Syntax & Grammars

Instructor: Wei Xu
Ohio State University

Slides adapted from Michael Collins, Marine Carpuat, Nathan Schneider

What's next in the class?

- From sequences to **trees**
- Syntax
  - Constituent, Grammatical relations, Dependency relations
- Formal Grammars
  - Context-free grammar
  - Dependency grammar

sýntaxis (setting out or arranging)

- The ordering of words and how they group into phrases
  - [[students][cook and serve][grandparents]]
  - [[students][cook][and][serve grandparents]]

Students Cook & Serve Grandparents

On Thursday, September 9, Gorman School hosted the first annual Grandparent's Day.

All Grandparents were invited to a school wide pancake breakfast. Upper grade students served as excellent chefs, as well as taking responsibility for serving the food and the clean up after-

Syntax and Grammar

- Goal of syntactic theory
  - “explain how people combine words to form sentences and how children attain knowledge of sentence structure”

- Grammar
  - implicit knowledge of a native speaker
  - acquired without explicit instruction
  - minimally able to generate all and only the possible sentences of the language

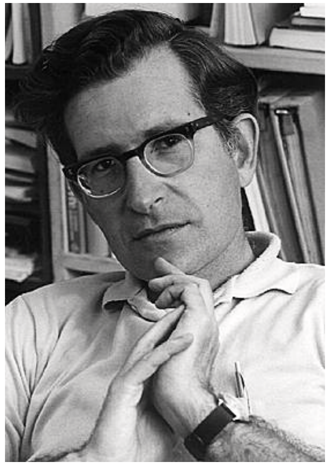
Colin Phillips, “Syntax”, 2003

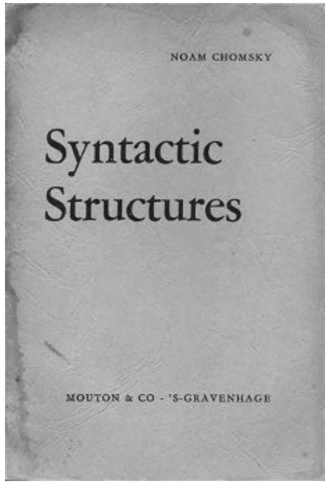
Syntax vs. Semantics

“Colorless green ideas sleep furiously.”

— Noam Chomsky (1957)

Contrast with: “sleep green furiously ideas colorless”

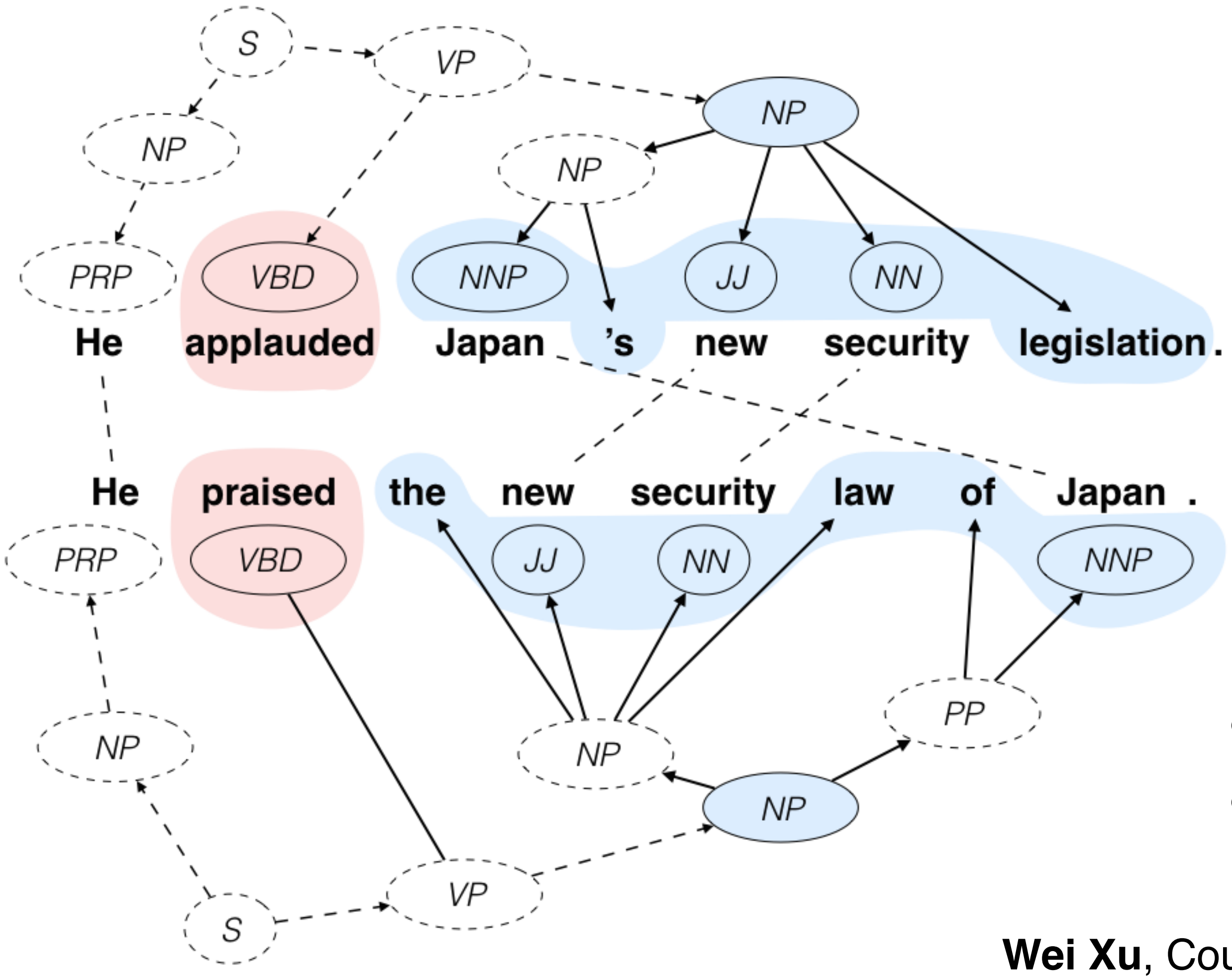




Syntax in NLP Applications

- Syntactic analysis is often a key component in applications
  - Grammar Checkers
  - Natural Language Generation:
    e.g. Sentence Compression, Fusion, Simplification, …
  - Information Extraction
  - Machine Translation
  - Question Answering
  - …

