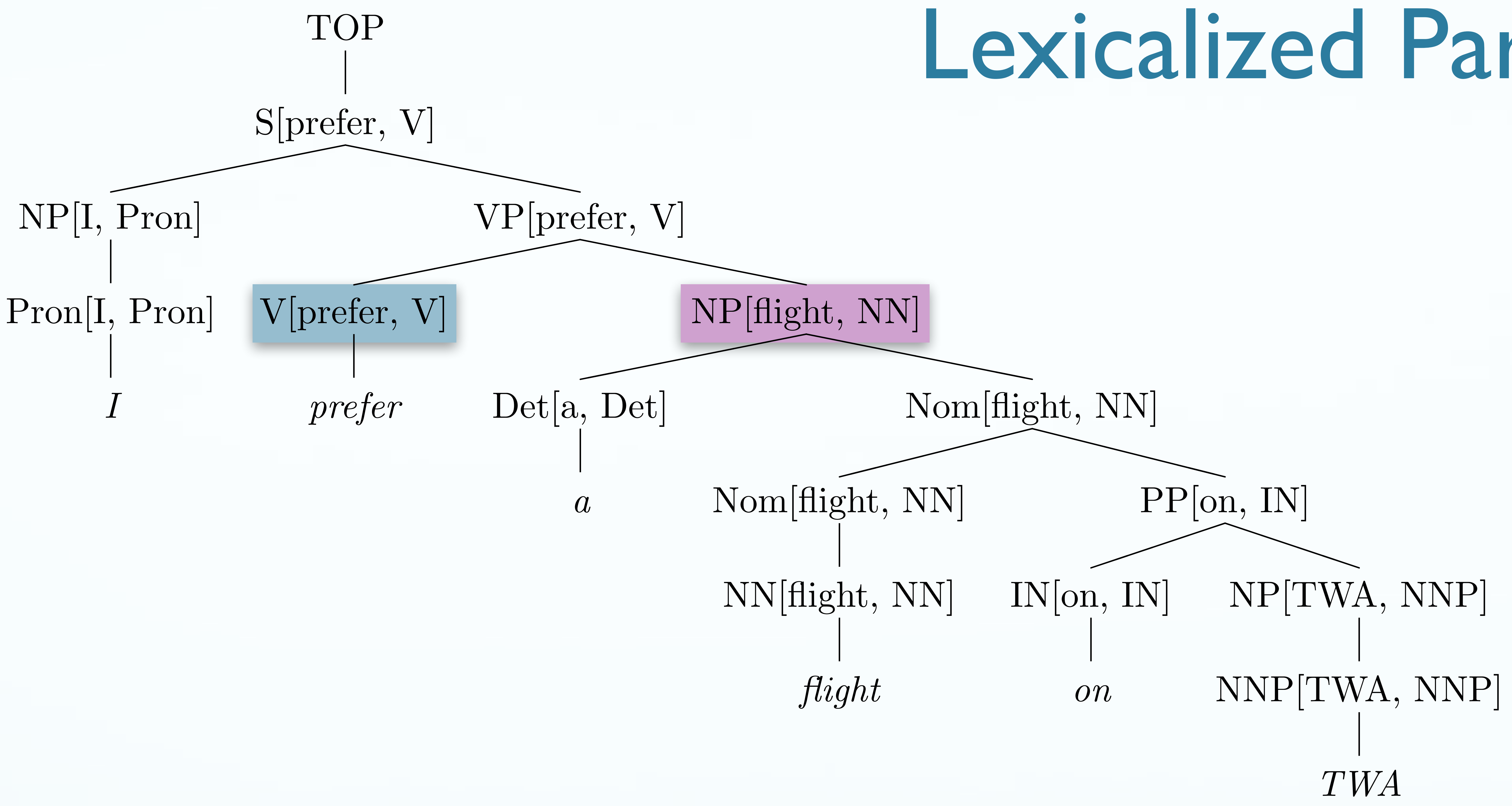
Lexicalized Parse Tree



Internal Rules			
TOP	→	*S(prefer, V)*	
S(prefer, V)	→	*NP(I, Pron)*	*VP(prefer, V)*
NP(I, Pron)	→	*Pron(I, Pron)*	
VP(prefer, V)	→	*V(prefer, V)*	*NP(flight, NN)*
NP(flight, NN)	→	*Det(a, Det)*	*Nom(flight, NN)*
PP(on, IN)	→	*IN(on, IN)*	*NP(TWA, NNP)*

