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 2021/22

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

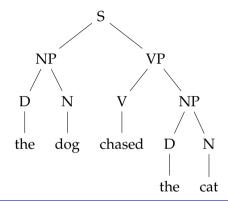
• A grammar is a finite specification of a possibly infinite language

Grammars

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

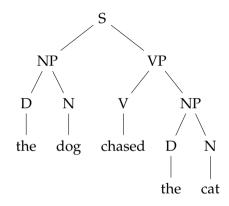
Grammars

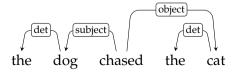
- A grammar is a finite specification of a possibly infinite language
- The most commonly studied type of grammars are *phrase structure* grammars
- Analysis using context-free grammars result in constituency or phrase structure trees



Why study parsing?

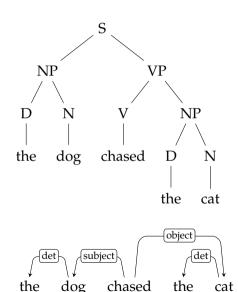
• In general, it is an intermediate step for interpreting sentences



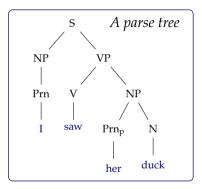


Why study parsing?

- 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

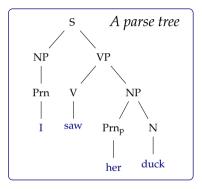


A history o	of derivations
Sentential form	derivation
S	(start)
NP VP	$S \Rightarrow NP VP$
Prn VP	$NP \Rightarrow Prn$
I VP	$Prn \Rightarrow I$
I V NP	$VP \Rightarrow \ V \ NP$
I saw NP	$V \Rightarrow saw$
I saw Prn _p N	$NP \Rightarrow Prn_p N$
I saw her N	$Prn_p \Rightarrow her$
I saw her duck	$N \rightarrow duck$

$$\textit{(Labeled) brackets: } \left[{}_{S} \left[{}_{NP} \left[{}_{Prn} \ I \right] \right] \left[{}_{VP} \left[{}_{V} \ saw \right] \left[{}_{NP} \left[{}_{Prn_p} \ her \right] \left[{}_{N} \ duck \right] \right] \right]$$

Ç. Çöltekin, SfS / University of Tübingen

Different ways to represent a context-free parse



A history o	of derivations
Sentential form	derivation
S	(start)
NP VP	$S \longrightarrow NP VP$
Prn VP	$NP \rightarrow Prn$
I VP	$Prn \rightarrow I$
I V NP	$VP \longrightarrow V NP$
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:
$$\left[\sum_{S} \left[\sum_{NP} \left[P_{rn} I \right] \right] \left[\sum_{VP} \left[V_{S} \text{ saw} \right] \left[\sum_{NP} \left[P_{rn_{\mathfrak{p}}} \text{ her} \right] \left[V_{N} \text{ duck} \right] \right] \right] \right]$$

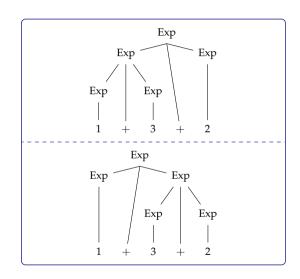
Ç. Çöltekin, SfS / University of Tübingen

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

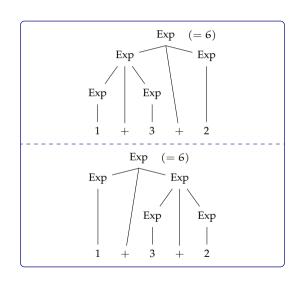
$$\begin{array}{ccc} Exp & \rightarrow & n \\ Exp & \rightarrow & Exp + Exp \\ \text{(terminal symbol 'n' stands for any number)} \end{array}$$

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



$$\begin{array}{ccc} Exp & \rightarrow & n \\ Exp & \rightarrow & Exp + Exp \\ \text{(terminal symbol 'n' stands for any number)} \end{array}$$

