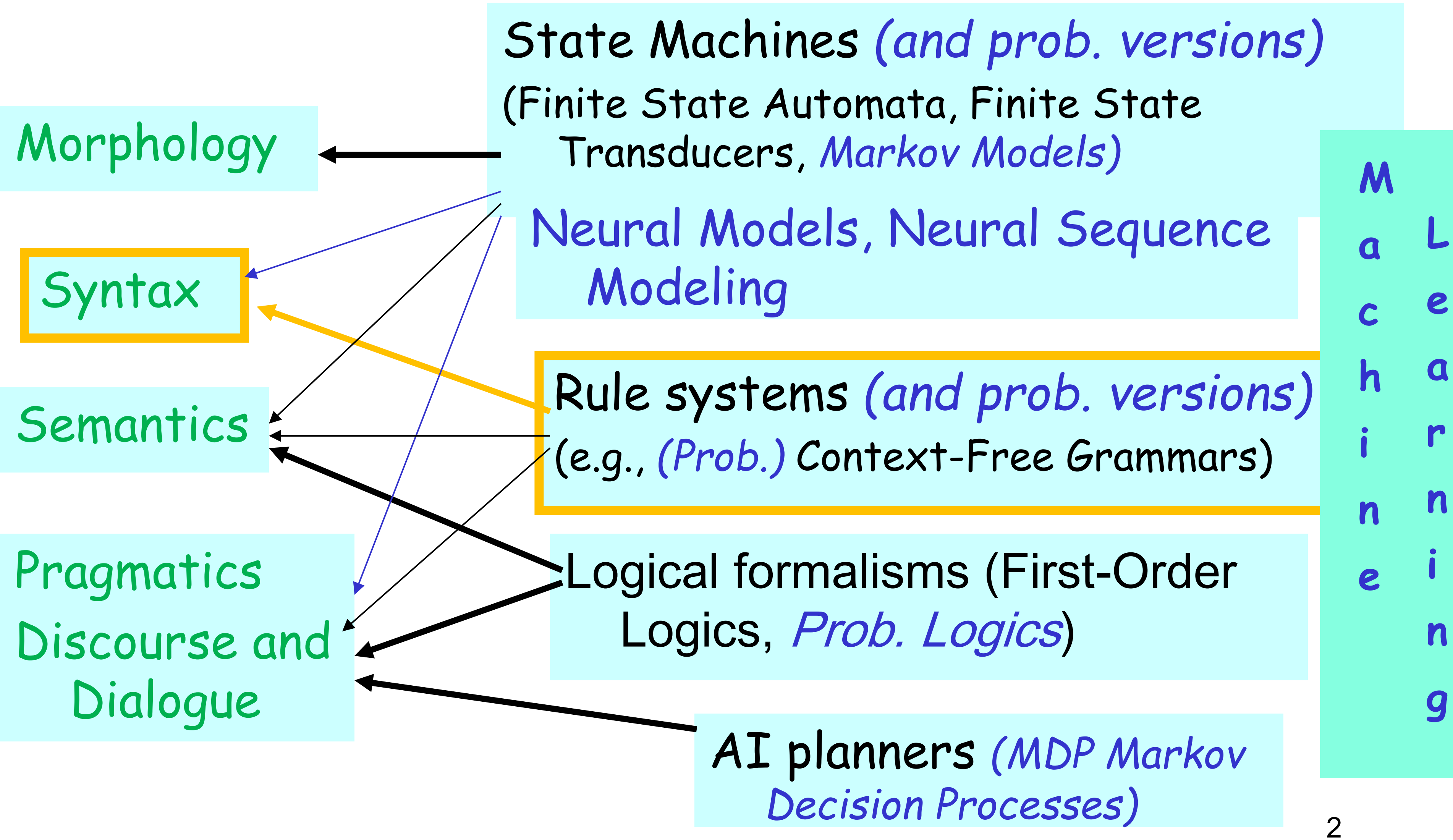
Intelligent Systems (AI-2)

Computer Science cpsc422, Lecture 25

NLP: Knowledge-Formalisms Map
(including probabilistic formalisms)



NLP Practical Goal for FOL: the ultimate Web question-answering system?

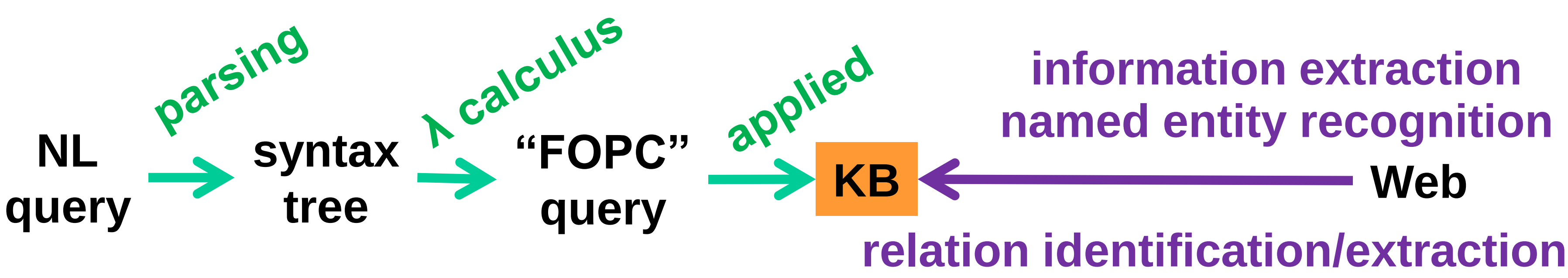
Map NL queries and the Web into FOL so that answers can be effectively computed

- *What African countries are not on the Mediterranean Sea?*

$$\exists c\ Country(c) \wedge \neg Borders(c, Med.Sea) \wedge In(c, Africa)$$

- *Was 2007 the first El Nino year after 2001?*

$$ElNino(2007) \wedge \neg \exists y\ Year(y) \wedge After(y, 2001) \wedge Before(y, 2007) \wedge ElNino(y)$$



Learning Goals for today's class

You can:

- Explain what is the syntax of a Natural Language
- Formally define a Context Free Grammar
- Justify why a CFG is a reasonable model for the English Syntax
- Apply a CFG as a Generative Formalism to
  - Impose structures (trees) on strings in the language (i.e. Trace Top-down and Bottom-up parsing on sentence given a grammar)
  - Reject strings not in the language (also part of parsing)
  - Generate strings in the language given a CFG

Lecture Outline

- English Syntax
- Context Free Grammars
- Parsing

Syntax of Natural Languages

Def. The study of how sentences are formed by grouping and ordering words

Part of speech: Noun, Verb….

