CKY Parsing & CNF Conversion

LING 571 — Deep Processing Techniques for NLP
October 3, 2018
Ryan Georgi

Announcements

- **HW #2** will be extended to Monday, 11/8 at **11:00pm**.
  - Then we will be caught up, so HW #3 will still be due that Friday.

- If you want to use `python3.6` on Patas:
  - **`/opt/python-3.6/bin/python3`**
  - `nltk` is installed.

Type Hinting in Python

- Supported in ≥3.6 [tutorial]

```
from typing import List
from nltk.grammar import Production

def fix_hybrid_production(hybrid_prod: Production) -> List[Production]:
    …
```

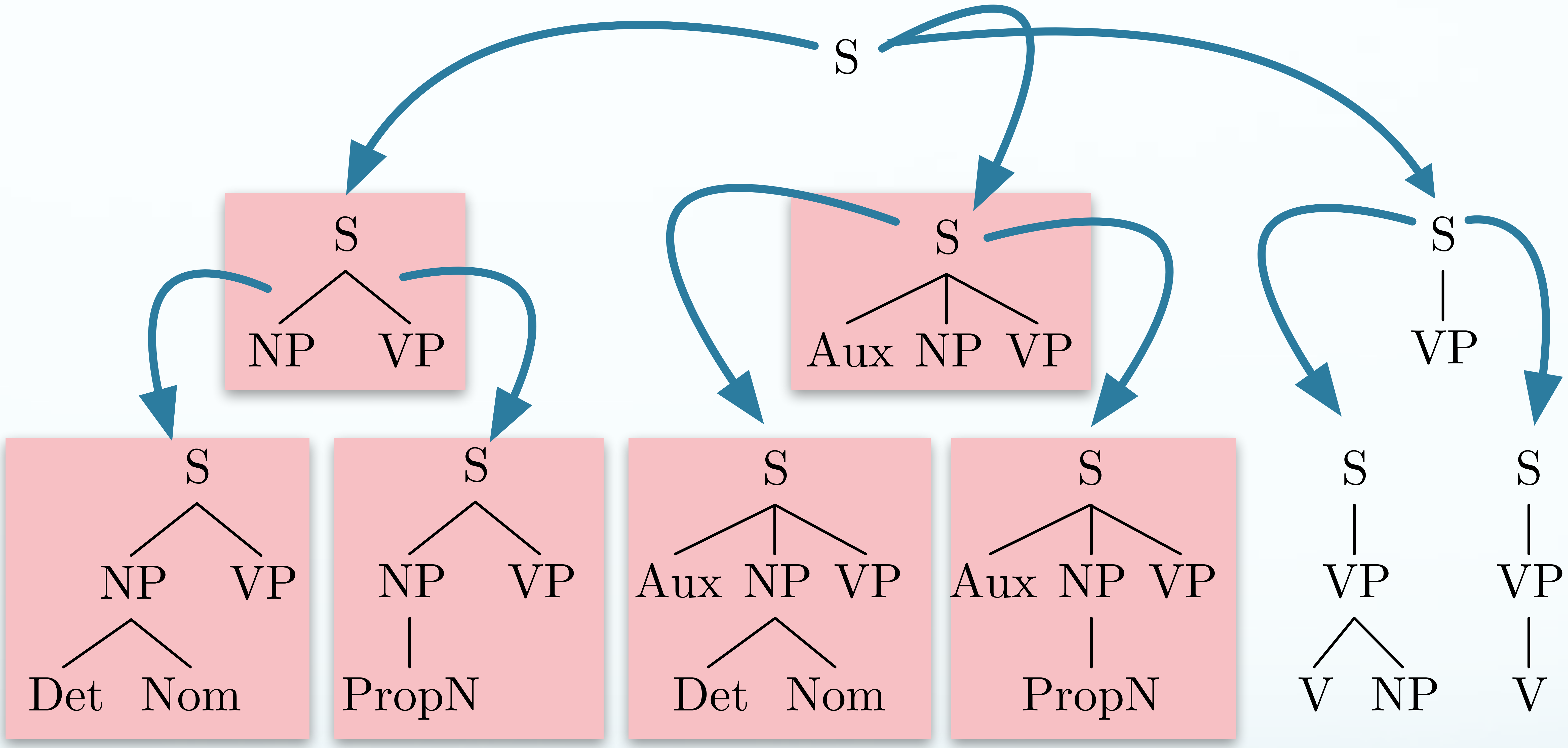
- Also available in PyCharm through docstrings and/or comments:

```
def fix_hybrid_productions(hybrid_prod):
    """
    This function takes a hybrid production and
    returns a list of new CNF productions
    :type hybrid_prod: Production
    :rtype: list[Production]
    """
```

Roadmap

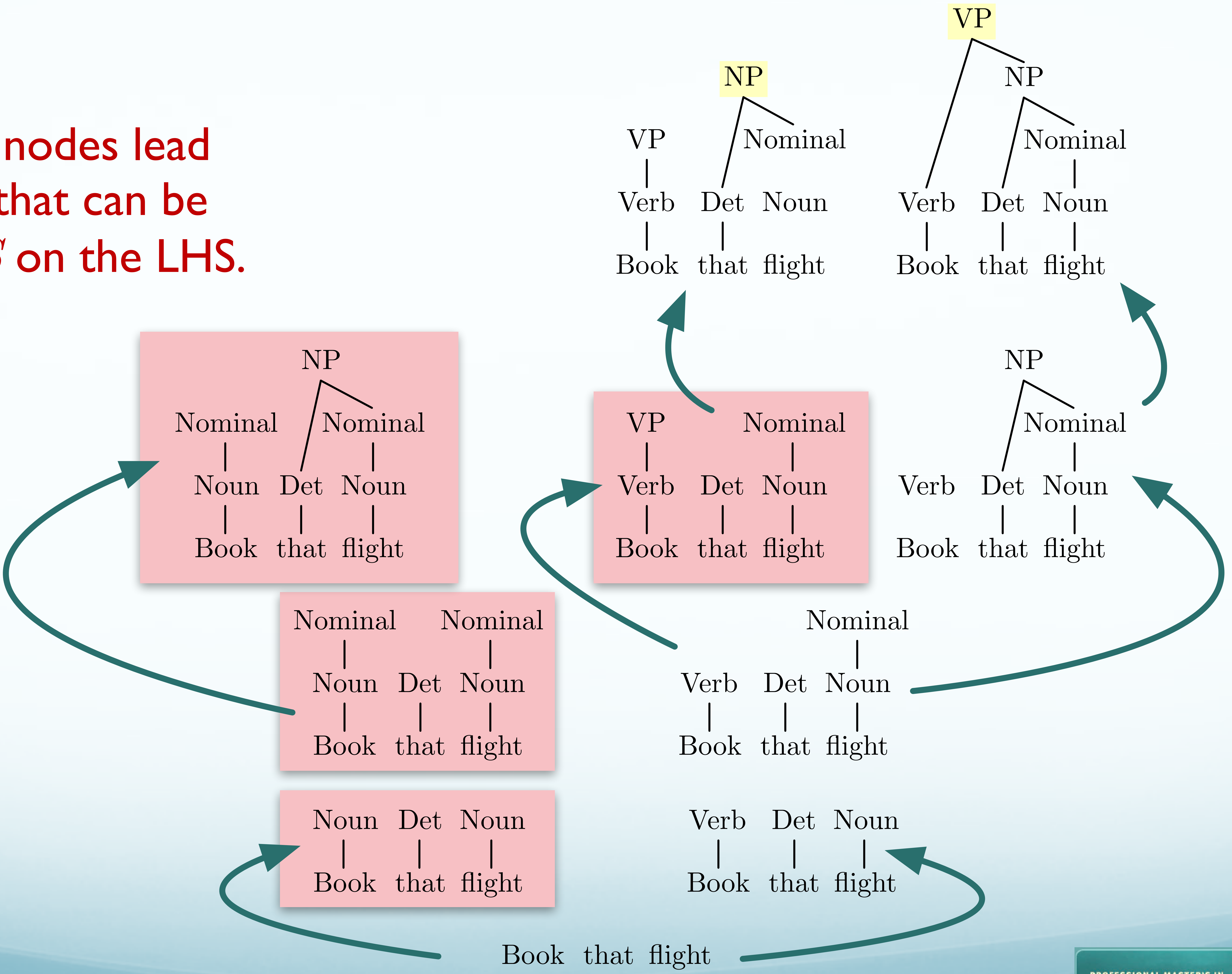
- **Recap: Parsing-as-Search**
- Parsing Challenges
- Strategy: Dynamic Programming
- Grammar Equivalence
- CKY parsing algorithm

Recap: Parsing as Search



None of these nodes can produce *book* as first terminal

None of these nodes lead lead to a RHS that can be combined with S on the LHS.