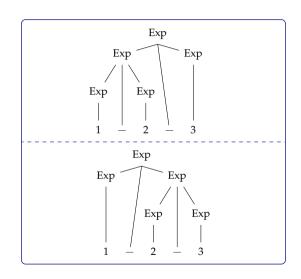
- If a grammar is ambiguous, some sentences produce multiple analyses
- If the resulting analysis lead to the same semantics, the ambiguity is spurious



$$\begin{array}{ccc} Exp & \rightarrow & n \\ Exp & \rightarrow & Exp - Exp \end{array}$$

(terminal symbol 'n' stands for any number)

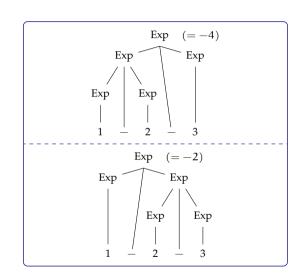
• Is this ambiguity spurious?



$$\begin{array}{l} Exp \ \rightarrow \ n \\ Exp \ \rightarrow \ Exp - Exp \end{array}$$

(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

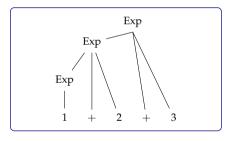
if the language is not ambiguous

$$egin{array}{l} \operatorname{Exp} &
ightarrow & \operatorname{n} \ \operatorname{Exp} &
ightarrow & \operatorname{Exp} + \operatorname{n} \ & ext{(terminal symbol 'n' stands for any number)} \end{array}$$

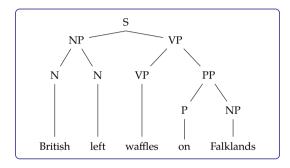
• The grammar above does not have the ambiguity of

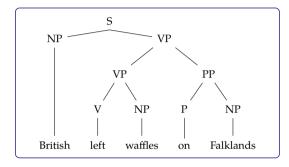
$$\begin{array}{l} Exp \ \rightarrow \ n \\ Exp \ \rightarrow \ Exp \ + Exp \end{array}$$

• 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

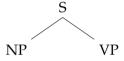
 $\begin{array}{ccc} S & \rightarrow & NP \ VP \\ NP & \rightarrow & Det \ N \\ VP & \rightarrow & V \ NP \\ VP & \rightarrow & V \\ Det & \rightarrow & a \\ Det & \rightarrow & the \\ N & \rightarrow & cat \\ N & \rightarrow & dog \\ V & \rightarrow & bites \\ \end{array}$

the cat bites a dog

S

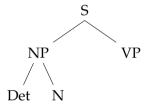
the cat bites a dog

 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \ \to \ V$ Det \rightarrow a Det \rightarrow the $N \rightarrow cat$ $N \quad \to \ dog$ \rightarrow bites



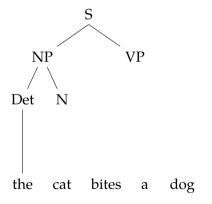
the cat bites a dog

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

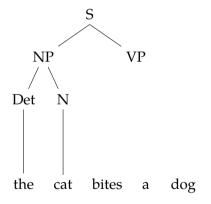


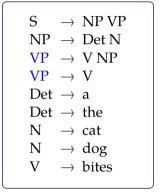
the cat bites a dog

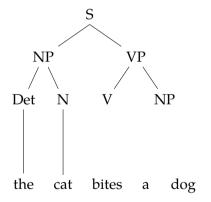
 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ $Det \rightarrow a$ $Det \rightarrow the$ \rightarrow cat $N \rightarrow dog$ \rightarrow bites

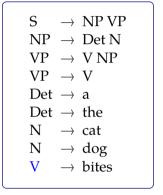


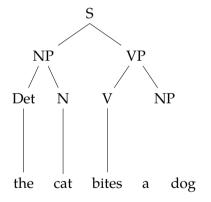
 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the \rightarrow cat \rightarrow dog \rightarrow bites

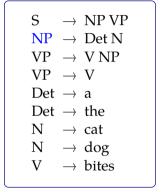


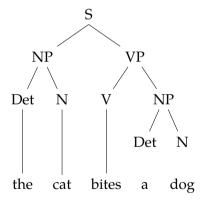




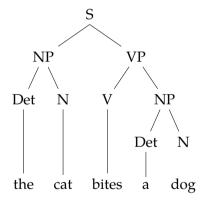


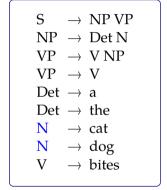


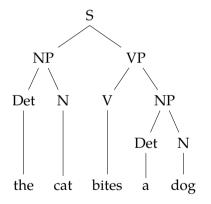




 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ $Det \rightarrow a$ $Det \rightarrow the$ \rightarrow cat \rightarrow dog \rightarrow bites