It is so …… The…… is

Example:

Ming and Sue prefer morning flights

*** Ming Sue flights morning and prefer**

Groups behave as single unit with respect to

- **Substitution** *they, it, do so*
- **Movement: passive, question**
- **Coordination** …. *and* ……

Syntax: Useful tasks

- Why should you care?
  - Grammar checkers
  - Basis for semantic interpretation
    - Question answering
    - Information extraction
    - Summarization
  - Discourse Parsing
  - Machine translation
  - ……

Key Constituents: Examples

Noun phrases	(Det)	N	(PP)
	the	cat	on the table
Verb phrases	(Qual)	V	(NP)
	never	eat	a cat
Prepositional phrases	(Deg)	P	(NP)
	almost	in	the net
Adjective phrases	(Deg)	A	(PP)
	very	happy	about it
Sentences	(NP)	(-)	(VP)
	a mouse	--	ate it

Context Free Grammar (Example)

Start-symbol

Non-terminal

Terminal

- S -> NP VP
- NP -> Det NOMINAL
- NOMINAL -> Noun
- VP -> Verb
- Det -> *a*
- Noun -> *flight*
- Verb -> *left*

- Backbone of many models of syntax
- Parsing is tractable

CFG more complex Example

Grammar with example phrases

S	$\rightarrow$	$NP\ VP$	I + want a morning flight
NP	$\rightarrow$	*Pronoun*	I
	$\mid$	*Proper-Noun*	Los Angeles
	$\mid$	*Det Nominal*	a + flight
Nominal	$\rightarrow$	*Noun Nominal*	morning + flight
	$\mid$	*Noun*	flights
VP	$\rightarrow$	*Verb*	do
	$\mid$	*Verb NP*	want + a flight
	$\mid$	*Verb NP PP*	leave + Boston + in the morning
	$\mid$	*Verb PP*	leaving + on Thursday
PP	$\rightarrow$	*Preposition NP*	from + Los Angeles

Lexicon

Noun	$\rightarrow$	*flights* | *breeze* | *trip* | *morning* | ...
Verb	$\rightarrow$	*is* | *prefer* | *like* | *need* | *want* | *fly*
Adjective	$\rightarrow$	*cheapest* | *non – stop* | *first* | *latest* | *other* | *direct* | ...
Pronoun	$\rightarrow$	*me* | *I* | *you* | *it* | ...
Proper-Noun	$\rightarrow$	*Alaska* | *Baltimore* | *Los Angeles* | *Chicago* | *United* | *American* | ...
Determiner	$\rightarrow$	*the* | *a* | *an* | *this* | *these* | *that* | ...
Preposition	$\rightarrow$	*from* | *to* | *on* | *near* | ...
Conjunction	$\rightarrow$	*and* | *or* | *but* | ...

CFGs

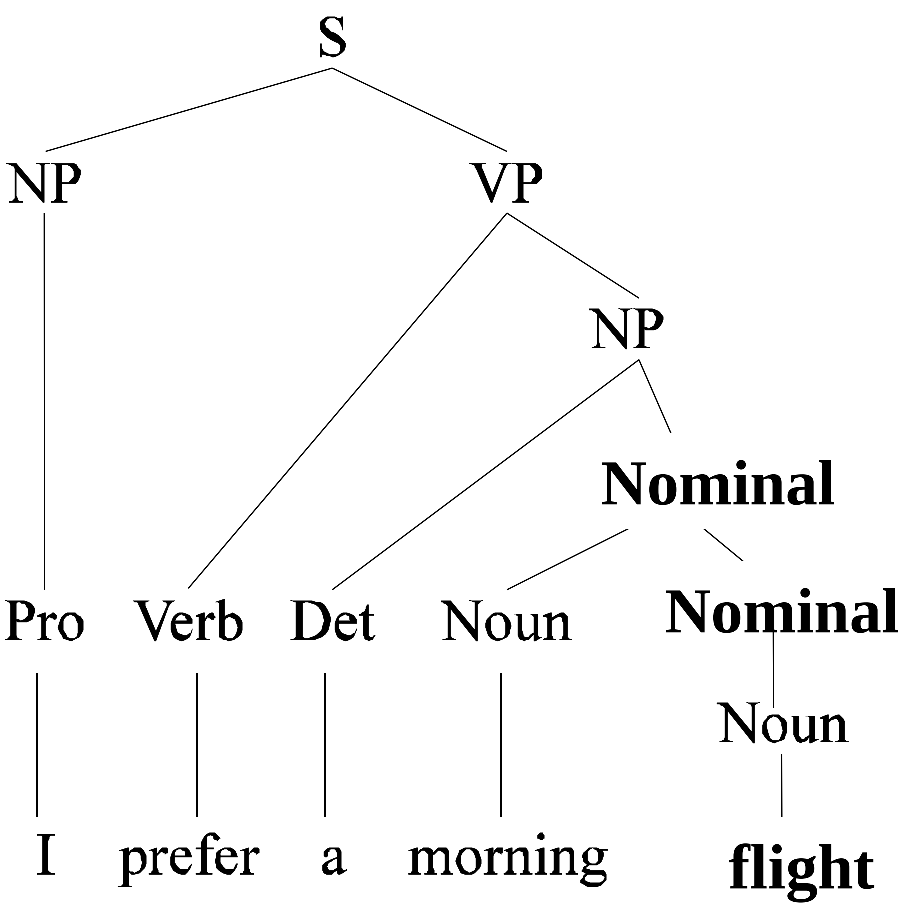
- Define a Formal Language (un/grammatical sentences)

- Generative Formalism
  - Generate strings in the language
  - Reject strings not in the language
  - Impose structures (trees) on strings in the language

CFG: Formal Definitions

- **4-tuple** (non-term., term., productions, start)
- **(N, Σ, P, S)**
- P is a set of rules $A \rightarrow \alpha$; $A \in N$, $\alpha \in (\Sigma \cup N)^*$
- **A derivation is the process of rewriting α_1 into α_m (both strings in $(\Sigma \cup N)^*$) by applying a sequence of rules**

Derivations as Trees



$$S \rightarrow NP\ VP$$

$$
\begin{aligned}
NP &\rightarrow Pronoun \\
&\mid\ Proper\text{-}Noun \\
&\mid\ Det\ Nominal \\
\text{Nominal} &\rightarrow Noun\ Nominal \\
&\mid\ Noun
\end{aligned}
$$

$$
\begin{aligned}
VP &\rightarrow Verb \\
&\mid\ Verb\ NP \\
&\mid\ Verb\ NP\ PP \\
&\mid\ Verb\ PP
\end{aligned}
$$

$$PP \rightarrow Preposition\ NP$$

Context Free?

Common Sentence-Types

- **Declaratives:** A plane left

  S -> NP VP

- **Imperatives:** Leave!

  S -> VP

- **Yes-No Questions:** Did the plane leave?

  S -> Aux NP VP

- **WH Questions:**

  Which flights serve breakfast?

  S -> WH NP VP

  When did the plane leave?

