

Introduction to Parsing

Data Structures and Algorithms for Computational Linguistics III
(ISCL-BA-07)

Çağrı Çöltekin

`ccoltekin@sfs.uni-tuebingen.de`

University of Tübingen
Seminar für Sprachwissenschaft

Winter Semester 2022/23

What is parsing?

- *Parsing* is the task of analyzing a string of symbols to discover its (inherent) structure
- Typically, the structure (and the valid strings in the language) is defined by a *grammar*
- The output of a parser is a structured representation of the input string, often a tree
- *Recognition* is an intimately related task which determines whether a given string is in a language

Ingredients of a parser

(for natural language parsing)

- A formal grammar defining a language of interest
- An algorithm that (efficiently) verifies whether a given string is in the language (recognizer) and enumerates the grammar rules used for verification (parser)
- A system for ambiguity resolution (not in this course)

Grammars

- A grammar is a finite specification of a possibly infinite language

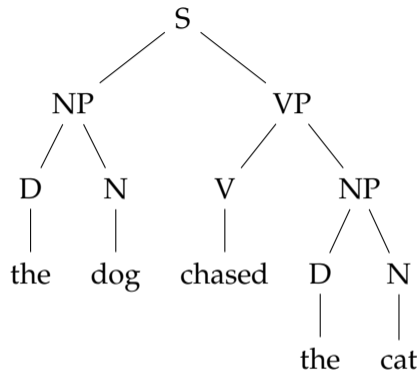
Grammars

- A grammar is a finite specification of a possibly infinite language
- The most commonly studied type of grammars are *phrase structure grammars*

$S \rightarrow NP VP$	$NP \rightarrow D N$	$VP \rightarrow V NP$	
$V \rightarrow \text{chased}$	$D \rightarrow \text{the}$	$N \rightarrow \text{cat}$	$N \rightarrow \text{dog}$

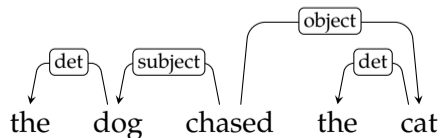
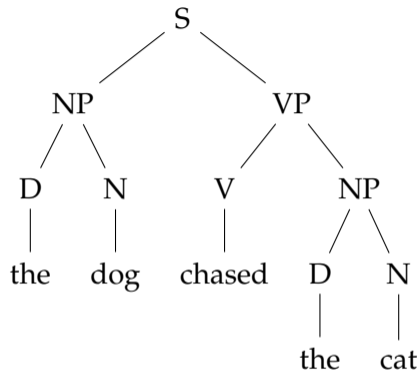
Grammars

- A grammar is a finite specification of a possibly infinite language
- The most commonly studied type of grammars are *phrase structure grammars*
- Analysis using context-free grammars result in *constituency* or *phrase structure trees*


 $S \rightarrow NP VP$
 $NP \rightarrow D N$
 $VP \rightarrow V NP$
 $V \rightarrow \text{chased}$
 $D \rightarrow \text{the}$
 $N \rightarrow \text{cat}$
 $N \rightarrow \text{dog}$

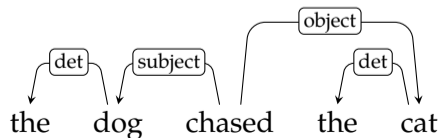
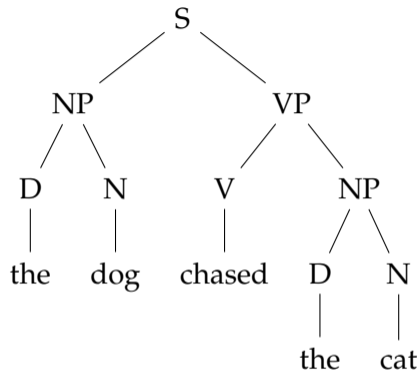
Why study parsing?