 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the \rightarrow cat \rightarrow dog \rightarrow 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

the grammar \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow VNP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the $N \rightarrow cat$ $N \rightarrow dog$ \rightarrow bites V

parse: the cat bites a dog

the grammar					
NP VP	$\overset{\longrightarrow}{\rightarrow}$	NP VP Det N V NP			
VP Det					
Det	\rightarrow	the			
N	\rightarrow	cat			
N		dog			
V	\rightarrow	bites			

m	atched	goal	production
		S	$S \Rightarrow NP VP$

parse: the cat bites a dog

the grammar		matched	goal
ene grammar			S
$S \to NP VP$			NP VP
$NP \rightarrow Det N$			
$ ext{VP} \ o \ ext{V NP}$			
$ ext{VP} \; o \; ext{V}$			
$\mathrm{Det} o \mathrm{a}$			
Det $ ightarrow$ the			
$N \rightarrow cat$			
$N \rightarrow dog$			
$ ext{V} ightarrow ext{bites}$			
	J		

parse: the cat bites a dog

production

 $S \Rightarrow NP VP$

 $NP \Rightarrow Det VP$

the gr	ammar	
NP VP	$ \rightarrow NP VP \rightarrow Det N \rightarrow V NP $	
Det	$ ightarrow V \ ightarrow a \ ightarrow the$	
N	\rightarrow the \rightarrow cat \rightarrow dog	
	\rightarrow bites	

matched	goai	production
	S NP VP Det N VP	$S \Rightarrow NP VP$ $NP \Rightarrow Det VP$ $Det \Rightarrow a X$

parse: the cat bites a dog

the grammar		matched	goal	production
$\begin{array}{c} \text{the grammar} \\ S & \rightarrow & \text{NP VP} \\ \text{NP} & \rightarrow & \text{Det N} \\ \text{VP} & \rightarrow & \text{V NP} \\ \text{VP} & \rightarrow & \text{V} \\ \text{Det} & \rightarrow & \text{a} \\ \text{Det} & \rightarrow & \text{the} \\ \text{N} & \rightarrow & \text{cat} \\ \text{N} & \rightarrow & \text{dog} \\ \text{V} & \rightarrow & \text{bites} \\ \end{array}$		matched	S NP VP	$S \Rightarrow NP VP$ $NP \Rightarrow Det VP$ $Det \Rightarrow a \times X$ $Det \Rightarrow the \checkmark$
$V \rightarrow \text{bites}$				

parse: the cat bites a dog

the grammar	n	natched	goal	production
$\begin{array}{c} S & \rightarrow & NP \ VP \\ NP & \rightarrow & Det \ N \\ VP & \rightarrow & V \ NP \\ VP & \rightarrow & V \\ Det & \rightarrow & a \\ Det & \rightarrow & the \\ N & \rightarrow & cat \\ N & \rightarrow & dog \\ V & \rightarrow & bites \\ \end{array}$		the	S NP VP Det N VP Det N VP N VP	$S \Rightarrow NP VP$ $NP \Rightarrow Det VP$ $Det \Rightarrow a X$ $Det \Rightarrow the \checkmark$ $N \Rightarrow dog X$

parse: the cat bites a dog

parse: the cat bites a dog

the grammar		matched	goal	production
$\begin{array}{cccccccccccccccccccccccccccccccccccc$		the the cat the cat	S NP VP Det N VP Det N VP N VP N VP	$S \Rightarrow NP \ VP$ $NP \Rightarrow Det \ VP$ $Det \Rightarrow a \ X$ $Det \Rightarrow the \ \checkmark$ $N \Rightarrow dog \ X$ $N \Rightarrow cat \ \checkmark$ $VP \Rightarrow V$