Parsing Challenges

- Recap: Parsing-as-Search
- **Parsing Challenges**
  - **Ambiguity**
  - Repeated Substructure
  - Recursion
- Strategy: Dynamic Programming
- Grammar Equivalence
- CKY parsing algorithm

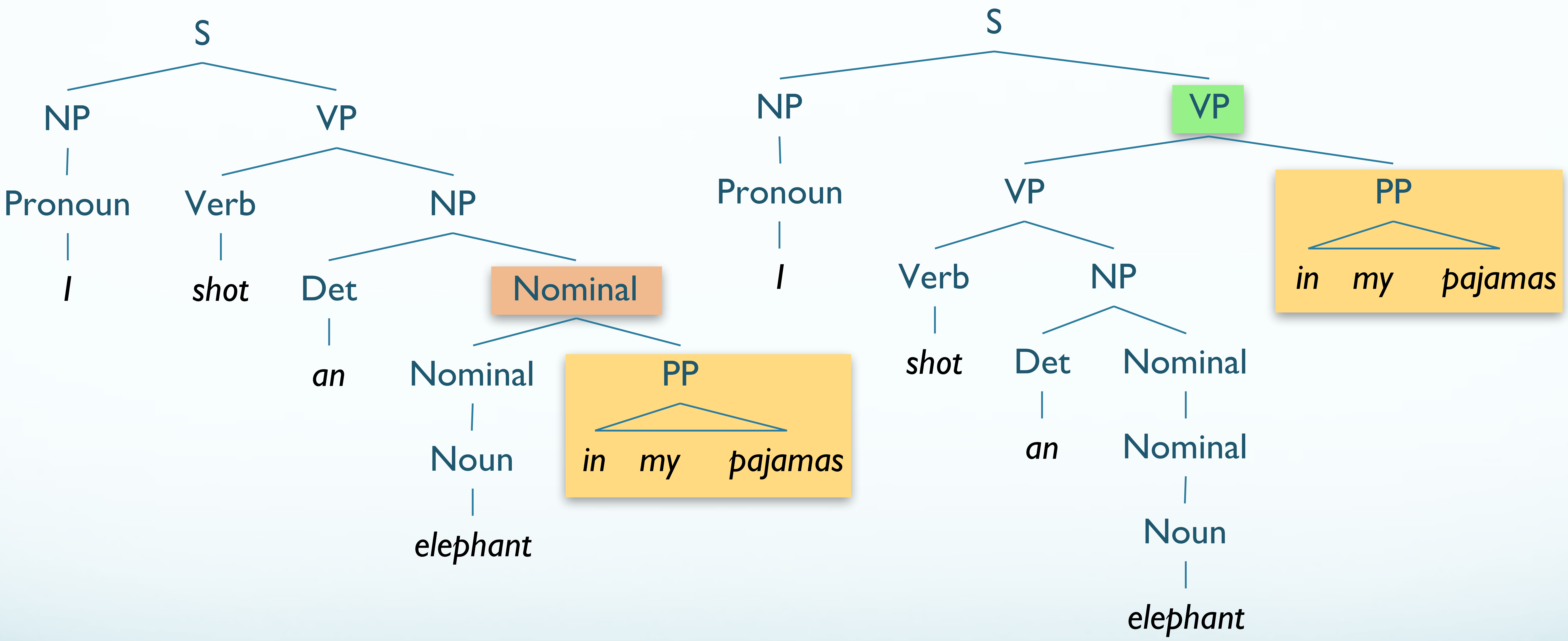
Parsing Ambiguity

- **Lexical Ambiguity**:
  - Book/NN → *I left a **book** on the table.*
  - Book/VB → ***Book** that flight.*
- Structural Ambiguity

Attachment Ambiguity

“One morning, I shot an elephant in my pajamas.
How he got into my pajamas, I’ll never know.” — *Groucho Marx*

Attachment Ambiguity



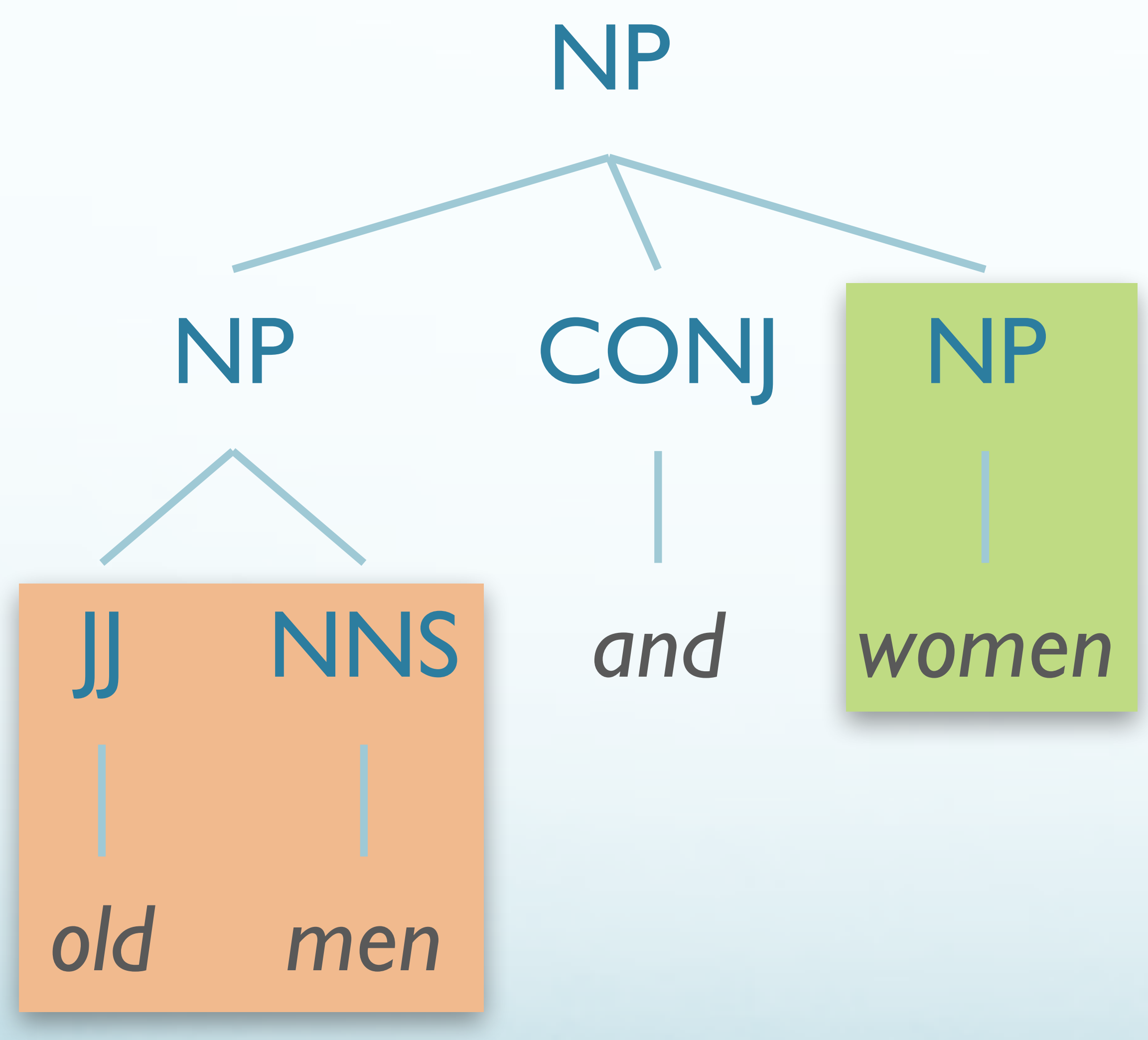
“We saw the Eiffel Tower flying to Paris”