- In general, it is an intermediate step for interpreting sentences

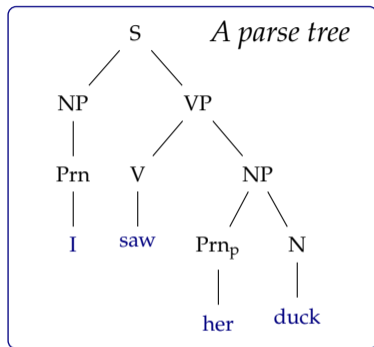


Why study parsing?

- In general, it is an intermediate step for interpreting sentences
- Applications include:
 - Compiler construction
 - Grammar checking
 - Sentiment analysis
 - Information (e.g., relation) extraction
 - Argument mining
 - ...



Different ways to represent a context-free parse

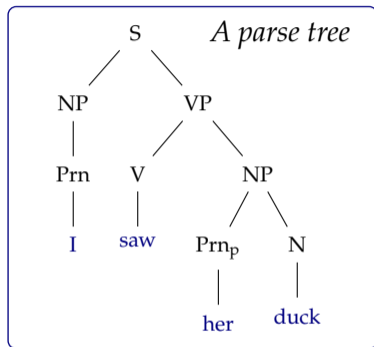


A history of derivations

Sentential form	derivation
S	(start)
NP VP	S \Rightarrow NP VP
Prn VP	NP \Rightarrow Prn
I VP	Prn \Rightarrow I
I V NP	VP \Rightarrow V NP
I saw NP	V \Rightarrow saw
I saw Prn _p N	NP \Rightarrow Prn _p N
I saw her N	Prn _p \Rightarrow her
I saw her duck	N \Rightarrow duck

(Labeled) brackets: $\left[{}_S \left[{}_{NP} \left[{}_{Prn} I \right] \right] \left[{}_{VP} \left[{}_V \text{ saw} \right] \left[{}_{NP} \left[{}_{Prn_p} \text{ her} \right] \left[{}_N \text{ duck} \right] \right] \right] \right]$

Different ways to represent a context-free parse



A history of derivations

Sentential form	derivation
S	(start)
NP VP	S → NP VP
Prn VP	NP → Prn
I VP	Prn → I
I V NP	VP → V NP
I saw NP	V → saw
I saw Prn _p N	NP → Prn _p N
I saw her N	Prn _p → her
I saw her duck	N → duck

(Labeled) brackets: $\left[{}_S \left[{}_{NP} \left[{}_{Prn} I \right] \right] \left[{}_{VP} \left[{}_V \text{ saw} \right] \left[{}_{NP} \left[{}_{Prn_p} \text{ her} \right] \left[{}_N \text{ duck} \right] \right] \right] \right]$

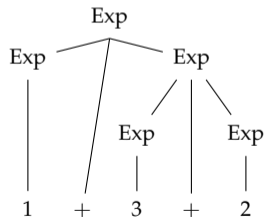
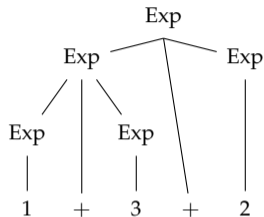
Relation between different representations

- The parse tree and the bracket representation is equivalent
 - parse trees are easier to read by humans
 - brackets are easier for computers
 - brackets are the typical representation for treebanks
- A parse tree (or bracket representation) can be obtained with a different order of production rules

Grammars and ambiguity

$\text{Exp} \rightarrow n$
 $\text{Exp} \rightarrow \text{Exp} + \text{Exp}$
 (terminal symbol 'n' stands for any number)

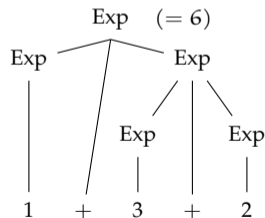
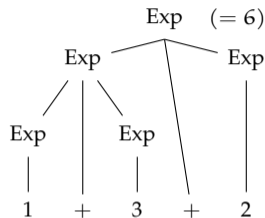
- If a grammar is ambiguous, some sentences produce multiple analyses
- If the resulting analysis lead to the same semantics, the ambiguity is *spurious*