the grammar	matched	goal	production
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	the the cat the cat the cat bites	VP	$S \Rightarrow NP VP$ $NP \Rightarrow Det VP$ $Det \Rightarrow a X$ $Det \Rightarrow the \checkmark$ $N \Rightarrow dog X$ $N \Rightarrow cat \checkmark$ $VP \Rightarrow V$ $V \Rightarrow bites \checkmark$

the grammar	matched	goal	production
$\begin{array}{ccc} S & \rightarrow & NP \ VP \\ NP & \rightarrow & Det \ N \end{array}$		S NP VP Det N VP	$S \Rightarrow NP VP$ $NP \Rightarrow Det VP$ $Det \Rightarrow a X$
$egin{array}{lll} {\sf VP} & ightarrow {\sf V} & {\sf NP} \\ {\sf VP} & ightarrow {\sf V} \\ {\sf Det} & ightarrow {\sf a} \\ \end{array}$	the the cat the cat	Det N VP N VP N VP	Det \Rightarrow the \checkmark $N \Rightarrow \deg \checkmark$ $N \Rightarrow \cot \checkmark$ $VP \Rightarrow V$
$egin{array}{lll} \operatorname{Det} & ightarrow & \operatorname{the} & & & & & & \\ N & ightarrow & \operatorname{cat} & & & & & & \\ N & ightarrow & \operatorname{dog} & & & & & & \\ V & ightarrow & \operatorname{bites} & & & & & & \end{array}$	the cat bites the cat bites	V	$V \Rightarrow \text{bites } \checkmark$ (not at the end) X

the grammar	matched	goal	production
the grammar $\begin{array}{ccc} S & \rightarrow & NP \ VP \\ NP & \rightarrow & Det \ N \\ VP & \rightarrow & V \ NP \\ VP & \rightarrow & V \\ Det & \rightarrow & a \\ Det & \rightarrow & the \\ N & \rightarrow & cat \\ N & \rightarrow & dog \\ V & \rightarrow & bites \\ \end{array}$	the the cat the cat	S NP VP Det N VP Det N VP N VP N VP VP	$S \Rightarrow NP \ VP$ $NP \Rightarrow Det \ VP$ $Det \Rightarrow a \ X$ $Det \Rightarrow the \ \checkmark$ $N \Rightarrow dog \ X$ $N \Rightarrow cat \ \checkmark$ $VP \Rightarrow V$ $V \Rightarrow bites \ \checkmark$ $(not at the end) \ X$ $VP \Rightarrow V \ NP$

the grammar		matched	goal	production
5110 81 211111			S	$S \Rightarrow NP VP$
$S \to NP VP$			NP VP	$NP \Rightarrow Det VP$
$NP \rightarrow Det N$			Det N VP	$\mathrm{Det} \Rightarrow \mathrm{a} \mathbf{X}$
$ ext{VP} \ o \ ext{V NP}$			Det N VP	Det \Rightarrow the \checkmark
$ ext{VP} \; o \; ext{V}$		the	N VP	$N \Rightarrow dog X$
$\mathrm{Det} o \mathrm{a}$		the cat	N VP	$N \Rightarrow cat \checkmark$
$Det \to the$		the cat		$VP \Rightarrow V$
			V	$V \Rightarrow bites \checkmark$
- 1		the cat bites		(not at the end) X
$ ext{N} o ext{dog}$		the cat	V NP	$VP \Rightarrow V NP$
$ ext{V} ightarrow ext{bites}$		the cat bites	NP	$V \Rightarrow bites \checkmark$
	J			

the grammar	matched	goal	production
5.10 B. 5.11		S	$S \Rightarrow NP VP$
$S \rightarrow NP VP$		NP VP	$NP \Rightarrow Det VP$
$NP \rightarrow Det N$		Det N VP	$Det \Rightarrow a X$
$ ext{VP} \ o \ ext{V} \ ext{NP}$		Det N VP	Det \Rightarrow the \checkmark
$ ext{VP} \ o \ ext{V}$	the	N VP	$N \Rightarrow dog X$
$Det \rightarrow a$	the cat	N VP	$N \Rightarrow cat \checkmark$
$Det \rightarrow the$	the cat		$VP \Rightarrow V$
		V	$V \Rightarrow \text{bites} \checkmark$
_ , _ , _ , _ , _ , _ , _ , _ , _ , _ ,	the cat bites	****	(not at the end) X
$N \rightarrow dog$	the cat	V NP	$VP \Rightarrow V NP$
$V \rightarrow bites$	the cat bites		$V \Rightarrow \text{bites} \checkmark$
	the cat bites	Det N	$NP \Rightarrow Det N$

