Syntax

Grammars

Linguistics

What are grammars?

Grammars are formal models of the composition of words.

They serve two purposes:

1. To semantically compose meaning
2. To encode grammatical relations

What are grammars?

Most importantly:

Grammars are only a model

& in cognitive science,

all models are probably wrong.

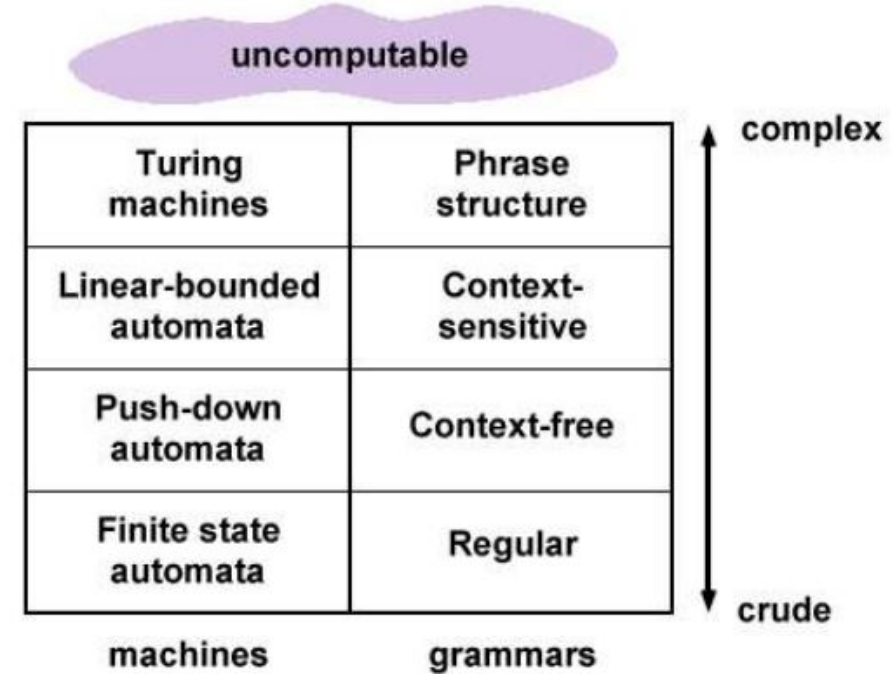
Types of Grammar

There are many types of grammars:

- Head-strong phrase structured grammar
- Dependency grammar
- Role and reference grammar
- Lexical-functional grammar
- The minimalist program

Grammars and Complexity

The Chomsky-Schützenberger hierarchy of languages:



Context Free Grammars

Rules have the form:

NonTerminal → {NonTerminal, Terminal}

For example,

S → NP VP

Example Grammar

$S \rightarrow NP\ VP$

$NP \rightarrow DT\ N$

$N \rightarrow N\ PP$

$N \rightarrow Adj\ N$

$PP \rightarrow P\ NP$

. . .

$VP \rightarrow IV$

$VP \rightarrow TV\ NP$

$VP \rightarrow DTV\ NP\ NP$

$VP \rightarrow SV\ S$

$VP \rightarrow Adv\ VP$

. . .

Tree Building Exercises

Ambiguity

Psychology

What is ambiguity?

- When the sentence has more than one meaning
  - In Linguistics terms: when the surface structure has multiple underlying structures
- Famous Example: *I saw the man with the binoculars.*
  - Meanings?

PP-Attachment

- Prepositional phrases can be attached “high” or “low”
  - High Attachment: modifying the verb
  - Low Attachment: modifying the object

PP-Attachment

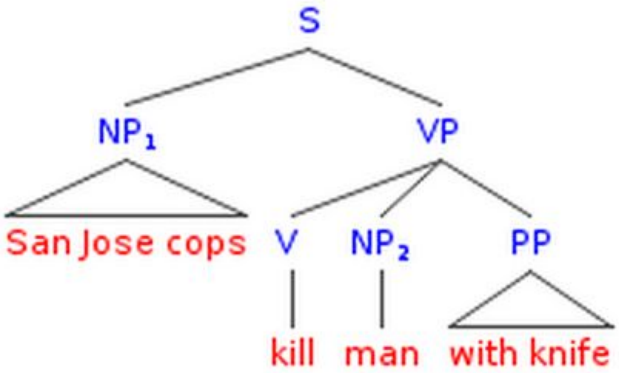
- Previous example:
  - High Attachment: *with the binoculars* modifies *saw*, meaning *Using the binoculars, I saw the man.*
  - Low Attachment: *with the binoculars* modifies *man*, meaning *I saw a man who had binoculars.*