Grammars and ambiguity

$\text{Exp} \rightarrow n$
 $\text{Exp} \rightarrow \text{Exp} + \text{Exp}$
 (terminal symbol 'n' stands for any number)

- If a grammar is ambiguous, some sentences produce multiple analyses
- If the resulting analysis lead to the same semantics, the ambiguity is *spurious*



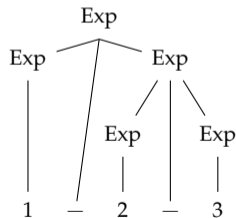
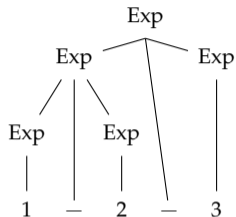
Grammars and ambiguity

$$\text{Exp} \rightarrow n$$

$$\text{Exp} \rightarrow \text{Exp} - \text{Exp}$$

(terminal symbol 'n' stands for any number)

- Is this ambiguity spurious?



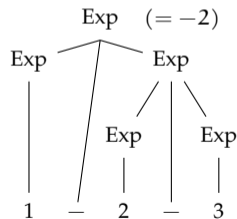
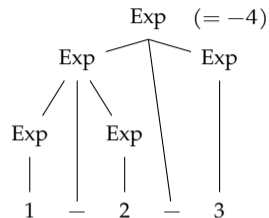
Grammars and ambiguity

$$\text{Exp} \rightarrow n$$

$$\text{Exp} \rightarrow \text{Exp} - \text{Exp}$$

(terminal symbol 'n' stands for any number)

- Is this ambiguity spurious?
- If different structures yield different semantics, the ambiguity is *essential*



Ambiguity can be removed from a grammar

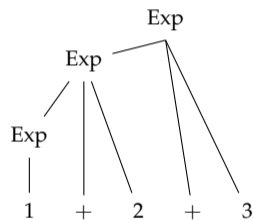
if the language is not ambiguous

$\text{Exp} \rightarrow n$
 $\text{Exp} \rightarrow \text{Exp} + n$
 (terminal symbol 'n' stands for any number)

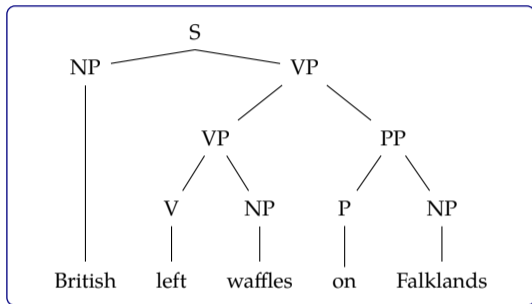
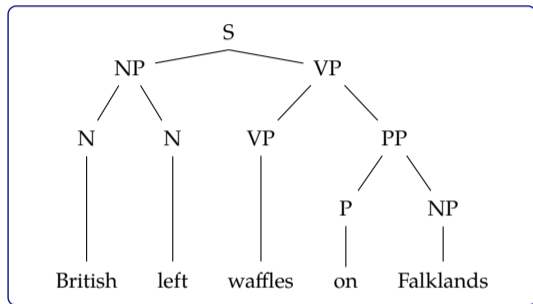
- The grammar above does not have the ambiguity of

$\text{Exp} \rightarrow n$
 $\text{Exp} \rightarrow \text{Exp} + \text{Exp}$

- Both grammars define the same language



Natural languages are ambiguous



- The grammars we define have to distinguish between two different structures
- We need methods for ranking analyses

Top-down parsing

general idea

- Start from S , find a sequence of derivations that yield the sentence
- This is simply the same as the generation procedure we discussed earlier
- Attempt to generate all strings from a grammar, but allow only the productions that 'produce' the input string

Top-down: demonstration

the cat bites a dog

S → NP VP
NP → Det N
VP → V NP
VP → V
Det → a
Det → the
N → cat
N → dog
V → bites