the grammar	matched	goal	production
sire gramma.		S	$S \Rightarrow NP VP$
$S \rightarrow NP VP$		NP VP	$NP \Rightarrow Det VP$
$NP \rightarrow Det N$		Det N VP	$\mathrm{Det} \Rightarrow a X$
$VP \rightarrow V NP$		Det N VP	Det \Rightarrow the \checkmark
$ ext{VP} ightarrow ext{V}$	the	N VP	$N \Rightarrow dog X$
$Det \rightarrow a$	the cat	N VP	$N \Rightarrow cat \checkmark$
	the cat	VP	$VP \Rightarrow V$
$\operatorname{Det} \to \operatorname{the}$	the cat bites	V	$V \Rightarrow bites \checkmark$
$N \rightarrow cat$	the cat bites		(not at the end) X
m N ightarrow dog	the cat	V NP	$VP \Rightarrow V NP$
$ ext{V} ightarrow ext{bites}$	the cat bites	NP	$V \Rightarrow bites \checkmark$
	the cat bites	Det N	$NP \Rightarrow Det N$
	the cat bites a	N	Det \Rightarrow a \checkmark

the grammar				
NP	$\overset{\longrightarrow}{\rightarrow}$	NP VP Det N V NP V		
Det	\rightarrow	a		
Det	\rightarrow	the		
N	\rightarrow	cat		
		dog		
V	\rightarrow	bites		
l				

matched	goal	production
	S	$S \Rightarrow NP VP$
	NP VP	$NP \Rightarrow Det VP$
	Det N VP	$\mathrm{Det} \Rightarrow a X$
	Det N VP	Det \Rightarrow the \checkmark
the	N VP	$N \Rightarrow dog X$
the cat	N VP	$N \Rightarrow cat \checkmark$
the cat	VP	$VP \Rightarrow V$
the cat bites	V	$V \Rightarrow bites \checkmark$
the cat bites		(not at the end) X
the cat	V NP	$VP \Rightarrow V NP$
the cat bites	NP	$V \Rightarrow bites \checkmark$
the cat bites	Det N	$NP \Rightarrow Det N$
the cat bites a	N	Det \Rightarrow a \checkmark
the cat bites a dog		$Det \Rightarrow dog \checkmark$

the grammar	matched	goai
g		S
$S \to NP VP$		NP VP
$\mathrm{NP} o \mathrm{Det} \mathrm{N}$		Det N VP
$ ext{VP} \ o \ ext{V} \ ext{NP}$		Det N VP
$VP \rightarrow V$	the	N VP
$\begin{array}{ccc} VI & \to & V \\ Det & \to & a \end{array}$	the cat	N VP
	the cat	VP
Det o the	the cat bites	V
$N \rightarrow cat$	the cat bites	
$N \rightarrow dog$	the cat	V NP
$ ext{V} ightarrow ext{bites}$	the cat bites	NP
	the cat bites	Det N
	the cat bites a	N

parse: the cat bites a dog

Note that the valid productions yield the parse tree.

the cat bites a dog

matched

Det \Rightarrow a \checkmark

Det \Rightarrow dog \checkmark

production $S \Rightarrow NP VP$ $NP \Rightarrow Det VP$ Det $\Rightarrow a X$ Det \Rightarrow the \checkmark $N \Rightarrow dog X$ $N \Rightarrow cat \checkmark$ $VP \Rightarrow V$ $V \Rightarrow bites \checkmark$ (not at the end) X $VP \Rightarrow V NP$ $V \Rightarrow bites \checkmark$ $NP \Rightarrow Det N$

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

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

- 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

 $NP \rightarrow NP PP$

- 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

 $NP \rightarrow NP PP$

- 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

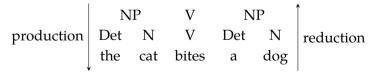
 $NP \rightarrow NP PP$

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

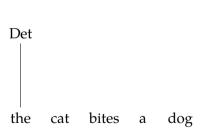
Bottom-up parsing general idea

- Start from from the input symbol, 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