An Example: Sentence Simplification



- current state-of-the-art system
- syntactic machine translation techniques

Wei Xu, Courtney Napoles, Ellie Pavlick, Quanze Chen, Chris Callison-Burch. "Optimizing Statistical Machine Translation for Simplification" in TACL (2016)

Two Views of Syntactic Structure

- Constituency (phrase structure)
  - Phrase structure organizes words in nested constituents

- Dependency structure
  - Shows which words depend on (modify or are arguments of) which on other words

Syntax

Constituency Parsing and Context Free Grammars

Constituency

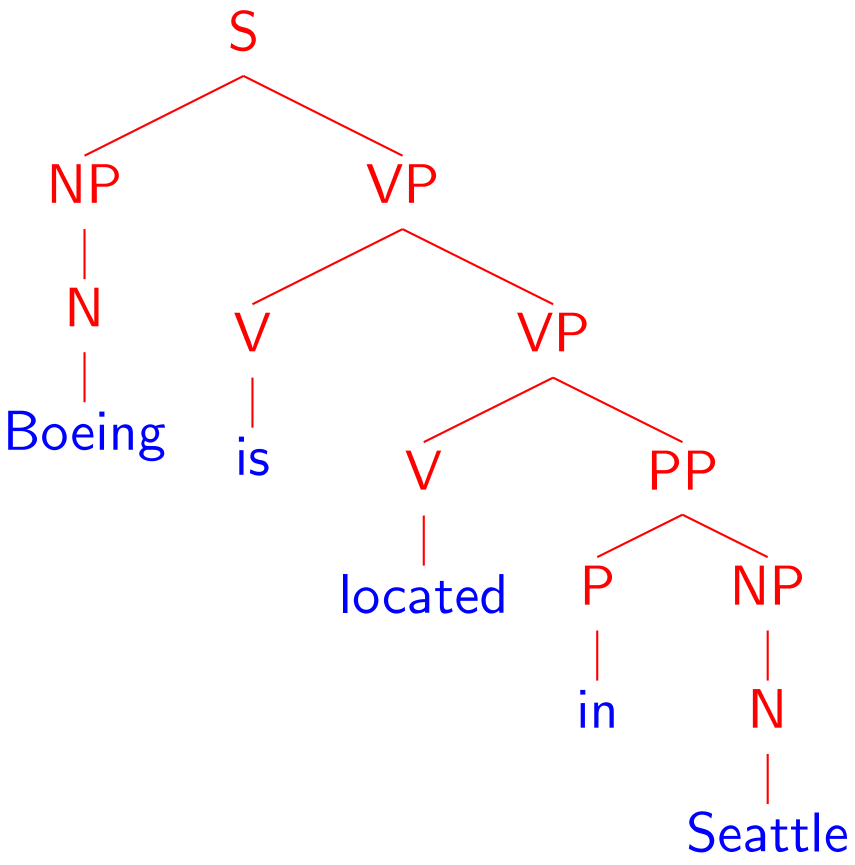
- Basic idea: groups of words act as a single unit
- Constituents form coherent classes that behave similarly
  - with respect to their internal structure:
    e.g. at the core of a noun phrase is a a noun
  - with respect to other constituents:
    e.g. noun phrases generally occur before verbs

Parsing (Syntactic Structure)

INPUT:

Boeing is located in Seattle.

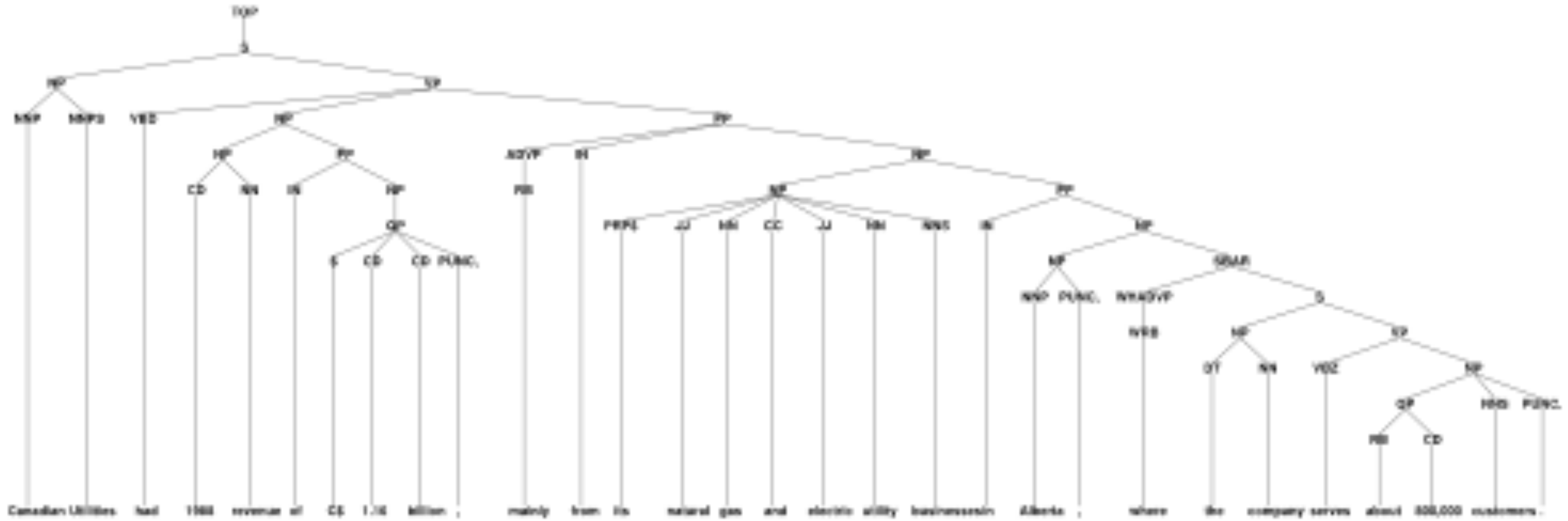
OUTPUT:



Data for Parsing Experiments

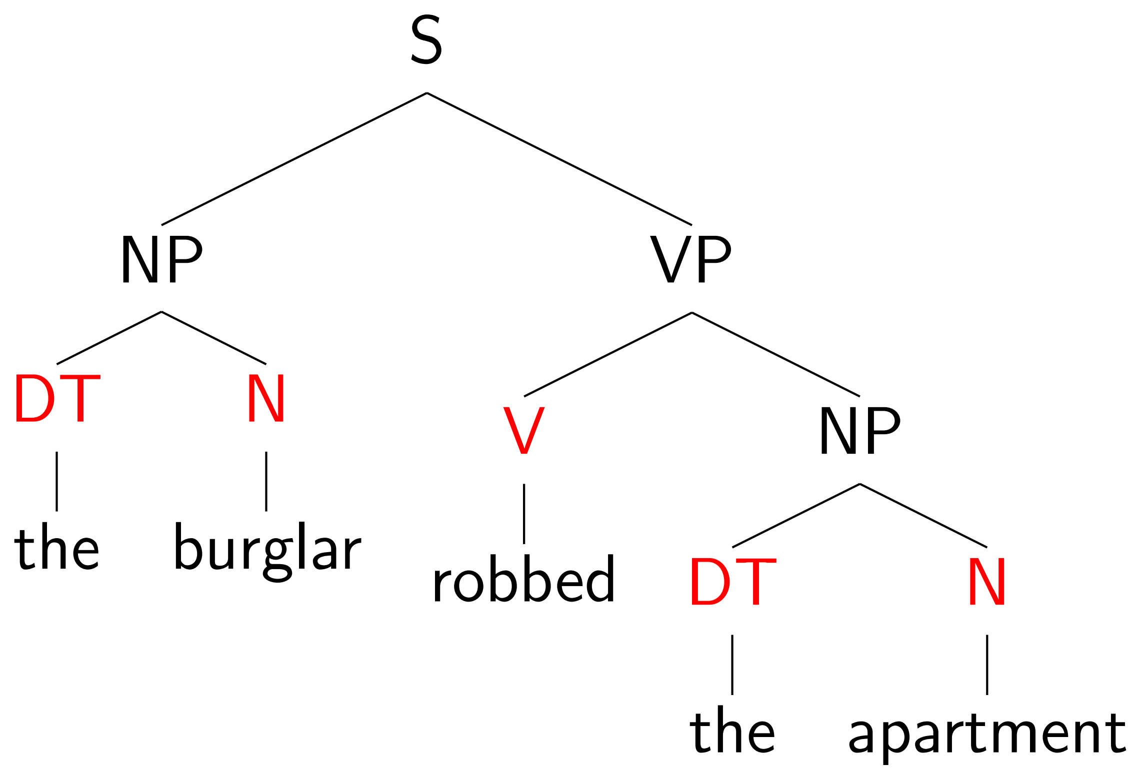
- Penn WSJ Treebank = 50,000 sentences with associated trees
- Usual set-up: 40,000 training sentences, 2400 test sentences

An example tree:



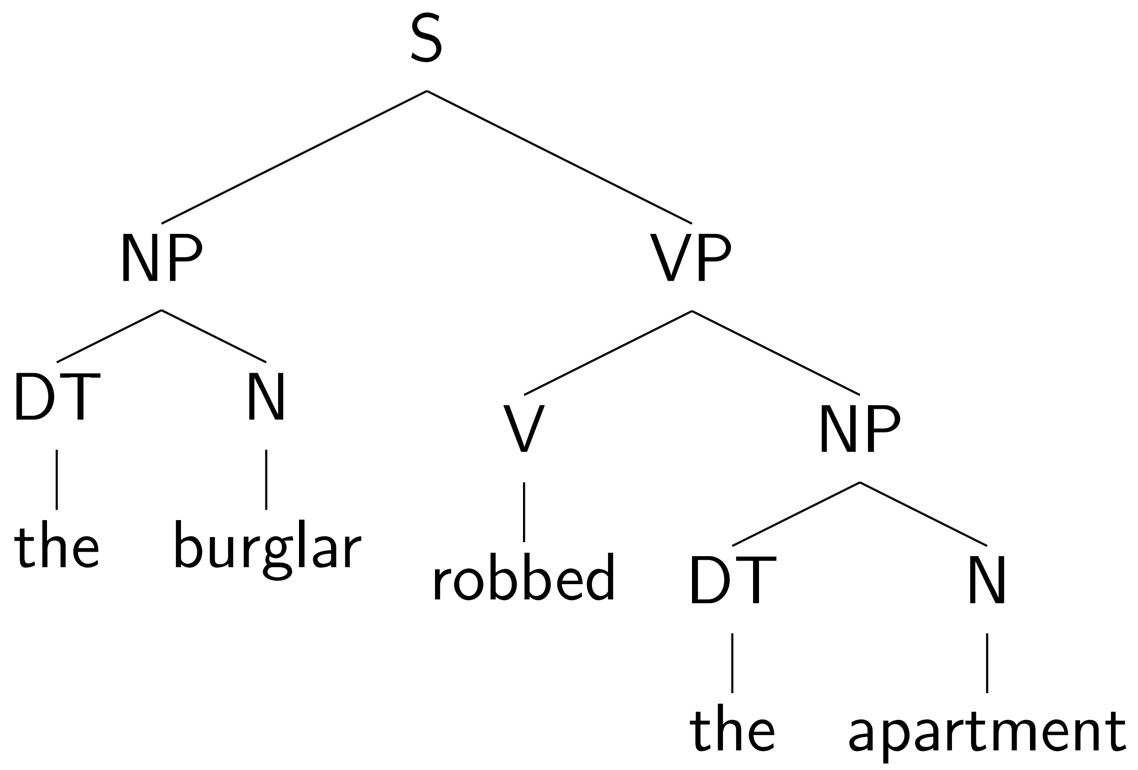
The Information Conveyed by Parse Trees

(1) Part of speech for each word
(N = noun, V = verb, DT = determiner)