Coordination Ambiguity:

“old men and women”

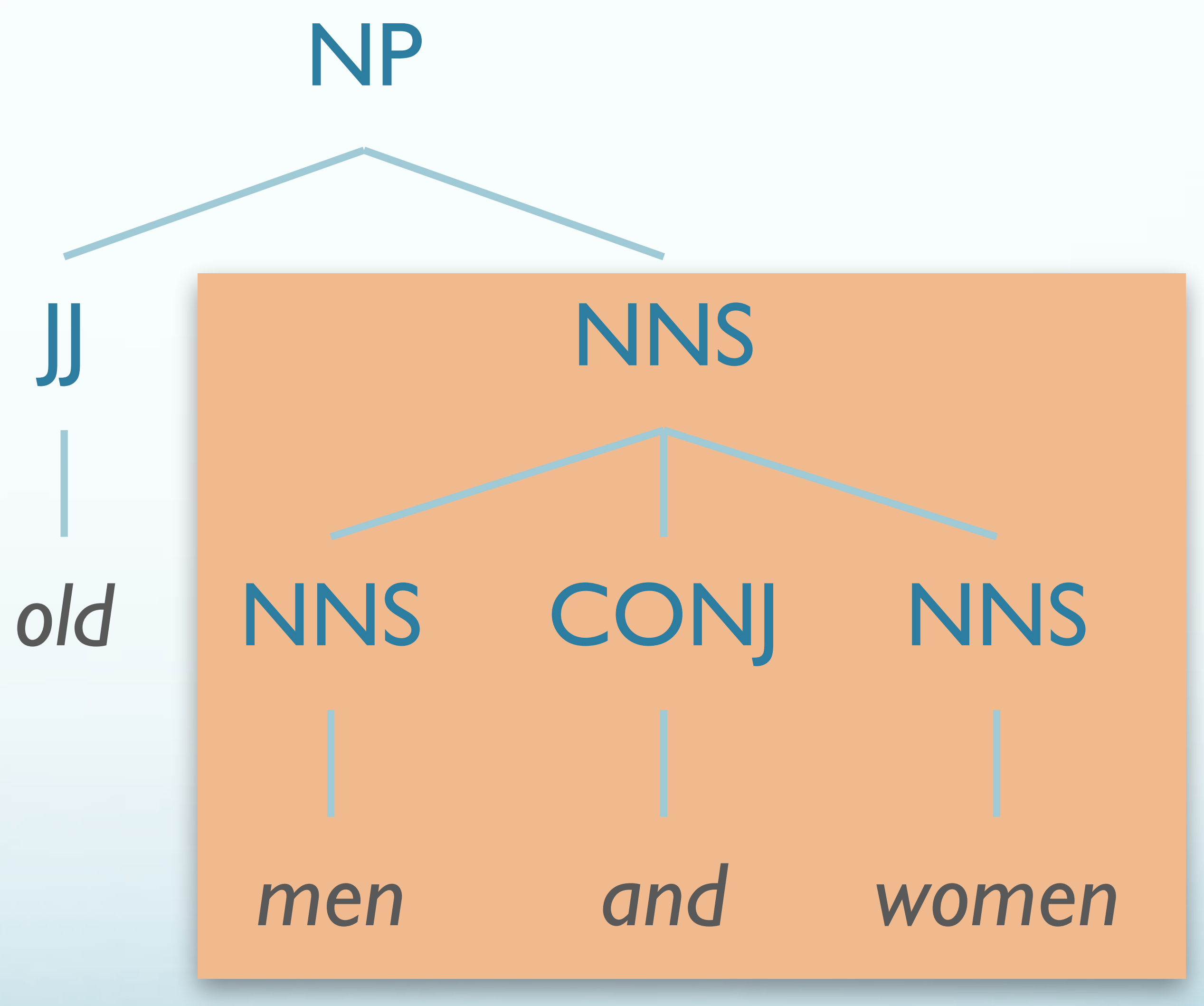
[old men] *and* **[women]**

(Only the men are old)


NP
NP CONJ NP
JJ NNS *and* *women*
old *men*

[old ***[men and women]****]*

(Both the men and women are old)


NP
JJ NNS
old NNS CONJ NNS
men *and* *women*

Local vs. Global Ambiguity

- ***Local*** ambiguity:
  - Ambiguity that cannot contribute to a full, valid parse
  - e.g. *Book/NN* in *"Book that flight"*
- ***Global*** ambiguity
  - Multiple valid parses

Why is Ambiguity a Problem?

- *Local* ambiguity:
  - increased processing time

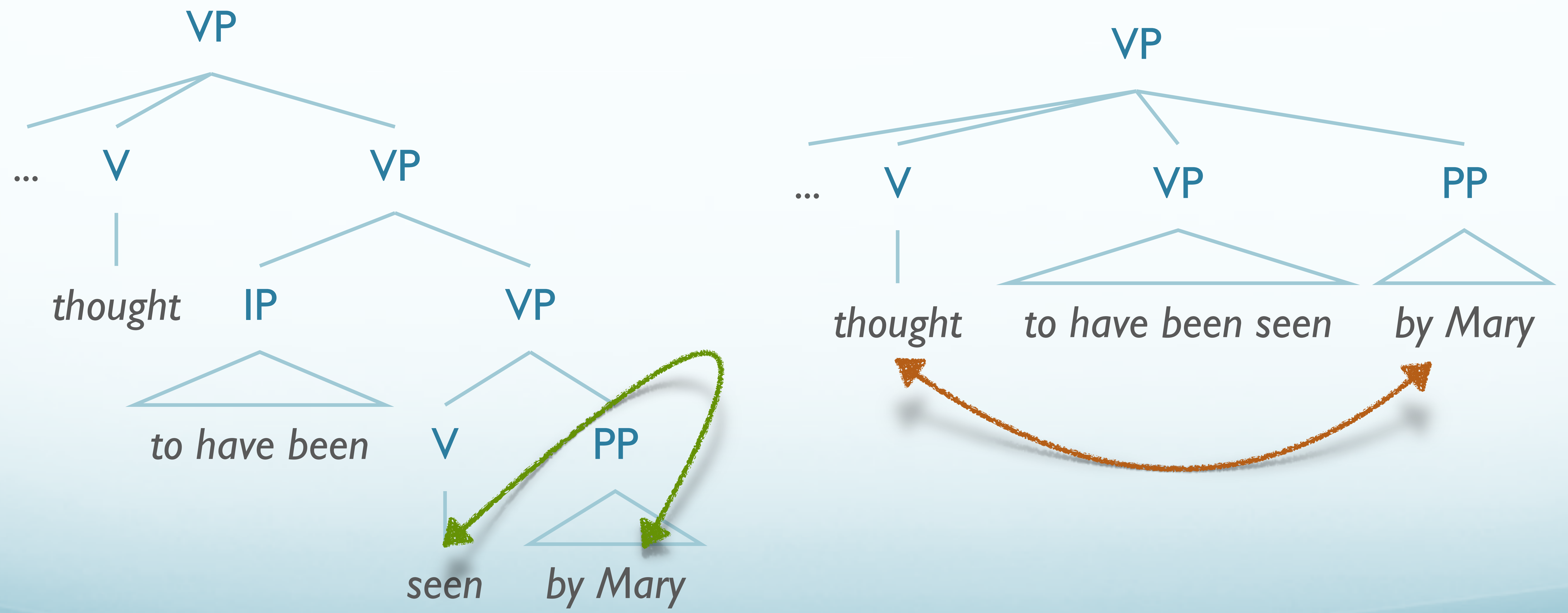
- *Global* ambiguity:
  - Would like to yield only "reasonable" parses
  - Ideally, the one that was intended*

Solution to Ambiguity?