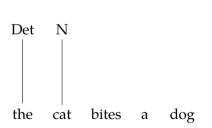


 $\begin{array}{ccc} S & \rightarrow & NP \ VP \\ NP & \rightarrow & Det \ N \\ VP & \rightarrow & V \ NP \\ VP & \rightarrow & V \\ Det & \rightarrow & a \\ Det & \rightarrow & the \\ N & \rightarrow & cat \\ N & \rightarrow & dog \\ V & \rightarrow & bites \\ \end{array}$

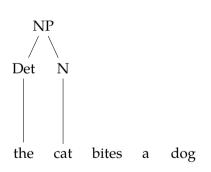
the cat bites a dog

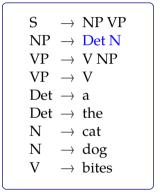


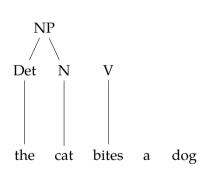
 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the \rightarrow cat $N \rightarrow dog$ \rightarrow bites

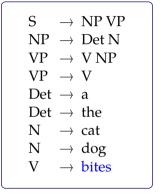


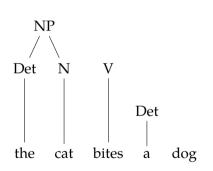
 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the \rightarrow cat $N \rightarrow dog$ \rightarrow bites

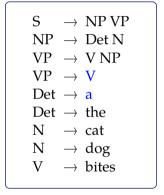


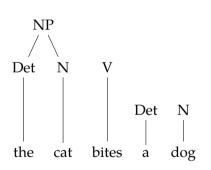


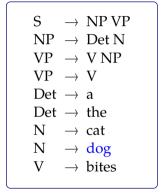


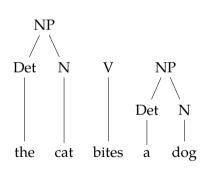




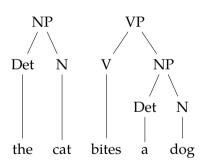




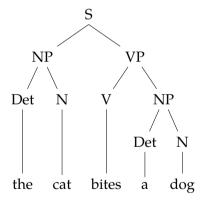




 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the \rightarrow cat $\rightarrow dog$ \rightarrow bites



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



 \rightarrow NP VP $NP \rightarrow Det N$ $VP \rightarrow V NP$ $VP \rightarrow V$ Det \rightarrow a Det \rightarrow the \rightarrow cat \rightarrow dog \rightarrow 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

$$NPV$$
 a dog \longrightarrow NPV a dog

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

$$\begin{array}{c|c}
NP V & a dog & \xrightarrow{reduce} & NP VP & a dog
\end{array}$$

stack input rule

stack	input	rule
	the cat bites a dog	shift

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

stack	input	rule
	the cat bites a dog cat bites a dog cat bites a dog	shift Det ⇒ the shift

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	$N \Rightarrow cat$

 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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$

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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift

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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift
NP bites	a dog	$V \Rightarrow bites$

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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift
NP bites	a dog	$V \Rightarrow bites$
NP V	a dog	$\mathrm{VP}\Rightarrow\mathrm{V}$

stack	stack input	
	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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift
NP bites	a dog	$V \Rightarrow bites$
NP V	a dog	$\mathrm{VP}\Rightarrow\mathrm{V}$
NP VP	a dog	$S \Rightarrow NP VP$

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	$N \Rightarrow cat$	
NP	bites a dog	$NP \Rightarrow Det N$	
NP	bites a dog	shift	
NP bites	a dog	$V \Rightarrow bites$	
NP V	a dog	$\mathrm{VP}\Rightarrow\mathrm{V}$	
NP VP	a dog	$S \Rightarrow NP VP$	
S	a dog	shift	

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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift
NP bites	a dog	$V \Rightarrow bites$
NP V	a dog	$\mathrm{VP} \Rightarrow \mathrm{V}$
NP VP	a dog	$S \Rightarrow NP VP$
S	a dog	shift
Sa	dog	$Det \Rightarrow A$

