Formal Grammars for human languages

What are grammar rules?

- It is important to never split infinitives.
- Prepositions are bad to end sentences with.
- When should you use who v. whom? ~~Whom~~ knows?
- Follow these rules and you'll make less (fewer) errors.

See more examples from Steven Pinker's "10 grammar rules it's OK to break"

Prescriptive grammar rules ← Important for paper writing

versus

Descritive grammar rules ← what you will write papers about.

A basic formalism for describing how human languages are structured is called a Context free grammar or CFG.

It was formalized by Noam Chompsky in 1956. A CFG allows us to describe the structure of language in terms of recursive tree structures.

CFGs consist of a set of rules (also called productions), which have terminal symbols (words) and non-terminal symbols (parts of speech or phrase structures)

Det → a

Det → the

Noun → flight

Verb → landed

Adverb → safely

Prep → in

ProperNoun → Philadelphia

Rules can also take the form

NP → Det Noun

VP → Verb PP

PP → Prep NP

Rules can also take the form

NP → Det Noun

VP → Verb PP

PP → Prep NP

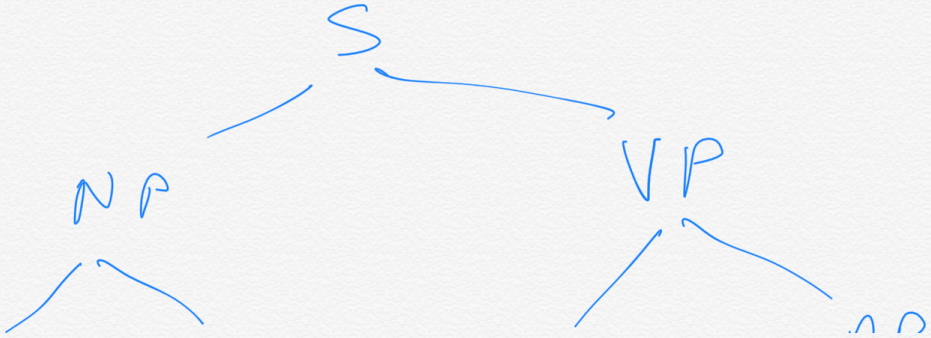
NP → Proper Noun

VP → VP Adverb

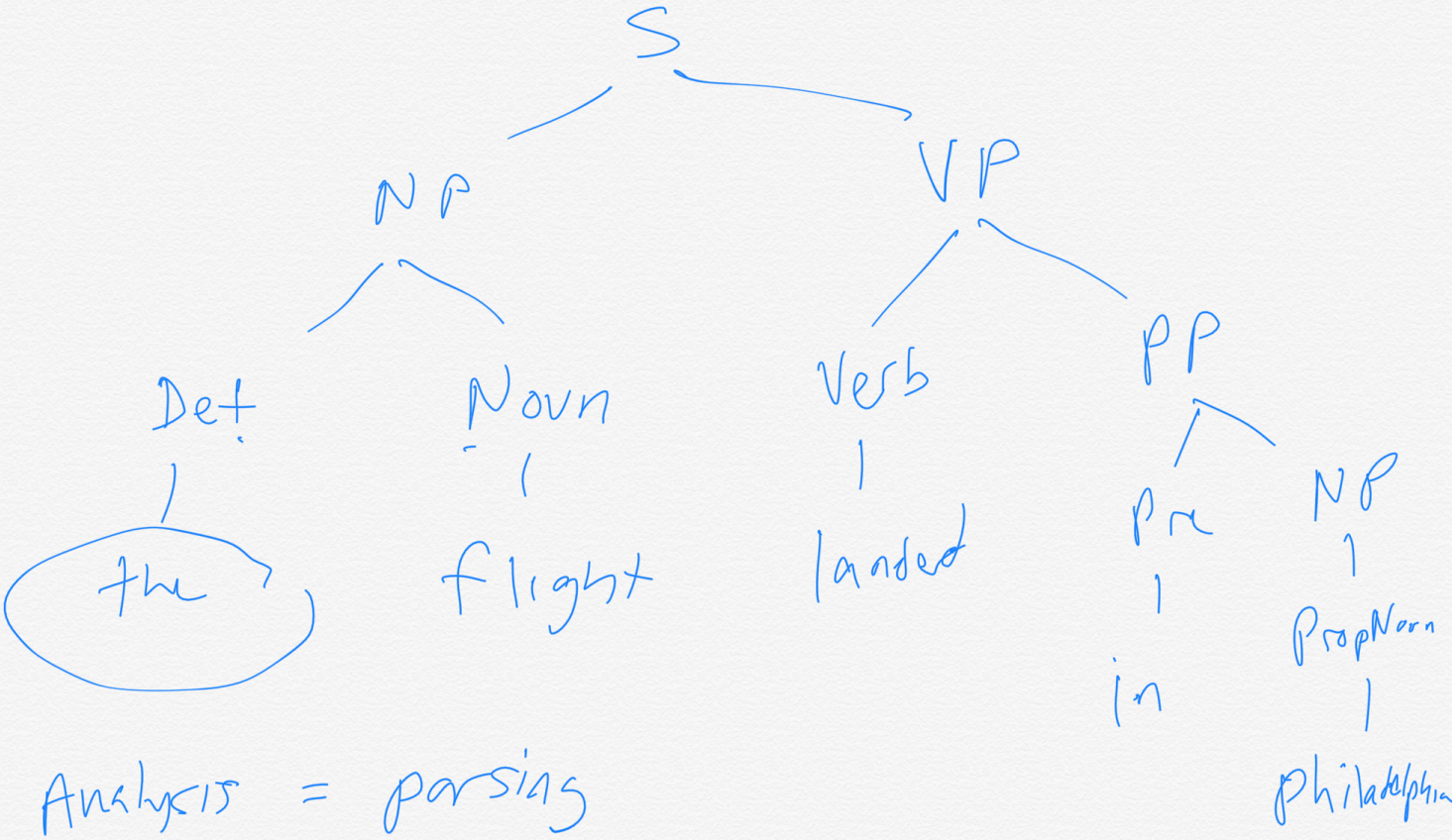
VP → Verb

S → NP VP

You can think of a CFG as a device for generating a sentence or as way to assign a structure to an input sentence.



You can think of a CFG as a device for generating a sentence or as way to assign a structure to an input sentence.



Analysis = parsing

Formal definition of a CFG.
Tuple with $\langle N, \Sigma, R, S \rangle$

N = set of non-terminal symbols

Formal definition of a CFG.

Tuple with $\langle N, \Sigma, R, S \rangle$

N - set of non-terminal symbols

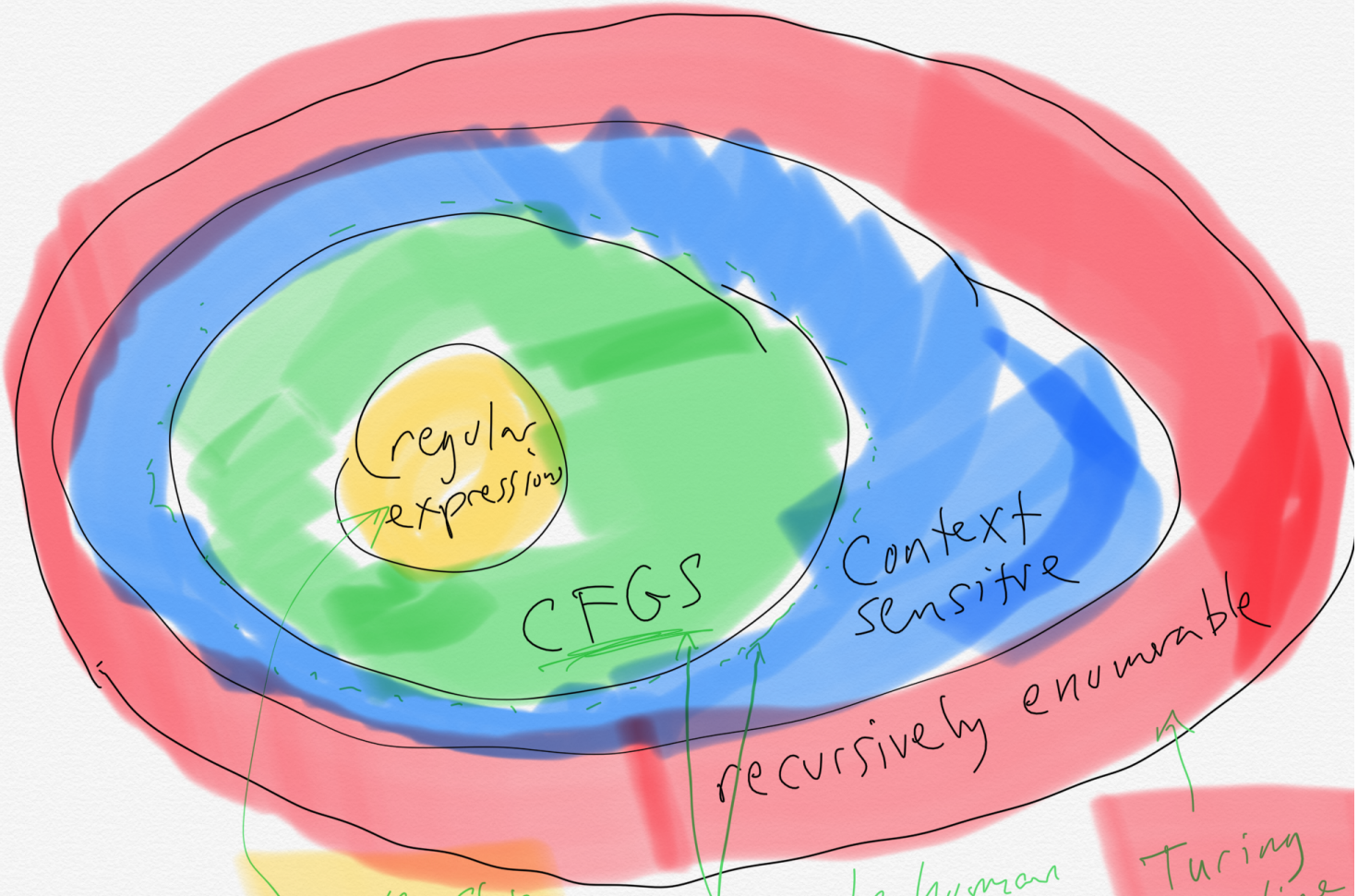
Σ - set of terminal symbols disjoint from N