- ***Dis*ambiguation!**
- Different possible strategies to select correct interpretation:

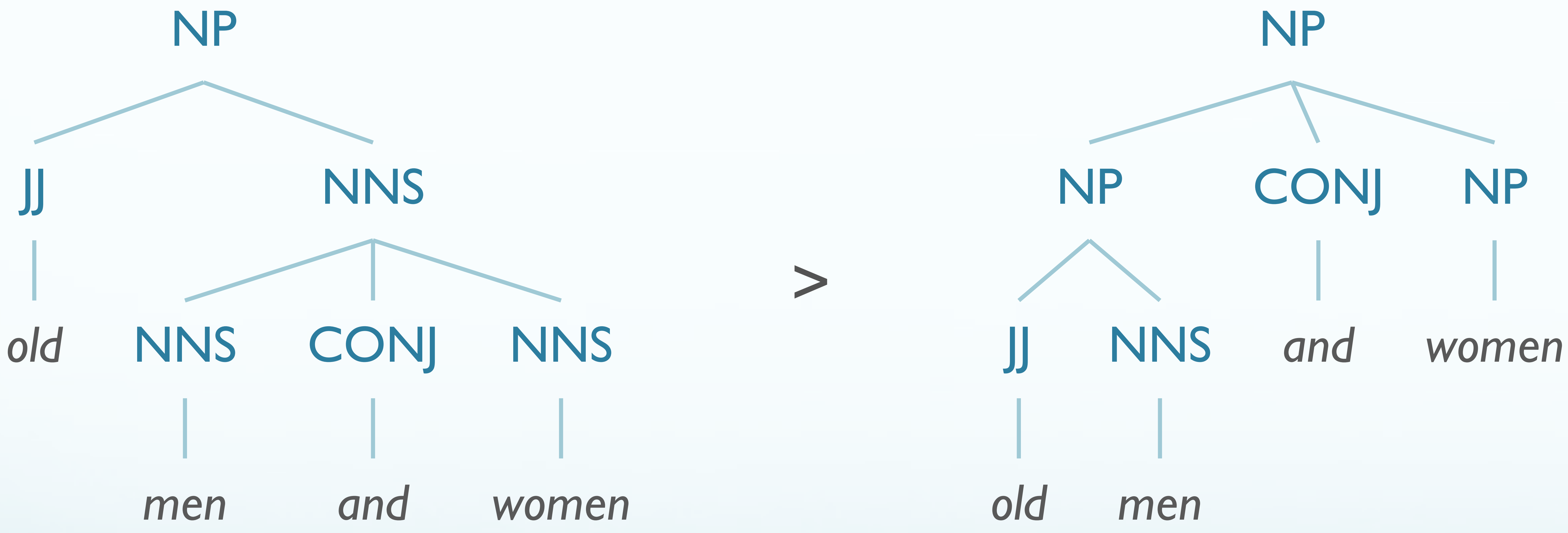
Disambiguation Strategy: Statistical

- Some prepositional structs more likely to attach high/low
  - *John was thought to have been seen by Mary*
    - Mary could be doing the seeing or thinking — seeing more likely



Disambiguation Strategy: **Statistical**

- Some phrases more likely overall
  - *[old [men and women]] is a more common construction than [old men] and [women]*



Disambiguation Strategy: **Semantic**

- Some interpretations we know to be semantically impossible
  - *Eiffel tower* as subject of *fly*

Disambiguation Strategy: **Pragmatic**

- Some interpretations are possible, unlikely given world knowledge
  - e.g. elephants and pajamas

Disambiguation Strategy:

🤷‍♂️

- Alternatively, keep all parses
  - *(Might even be the appropriate action for some jokes)*

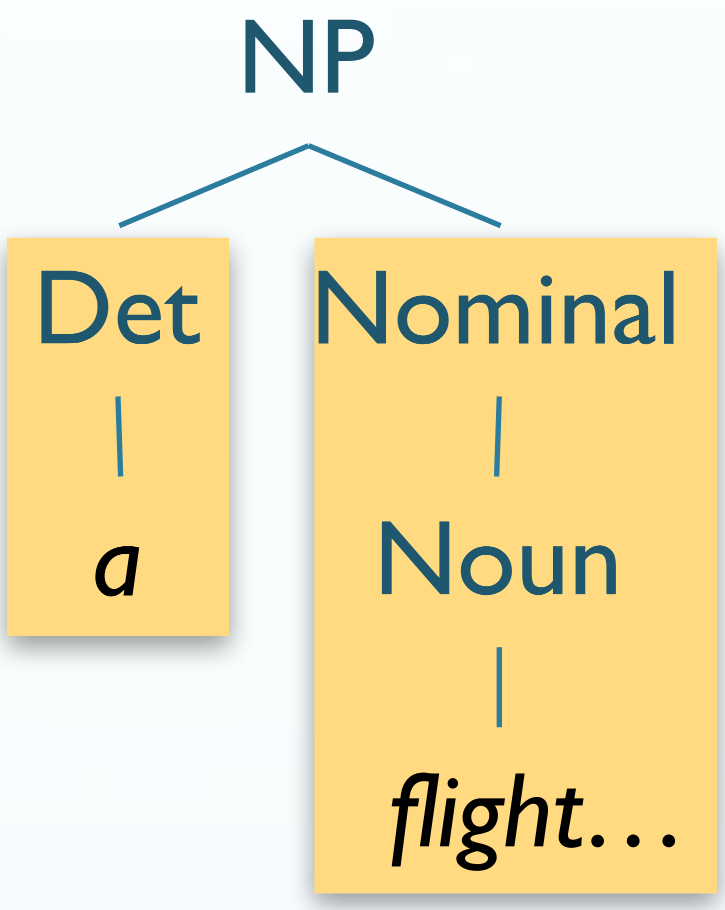
Parsing Challenges

- Recap: Parsing-as-Search
- **Parsing Challenges**
  - Ambiguity
  - **Repeated Substructure**
  - Recursion
- Strategy: Dynamic Programming
- Grammar Equivalence
- CKY parsing algorithm