The Information Conveyed by Parse Trees (continued)

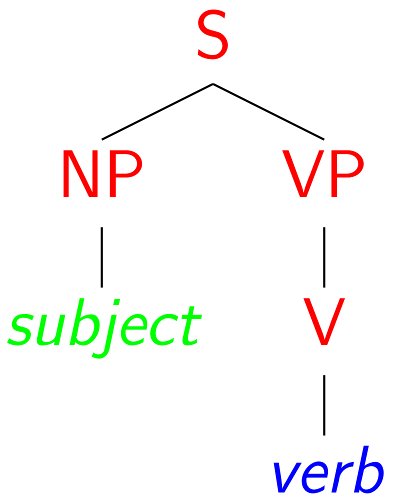
(2) Phrases

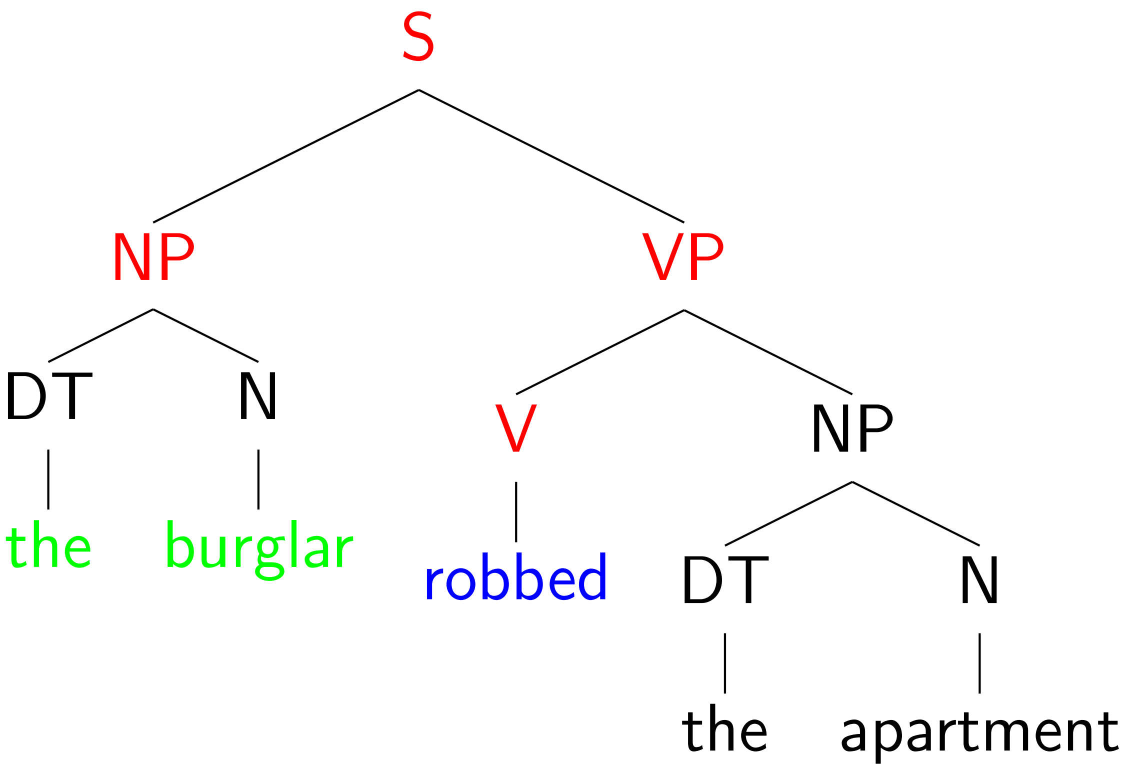


Noun Phrases (NP): “the burglar”, “the apartment”
Verb Phrases (VP): “robbed the apartment”
Sentences (S): “the burglar robbed the apartment”

The Information Conveyed by Parse Trees (continued)

(3) Useful Relationships





$\Rightarrow$ "the burglar" is the subject of "robbed"

Grammars and Constituency

- For a particular language:
  - What are the “right” set of constituents?
  - What rules govern how they combine?

- Answer: not obvious and difficult
  - That’s why there are many different theories of grammar and competing analyses of the same data!

Syntactic Formalisms

- Work in formal syntax goes back to Chomsky's PhD thesis in the 1950s
- Examples of current formalisms: minimalism, lexical functional grammar (LFG), head-driven phrase-structure grammar (HPSG), tree adjoining grammars (TAG), categorial grammars

Regular Grammar

- You've already seen one class of grammars: **regular expressions**
  - A pattern like ^[a-z][0-9]$ corresponds to a grammar which accepts (matches) some strings but not others.

- **Q:** Can regular languages define infinite languages?

- **Q:** Can regular languages define arbitrarily complex languages?

Regular Grammar

- You've already seen one class of grammars: **regular expressions**
  - A pattern like ^[a-z][0-9]$ corresponds to a grammar which accepts (matches) some strings but not others.

- **Q:** Can regular languages define infinite languages?

  Yes, e.g. a*

- **Q:** Can regular languages define arbitrarily complex languages?

  No. Cannot match all strings with matched parentheses or in $a^n b^n$ forms in general (recursion/arbitrary nesting).

https://en.wikipedia.org/wiki/Pumping_lemma_for_regular_languages

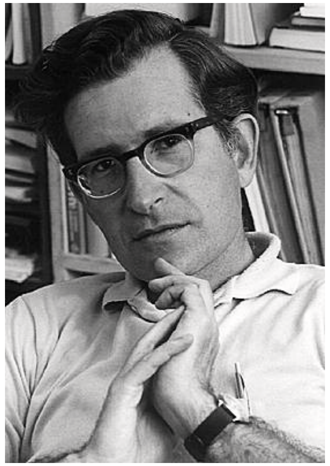
English is not a regular language

- There are certain types of sentences in English that look like $a^n b^n$
  - For example, "The dog that the man that the cat saw kicked barked" could be extended indefinitely.