  S -> WH Aux NP VP

Conjunctive Constructions

- S -> S and S
  - John went to NY, and Mary followed him
- NP -> NP and NP
  - John went to NY and Boston
- VP -> VP and VP
  - John went to NY and visited MOMA
- …
- In fact the right rule for English is
  X -> X and X

CFG for NLP: summary

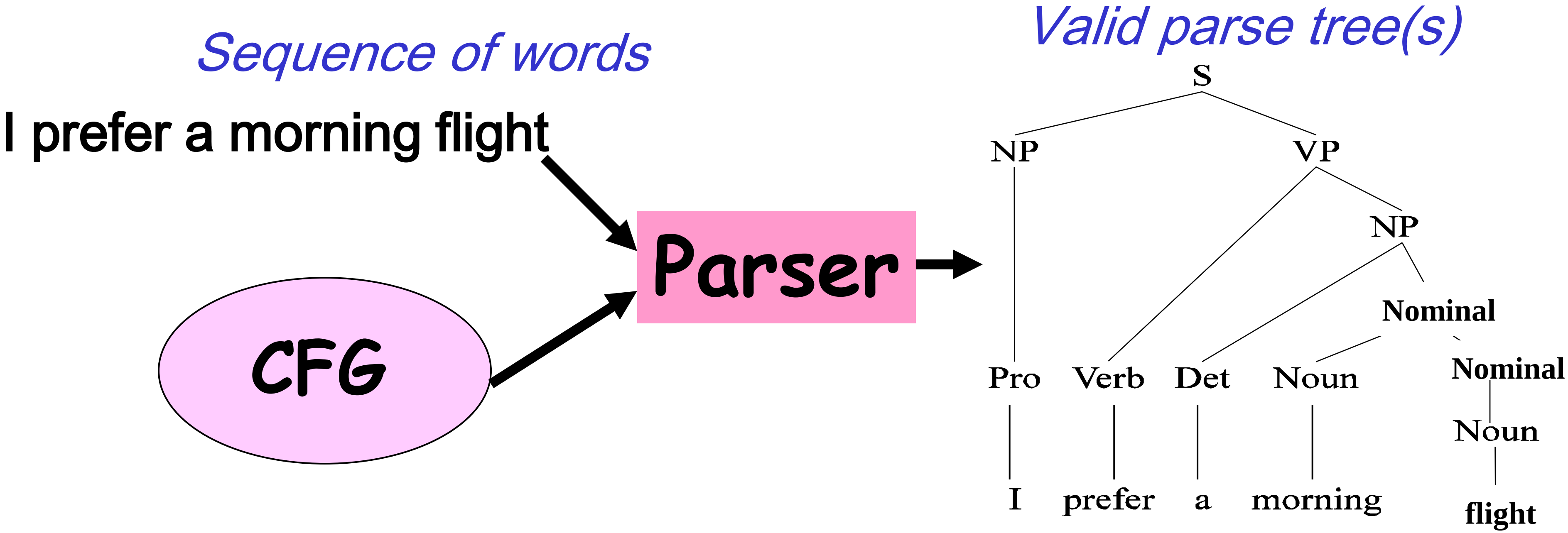
- **CFGs cover most syntactic structure in English.**

- **Many practical computational grammars simply rely on CFG**

Lecture Outline

- Context Free Grammars / English Syntax
- **Parsing**

Parsing with CFGs



Assign valid trees: covers all and only the elements of the input and has an S at the top

How many **valid** *trees are obtainable for a given input sequence?*

A. 0 B. 1 C. >1 D. any of A-C E. I'm trying to sleep here

Parsing as Search

CFG

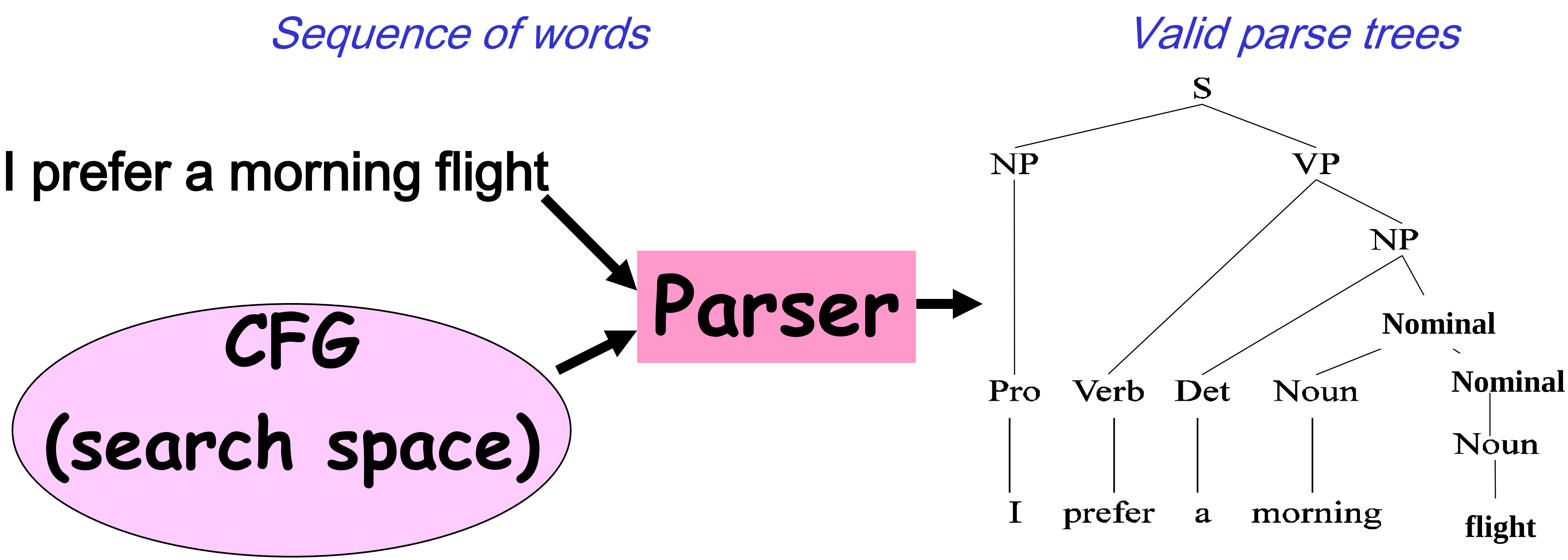
- S -> NP VP
- S -> Aux NP VP
- NP -> Det Noun
- VP -> Verb
- Det -> *a*
- Noun -> *flight*
- Verb -> *left, arrive*
- Aux -> *do, does*

defines

Search space of possible parse trees

Parsing: find all trees that cover all and only the words in the input

Constraints on Search



Search Strategies:

- **Top-down** or goal-directed
- **Bottom-up** or data-directed

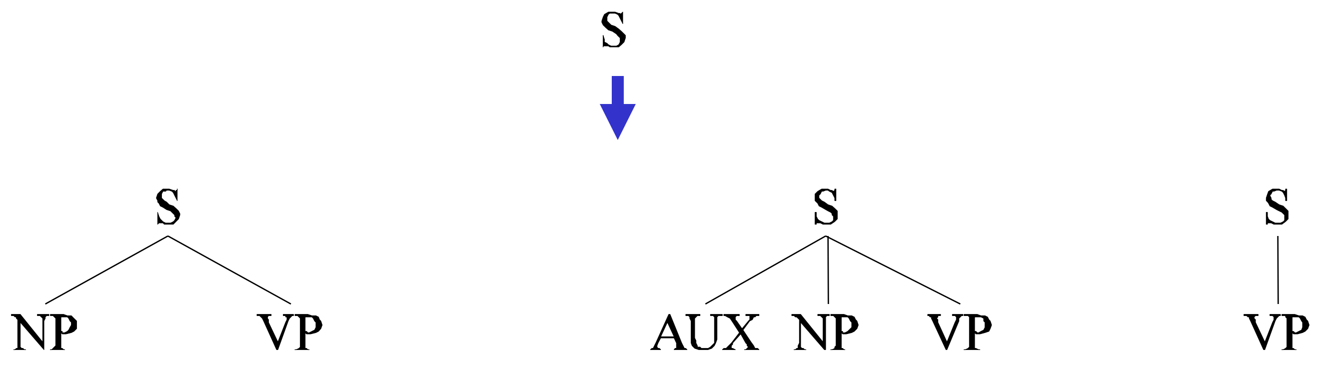
Context Free Grammar (Used in parsing Example)

$S \rightarrow NP\ VP$	$Det \rightarrow that \mid this \mid a$
$S \rightarrow Aux\ NP\ VP$	$Noun \rightarrow book \mid flight \mid meal \mid money$
$S \rightarrow VP$	$Verb \rightarrow book \mid include \mid prefer$
$NP \rightarrow Proper\text{-}Noun$	$Proper\text{-}Noun \rightarrow Houston \mid TWA$
$NP \rightarrow Det\ Nominal$	$Aux \rightarrow does$
$Nominal \rightarrow Noun$	
$Nominal \rightarrow Nominal\ Noun$	
$Nominal \rightarrow Nominal\ PP$	
$VP \rightarrow Verb$	
$VP \rightarrow Verb\ NP$	

Top-Down Parsing

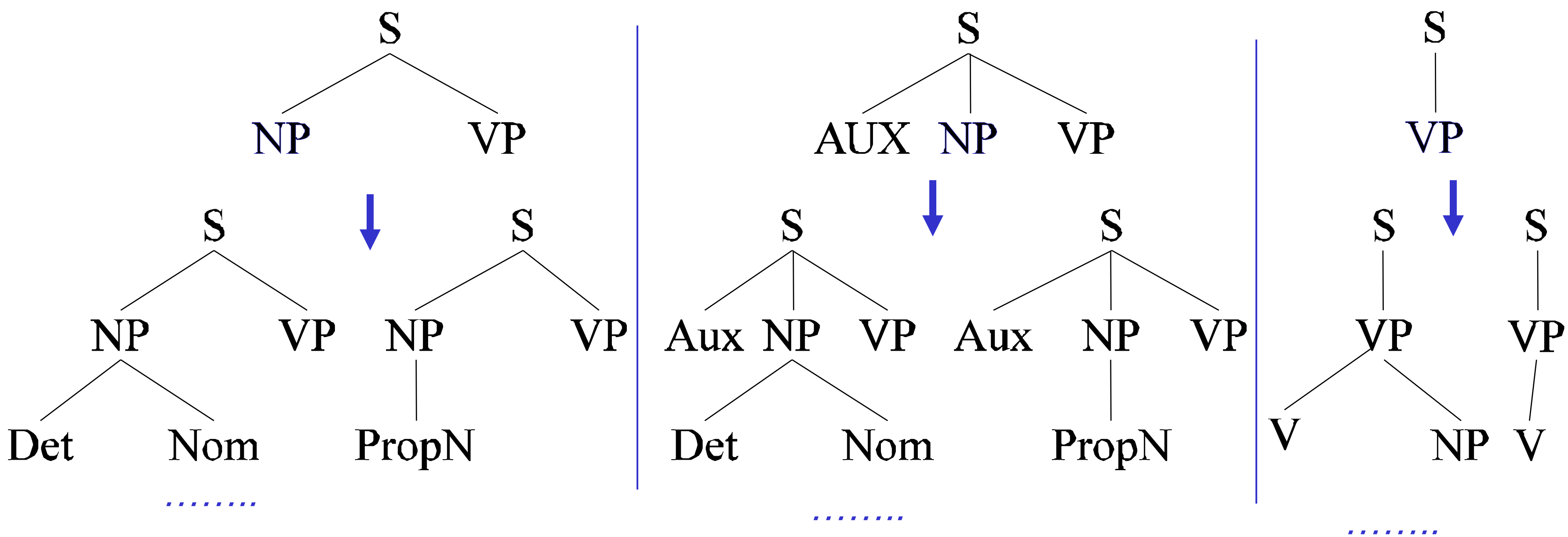
- Since we're trying to find trees rooted with S (Sentences) start with the rules that rewrite S.
- Then work your way down from there to the words.