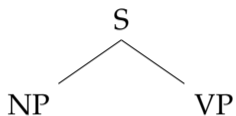
Top-down: demonstration

S

the cat bites a dog

S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

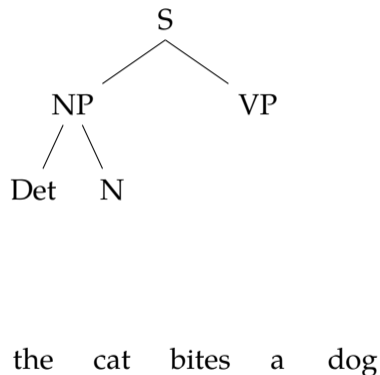
Top-down: demonstration



the cat bites a dog

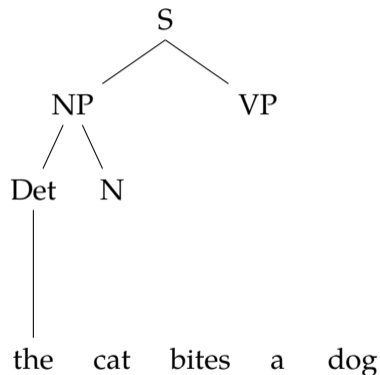
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



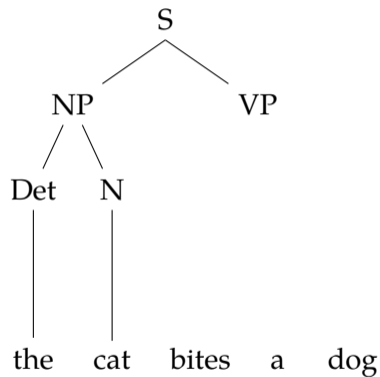
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



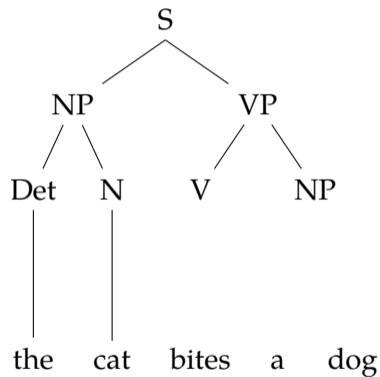
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



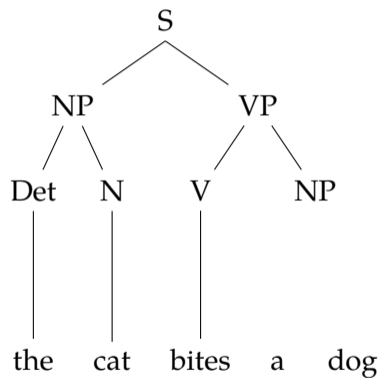
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



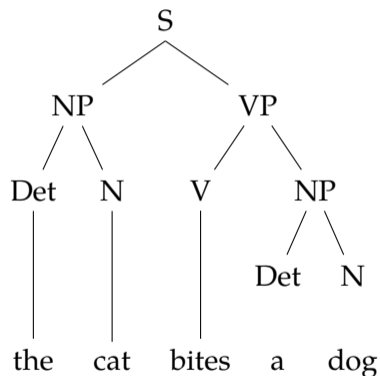
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



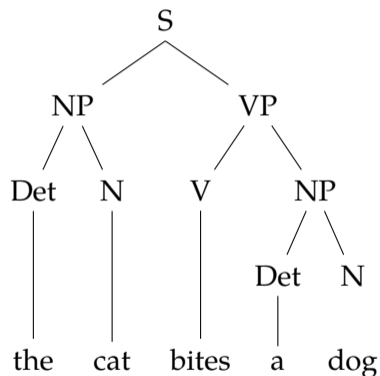
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