- If syntax were regular, we should be able to reach a length after which we can just insert nouns, without adding the corresponding verb (by the Pumping Lemma).
  - For example, "The dog that the man that the cat that the rat that the mouse ______ feared saw kicked barked"

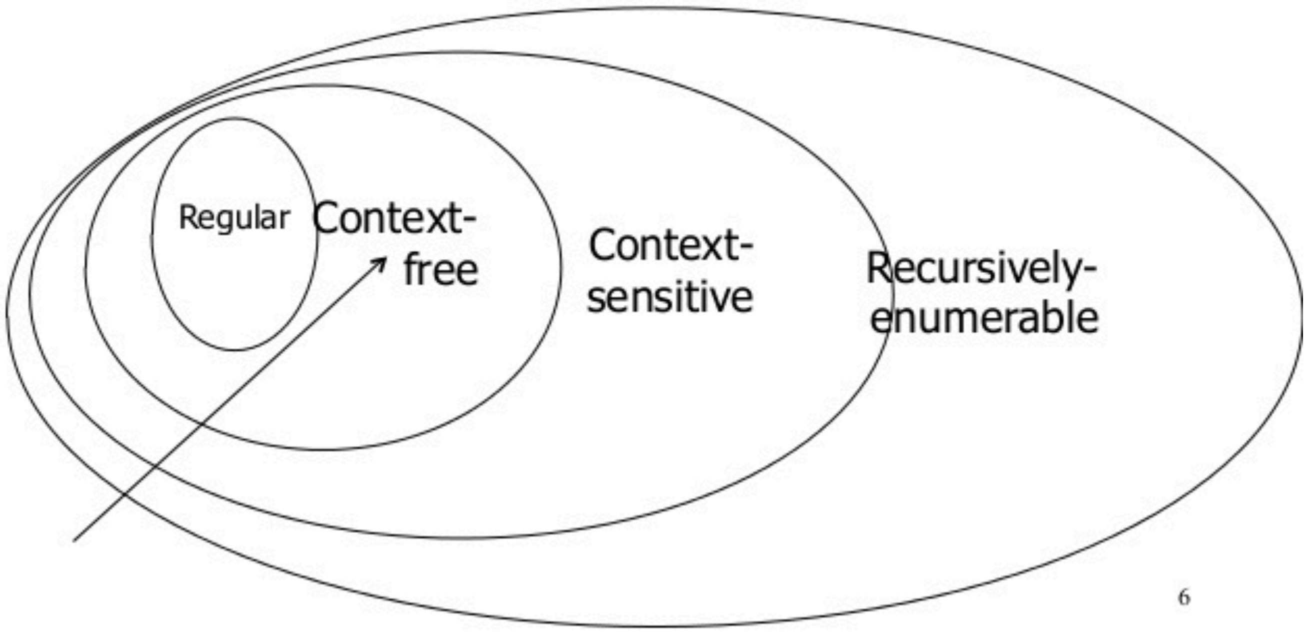
Noah Chomsky. 1956. The range of adequacy of various types of grammars.

The Chomsky Hierarchy



- Hierarchy of classes of formal languages

One language is of greater generative power or complexity than another if it can define a language that other cannot define. Context-free grammars are more powerful than regular grammars



Context-Free Grammars

a.k.a phrase structure grammars,
Backus-Naur form (BNF)

Hopcroft and Ullman, 1979

A context free grammar $G = (N, \Sigma, R, S)$ where:

- N is a set of non-terminal symbols
- Σ is a set of terminal symbols
- R is a set of rules of the form $X \rightarrow Y_1 Y_2 \ldots Y_n$ for $n \geq 0$, $X \in N$, $Y_i \in (N \cup \Sigma)$
- $S \in N$ is a distinguished start symbol

A Context-Free Grammar for English

$N = \{$S, NP, VP, PP, DT, Vi, Vt, NN, IN$\}$
$S\ =$ S
$\Sigma\ = \{$sleeps, saw, man, woman, telescope, the, with, in$\}$

$R =$

S	$\rightarrow$	NP	VP
VP	$\rightarrow$	Vi	
VP	$\rightarrow$	Vt	NP
VP	$\rightarrow$	VP	PP
NP	$\rightarrow$	DT	NN
NP	$\rightarrow$	NP	PP
PP	$\rightarrow$	IN	NP

Vi	$\rightarrow$	sleeps
Vt	$\rightarrow$	saw
NN	$\rightarrow$	man
NN	$\rightarrow$	woman
NN	$\rightarrow$	telescope
DT	$\rightarrow$	the
IN	$\rightarrow$	with
IN	$\rightarrow$	in

Note: S=sentence, VP=verb phrase, NP=noun phrase, PP=prepositional phrase, DT=determiner, Vi=intransitive verb, Vt=transitive verb, NN=noun, IN=preposition

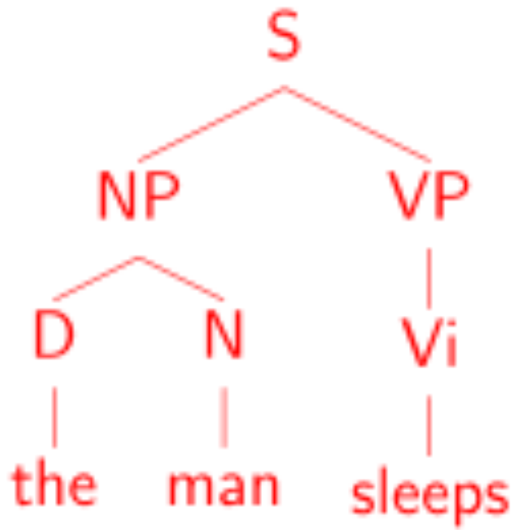
Left-Most Derivations

A left-most derivation is a sequence of strings $s_1 \dots s_n$, where

- $s_1 = S$, the start symbol
- $s_n \in \Sigma^*$, i.e. s_n is made up of terminal symbols only
- Each s_i for $i = 2 \dots n$ is derived from s_{i-1} by picking the left-most non-terminal X in s_{i-1} and replacing it by some β where $X \to \beta$ is a rule in R

For example: [S], [NP VP], [D N VP], [the N VP], [the man VP], [the man Vi], [the man sleeps]

Representation of a derivation as a tree:



An Example

DERIVATION	RULES USED
S	

An Example

DERIVATION	RULES USED
S	$S \rightarrow NP\ VP$
NP VP	

An Example

DERIVATION	RULES USED
S	$S \rightarrow NP\ VP$
NP VP	$NP \rightarrow DT\ N$
DT N VP	

An Example

DERIVATION	RULES USED
S	
	S $\rightarrow$ NP VP
NP VP	
	NP $\rightarrow$ DT N
DT N VP	
	DT $\rightarrow$ the
the N VP	

An Example

DERIVATION	RULES USED
S	$S \rightarrow NP\ VP$
NP VP	$NP \rightarrow DT\ N$
DT N VP	$DT \rightarrow$ the
the N VP	$N \rightarrow$ dog
the dog VP	

An Example

DERIVATION	RULES USED
S	$S \to NP\ VP$
NP VP	$NP \to DT\ N$
DT N VP	$DT \to the$
the N VP	$N \to dog$
the dog VP	$VP \to VB$
the dog VB	

An Example

DERIVATION	RULES USED
S	
NP VP	$S \rightarrow NP\ VP$
DT N VP	$NP \rightarrow DT\ N$
the N VP	$DT \rightarrow$ the
the dog VP	$N \rightarrow$ dog
the dog VB	$VP \rightarrow VB$
the dog laughs	$VB \rightarrow$ laughs