Lexical Rules		
Pron(I, Pron)	→	I
V(prefer, V)	→	prefer
Det(a, Det)	→	a
NN(flight, NN)	→	flight
IN(on, IN)	→	on
NNP(NWA, NNP)	→	TWA

Lexicalized Parse Tree

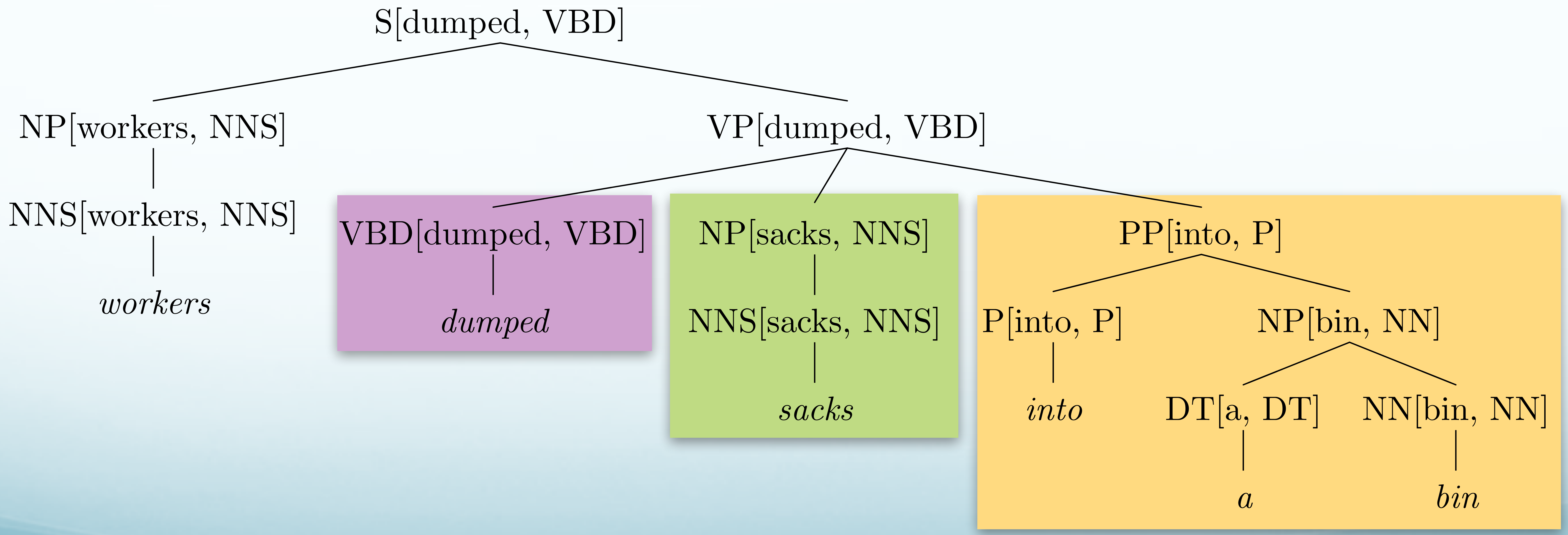


Internal Rules			
TOP	→	*S(prefer, V)*	
S(prefer, V)	→	*NP(I, Pron)*	*VP(prefer, V)*
NP(I, Pron)	→	*Pron(I, Pron)*	
VP(prefer, V)	→	*V(prefer, V)*	*NP(flight, NN)*
NP(flight, NN)	→	*Det(a, Det)*	*Nom(flight, NN)*
PP(on, IN)	→	*IN(on, IN)*	*NP(TWA, NNP)*

Lexical Rules		
Pron(I, Pron)	→	I
V(prefer, V)	→	prefer
Det(a, Det)	→	a
NN(flight, NN)	→	flight
IN(on, IN)	→	on
NNP(NWA, NNP)	→	TWA

