Syntax: Introduction, Parsing and CFG

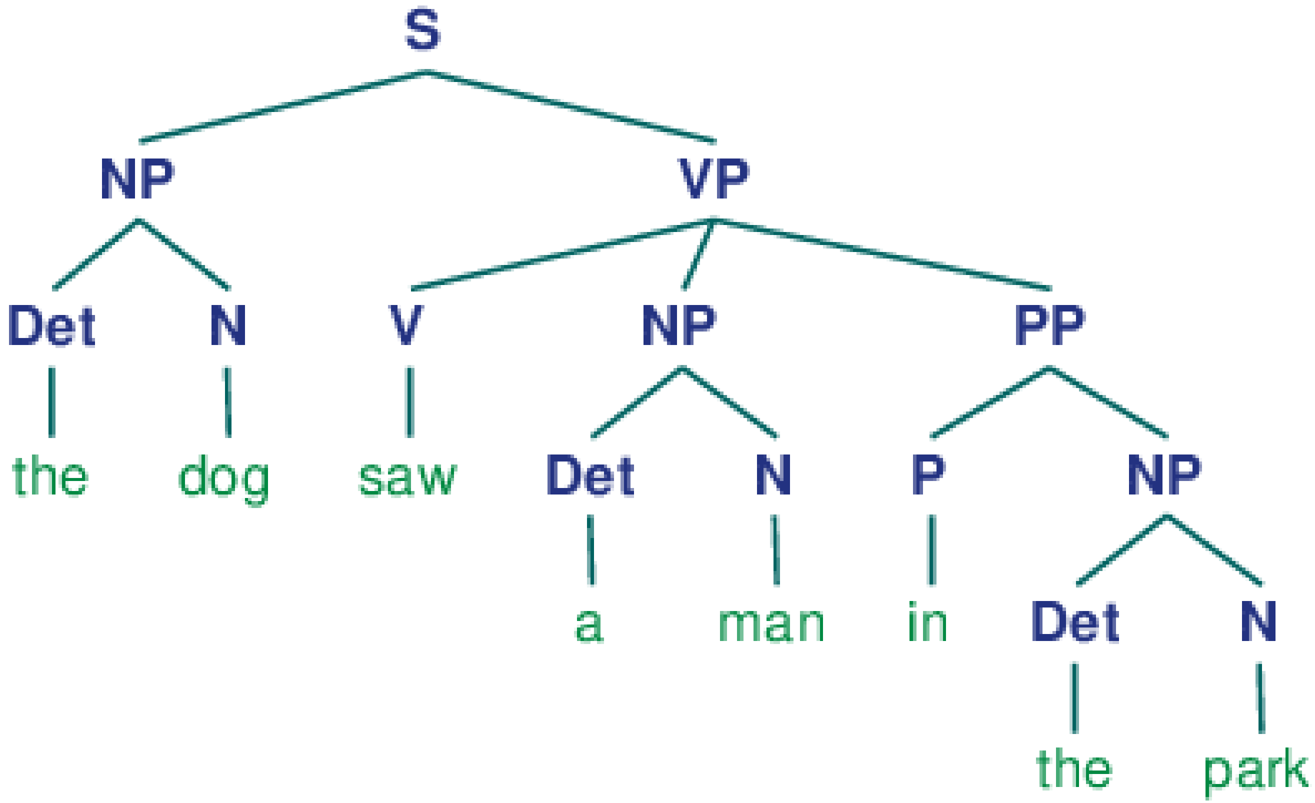
Outline

- Syntax
- Grammar
- CFG
- Parsing- top down and bottom up
- CYK algorithm

Syntax

- We discussed:
  - Language model --> word order
  - POS tagging --> word category
- Syntax: setting out togather
- the way words are arranged together
- How to order the words accordinng to rules (grammar)