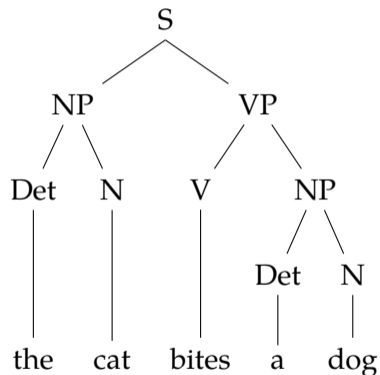
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Top-down: demonstration



S → NP VP
 NP → Det N
 VP → V NP
 VP → V
 Det → a
 Det → the
 N → cat
 N → dog
 V → bites

From demonstration to parsing

- There may be multiple productions applicable
- We need an automatic mechanism to select the correct productions
- We have two actions:
 - predict generate a hypothesis based on the grammar
 - match when a terminal symbol is produced, check if it matches with the one in the expected position
 - if matched, continue
 - otherwise, backtrack
- if we eliminate all non terminals from the sentential form, and the complete input string is matched (produced), then parsing successful

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

matched	goal	production
	S	$S \Rightarrow NP VP$

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

matched	goal	production
	S	$S \Rightarrow NP VP$
	NP VP	$NP \Rightarrow Det VP$

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a X

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗
the cat	VP	VP ⇒ V NP

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗
the cat	VP	VP ⇒ V NP
the cat	V NP	V ⇒ bites ✓

parse: *the cat bites a dog*

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗
the cat	VP	VP ⇒ V NP
the cat	V NP	V ⇒ bites ✓
the cat bites	Det N	NP ⇒ Det N

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗
the cat	VP	VP ⇒ V NP
the cat	V NP	V ⇒ bites ✓
the cat bites	Det N	NP ⇒ Det N
the cat bites a	N	Det ⇒ a ✓

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗
the cat	VP	VP ⇒ V NP
the cat	V NP	V ⇒ bites ✓
the cat bites	Det N	NP ⇒ Det N
the cat bites a	N	Det ⇒ a ✓
the cat bites a dog		Det ⇒ dog ✓

Top-down parsing: another demonstration

the grammar	
S	→ NP VP
NP	→ Det N
VP	→ V NP
VP	→ V
Det	→ a
Det	→ the
N	→ cat
N	→ dog
V	→ bites

parse: *the cat bites a dog*

matched	goal	production
	S	S ⇒ NP VP
	NP VP	NP ⇒ Det VP
	Det N VP	Det ⇒ a ✗
	Det N VP	Det ⇒ the ✓
the	N VP	N ⇒ dog ✗
the cat	N VP	N ⇒ cat ✓
the cat	VP	VP ⇒ V
the cat bites	V	V ⇒ bites ✓
the cat bites		(not at the end) ✗
the cat	VP	VP ⇒ V NP
the cat	V NP	V ⇒ bites ✓
the cat bites	Det N	NP ⇒ Det N
the cat bites a	N	Det ⇒ a ✓
the cat bites a dog		Det ⇒ dog ✓

Note that the valid productions yield the parse tree.

Top-down parsing: problems and possible solutions

- The trial-and-error procedure leads to exponential time parsing

Top-down parsing: problems and possible solutions

- The trial-and-error procedure leads to exponential time parsing
- But lots of repeated work: dynamic programming may help avoid it

Top-down parsing: problems and possible solutions

- The trial-and-error procedure leads to exponential time parsing
- But lots of repeated work: dynamic programming may help avoid it
- What happens if we had a rule like

$$\text{NP} \rightarrow \text{NP PP}$$

Top-down parsing: problems and possible solutions

- The trial-and-error procedure leads to exponential time parsing
- But lots of repeated work: dynamic programming may help avoid it
- What happens if we had a rule like

$$\text{NP} \rightarrow \text{NP PP}$$

Top-down parsing: problems and possible solutions

- The trial-and-error procedure leads to exponential time parsing
- But lots of repeated work: dynamic programming may help avoid it
- What happens if we had a rule like

$$\text{NP} \rightarrow \text{NP PP}$$

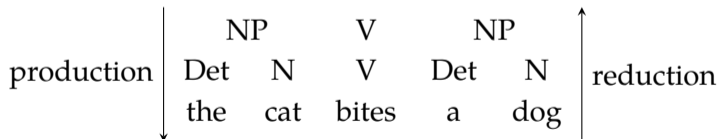
some rules may cause infinite loops

- Notice that if we knew which terminals are possible as the initial part of a non-terminal symbol, we can eliminate the unsuccessful matches earlier