Improving PCFGs: Lexical Dependencies

- Upshot: heads propagate up tree:
  - $VP \rightarrow VBD(dumped,\ VBD)\ NP(sacks,\ NNS)\ PP(into,\ P)$
  - $NP \rightarrow NNS(sacks,\ NNS)\ PP(into,\ P)$



Improving PCFGs: Lexical Dependencies

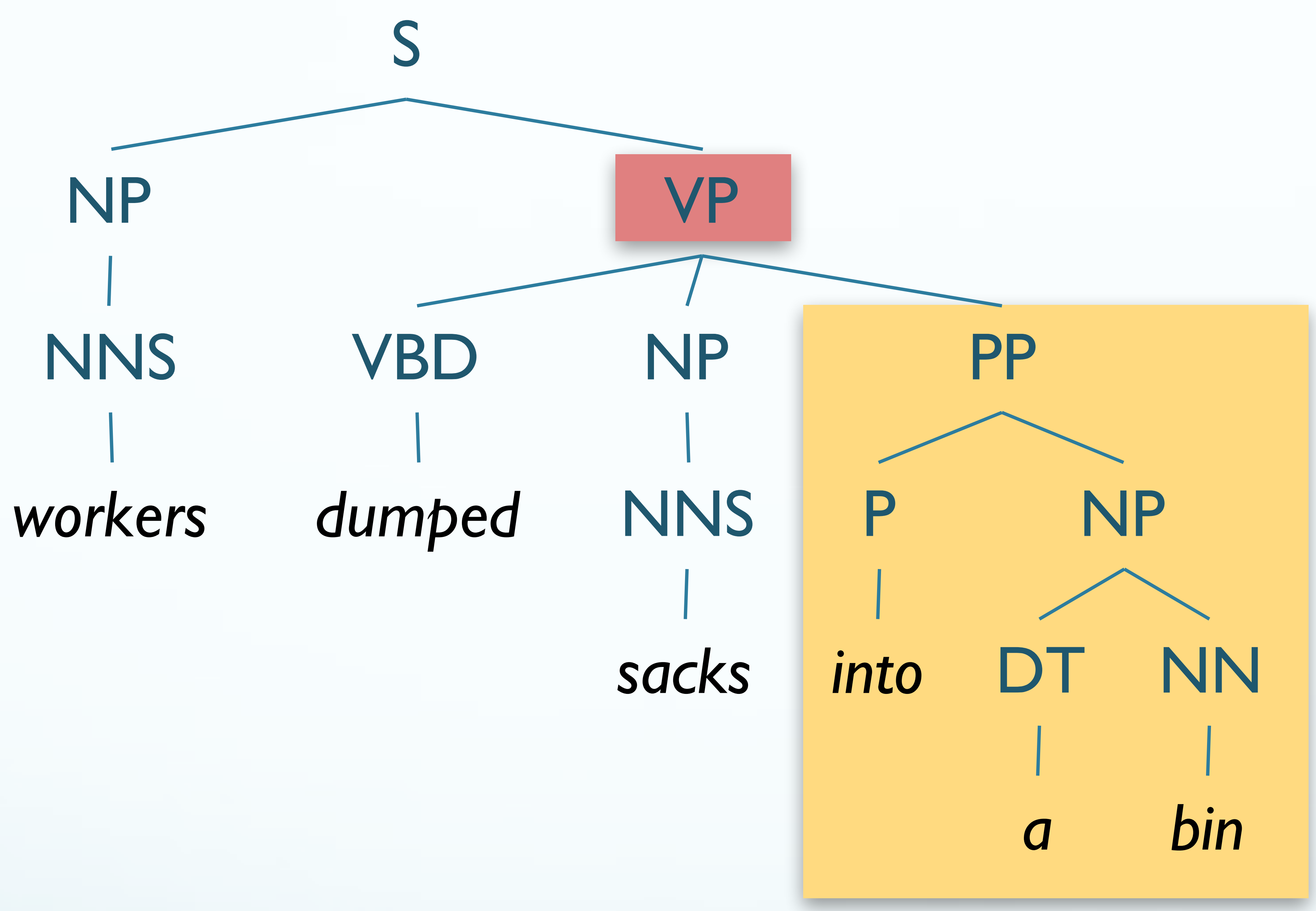
- Downside:
  - Rules far too specialized — will be sparse
- Solution:
  - Assume ***conditional*** independence
  - Create more rules

