Language and Computation

week 7, Thursday, February 27, 2014

Tamás Biró
Yale University
tamas.biro@yale.edu
http://www.birot.hu/courses/2014-LC/

Practical matters

- **Post-reading:** Chapters 12 and 16.
- **Pre-reading:** Sections 13.1-3
- **Sections**
- **Homework 3** posted, due 03/04.
- To come: Viterbi and Forward-Backward – an example
- To come: proof of HW 2, part 3.

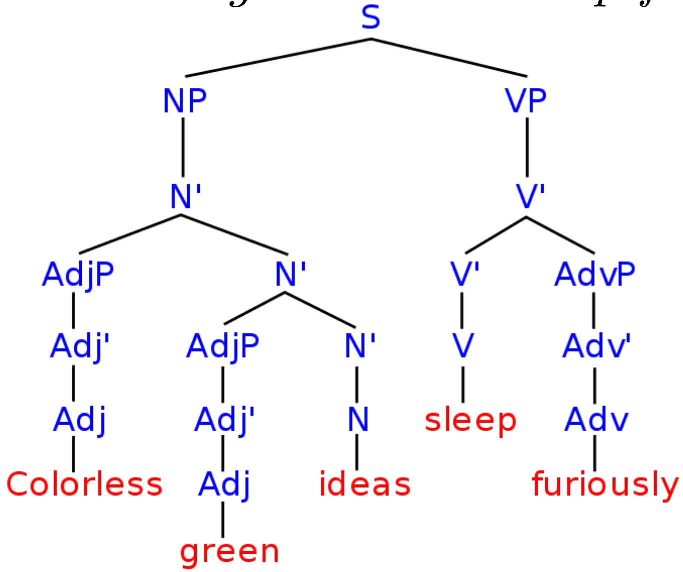
Today

- Syntax in a nutshell
- Formal Grammars and the Chomsky Hierarchy
- A note still on regular languages: The Pumping Lemma
- Beyond regular languages: Context-Free Grammars
- Parsing CFGs – basics

Syntax in a nutshell

Regular languages and Markov Models are not sufficient

Colorless green ideas sleep furiously



`http://en.wikipedia.org/wiki/File:Syntax_tree.svg`

Formal Grammars

Formal Grammars

N	a set of **non-terminal symbols** (or **variables**)
Σ	a set of **terminal symbols** (disjoint from N)
R	a set of **rules** or productions, each of the form $A \rightarrow \beta$, where A is a non-terminal, β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$
S	a designated **start symbol**

Capital letters like A, B, and S	Non-terminals
S	The start symbol
Lower-case Greek letters like α, β, and γ	Strings drawn from $(\Sigma \cup N)*$
Lower-case Roman letters like u, v, and w	Strings of terminals

Formal Grammars

A toy grammar for English:

$S \rightarrow NP \;\; VP$

$VP \rightarrow V$

$VP \rightarrow V \;\; NP$

$NP \rightarrow N$

$NP \rightarrow Det \;\; N$

$V \rightarrow \{$ eat, love, walk,. . . $\}$

$V \rightarrow \{$ eats, loves, walks,. . . $\}$

$N \rightarrow \{$ John, Marry. . . $\}$

$N \rightarrow \{$ apple, pear. . . $\}$

$N \rightarrow \{$ apples, pears. . . $\}$

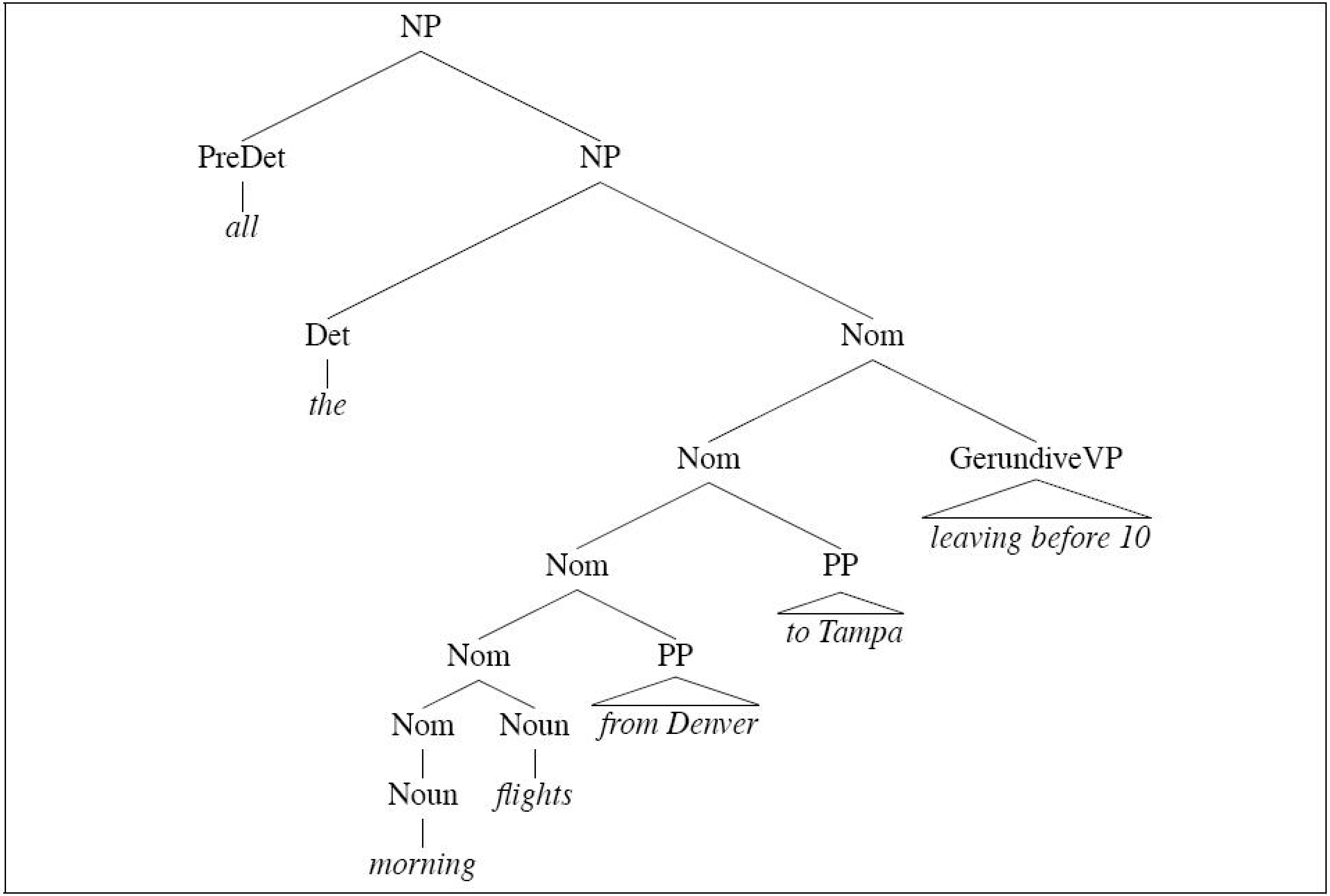
$Det \rightarrow \{$ the, a, an, $\emptyset$ $\}$

Formal Grammars

A toy grammar for English – lessons:

- Introduce additional categories: $V_{\text{transitive}}$ vs. $V_{\text{intransitive}}$.
- Proper names as NPs.
- Agreement

$\rightarrow$ more general formalism needed (feature structures: Ch. 15)


NP
PreDet
NP
all
Det
Nom
the
Nom
GerundiveVP
leaving before 10
Nom
PP
to Tampa
Nom
PP
from Denver
Nom
Noun
Noun
flights
morning

Formal Grammars: an example

$V \rightarrow V$ and. . . what?

Subcategorization frames for a set of example verbs:

Frame	Verb	Example
$\emptyset$	eat, sleep	I ate
NP	prefer, find, leave	Find [$_{NP}$ the flight from Pittsburgh to Boston]
NP NP	show, give	Show [$_{NP}$ me] [$_{NP}$ airlines with flights from Pittsburgh]
$PP_{\text{from}}\ PP_{\text{to}}$	fly, travel	I would like to fly [$_{PP}$ from Boston] [$_{PP}$ to Philadelphia]
$NP\ PP_{\text{with}}$	help, load	Can you help [$_{NP}$ me] [$_{PP}$ with a flight]
VPto	prefer, want, need	I would prefer [$_{VPto}$ to go by United airlines]
VPbrst	can, would, might	I can [$_{VPbrst}$ go from Boston]
S	mean	Does this mean [$_{S}$ AA has a hub in Boston]

The Chomsky Hierarchy

Generative power of a formalism

What is the set of languages generated by a formalism?