Input: Book that **flight**



Next step: Top Down Space

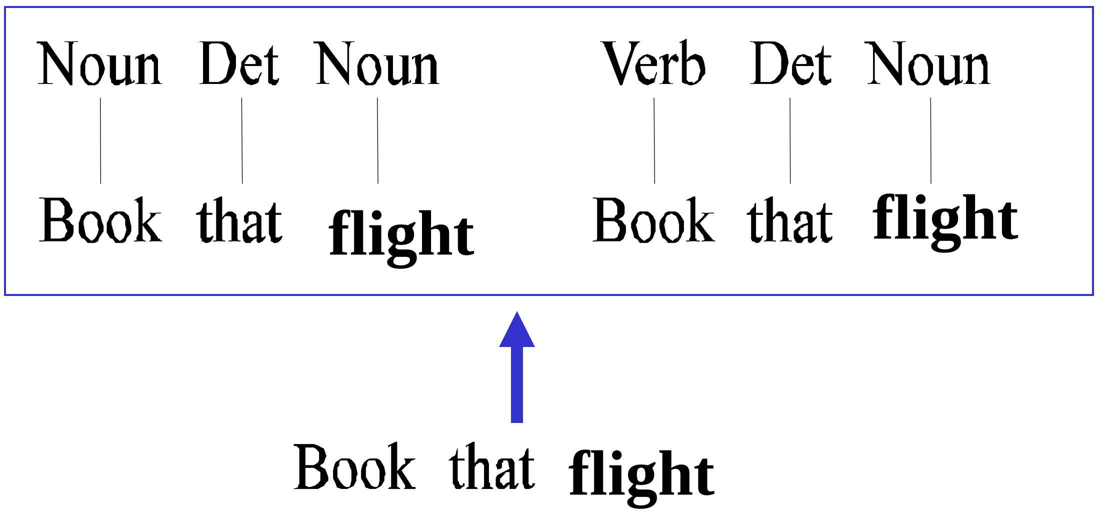
Input: Book that **flight**



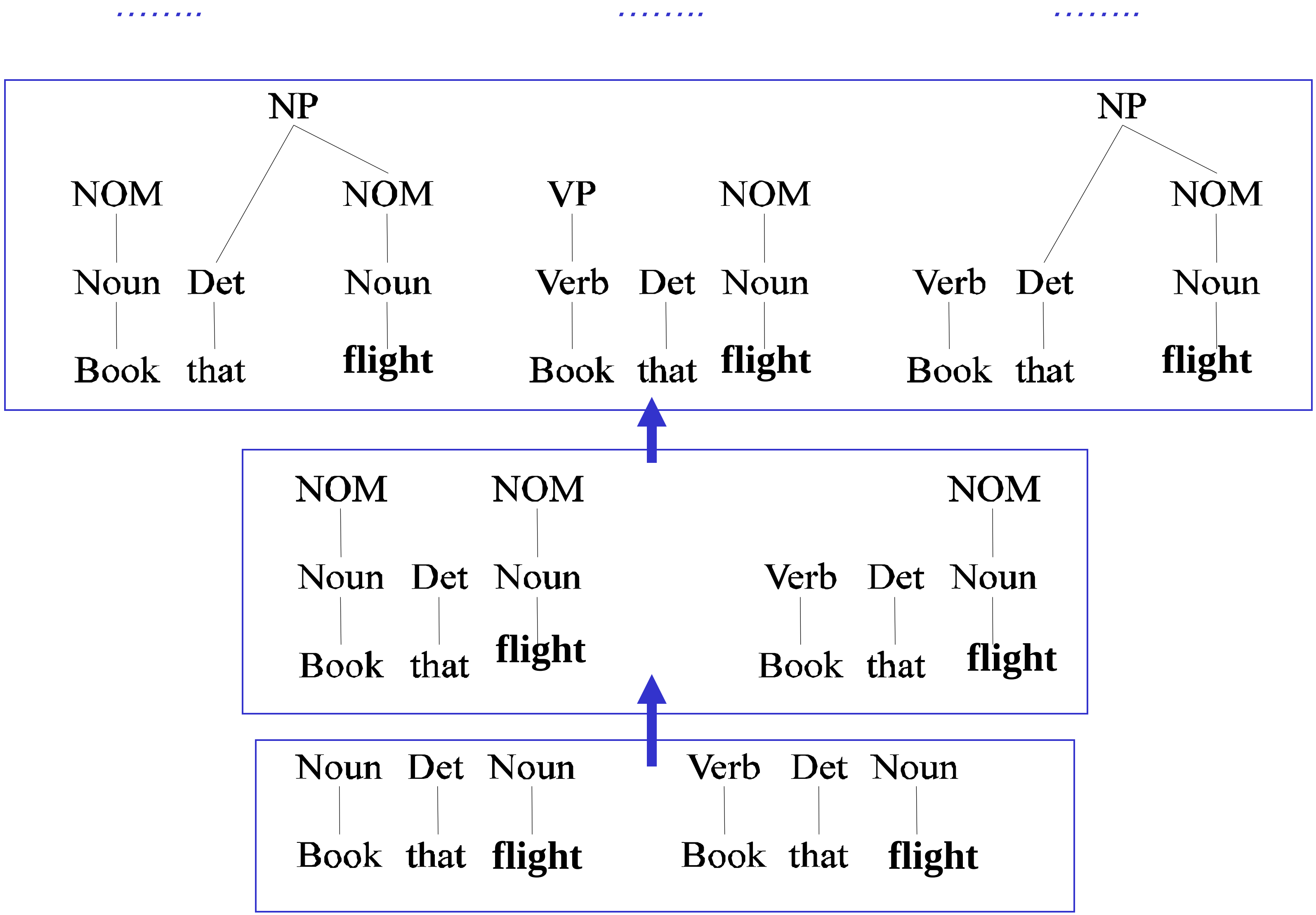
- **When POS categories are reached, reject trees whose leaves fail to match all words in the input**

Bottom-Up Parsing

- Of course, we also want trees that cover the input words. So start with trees that link up with the words in the right way.
- Then work your way up from there.



Two more steps: Bottom-Up Space



Top-Down vs. Bottom-Up

- Top-down
  - Only searches for trees that can be answers +
  - But suggests trees that are not consistent with the words −
- Bottom-up
  - Only forms trees consistent with the words +
  - Suggest trees that make no sense globally −

So Combine Them (from here to slide 35 not required for 422 - just for your interest)

- Top-down: control strategy to generate trees
- Bottom-up: to filter out inappropriate parses

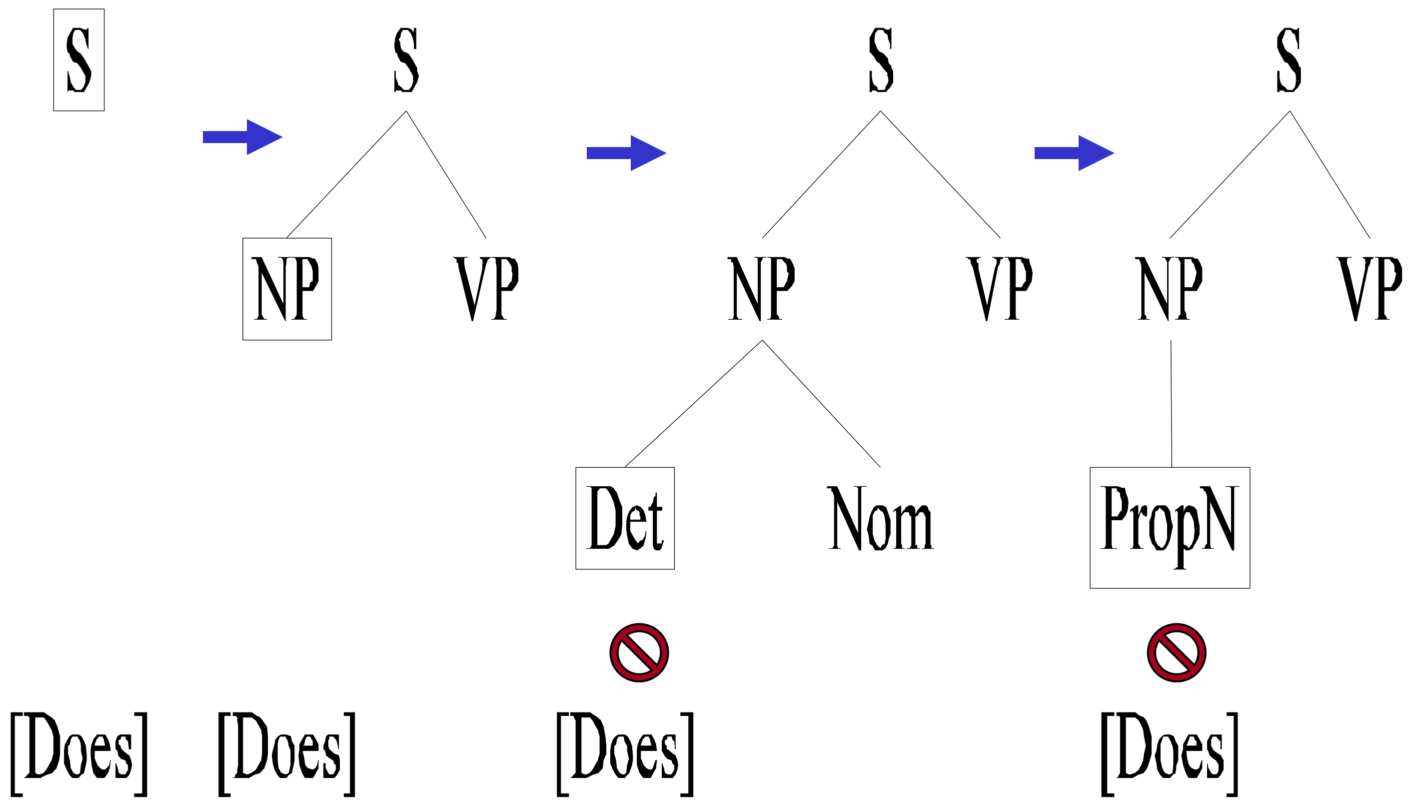
Top-down Control strategy:

- Depth vs. Breadth first
- Which node to try to expand next *(left-most)*
- Which grammar rule to use to expand a node

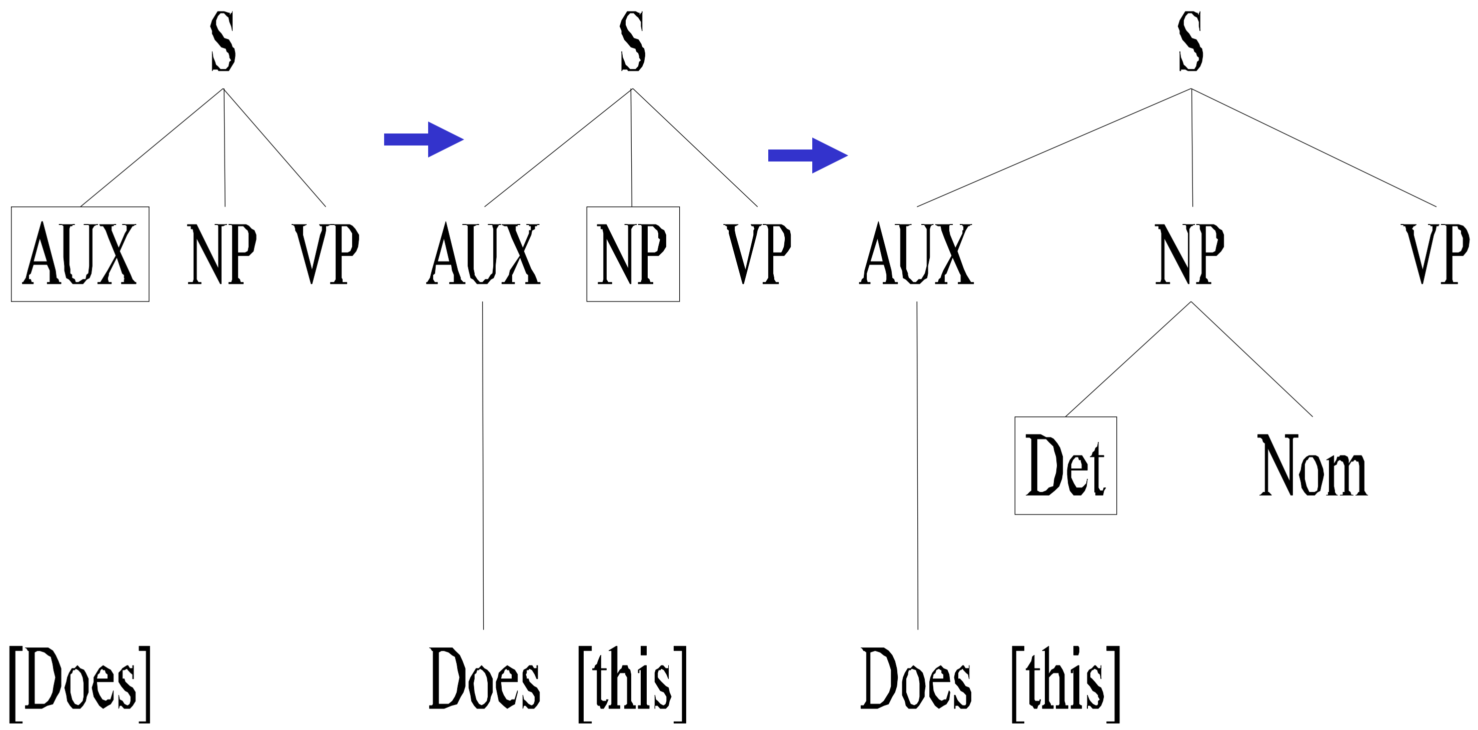
(textual order)

Top-Down, Depth-First, Left-to-Right Search

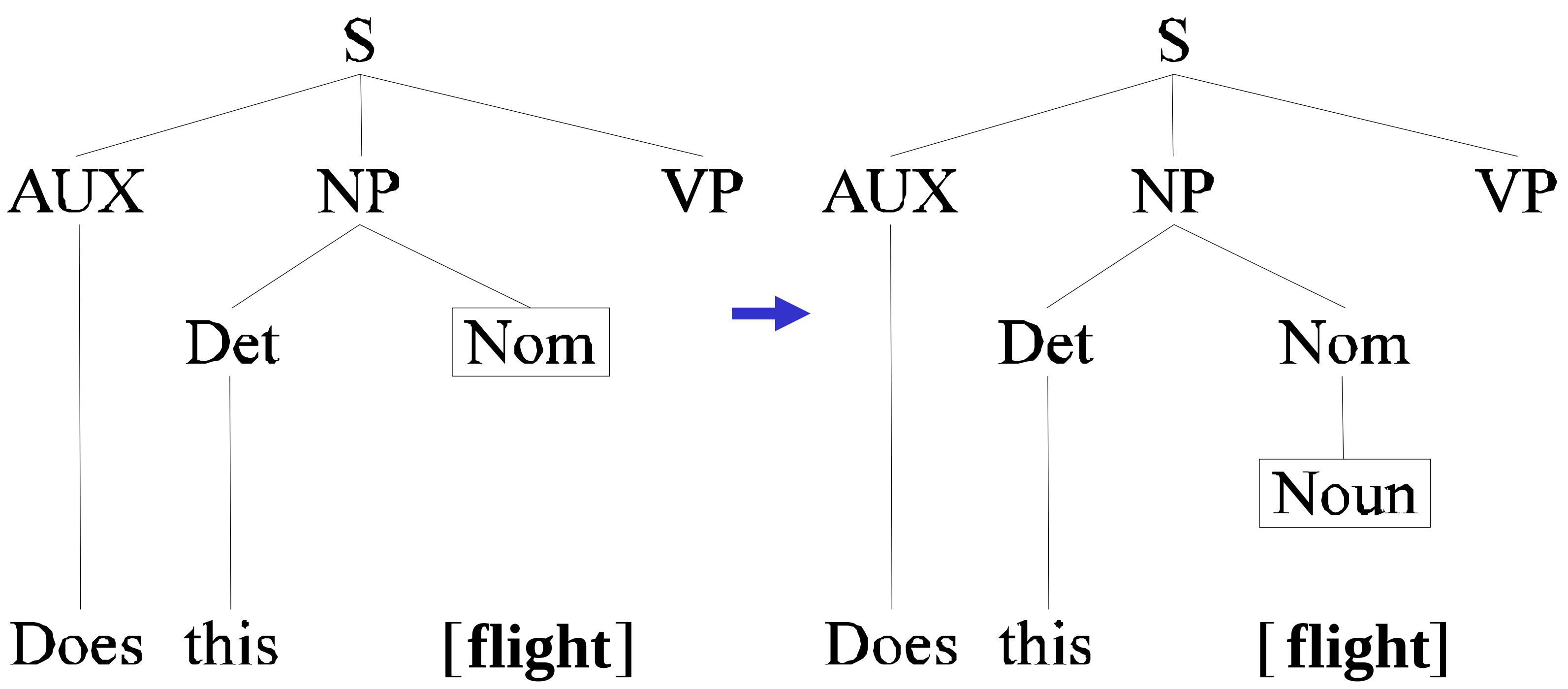
Sample sentence: **"Does this flight include a meal?"**