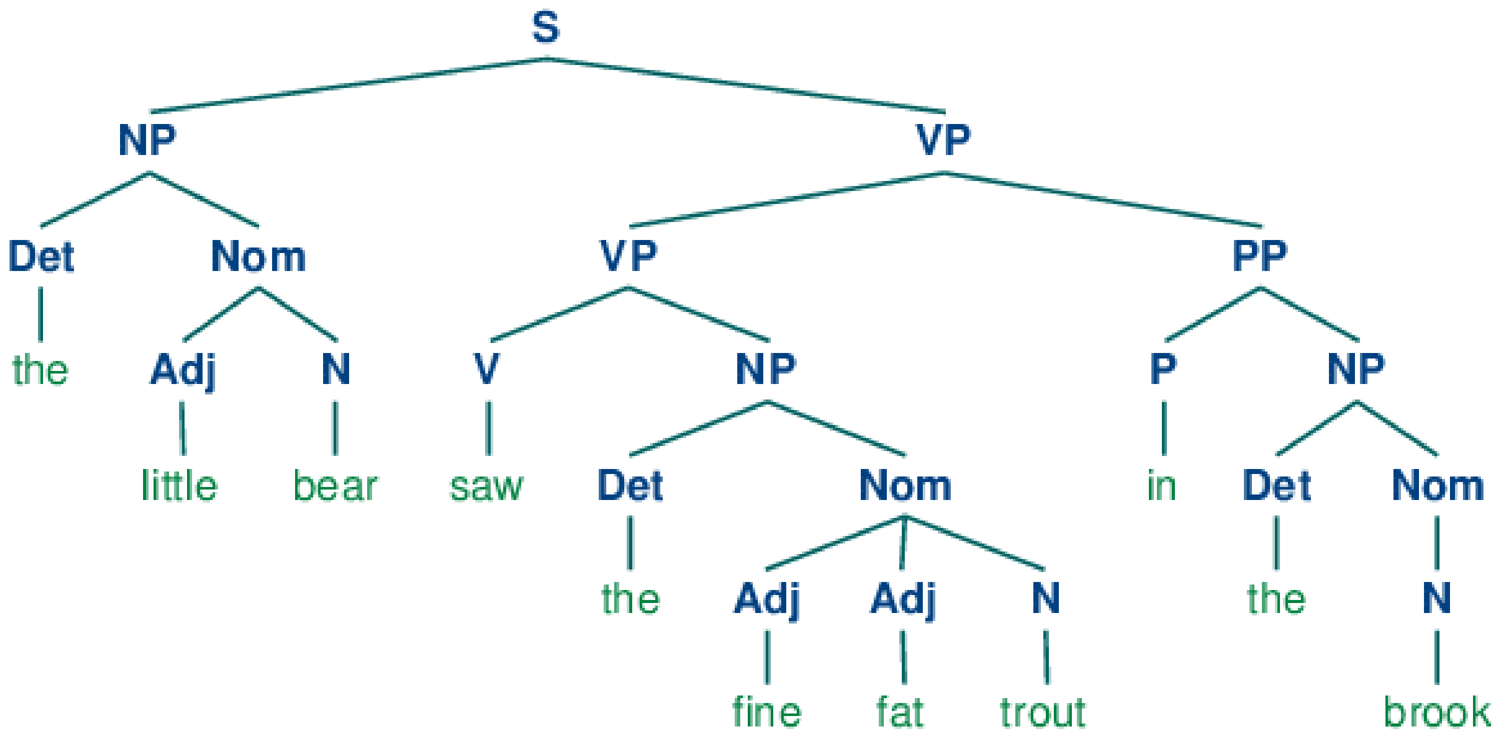
Parse Tree



Some basics

- **Constituency**: groups of words can behave as single units
- **Noun Phrase**: Harry the Horse, the Broadway coppers, they
- **Preposition Phrase**: On september, from Chickago,...
- **Verb Phrase**: book that flight, killed by Rama
- Constituent is identified by its head word