Bottom-up parsing

general idea

- Start from from the input symbols, and try to *reduce* the input to start symbol
- We need to match parts of the sentential form (starting from the input) to the RHS of the grammar rules
- While top-down process relies on *productions* the bottom-up process relies on *reductions*

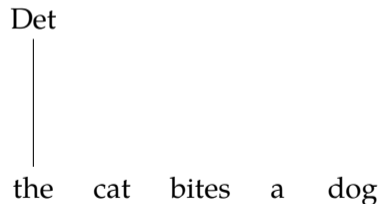


Bottom-up: demonstration

the cat bites a dog

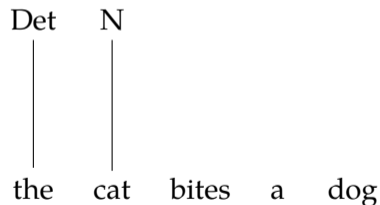
S → NP VP
NP → Det N
VP → V NP
VP → V
Det → a
Det → the
N → cat
N → dog
V → bites

Bottom-up: demonstration



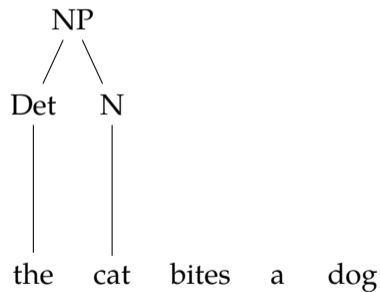
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Bottom-up: demonstration



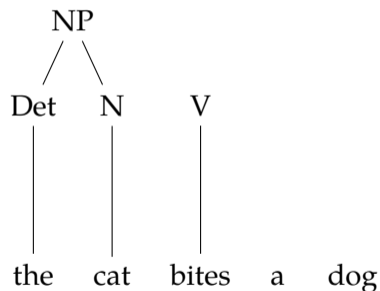
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Bottom-up: demonstration



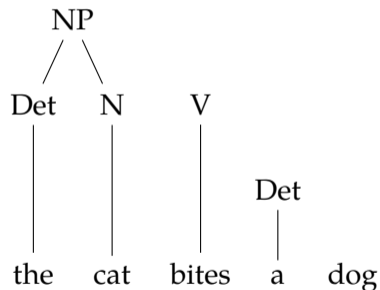
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Bottom-up: demonstration



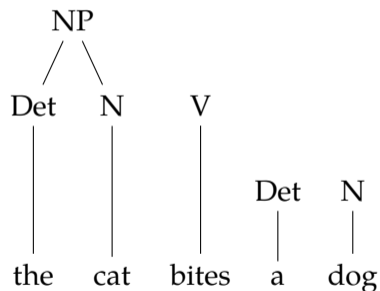
S \rightarrow NP VP
 NP \rightarrow Det N
 VP \rightarrow V NP
 VP \rightarrow V
 Det \rightarrow a
 Det \rightarrow the
 N \rightarrow cat
 N \rightarrow dog
 V \rightarrow bites

Bottom-up: demonstration



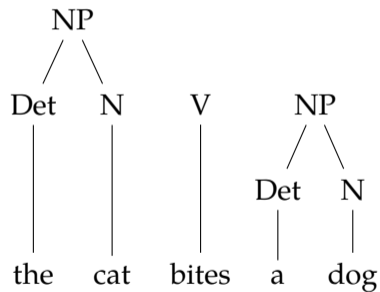
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Bottom-up: demonstration