R - a set of rules of the form $A \to B$ where $A \in N$ and B is a string of symbols $(\Sigma \cup N)^*$

S - start symbol ($\in$ of N)

Chompsky hierarchy

Chomsky hierarchy

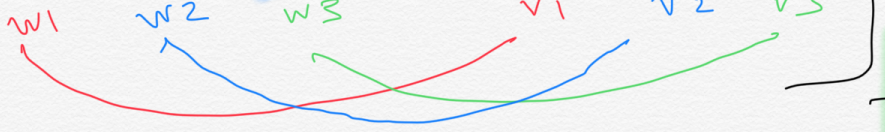


Finite-state machines (Least complex)

Where do human languages fit?

Turing Machine (most complex)

⇒ Dutch cross serial dependencies



not possible to represent w/a CFG

therefore NL is more complex than context

Sentence level constructions in English

Sentence level constructions in English

- Declarative
- Imperative
- Yes/No questions
- Wh- questions
  - subject wh-questions
  - non-subject wh-questions

phrases

Noun phrases

Determiners

the cat

Posessives instead of determiners

CCB's class

Posessives instead of determiners

CCB's class

Mass nouns

£ music

pre-modifiers (numbers, adjective phrases, quantifiers)

all flights

post-modifiers (PPs, gerundives, relative pronouns that/who)