<table>
<tr><td>Det
the</td><td>Adj
little</td><td>N
bear</td><td>V
saw</td><td>Det
the</td><td>Adj
fine</td><td>Adj
fat</td><td>N
trout</td><td>P
in</td><td>Det
the</td><td>N
brook</td></tr>
<tr><td>Det
the</td><td colspan="2">Nom
bear</td><td>V
saw</td><td>Det
the</td><td colspan="3">Nom
trout</td><td>P
in</td><td colspan="2">NP
it</td></tr>
<tr><td colspan="3">NP
He</td><td>V
saw</td><td colspan="4">NP
it</td><td colspan="3">PP
there</td></tr>
<tr><td colspan="3">NP
He</td><td colspan="5">VP
ran</td><td colspan="3">PP
there</td></tr>
<tr><td colspan="3">NP
He</td><td colspan="8">VP
ran</td></tr>
</table>


S
NP
VP
Det
Nom
VP
PP
the
Adj
N
V
NP
P
NP
little
bear
saw
Det
Nom
in
Det
Nom
the
Adj
Adj
N
the
N
fine
fat
trout
brook

Context Free Grammars

- Formal system for modeling constituent structure in English
- Aka Phrase structure grammar
- Set of rules (or productions)
- Example:
  - NP --> DET NOM
  - NP --> PropNoun
  - NOM --> NOM NOUN
  - Det --> a|the
  - NOUN --> flight

CFG

- CFG = {T, N, S, R}
  - T : Terminals
  - N: Non-terminals
  - S: Start symbol (belongs to T)
  - R: set of production rules X--> Y

CFG: Derivation

- A CFG can be used to generate a set of strings. This sequence of rule expansions is called a derivation of the string of words.
- Commonly represent as Parse Tree
- If sentence can be represented by grammart--> its grammatical otherwise not grammatical (Panineeyam and Apanineeyam)

$$
\begin{aligned}
NP &\rightarrow Det\ Nominal \\
NP &\rightarrow ProperNoun \\
Nominal &\rightarrow Noun \mid Nominal\ Noun
\end{aligned}
$$

Context-free rules can be hierarchically embedded, so we can combine the previous rules with others, like the following, that express facts about the lexicon:

$$
\begin{aligned}
Det &\rightarrow a \\
Det &\rightarrow the \\
Noun &\rightarrow flight
\end{aligned}
$$

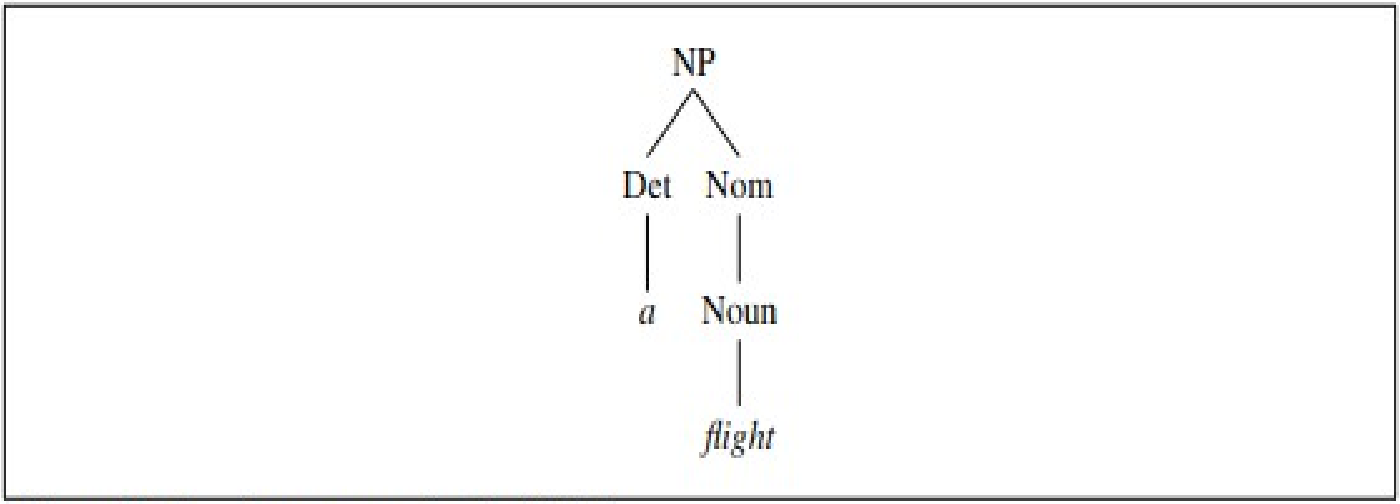


Figure 12.1 A parse tree for “a flight”.