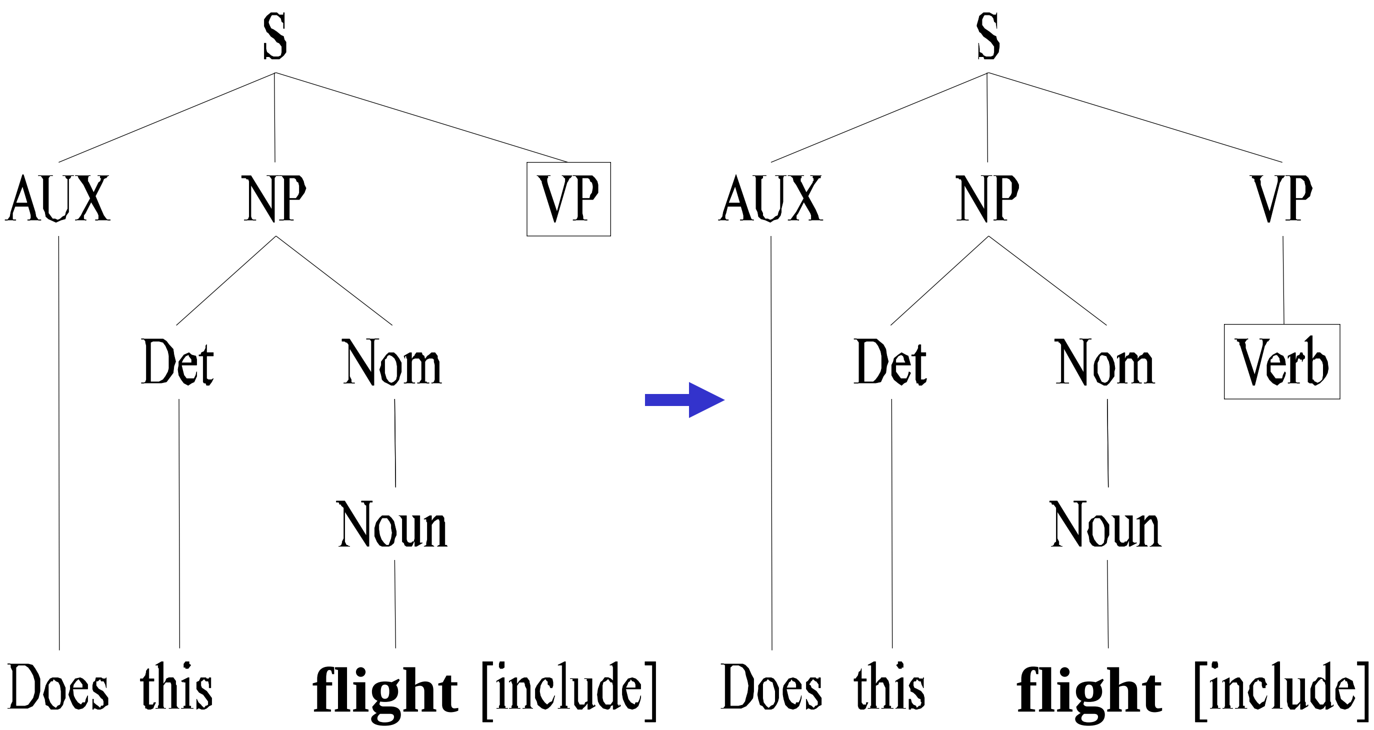
Example "Does this flight include a meal?"



Example **"Does this flight include a meal?"**

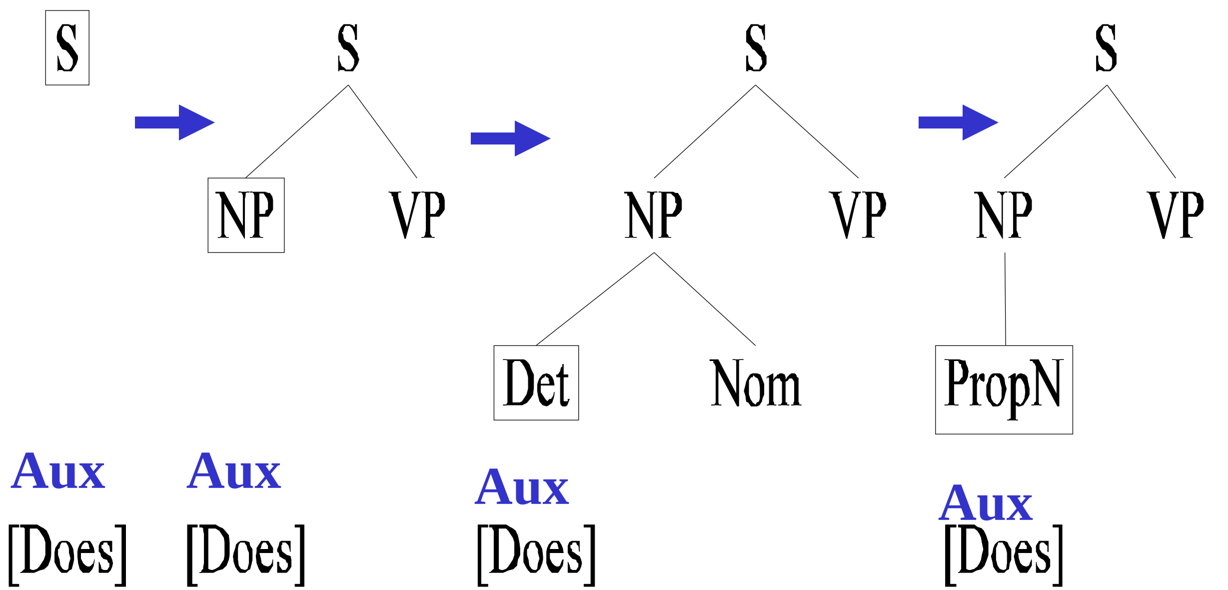


Example "Does this flight include a meal?"