PP-Attachment

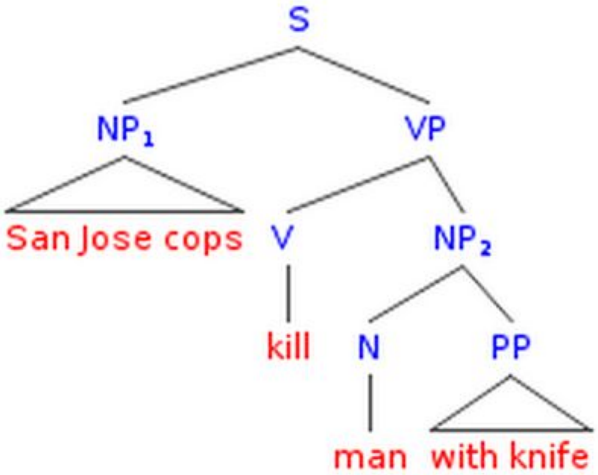
- Example: *San Jose cops kill man with knife* (newspaper headline found by Mark Liberman and posted on Language Log)
  - What is the PP?
  - What would be high attachment? What would be low attachment?

PP-Attachment

High Attachment



Low Attachment



Garden Path Sentences

- The initial reading of the sentence turns out to be wrong and requires a modification of structure being built
  - Frequently due to reduced relative clauses (omission of the *that*, *was, that was*)

Garden Path Sentences

- Famous Example: *The horse raced past the barn fell.*
  - Easier to interpret: *The horse that was raced past the barn fell.*
- Tree Structure: https://lh5.googleusercontent.com/-En54raKghVk/T8dHCm5XuVI/AAAAAAAAAfs/_w-hQOH7THI/s1600/the%2520horse%2520raced%2520full.gif

Parsing Algorithms

Computer Science

Parsing as Search

Any sequence of words *n* can be grouped with over 2^*n* syntactic interpretations (i.e. Catalan numbers)

This is actually a finite set and, therefore, parsing can be thought of as a search through the set of possible trees.

Bottom up vs. Top down

Bottom up (e.g., shift reduce, b-up chart):

POS tag each word, apply rules

Top down (e.g., left-corner, Earley) :

Start with end symbol and reverse engineer parse

Shift-Reduce Parser Yngve (1955)

While stack not S and input not empty:

Reduce: If there is a rule X → Y..Yn and Y..Yn is on top of the stack replace it with X.

Shift: Else, move the top token of the input onto the stack

Input	Stack	Action
Hacker Barbie is an . . .	_	Initial State
...	...	...

Input	Stack	Action
Hacker Barbie is an . . .	_	Initial State
Barbie is an inspiration	Hacker	Shift
...	...	...

Input	Stack	Action
Hacker Barbie is an . . .	__	Initial State
Barbie is an inspiration	Hacker	Shift
Barbie is an inspiration	Adj	Reduce
...	...	...

Input	Stack	Action
Hacker Barbie is an . . .	__	Initial State
Barbie is an inspiration	Hacker	Shift
Barbie is an inspiration	Adj	Reduce
is an inspiration	Barbie Adj	Shift
...	...	...

Input	Stack	Action
Hacker Barbie is an . . .	__	Initial State
Barbie is an inspiration	Hacker	Shift
Barbie is an inspiration	Adj	Reduce
is an inspiration	Barbie Adj	Shift
is an inspiration	PN Adj	Reduce
...	...	...

Input	Stack	Action
Hacker Barbie is an . . .	__	Initial State
Barbie is an inspiration	Hacker	Shift
Barbie is an inspiration	Adj	Reduce
is an inspiration	Barbie Adj	Shift
is an inspiration	PN Adj	Reduce
is an inspiration	NP	Reduce
...	...	...

Input	Stack	Action
Hacker Barbie is an . . .	__	Initial State
Barbie is an inspiration	Hacker	Shift
Barbie is an inspiration	Adj	Reduce
is an inspiration	Barbie Adj	Shift
is an inspiration	PN Adj	Reduce
is an inspiration	NP	Reduce
an inspiration	is NP	Shift
...	...	...