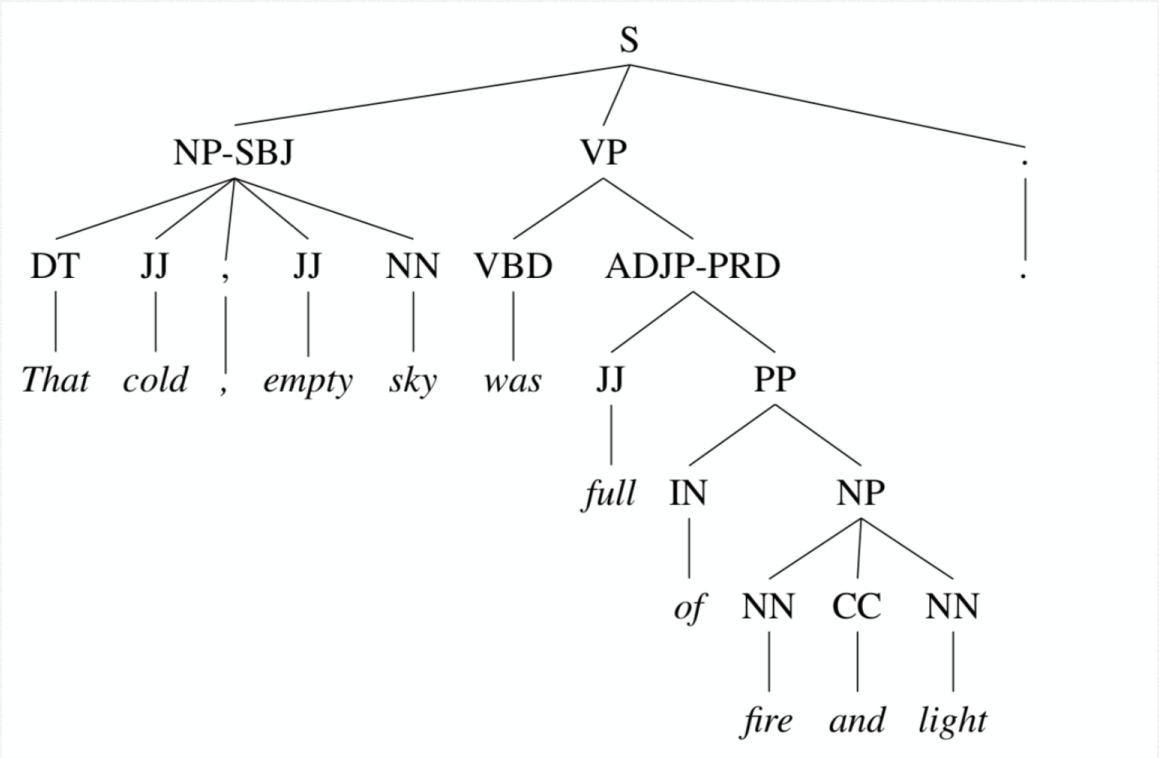
all classes holding midterms in CIS

Verb phrases

arguments / valency / subcategorization frames

Q: How do we construct a grammar for a language? write it by hand?

Treebanks.



Grammar	Lexicon
$S \rightarrow$ *NP VP .*	*PRP* $\rightarrow$ *we* | *he*
$S \rightarrow$ *NP VP*	*DT* $\rightarrow$ *the* | *that* | *those*
$S \rightarrow$ *" S " , NP VP .*	*JJ* $\rightarrow$ *cold* | *empty* | *full*
$S \rightarrow$ *-NONE-*	*NN* $\rightarrow$ *sky* | *fire* | *light* | *flight* | *tomorrow*
NP $\rightarrow$ *DT NN*	*NNS* $\rightarrow$ *assets*
NP $\rightarrow$ *DT NNS*	*CC* $\rightarrow$ *and*
NP $\rightarrow$ *NN CC NN*	*IN* $\rightarrow$ *of* | *at* | *until* | *on*
NP $\rightarrow$ *CD RB*	*CD* $\rightarrow$ *eleven*
NP $\rightarrow$ *DT JJ , JJ NN*	*RB* $\rightarrow$ *a.m.*
NP $\rightarrow$ *PRP*	*VB* $\rightarrow$ *arrive* | *have* | *wait*
NP $\rightarrow$ *-NONE-*	*VBD* $\rightarrow$ *was* | *said*
VP $\rightarrow$ *MD VP*	*VBP* $\rightarrow$ *have*
VP $\rightarrow$ *VBD ADJP*	*VBN* $\rightarrow$ *collected*
VP $\rightarrow$ *VBD S*	*MD* $\rightarrow$ *should* | *would*
VP $\rightarrow$ *VBN PP*	*TO* $\rightarrow$ *to*
VP $\rightarrow$ *VB S*	
VP $\rightarrow$ *VB SBAR*	
VP $\rightarrow$ *VBP VP*	
VP $\rightarrow$ *VBN PP*	
VP $\rightarrow$ *TO VP*	
SBAR $\rightarrow$ *IN S*	
ADJP $\rightarrow$ *JJ PP*	
PP $\rightarrow$ *IN NP*	

Figure 11.10 A sample of the CFG grammar rules and lexical entries that would be extracted from the three treebank sentences in Fig. 11.7 and Fig. 11.9.

Treebanks as grammars.