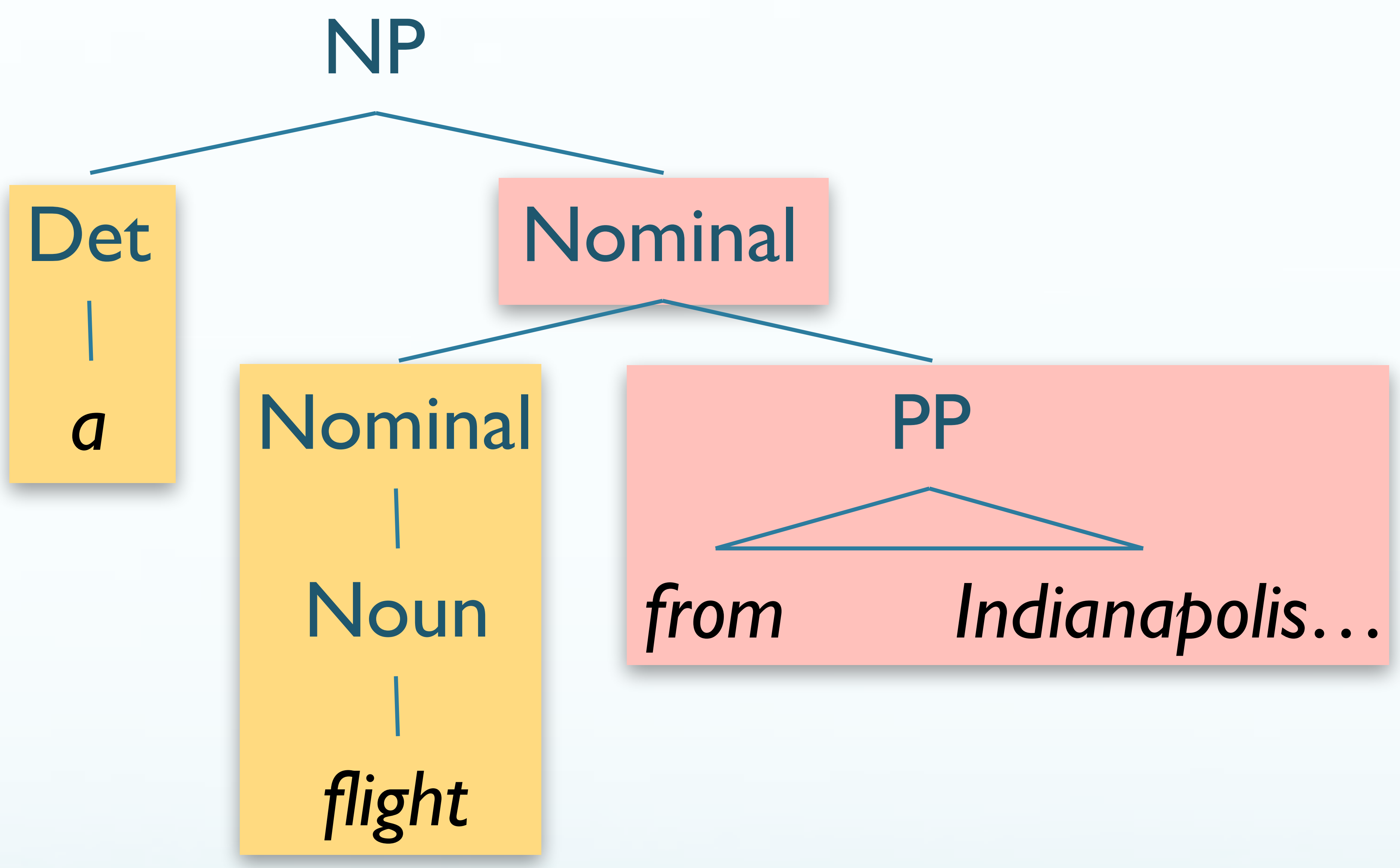
Repeated Work

- Search (top-down/bottom-up) both lead to repeated substructures
  - Globally bad parses can construct good subtrees
  - …will reconstruct along another branch
  - No static backtracking can avoid
- Efficient parsing techniques require storage of partial solutions
- Example: *a flight from Indianapolis to Houston on TWA*