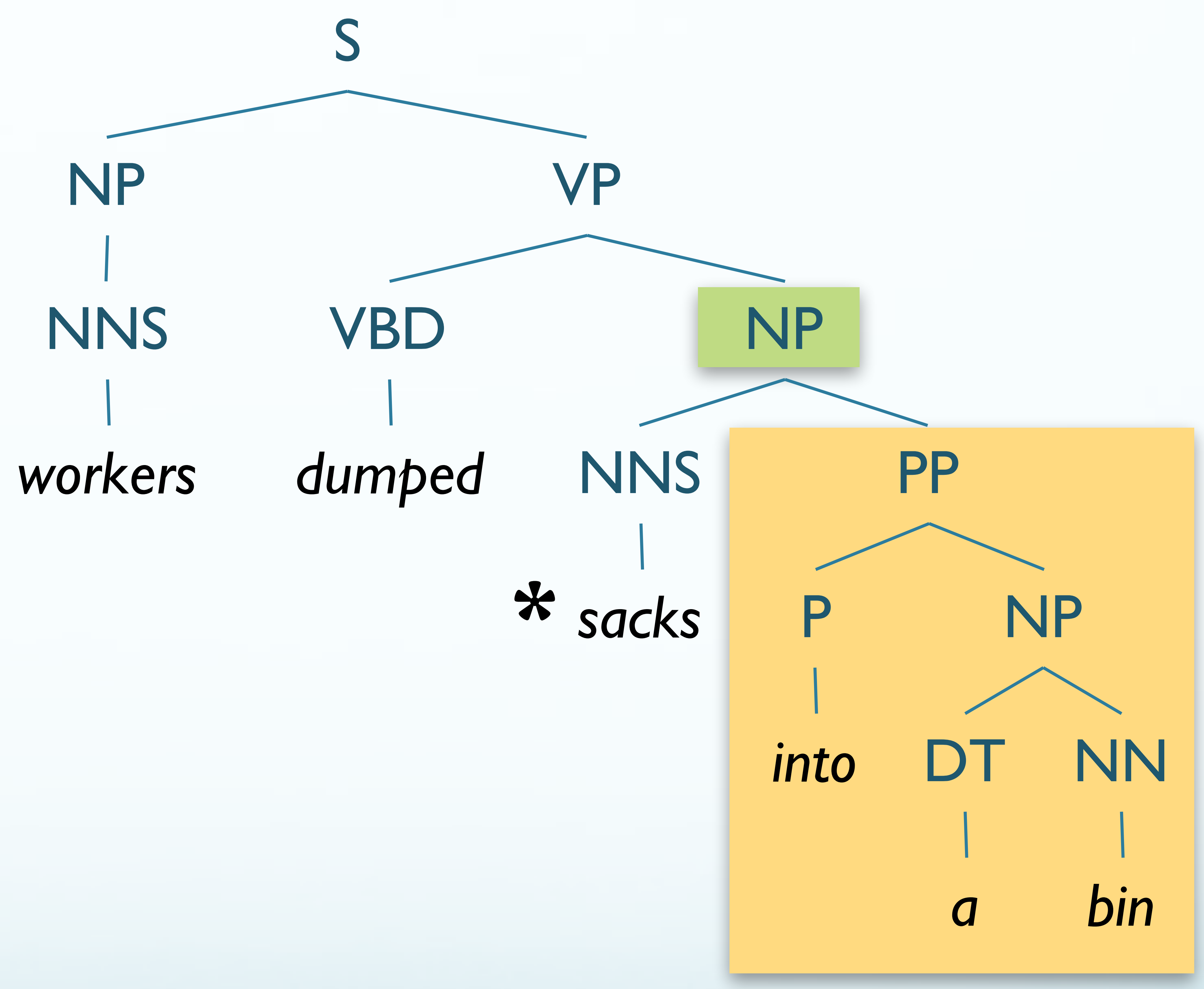
Improving PCFGs: Collins Parser

- Proposal:
  - $\boldsymbol{LHS} \rightarrow \boldsymbol{LeftOfHead} \ldots \quad \boldsymbol{Head} \quad \ldots \boldsymbol{RightOfHead}$
  - Instead of calculating $P(EntireRule)$, which is sparse:
  - Calculate:
    - Probability that $\boldsymbol{LHS}$ has nonterminal phrase $\boldsymbol{H}$ given head-word $hw\ldots$
    - × Probability of modifiers to the **left** given head-word $hw\ldots$
    - × Probability of modifiers to the **right** given head-word $hw\ldots$

$$P(T,S) = \prod_{n \in T} p\left(r(n) \mid n, h(n)\right) \cdot p\left(h(n) \mid n, h\left(m(n)\right)\right)$$

Collins Parser Example





Collins Parser Example

$$P(VP \rightarrow VBD\ NP\ PP \mid VP, dumped)$$
$$= \frac{Count\left(VP\left(dumped\right) \rightarrow VBD\ NP\ PP\right)}{\sum_{\beta} Count\left(VP\left(dumped\right) \rightarrow \beta\right)}$$
$$= \frac{6}{9} = 0.67$$

$$P(VP \rightarrow VBD\ NP \mid VP, dumped)$$