PennTreebank has 4500 VP rules and 1000s NP rules

Common modifications to CFGs

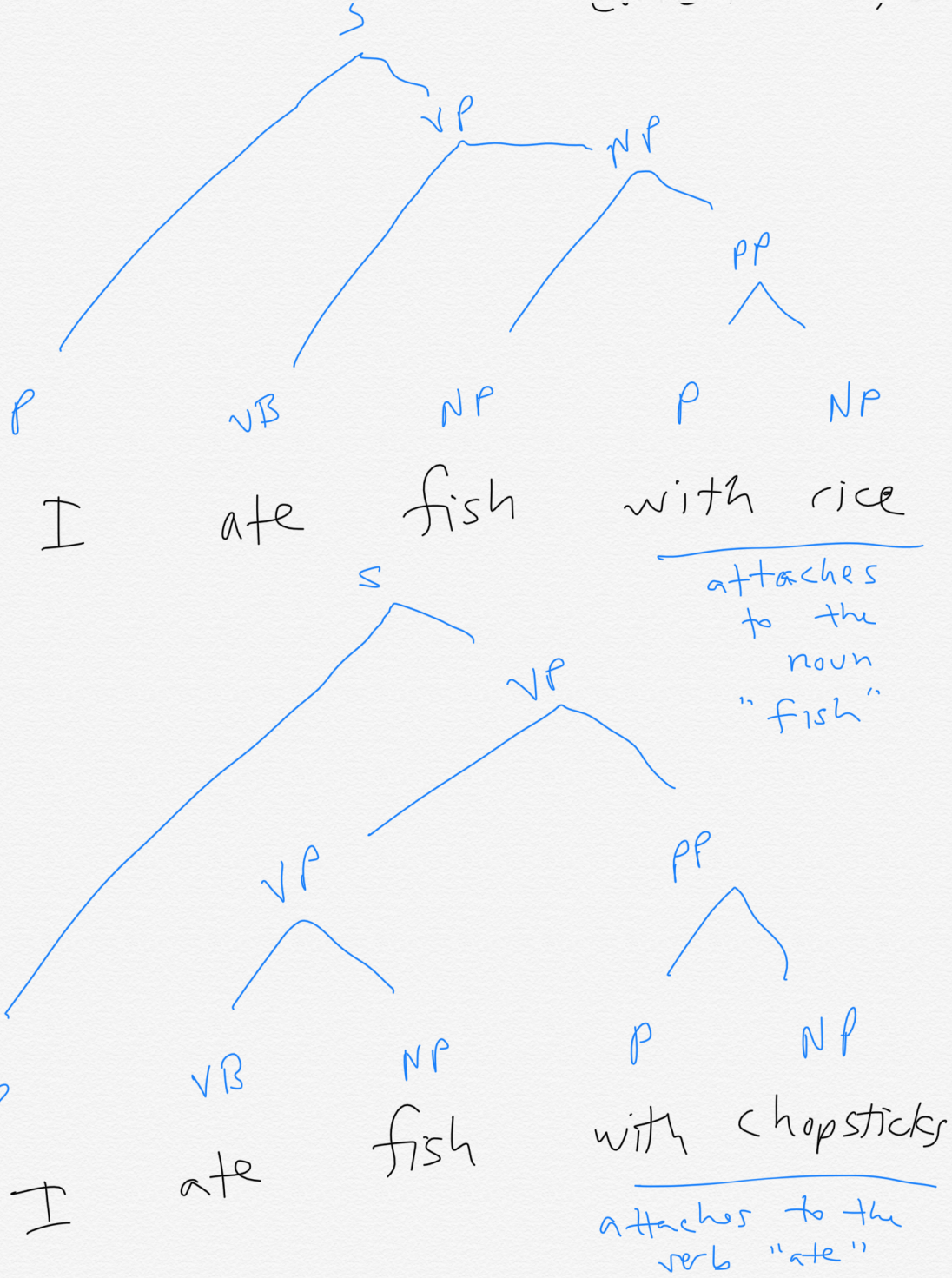
- lexicalized rules that encode the head of the phrase

Common modifications to CFGs

- lexicalized rules that encode the head of the phrase
- estimating probabilities for all production rules. (see Ch. 13)
- Synchronous CFGs for translation

Important concepts

- Parsing: derive one or more parse trees for an input sentence.
- Language is ambiguous so this is hard to find the correct analysis.



Different syntactic analyses encode different semantics.