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

Figure 12.2 The lexicon for $\mathscr{L}_0$.

Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

Figure 12.3 The grammar for $\mathscr{L}_0$, with example phrases for each rule.

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

Figure 12.2 The lexicon for $\mathscr{L}_0$.

Grammar Rules			Examples
S	→	*NP VP*	I + want a morning flight
NP	→	*Pronoun*	I
	|	*Proper-Noun*	Los Angeles
	|	*Det Nominal*	a + flight
Nominal	→	*Nominal Noun*	morning + flight
	|	*Noun*	flights
VP	→	*Verb*	do
	|	*Verb NP*	want + a flight
	|	*Verb NP PP*	leave + Boston + in the morning
	|	*Verb PP*	leaving + on Thursday
PP	→	*Preposition NP*	from + Los Angeles

Figure 12.3 The grammar for $\mathscr{L}_0$, with example phrases for each rule.

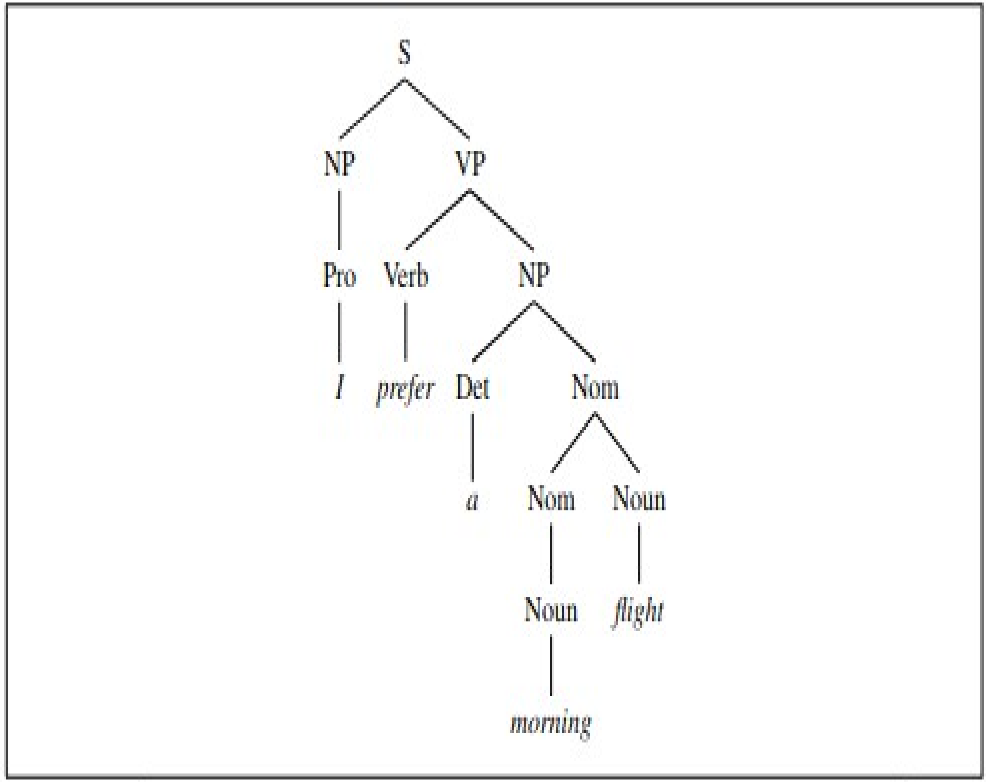
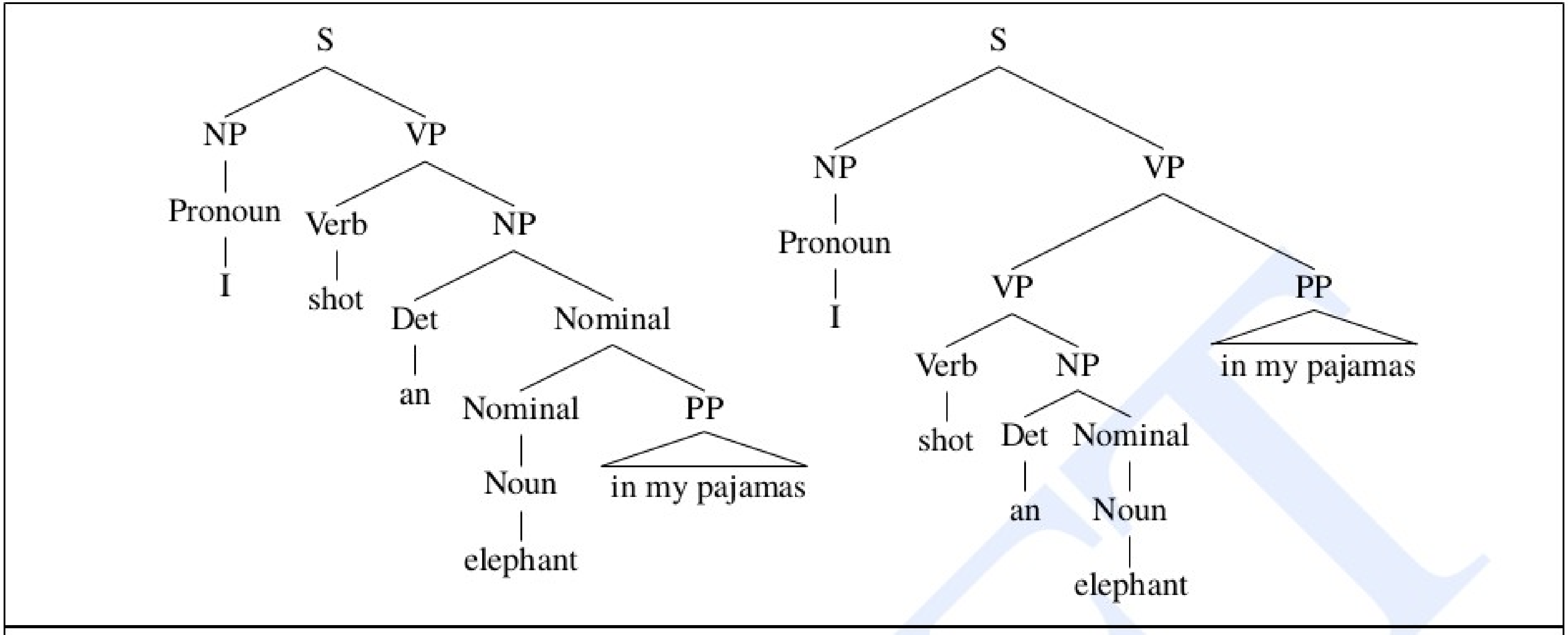


Figure 12.4 The parse tree for "I prefer a morning flight" according to grammar $\mathscr{L}_0$.