- **Overgeneration:** too powerful a formalism,
  also generating languages that we don't want.

- **Undergeneration:** too weak a formalism,
  not generating the languages we would like to.

Generative power of a formalism

What is the set of languages generated by a formalism?

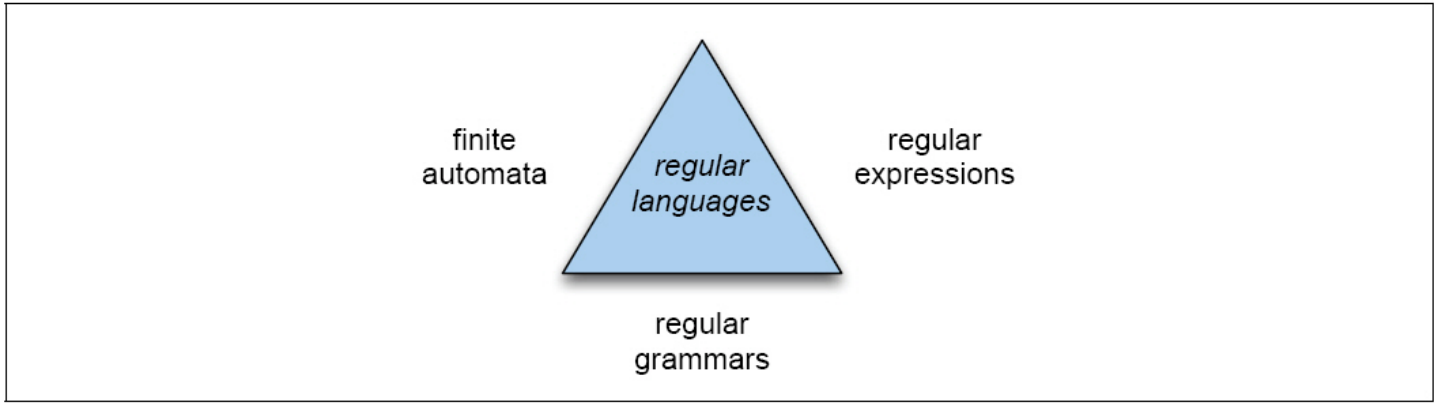
Goal: generate exactly the attested human languages.

If reached: our formalism *accounts* for human languages.

Making happy

- the theoretical linguist wishing to characterize the possible languages of the world, who is now offered a mathematical tool to do so.
- the cognitive scientist
  wishing to decipher the "mental software" run by our brain.

Regular languages



But this is too weak a formalism for natural languages!

What can we do with formal grammars?

Chomsky hierarchy

Type	Common Name	Rule Skeleton	Linguistic Example
0	Turing Equivalent	$\alpha \rightarrow \beta$, s.t. $\alpha \neq \epsilon$	HPSG, LFG, Minimalism
1	Context Sensitive	$\alpha A \beta \rightarrow \alpha \gamma \beta$, s.t. $\gamma \neq \epsilon$	
–	Mildly Context Sensitive		TAG, CCG
2	Context Free	$A \rightarrow \gamma$	Phrase-Structure Grammars
3	Regular	$A \rightarrow xB$ or $A \rightarrow x$	Finite-State Automata

NB:

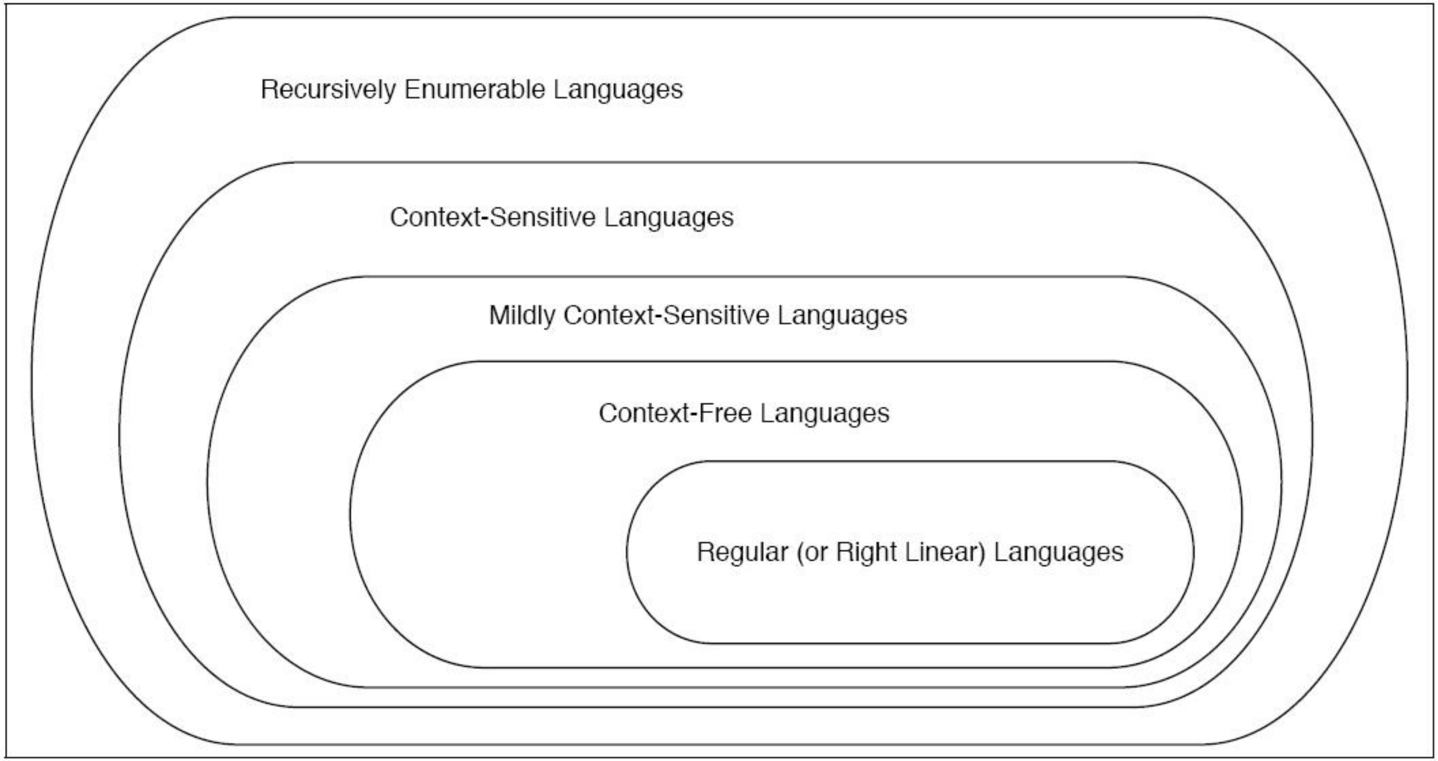
0: Turing machine

1: Linear bounded automaton

2: Non-deterministic push-down automaton

3: Finite-state automaton

The Chomsky Hierarchy



Weak and strong equivalence

$\{a^n b^m | n, m \in \mathbb{N}^+\}$

- Regular expression: `/a+ b+/`
- Finite State Automaton: initial state q_0, state q_1, end state q_2, arc $q_0 \rightarrow q_0$ with label `a`, arc $q_1 \rightarrow q_1$ with label `b`, arc loop $q_0 \rightarrow q_1$ with label `a`, arc loop $q_1 \rightarrow q_2$ with label `b`.
- Regular grammar: $S \rightarrow a \;\; S$, $S \rightarrow a \;\; A$ $A \rightarrow b \;\; A$, $A \rightarrow b$
- Context Free Grammar: $S \rightarrow S \quad A \quad B$, $A \rightarrow A \quad A$, $B \rightarrow B \;\; B$, $A \rightarrow a$, $B \rightarrow b$

The Pumping Lemma

For all L infinite regular languages,

there are strings x, y and z such that

$y \neq \epsilon$ and

$xy^n z \in L$ for all $N \geq 0$.

Examples: $\{a^n b^n\}$ is not regular. $\{xx^{rev}\}$ is not regular.

See you next week!