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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift
NP bites	a dog	$V \Rightarrow bites$
NP V	a dog	$\mathrm{VP}\Rightarrow\mathrm{V}$
NP VP	a dog	$S \Rightarrow NP VP$
S	a dog	shift
Sa	dog	$Det \Rightarrow A$
S Det dog	-	$N \Rightarrow dog$

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	$N \Rightarrow cat$
NP	bites a dog	$NP \Rightarrow Det N$
NP	bites a dog	shift
NP bites	a dog	$V \Rightarrow bites$
NP V	a dog	$\mathrm{VP} \Rightarrow \mathrm{V}$
NP VP	a dog	$S \Rightarrow NP VP$
S	a dog	shift
Sa	dog	$Det \Rightarrow A$
S Det dog	-	$N \Rightarrow dog$
S Det N		$NP \Rightarrow Det N$

e
t N
3
/P
t N

stack	input	rule		stack	input	rule	
	the cat bites a dog	shift		NP V	a dog	shift	
the	cat bites a dog	$Det \Rightarrow the$			O		
Det	cat bites a dog	shift					
Det cat	bites a dog	$N \Rightarrow cat$					
NP	bites a dog	$NP \Rightarrow Det N$					
NP	bites a dog	shift					
NP bites	a dog	$V \Rightarrow bites$					
NP V	a dog	$\mathrm{VP} \Rightarrow \mathrm{V}$					
NP VP	a dog	$S \Rightarrow NP VP$					
S	a dog	shift					
Sa	dog	$Det \Rightarrow A$					
S Det dog		$N \Rightarrow dog$					
S Det N		$NP \Rightarrow Det N$					
SNP		(stuck)					

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	$N \Rightarrow cat$					
NP	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					
Sa	dog	$Det \Rightarrow A$					
S Det dog		$N \Rightarrow dog$					
S Det N		$NP \Rightarrow Det N$					
SNP		(stuck)					

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	$\mathrm{Det} \Rightarrow \mathrm{a}$	
Det	cat bites a dog	shift	NP V Det	dog	shift	
Det cat	bites a dog	$N \Rightarrow cat$		Ü		
NP	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				
Sa	dog	$Det \Rightarrow A$				
S Det dog		$N \Rightarrow dog$				
S Det N		$NP \Rightarrow Det N$				
SNP		(stuck)				

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	O	$N \Rightarrow dog$
NP	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			
Sa	dog	$Det \Rightarrow A$			
S Det dog		$N \Rightarrow dog$			
S Det N		$NP \Rightarrow Det N$			
SNP		(stuck)			

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		$N \Rightarrow dog$
NP	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			
Sa	dog	$Det \Rightarrow A$			
S Det dog		$N \Rightarrow dog$			
S Det N		$NP \Rightarrow Det N$			
S NP		(stuck)			

	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		$N \Rightarrow dog$
	NP	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			
	Sa	dog	$Det \Rightarrow A$			
	S Det dog		$N \Rightarrow dog$			
	S Det N		$NP \Rightarrow Det N$			
	SNP		(stuck)			

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		$N \Rightarrow dog$
NP	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			
Sa	dog	$Det \Rightarrow A$			
S Det dog		$N \Rightarrow dog$			
S Det N		$NP \Rightarrow Det N$			
SNP		(stuck)			

stack	input	rule	sta	ack	input	rule	
	the cat bites a dog	shift	NF	PV	a dog	shift	
the	cat bites a dog	$Det \Rightarrow the$	NP V	V a	dog	$Det \Rightarrow a$	
Det	cat bites a dog	shift	NP V I	Det	dog	shift	
Det cat	bites a dog	$N \Rightarrow cat$	NP V Det d	log		$N \Rightarrow dog$	
NP	bites a dog	$NP \Rightarrow Det N$	NP V Det	t Ň		$NP \Rightarrow Det N$	
NP	bites a dog	shift	NP V I	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	 All inpu 	 All input reduced to S, accept 			
Sa	dog	$Det \Rightarrow A$	 Rules for 	 Rules form the parse tree 			
S Det dog		$N \Rightarrow dog$			F		
S Det N		$NP \Rightarrow Det N$					
SNP		(stuck)					

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)