An Example

DERIVATION

S

NP VP

DT N VP

the N VP

the dog VP

the dog VB

the dog laughs

RULES USED

S $\rightarrow$ NP VP

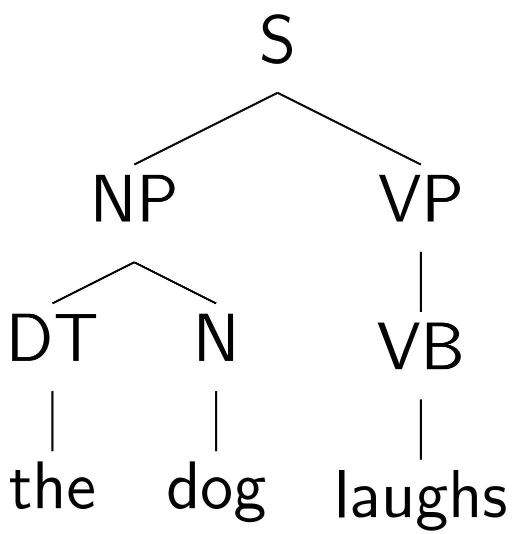
NP $\rightarrow$ DT N

DT $\rightarrow$ the

N $\rightarrow$ dog

VP $\rightarrow$ VB

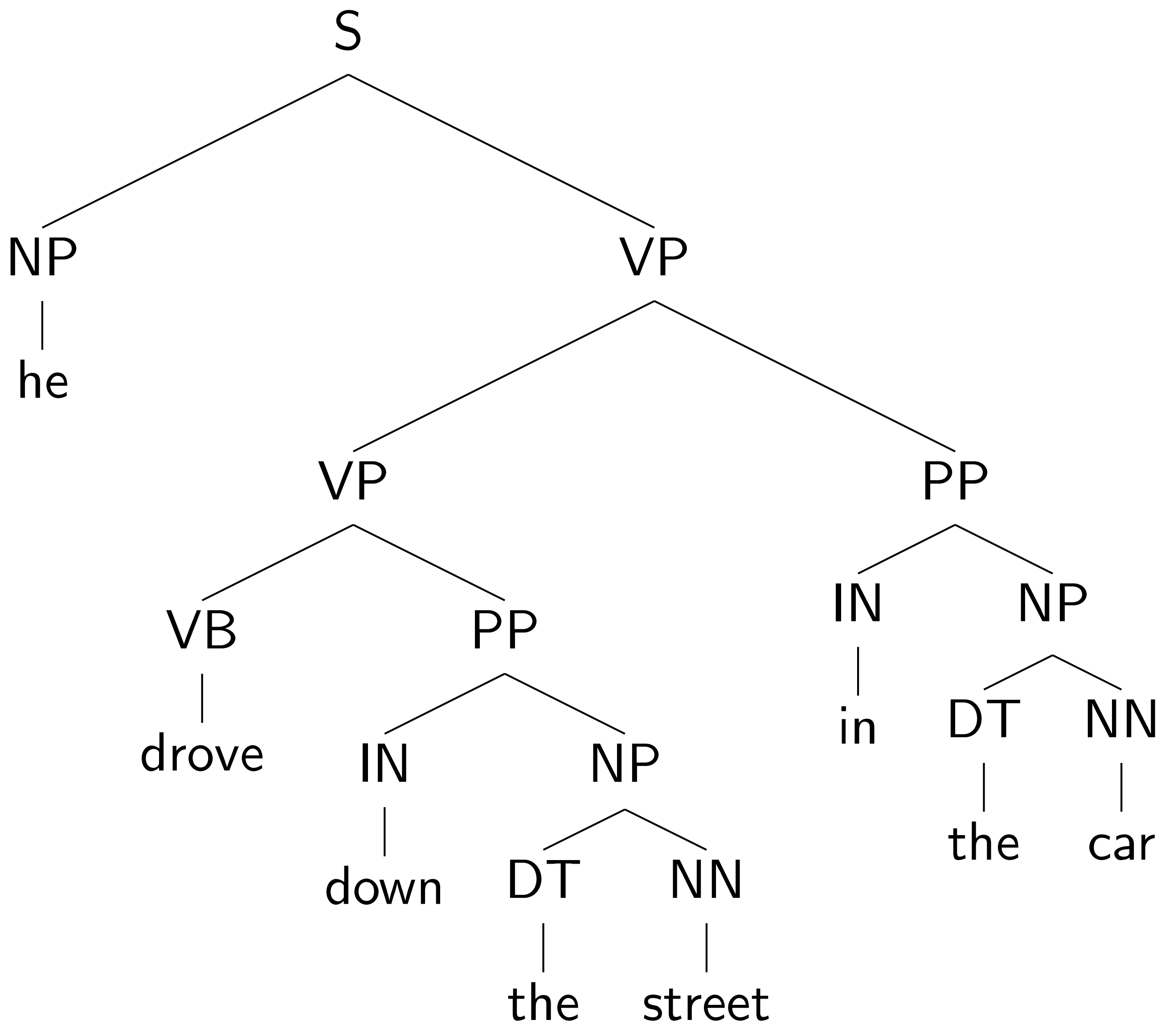
VB $\rightarrow$ laughs



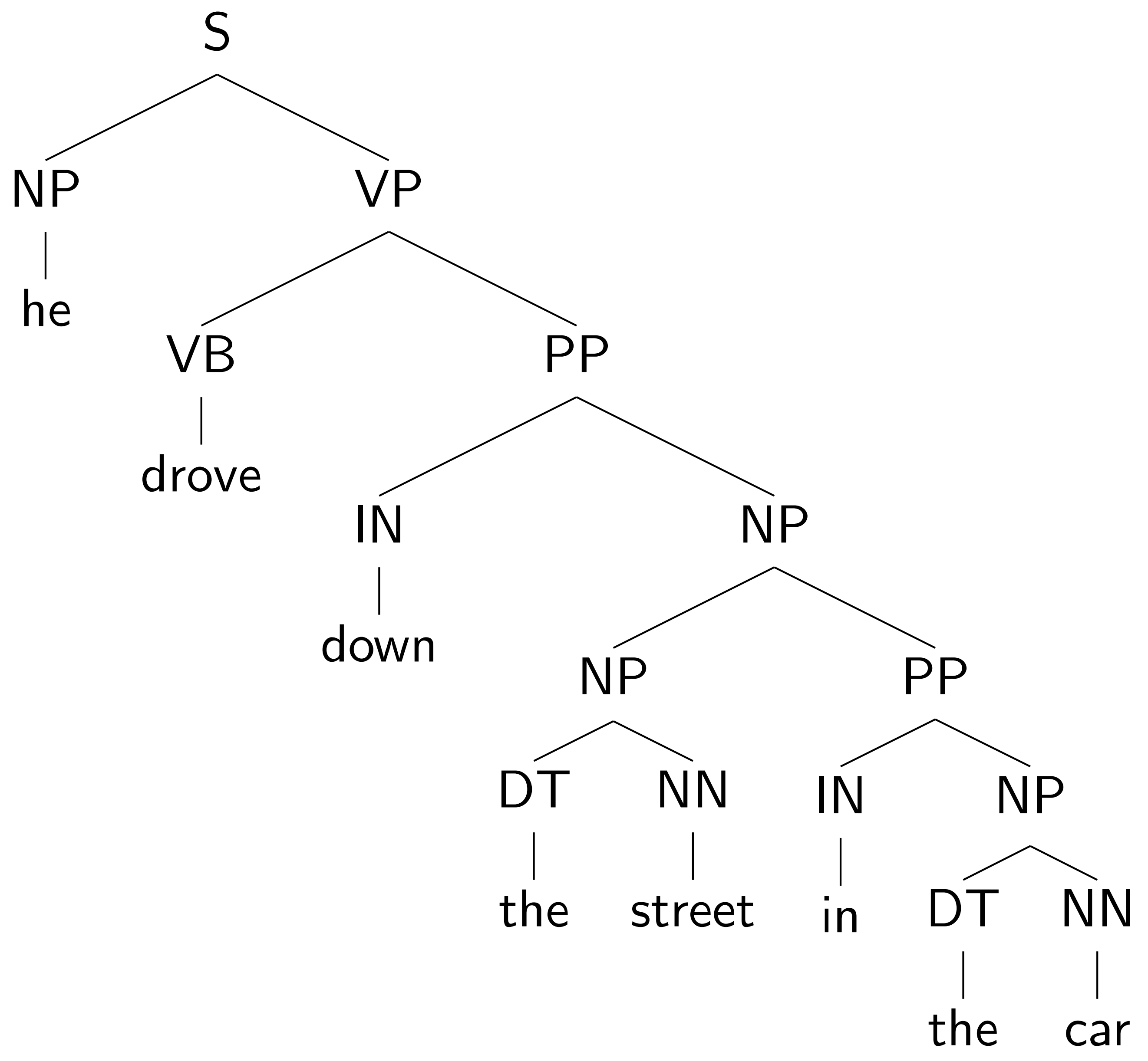
Properties of CFGs

- A CFG defines a set of possible derivations
- A string $s \in \Sigma^*$ is in the *language* defined by the CFG if there is at least one derivation that yields s
- Each string in the language generated by the CFG may have more than one derivation ("ambiguity")