Input	Stack	Action
Barbie is an inspiration	Hacker	Shift
Barbie is an inspiration	Adj	Reduce
is an inspiration	Barbie Adj	Shift
is an inspiration	PN Adj	Reduce
is an inspiration	NP	Reduce
an inspiration	is NP	Shift
an inspiration	TV NP	Reduce
...	...	...

Input	Stack	Action
Barbie is an inspiration	Adj	Reduce
is an inspiration	Barbie Adj	Shift
is an inspiration	PN Adj	Reduce
is an inspiration	NP	Reduce
an inspiration	is NP	Shift
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
...	...	...

Input	Stack	Action
is an inspiration	Barbie Adj	Shift
is an inspiration	PN Adj	Reduce
is an inspiration	NP	Reduce
an inspiration	is NP	Shift
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
inspiration	DT TV NP	Reduce
...	...	...

Input	Stack	Action
is an inspiration	PN Adj	Reduce
is an inspiration	NP	Reduce
an inspiration	is NP	Shift
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
inspiration	DT TV NP	Reduce
{}	inspiration DT TV NP	Shift
...	...	...

Input	Stack	Action
is an inspiration	NP	Reduce
an inspiration	is NP	Shift
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
inspiration	DT TV NP	Reduce
{}	inspiration DT TV NP	Shift
{}	N DT TV NP	Reduce
...	...	...

Input	Stack	Action
an inspiration	is NP	Shift
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
inspiration	DT TV NP	Reduce
{}	inspiration DT TV NP	Shift
{}	N DT TV NP	Reduce
{}	NP TV NP	Reduce
...	...	...

Input	Stack	Action
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
inspiration	DT TV NP	Reduce
{}	inspiration DT TV NP	Shift
{}	N DT TV NP	Reduce
{}	NP TV NP	Reduce
{}	VP NP	Reduce
...	...	...

Input	Stack	Action
an inspiration	TV NP	Reduce
inspiration	an TV NP	Shift
inspiration	DT TV NP	Reduce
{}	inspiration DT TV NP	Shift
{}	N DT TV NP	Reduce
{}	NP TV NP	Reduce
{}	VP NP	Reduce
{}	S	Victory

Passive Chart-Parsing

Charts keep records of all the recognized phrases as you read in the input.

While input,

- Read in a token
- Apply all rules

Goal is to find a sentence that spans the input.

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

__	Initial State
(0,1,Hacker)	Read
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

__	Initial State
(0,1,Hacker)	Read
(0,1,Hacker) (0,1,Adj)	Apply
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

—	Initial State
(0,1,Hacker)	Read
(0,1,Hacker) (0,1,Adj)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie)	Read
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

__	Initial State
(0,1,Hacker)	Read
(0,1,Hacker) (0,1,Adj)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP)	Apply
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

—	Initial State
(0,1,Hacker)	Read
(0,1,Hacker) (0,1,Adj)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is)	Read
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

—	Initial State
(0,1,Hacker)	Read
(0,1,Hacker) (0,1,Adj)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV)	Apply
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV)	Apply
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an)	Read
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an) (3,4,DT)	Apply
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an) (3,4,DT)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an) (3,4,DT) (4,5,inspiration)	Read
...	...

$_0$ Hacker $_1$ Barbie $_2$ is $_3$ an $_4$ inspiration $_5$

(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an) (3,4,DT)	Apply
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an) (3,4,DT) (4,5,inspiration)	Read
(0,1,Hacker) (0,1,Adj) (1,2,Barbie) (1,2,PN) (0,2,NP) (2,3,is) (2,3,TV) (3,4,an) (3,4,DT) (4,5,inspiration) (4,5,N) (3,5,NP) (2,5,VP) **(0,5,S)**	Apply

Penn TreeBank

- https://www.cis.upenn.edu/~treebank/switch-samp-pos.html
- https://catalog.ldc.upenn.edu/search

Penn TreeBank

Table 2:
The Penn Treebank POS tagset

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subord. conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	`	Left open single quote
22.	RBS	Adverb, superlative	46.	``	Left open double quote
23.	RP	Particle	47.	'	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	''	Right close double quote

How to Evaluate a Parser

- Compare to hand-parsed information that are both correct and incorrect
  - Like the Penn TreeBank

How to Evaluate a Parser

- Look for:
  - Precision: number parser gets correct/number of answers given by the parser (removes parsing failures (meaning no output))
  - Recall: number parser gets correct/number it could have gotten correct (checks that it parses the hand-coding mistakes accurately)