$$= \frac{Count\left(VP\left(dumped\right) \rightarrow VBD\ NP\right)}{\sum_{\beta} Count\left(VP\left(dumped\right) \rightarrow \beta\right)}$$
$$= \frac{1}{9} = 0.11$$

$$P_R(into \mid PP, dumped)$$
$$= \frac{Count\left(X\left(dumped\right) \rightarrow \ldots\ PP\left(into\right)\ \ldots\right)}{\sum_{\beta} Count\left(X\left(dumped\right) \rightarrow \ldots\ PP\ \ldots\right)}$$
$$= \frac{2}{9} = 0.22$$

$$P_R(into \mid PP, sacks)$$
$$= \frac{Count\left(X\left(sacks\right) \rightarrow \ldots\ PP\left(into\right)\ \ldots\right)}{\sum_{\beta} Count\left(X\left(sacks\right) \rightarrow \ldots\ PP\ \ldots\right)}$$
$$= \frac{0}{0}$$

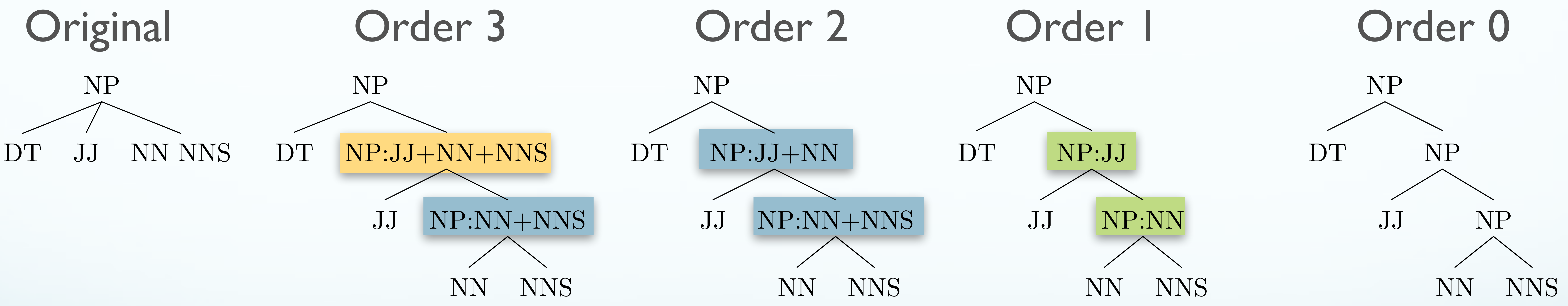
Improving PCFGs

- Parent Annotation
- Lexicalization
- **Markovization**
- Reranking

CNF Factorization & Markovization

- CNF Factorization:
  - Converts n-ary branching to binary branching
  - Can maintain information about original structure
    - Neighborhood history and parent