CFG

- What is context?
- The non-terminal in the LHS of the grammar is independent of other terminals or non-ternminals

Ambiguity

- When more than one parse tree(derivation) for a single sentence
- 

Parsing Approaches

- Top-down
- Bottom-up

Top-down

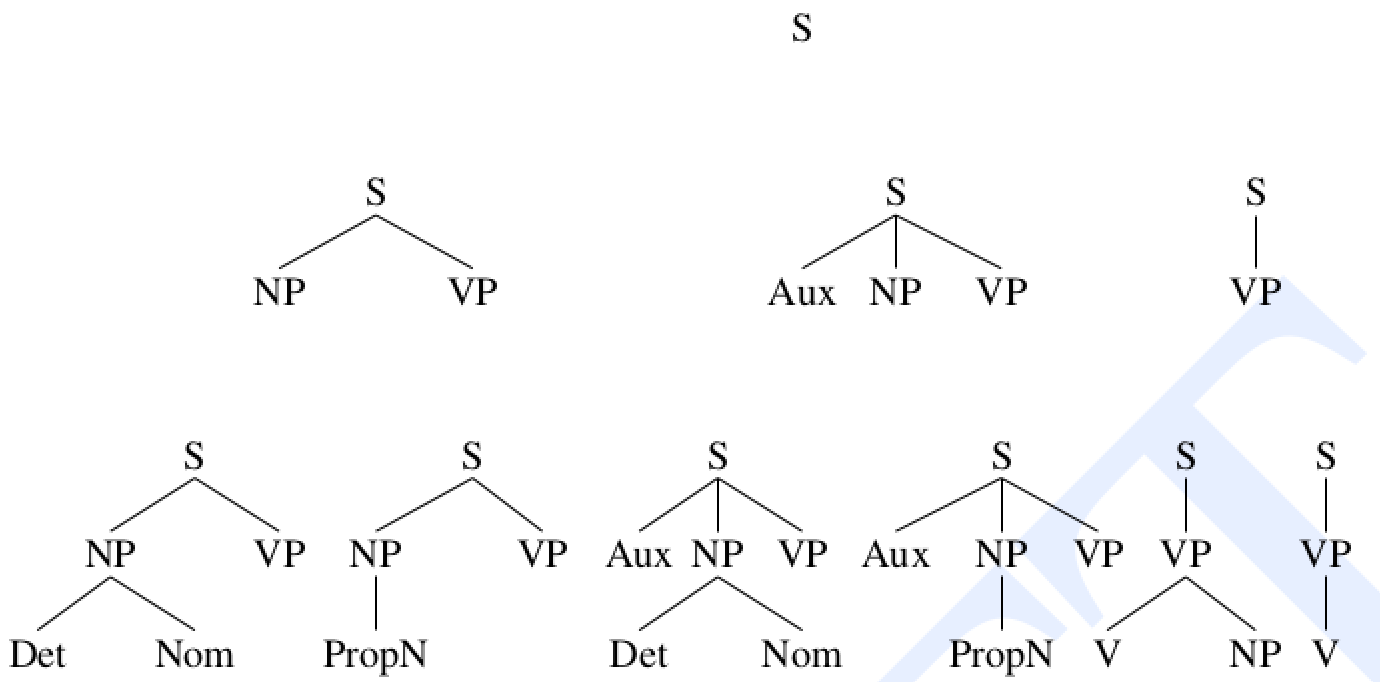
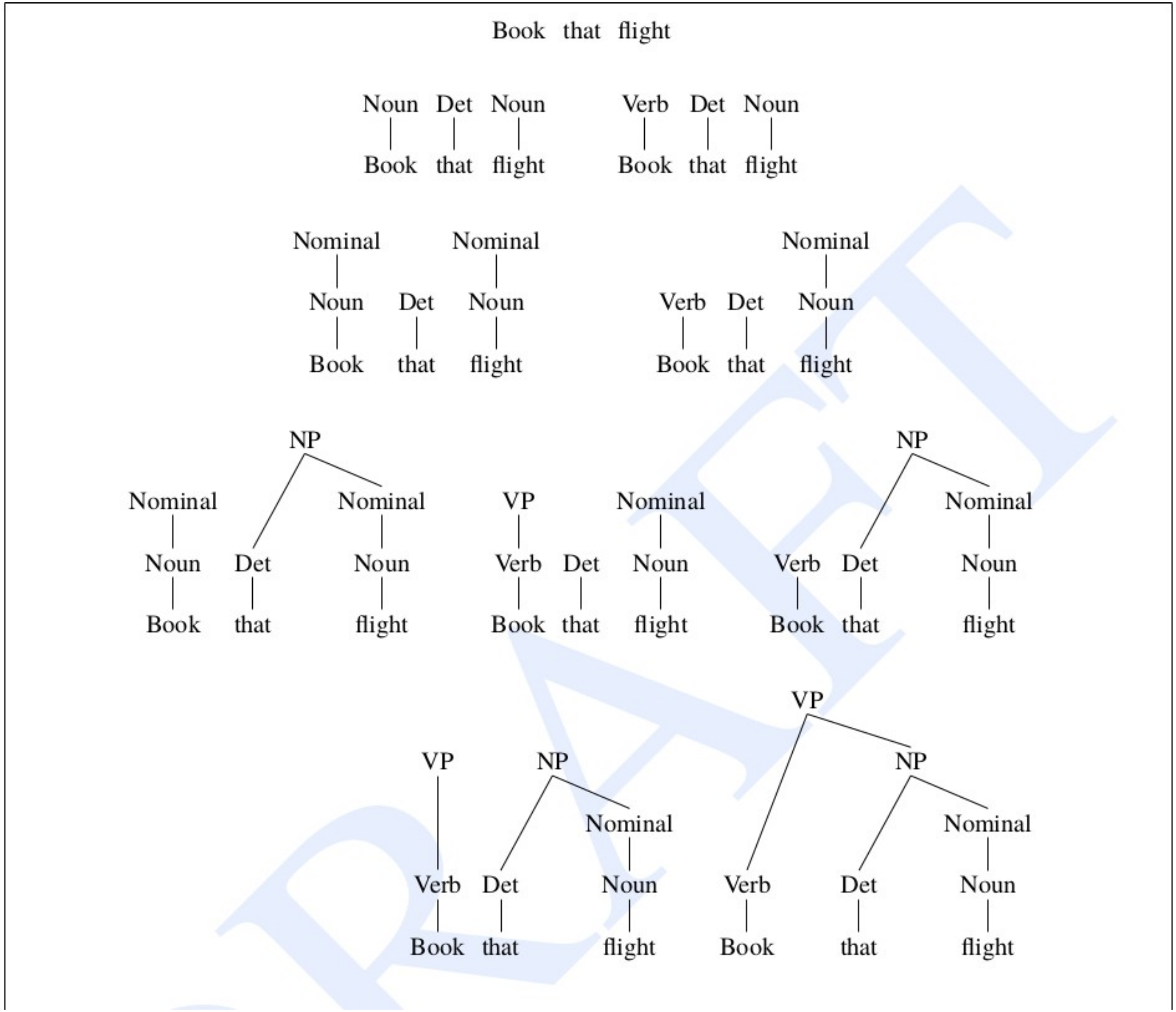


Figure 13.3 An expanding top-down search space. Each ply is created by taking each tree from the previous ply, replacing the leftmost non-terminal with each of its possible expansions, and collecting each of these trees into a new ply.

Bottom-up



Book that flight
Noun Det Noun
Book that flight
Verb Det Noun
Book that flight
Nominal Nominal
Noun Det Noun
Book that flight
Nominal
Verb Det Noun
Book that flight
NP
Nominal Nominal
Noun Det Noun
Book that flight
VP Nominal
Verb Det Noun
Book that flight
NP
Nominal
Verb Det Noun
Book that flight
VP NP
Nominal
Verb Det Noun
Book that flight
VP
NP
Nominal
Verb Det Noun
Book that flight

Top-down vs Bottom Up

- Top down never explores options that will not lead to a full parse, but can explore many options that never connect to the actual sentence.
- Bottom up never explores options that do not connect to the actual sentence but can explore options that can never lead to a full parse.
- Relative amounts of wasted search depend on how much the grammar branches in each direction.

How efficiently search tree space?

- Dynamic Programming
- CYK algorithm
- O(n^3) complexity

Next

- CYK (brief)
- Probabilistic CFG