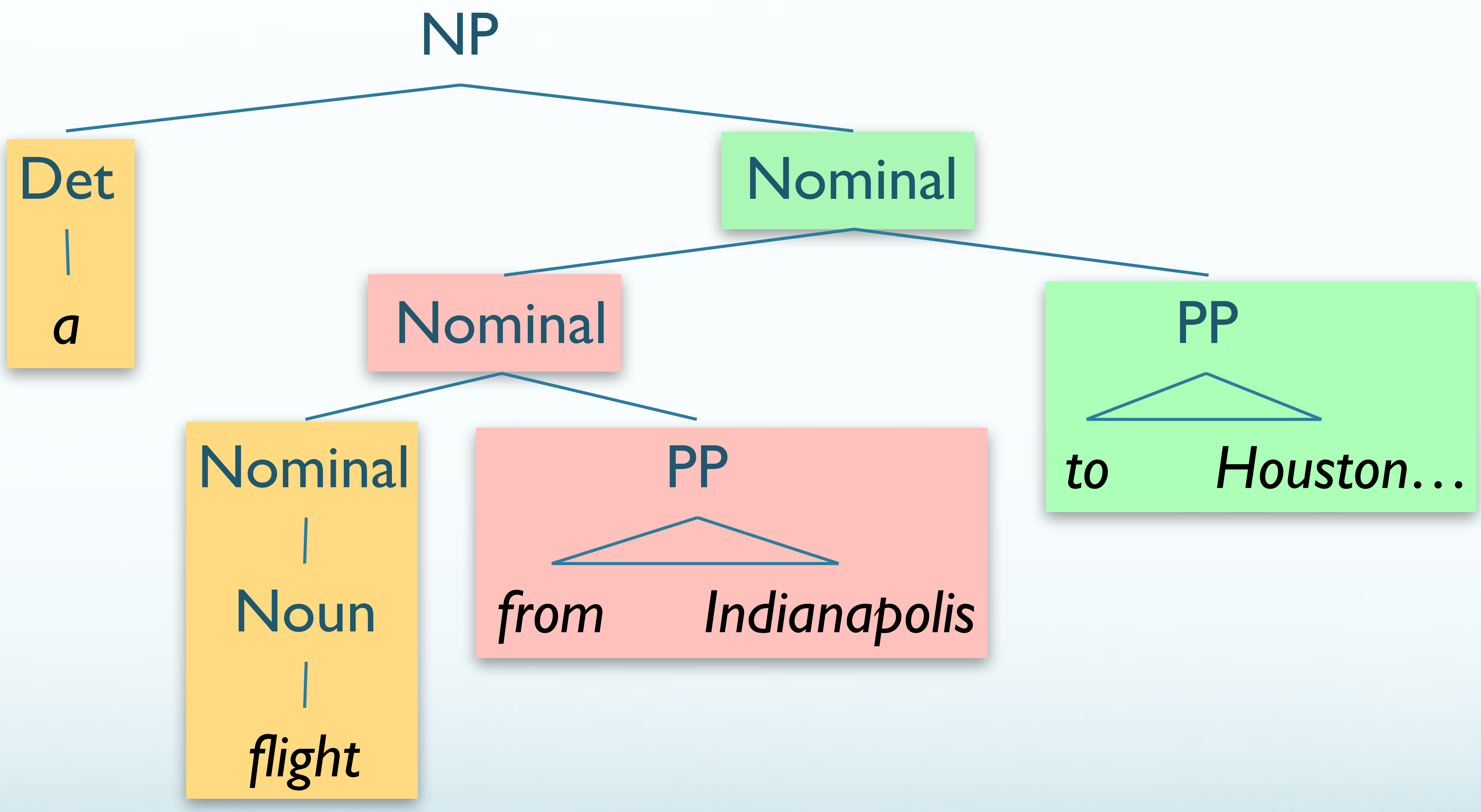
Shared Sub-Problems



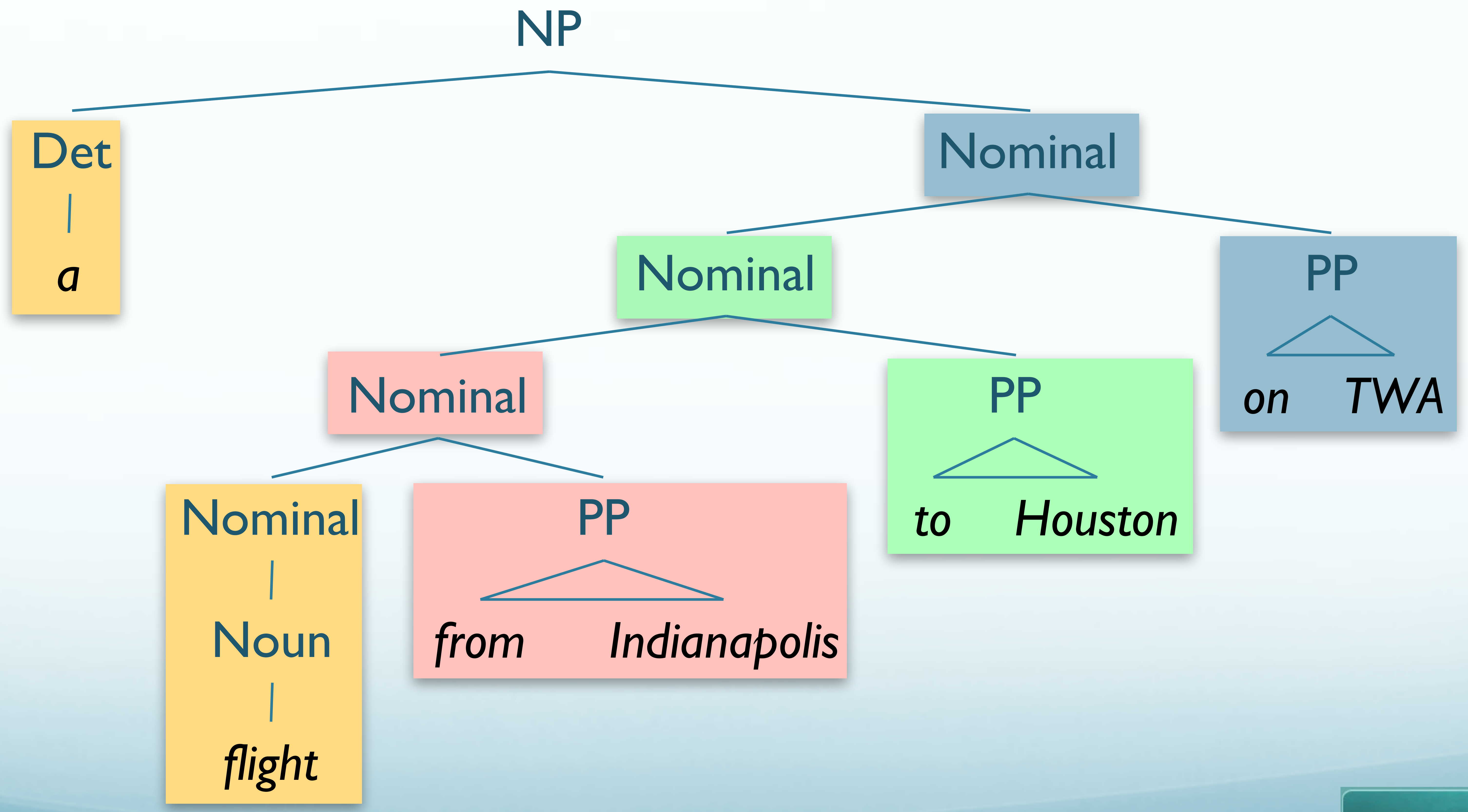
Shared Sub-Problems



Shared Sub-Problems



Shared Sub-Problems



Parsing Challenges

- Recap: Parsing-as-Search
- **Parsing Challenges**
  - Ambiguity
  - Repeated Substructure
  - **Recursion**
- Strategy: Dynamic Programming
- Grammar Equivalence
- CKY parsing algorithm

Recursion

- Many grammars have recursive rules
  - $S \rightarrow S \ Conj \ S$
- In search approaches, recursion is problematic
  - Can yield infinite searches
  - Top-down especially vulnerable

Roadmap

- Recap: Parsing-as-Search
- Parsing Challenges
- **Strategy: Dynamic Programming**
- Grammar Equivalence
- CKY parsing algorithm

Dynamic Programming

- Challenge:
  - Repeated substructure → Repeated Work
- Insight:
  - Global parse composed of sub-parses
  - Can record these sub-parses and re-use
- Dynamic programming avoids repeated work by recording the subproblems
  - Here, stores subtrees

Parsing w/Dynamic Programming

- Avoids repeated work
- Allows implementation of (relatively) efficient parsing algorithms
  - Polynomial time in input length
  - Typically cubic (n^3) or less
- Several different implementations
  - Cocke-Kasami-Younger (CKY) algorithm
  - Earley algorithm
  - Chart parsing

Roadmap

- Recap: Parsing-as-Search
- Parsing Challenges
- Strategy: Dynamic Programming
- **Grammar Equivalence**
- CKY parsing algorithm

Grammar Equivalence and Form

- ***Weak*** Equivalence
  - **Accepts** same language
  - May produce **different** structures

- ***Strong*** Equivalence
  - Accepts same language
  - Produces **same** structures

Grammar Equivalence and Form

- Reason?
  - We can create a weakly-equivalent grammar that allows for greater efficiency
  - This is required by the CKY algorithm

Chomsky Normal Form (CNF)

- Required by CKY Algorithm
- All productions are of the form:
  - $A \rightarrow B\ C$
  - $A \rightarrow \mathbf{a}$
- Most of our grammars are not of this form:
  - $S \rightarrow \textit{Wh-NP}\ \textit{Aux}\ \textit{NP}\ \textit{VP}$
- Need a general conversion procedure

CNF Conversion

1) Hybrid productions:

$$INF\text{-}VP \rightarrow \textbf{to}\ VP$$

2) Unit productions:

$$A \rightarrow B$$

3) Long productions:

$$A \rightarrow B\ C\ D\ \ldots$$

CNF Conversion: Hybrid Productions

- Hybrid production:
  - Replace all terminals with dummy non-terminal
  - $INF\text{-}VP \rightarrow \textbf{to}\ VP$
    - $INF\text{-}VP \rightarrow TO\ VP$
    - $TO \rightarrow \textbf{to}$

CNF Conversion: Unit Productions

- Unit productions:
  - Rewrite RHS with RHS of all derivable, non-unit productions
  - If $A \overset{*}{\Rightarrow} B$ and $B \rightarrow \mathbf{w}$, add $A \rightarrow \mathbf{w}$
- $Nominal \rightarrow Noun,\ Noun \rightarrow \mathbf{dog}$
  - $Nominal \rightarrow \mathbf{dog}$
  - $Noun \rightarrow \mathbf{dog}$

CNF Conversion: Long Productions

- Long productions
  - Introduce unique nonterminals, and spread over rules

$S \rightarrow Aux\ NP\ VP$	
$S \rightarrow \boldsymbol{X1}\ VP$	$\boldsymbol{X1} \rightarrow Aux\ NP$

CNF Conversion

1) Convert terminals in hybrid rules to dummy non-terminals

2) Convert unit productions

3) Binarize long production rules

$\mathscr{L}_1$ **Grammar**	$\mathscr{L}_1$ **in CNF**
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

$\mathscr{L}_1$ **Grammar**	$\mathscr{L}_1$ **in CNF**
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

$\mathscr{L}_1$ **Grammar**	$\mathscr{L}_1$ **in CNF**
$S \rightarrow NP\ VP$	$S \rightarrow NP\ VP$
$S \rightarrow Aux\ NP\ VP$	$S \rightarrow X1\ VP$
	$X1 \rightarrow Aux\ NP$
$S \rightarrow VP$	$S \rightarrow book \mid include \mid prefer$
	$S \rightarrow Verb\ NP$
	$S \rightarrow X2\ PP$
	$S \rightarrow Verb\ PP$
	$S \rightarrow VP\ PP$
$NP \rightarrow Pronoun$	$NP \rightarrow I \mid she \mid me$
$NP \rightarrow Proper\text{-}Noun$	$NP \rightarrow TWA \mid Houston$
$NP \rightarrow Det\ Nominal$	$NP \rightarrow Det\ Nominal$
$Nominal \rightarrow Noun$	$Nominal \rightarrow book \mid flight \mid meal \mid money$
$Nominal \rightarrow Nominal\ Noun$	$Nominal \rightarrow Nominal\ Noun$
$Nominal \rightarrow Nominal\ PP$	$Nominal \rightarrow Nominal\ PP$
$VP \rightarrow Verb$	$VP \rightarrow book \mid include \mid prefer$
$VP \rightarrow Verb\ NP$	$VP \rightarrow Verb\ NP$
$VP \rightarrow Verb\ NP\ PP$	$VP \rightarrow X2\ PP$
	$X2 \rightarrow Verb\ NP$
$VP \rightarrow Verb\ PP$	$VP \rightarrow Verb\ PP$
$VP \rightarrow VP\ PP$	$VP \rightarrow VP\ PP$
$PP \rightarrow Preposition\ NP$	$PP \rightarrow Preposition\ NP$