Different Markov Orders



Markovization and Costs

PCFG	Time(s)	Words/s	|V|	|P|	LR	LP	F_1
Right-factored	4848	6.7	10105	23220	69.2	73.8	71.5
Right-factored, Markov order-2	1302	24.9	2492	11659	68.8	73.8	71.3
Right-factored, Markov order-1	445	72.7	564	6354	68.0	730	70.5
Right-factored, Markov order-0	206	157.1	99	3803	61.2	65.5	63.3
Parent-annotated, Right-factored, Markov order-2	7510	4.3	5876	22444	76.2	78.3	77.2

from Mohri & Roark 2006

Improving PCFGs

- Parent Annotation
- Lexicalization
- Markovization
- **Reranking**

Reranking

- Issue: Locality
  - PCFG probabilities associated with rewrite rules
  - Context-free grammars are, well, context-free
  - Previous approaches create new rules to incorporate context
- Need approach that incorporates broader, global info

Discriminative Parse Reranking

- General approach:
  - Parse using (L)PCFG
  - Obtain top-N parses
  - Re-rank top-N using better features
- Use discriminative model (e.g. MaxEnt) to rerank with features:
  - right-branching vs. left-branching
  - speaker identity
  - conjunctive parallelism
  - fragment frequency
  - …

Reranking Effectiveness

- How can reranking improve?
  - …assuming N-best includes the correct parse
- Results from Collins (2000), with 50-best

System	Accuracy
Baseline	0.897
Oracle	0.968
Discriminative	0.917

- "Oracle" is to automatically choose the correct parse if in N-best

Improving PCFGs: Tradeoffs

- **Pros:**
  - Increased accuracy/specificity
  - e.g. Lexicalization, Parent annotation, Markovization, etc
- **Cons**:
  - Explode grammar size
  - Increased processing time
  - Increased data requirements
- *How can we balance?*

Improving PCFGs: Efficiency

- **Beam thresholding**
- Heuristic Filtering

Efficiency

- PCKY is $|G| \cdot n^3$
  - Grammar can be huge
  - Grammar can be extremely ambiguous
  - Hundreds of analyses not unusual
- …but only care about best parses
- Can we use this to improve efficiency?

Beam Thresholding

- Inspired by Beam Search
- Assume low probability parses unlikely to yield high probability overall
  - Keep only top k most probable partial parses
  - Retain only k choices per cell
    - For large grammars, maybe 50-100
    - For small grammars, 5 or 10

Heuristic Filtering

- **Intuition**: Some rules/partial parses unlikely to create best parse
- **Proposal**: Don't store these in table.
- Exclude:
  - Low frequency: (singletons)
  - Low probability: constituents $\boldsymbol{X}$ s.t. $P(\boldsymbol{X}) < 10^{-200}$
  - Low relative probability:
    - Exclude $\boldsymbol{X}$ if there exists $\boldsymbol{Y}$ s.t. $P(\boldsymbol{Y}) > 100 \times P(\boldsymbol{X})$

HW #4

Probabilistic Parsing

- Goals:
  - Learn about PCFGs
  - Implement PCKY
  - Analyze Parsing Evaluation
  - Assess improvements to PCFG Parsing

Tasks

1. Train a PCFG
   - Estimate rule probabilities from treebank
   - Treebank is already in CNF
   - More ATIS data from Penn Treebank
2. Build CKY Parser
   - Modify (your) existing CKY implementation

Tasks

3. Evaluation
   - Evaluate your parser using standard metric
   - We will provide **evalb** program and gold standard
4. Improvement
   - Improve your parser in some way:
     - Coverage
     - Accuracy
     - Speed
   - Evaluate new parser

Improvement Possibilities

- Coverage:
  - Some test sentences won't parse as is!
    - Lexical gaps (aka out-of-vocabulary [OOV] tokens)
      - …remember to model the probabilities, too
- Better context modeling
  - e.g. — Parent Annotation
- Better Efficiency
  - e.g. — Heuristic Filtering, Beam Search
- No "cheating" improvements:
  - improvement can't change training by looking at test data