An Example of Ambiguity

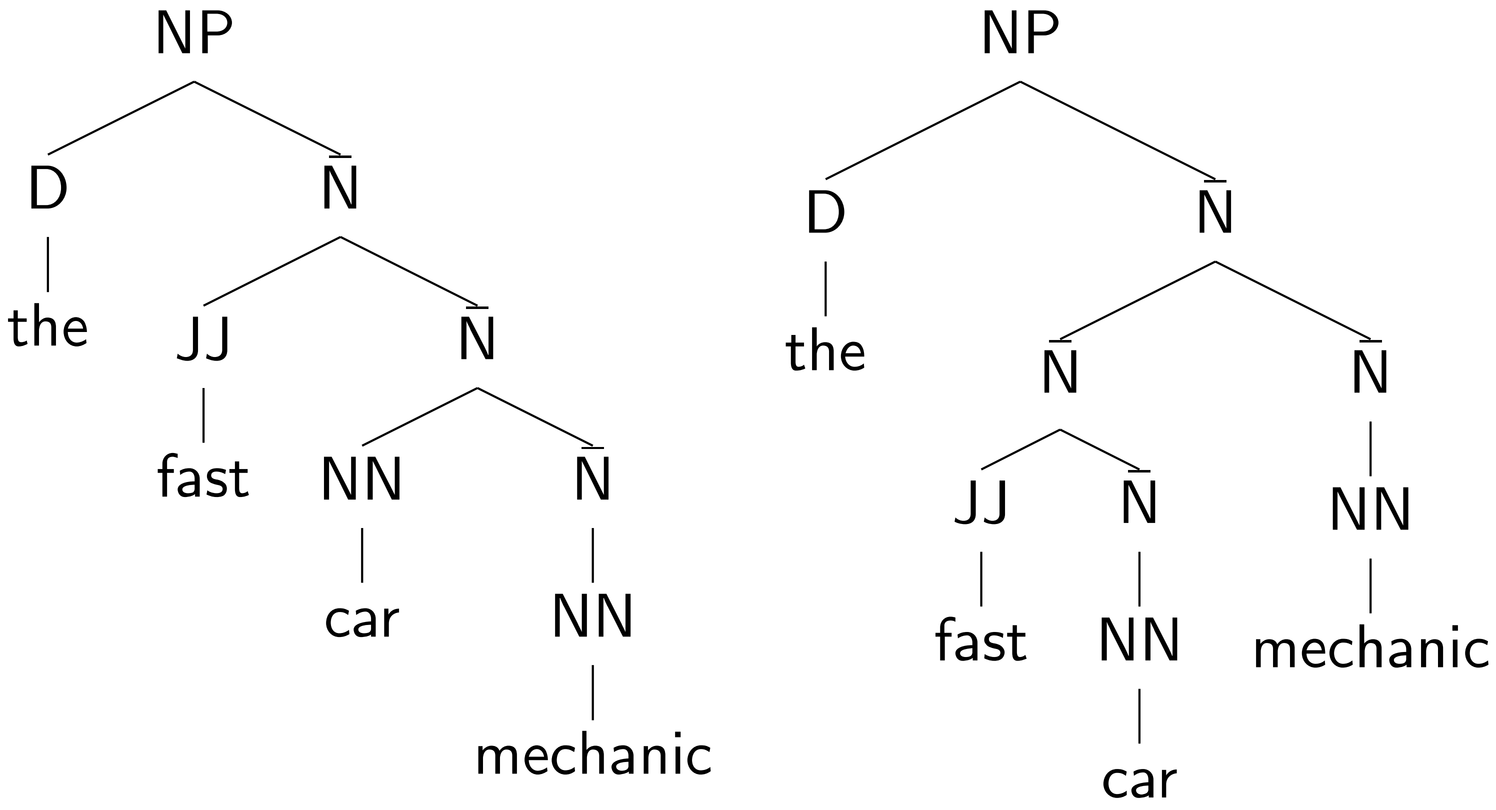


An Example of Ambiguity (continued)



Sources of Ambiguity: Noun Premodifiers

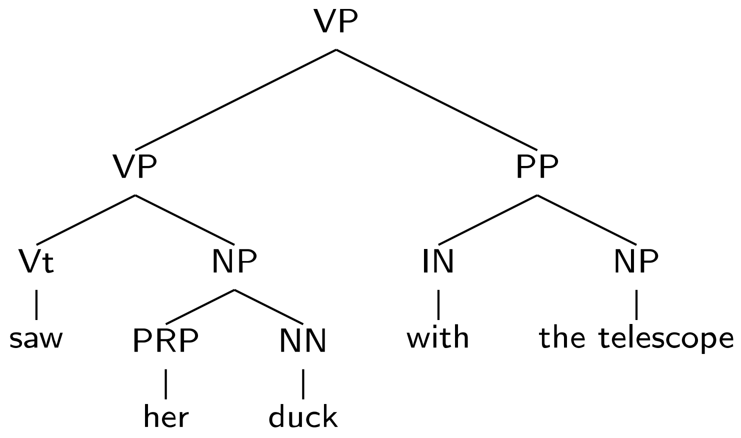
- Noun premodifiers:

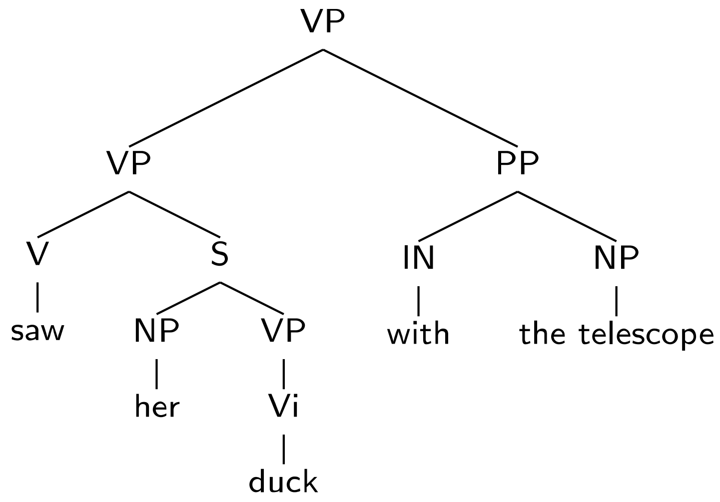


Sources of Ambiguity

- Part-of-Speech ambiguity

$$\begin{array}{lll} \text{NN} & \rightarrow & \text{duck} \\ \text{Vi} & \rightarrow & \text{duck} \end{array}$$





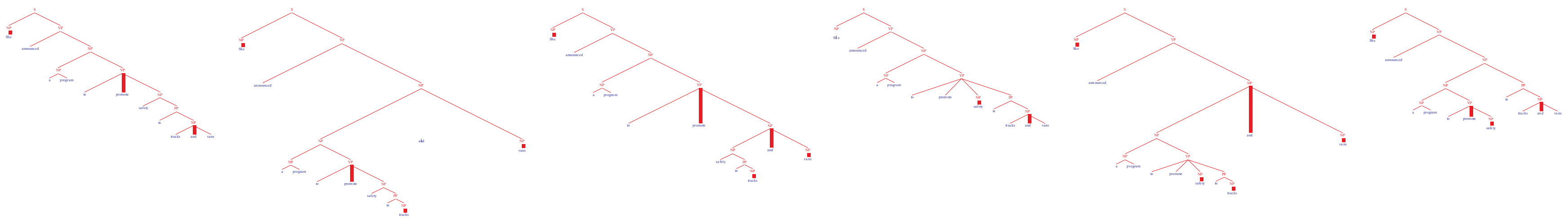
The Problem with Parsing: Ambiguity

INPUT:

She announced a program to promote safety in trucks and vans

+

POSSIBLE OUTPUTS:



And there are more...