Roadmap

- Recap: Parsing-as-Search
- Parsing Challenges
- Strategy: Dynamic Programming
- Grammar Equivalence
- **CKY parsing algorithm**

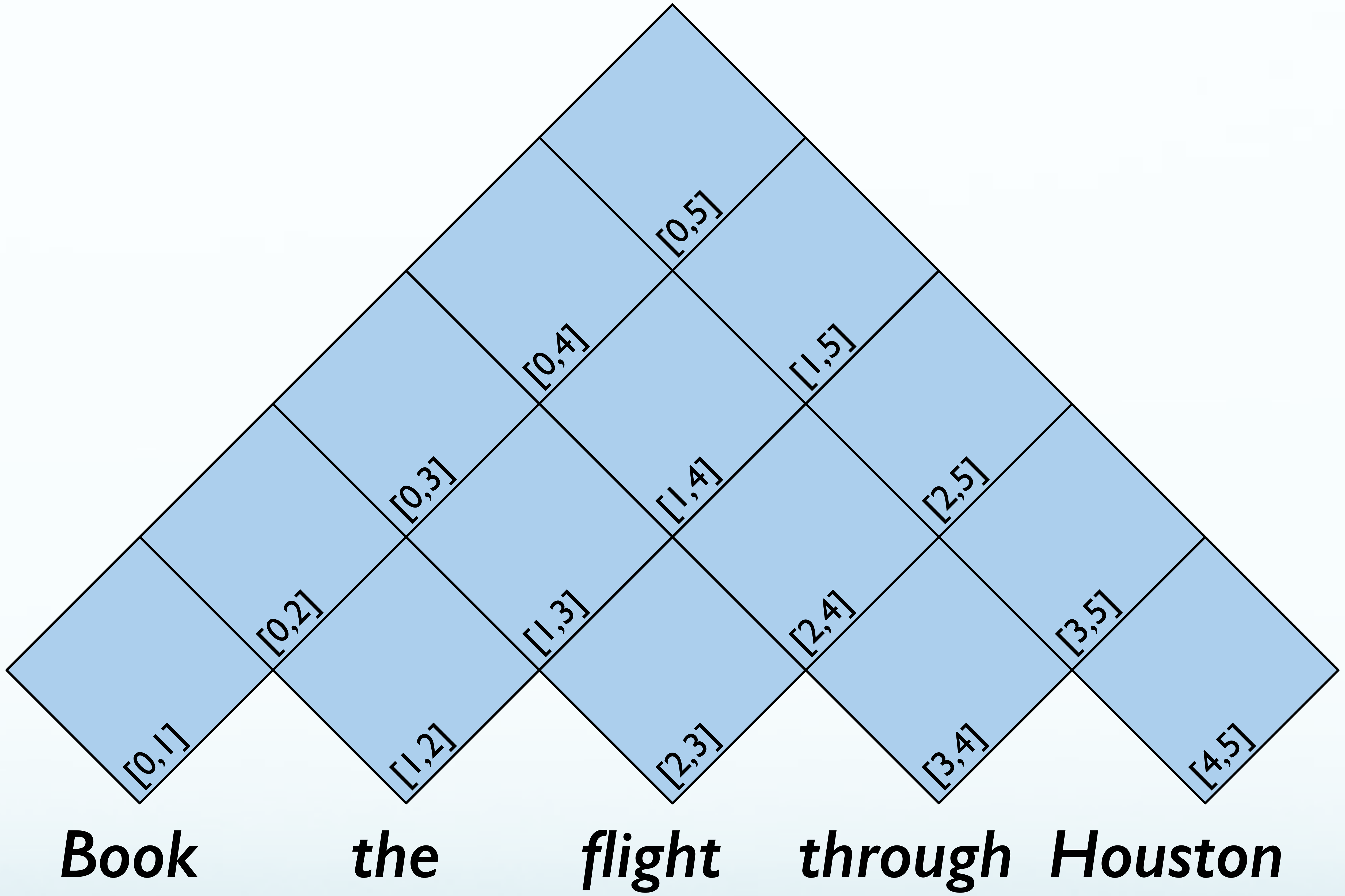
CKY Parsing

- (Relatively) efficient bottom-up parsing algorithm
- Based on tabulating substring parses to avoid repeat work
- Approach:
  - Use CNF Grammar
  - Build an $(n+1) \times (n+1)$ matrix to store subtrees
    - Upper triangular portion
  - Incrementally build parse spanning whole input string

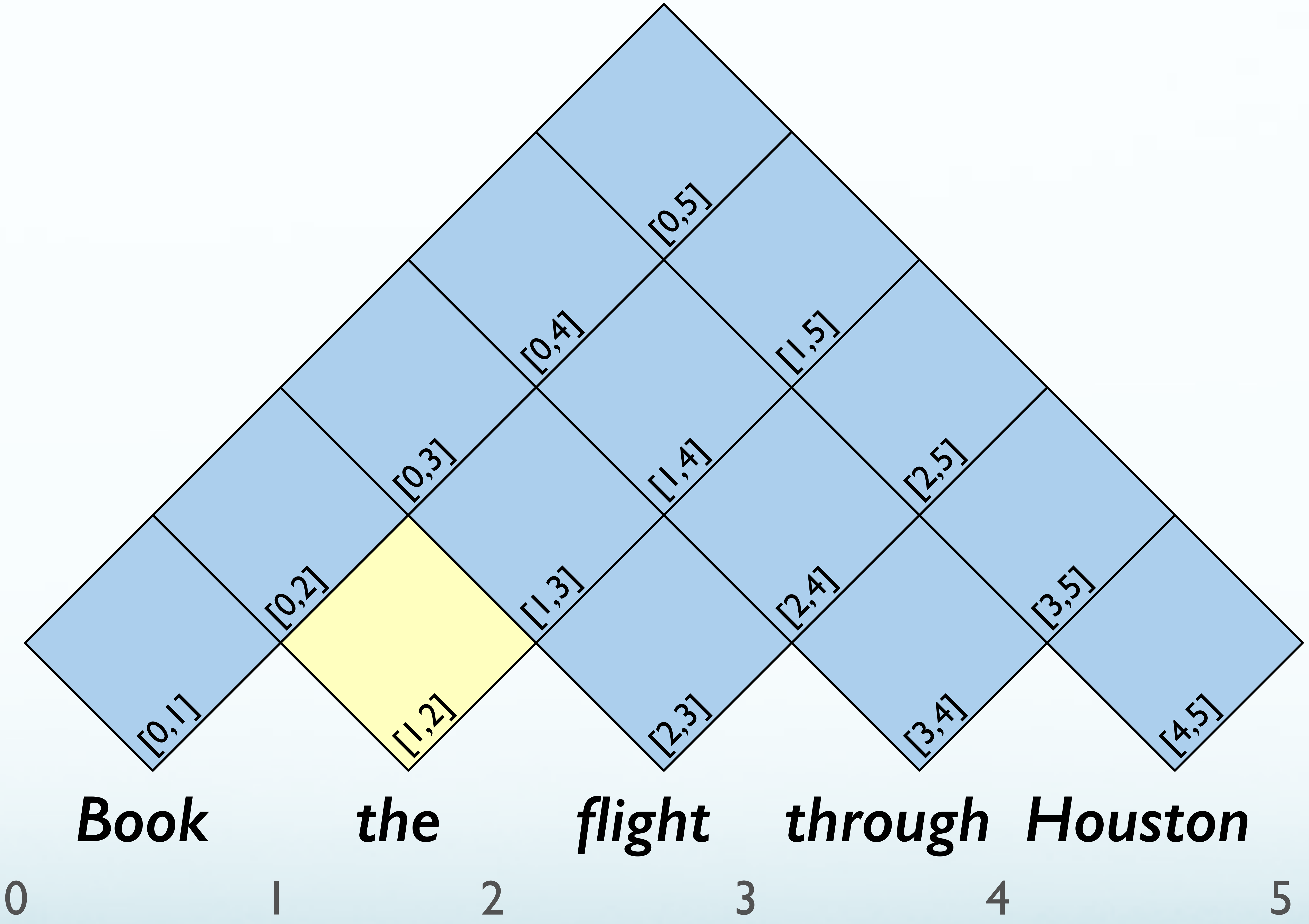
CKY Matrix

Book	*the*	*flight*	*through*	*Houston*
[0,1]	[0,2]	[0,3]	[0,4]	[0,5]
	[1,2]	[1,3]	[1,4]	[1,5]
		[2,3]	[2,4]	[2,5]
			[3,4]	[3,5]
				[4,5]