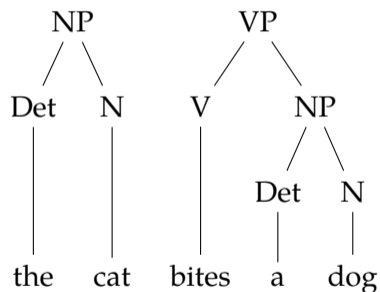
S	→	NP VP
NP	→	Det N
VP	→	V NP
VP	→	V
Det	→	a
Det	→	the
N	→	cat
N	→	dog
V	→	bites

Bottom-up: demonstration



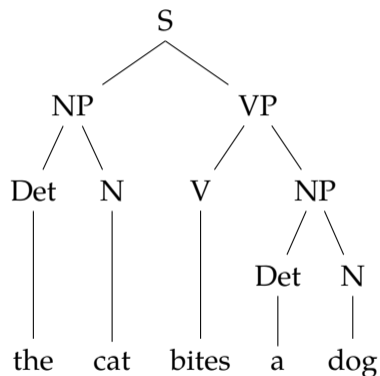
S → NP VP
 NP → Det N
 VP → V NP
 VP → V
 Det → a
 Det → the
 N → cat
 N → dog
 V → bites

Bottom-up: demonstration



S → NP VP
 NP → Det N
 VP → V NP
 VP → V
 Det → a
 Det → the
 N → cat
 N → dog
 V → bites

Bottom-up: demonstration



S → NP VP
 NP → Det N
 VP → V NP
 VP → V
 Det → a
 Det → the
 N → cat
 N → dog
 V → bites

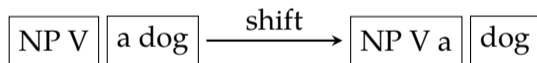
A (first) introduction to shift-reduce parsing

- We keep two data structures:
 - a stack for the (partially) reduced sentential form
 - an input queue that contains only terminal symbols

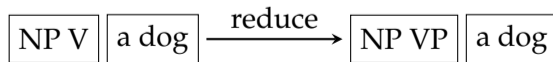


- We use two operations:

shift shifts a terminal to stack



reduce when top symbols on stack match a RHS, replace them with the LHS of the rule



Shift-reduce (bottom-up) parsing a demonstration

stack input

rule

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V
NP VP	a dog	S \Rightarrow NP VP

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V
NP VP	a dog	S \Rightarrow NP VP
S	a dog	shift

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V
NP VP	a dog	S \Rightarrow NP VP
S	a dog	shift
S a	dog	Det \Rightarrow A

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V
NP VP	a dog	S \Rightarrow NP VP
S	a dog	shift
S a	dog	Det \Rightarrow A
S Det dog		N \Rightarrow dog

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V
NP VP	a dog	S \Rightarrow NP VP
S	a dog	shift
S a	dog	Det \Rightarrow A
S Det dog		N \Rightarrow dog
S Det N		NP \Rightarrow Det N

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule
	the cat bites a dog	shift
the	cat bites a dog	Det \Rightarrow the
Det	cat bites a dog	shift
Det cat	bites a dog	N \Rightarrow cat
Det N	bites a dog	NP \Rightarrow Det N
NP	bites a dog	shift
NP bites	a dog	V \Rightarrow bites
NP V	a dog	VP \Rightarrow V
NP VP	a dog	S \Rightarrow NP VP
S	a dog	shift
S a	dog	Det \Rightarrow A
S Det dog		N \Rightarrow dog
S Det N		NP \Rightarrow Det N
S NP		(stuck)