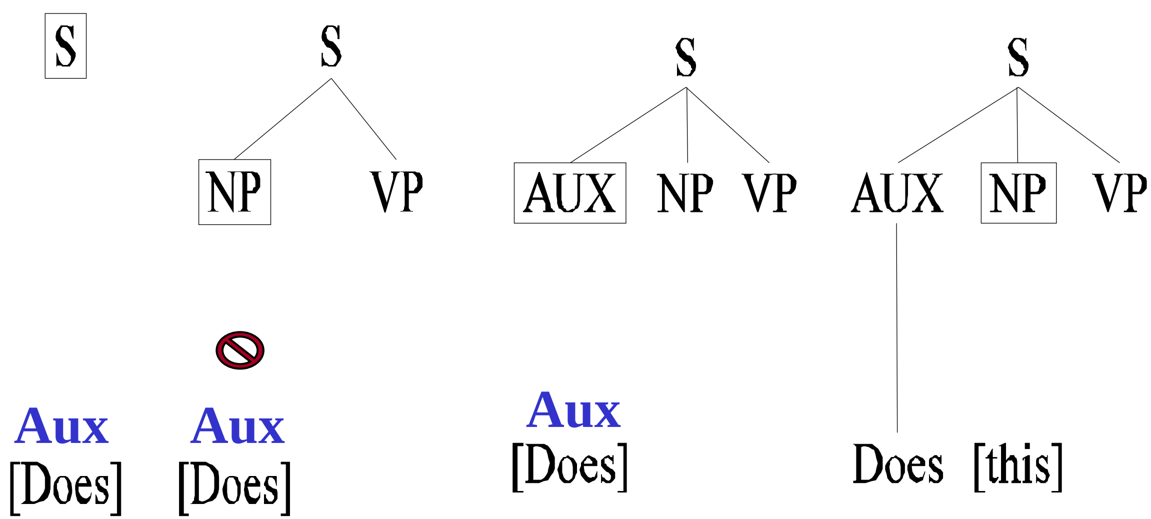
Adding Bottom-up Filtering

The following sequence was a waste of time because an NP cannot generate a parse tree starting with an AUX



Bottom-Up Filtering

Category	Left Corners
S	Det, Proper-Noun, Aux, Verb
NP	Det, Proper-Noun
Nominal	Noun
VP	Verb



Effective Parsing

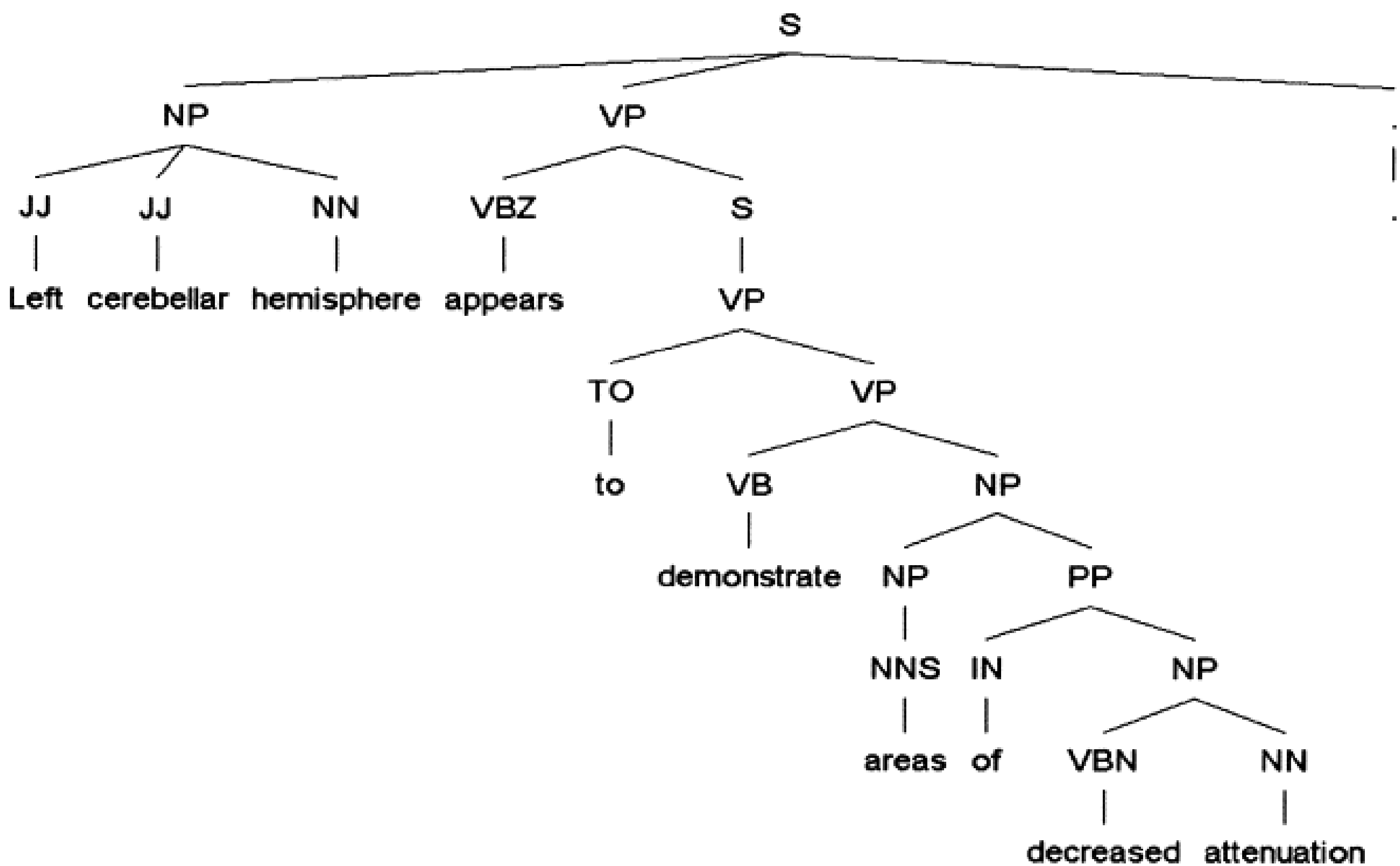
- Top-down and Bottom-up can be effectively combined but still cannot deal with ambiguity and repeated parsing
  - PARTIAL SOLUTION: Dynamic Programming approaches (you'll see one applied to Prob. CFG)

  Fills tables with solution to sub-problems

Parsing: sub-trees consistent with the input, once discovered, are stored and can be reused

1. Stores ambiguous parse compactly (but cannot select best one)
2. Does not do (avoidable) repeated work

Example of relatively complex parse tree



Journal of the American Medical Informatics Association, 2005, Improved Identification of Noun Phrases in Clinical Radiology Reports ….

Check out demos on course web page

- Berkeley Parser
- Stanford Parser

Next class

- **Probabilistic CFG…**