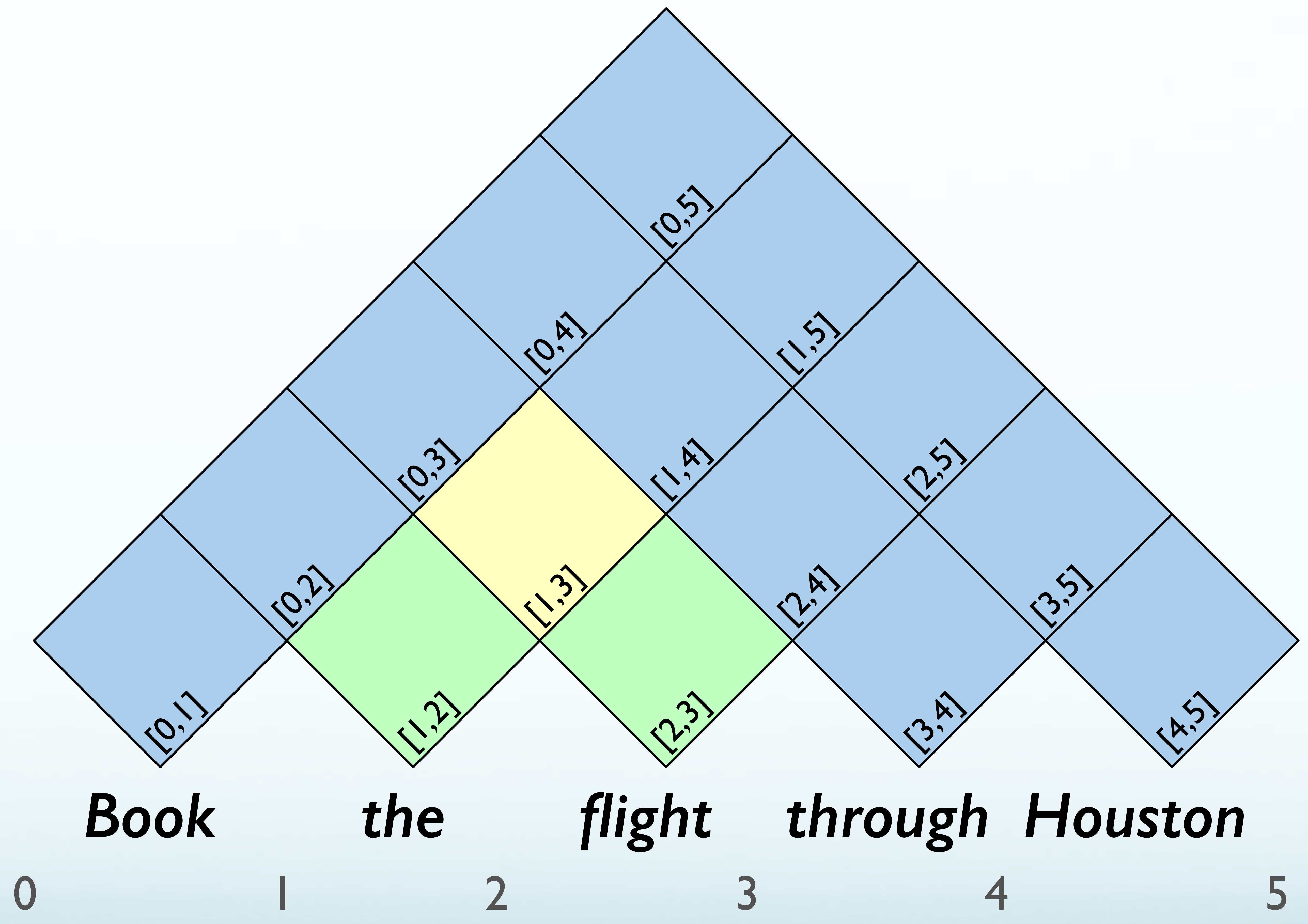
CKY Matrix


[0,5]
[0,4]
[1,5]
[0,3]
[1,4]
[2,5]
[0,2]
[1,3]
[2,4]
[3,5]
[0,1]
[1,2]
[2,3]
[3,4]
[4,5]
Book
the
flight
through
Houston

CKY Matrix


[0,5]
[0,4]
[1,5]
[0,3]
[1,4]
[2,5]
[0,2]
[1,3]
[2,4]
[3,5]
[0,1]
[1,2]
[2,3]
[3,4]
[4,5]
Book
the
flight
through
Houston
0
1
2
3
4
5

CKY Matrix



[0,5]

[0,4] [1,5]

[0,3] [1,4] [2,5]

[0,2] [1,3] [2,4] [3,5]

[0,1] [1,2] [2,3] [3,4] [4,5]

Book the flight through Houston

0 1 2 3 4 5

Dynamic Programming in CKY

- Key idea:
  - for $i < k < j$
  - …and a parse spanning substring [i, j]
  - $\exists k$ such that there are parses spanning [i, k] and [k, j]
  - We can construct parses for whole sentences by building from these partial parses
- So to have a rule $A \rightarrow B\ \ C$ in [i, j]
  - Must have B in [i, j] and C in [k, j] for some $i < k < j$
  - CNF forces this for all $j > i + 1$

HW #2

LING 571
Deep Processing Techniques for NLP
January 10, 2018

Goals

- Begin development of CKY parser

- First stage: Conversion to CNF
  - Develop Representation for CFG
  - Manipulate/Transform Grammars
  - Investigate weakly equivalent grammars

Task

- Conversion:
  - Read in grammar rules from arbitrary CFG
  - Convert to CNF
  - Write out new grammar
- Validation:
  - Parse test sentences with original CFG
  - Parse test sentences with CFG in CNF

Approach

- May use any programming language
  - In keeping with course policies
- May use existing models/packages to represent rules
  - Need RULE, RHS, LHS, etc
  - NLTK, Stanford
- ***Conversion code must be your own***

Data

- ATIS (Air Travel Information System) data
  - Grammar provided in nltk-data
  - Terminals in double-quotes
    - $the \rightarrow$ "the"
  - All required files on patas dropbox
- **NOTE**:
  - Grammar is fairly large (~193K Productions)
  - Grammar is fairly ambiguous (Test sentences may have 100 parses)
  - You will likely want to develop against a smaller grammar

NLTK Grammars

```
>>> gr1 = nltk.data.load('grammars/large_grammars/atis.cfg')

>>> gr1.productions()[0]
ABBCL_NP -> QUANP_DTI QUANP_DTI QUANP_CD AJP_JJ NOUN_NP
PRPRTCL_VBG

>>> gr1.productions()[0].lhs()
ABBCL_NP

>>> gr1.productions(lhs=gr1.productions()[1].lhs())
[ADJ_ABL -> only, ADJ_ABL->such]
```