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the			
Det	cat bites a dog	shift			
Det cat	bites a dog	N \Rightarrow cat			
Det N	bites a dog	NP \Rightarrow Det N			
NP	bites a dog	shift			
NP bites	a dog	V \Rightarrow bites			
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift			
Det cat	bites a dog	N \Rightarrow cat			
Det N	bites a dog	NP \Rightarrow Det N			
NP	bites a dog	shift			
NP bites	a dog	V \Rightarrow bites			
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift	NP V Det	dog	shift
Det cat	bites a dog	N \Rightarrow cat			
Det N	bites a dog	NP \Rightarrow Det N			
NP	bites a dog	shift			
NP bites	a dog	V \Rightarrow bites			
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift	NP V Det	dog	shift
Det cat	bites a dog	N \Rightarrow cat	NP V Det dog		N \Rightarrow dog
Det N	bites a dog	NP \Rightarrow Det N			
NP	bites a dog	shift			
NP bites	a dog	V \Rightarrow bites			
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift	NP V Det	dog	shift
Det cat	bites a dog	N \Rightarrow cat	NP V Det dog		N \Rightarrow dog
Det N	bites a dog	NP \Rightarrow Det N	NP V Det N		NP \Rightarrow Det N
NP	bites a dog	shift			
NP bites	a dog	V \Rightarrow bites			
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift	NP V Det	dog	shift
Det cat	bites a dog	N \Rightarrow cat	NP V Det dog		N \Rightarrow dog
Det N	bites a dog	NP \Rightarrow Det N	NP V Det N		NP \Rightarrow Det N
NP	bites a dog	shift	NP V NP		VP \Rightarrow V NP
NP bites	a dog	V \Rightarrow bites			
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift	NP V Det	dog	shift
Det cat	bites a dog	N \Rightarrow cat	NP V Det dog		N \Rightarrow dog
Det N	bites a dog	NP \Rightarrow Det N	NP V Det N		NP \Rightarrow Det N
NP	bites a dog	shift	NP V NP		VP \Rightarrow V NP
NP bites	a dog	V \Rightarrow bites	NP VP		S \Rightarrow NP VP
NP V	a dog	VP \Rightarrow V			
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

Shift-reduce (bottom-up) parsing a demonstration

stack	input	rule	stack	input	rule
	the cat bites a dog	shift	NP V	a dog	shift
the	cat bites a dog	Det \Rightarrow the	NP V a	dog	Det \Rightarrow a
Det	cat bites a dog	shift	NP V Det	dog	shift
Det cat	bites a dog	N \Rightarrow cat	NP V Det dog		N \Rightarrow dog
Det N	bites a dog	NP \Rightarrow Det N	NP V Det N		NP \Rightarrow Det N
NP	bites a dog	shift	NP V NP		VP \Rightarrow V NP
NP bites	a dog	V \Rightarrow bites	NP VP		S \Rightarrow NP VP
NP V	a dog	VP \Rightarrow V	S		(done)
NP VP	a dog	S \Rightarrow NP VP			
S	a dog	shift			
S a	dog	Det \Rightarrow A			
S Det dog		N \Rightarrow dog			
S Det N		NP \Rightarrow Det N			
S NP		(stuck)			

- All input reduced to S, accept
- Rules form the parse tree

Summary

- Parsing can be formulated as a top-down or bottom-up search (the search may also be depth-first or breadth first)
- Naive parsing algorithms are inefficient (exponential time complexity)
- There are some directions: dynamic programming, filtering
- Suggested reading (for constituency parsing): Jurafsky and Martin (2009, draft 3rd ed, chapters 12 & 13)
- A general reference for parsing: Grune and Jacobs (2007)

Summary

- Parsing can be formulated as a top-down or bottom-up search (the search may also be depth-first or breadth first)
- Naive parsing algorithms are inefficient (exponential time complexity)
- There are some directions: dynamic programming, filtering
- Suggested reading (for constituency parsing): Jurafsky and Martin (2009, draft 3rd ed, chapters 12 & 13)
- A general reference for parsing: Grune and Jacobs (2007)

Next:

- Bottom-up chart parsing: CKY algorithm
- Suggested reading: Jurafsky and Martin (2009, draft 3rd ed, section 13.2)

