Language and Computation

week 8, Thursday, March 06, 2014

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

Practical matters

- **Post-reading:** Chapters 12, 16 and 14.1
- **Pre-reading:** Sections 13.1-3
- **Homework 3** returned after the break.
- **Midterm:** will be posted tomorrow.
- Proof of HW 2, part 3 posted.
- To come: Viterbi and Forward-Backward – an example

Today

- Formal definition of Formal Grammars
- Chomsky Hierarchy
- The Pumping Lemma
- Beyond regular languages: Context-Free Grammars
- Probabilistic Context Free Grammars

(Parsing to come after the break)

Formal Grammars: an example

Formal Grammars

A toy grammar for English:

Phrase Structure Rules:

$S \rightarrow NP \quad VP$

$VP \rightarrow V$

$VP \rightarrow V \quad NP$

$NP \rightarrow N$

$NP \rightarrow Det \quad N$

Lexical Insertion Rules:

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

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

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

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

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

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

Formal Grammars

A toy grammar for English:

$$S \Rightarrow NP\ \ VP \Rightarrow Det\ \ N\ \ VP \Rightarrow$$

$$\Rightarrow Det\ \ N\ \ V\ \ NP \Rightarrow Det\ \ N\ \ V\ \ N \Rightarrow$$

$$\Rightarrow The\ \ N\ \ V\ \ N \Rightarrow The\ \ John\ \ V\ \ N \Rightarrow$$

$$\Rightarrow The\ \ John\ \ sleep\ \ N \Rightarrow The\ \ John\ \ sleep\ \ apple$$

This is a sentence derived from this grammar.

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)

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_{from} PP_{to}	fly, travel	I would like to fly [$_{PP}$ from Boston] [$_{PP}$ to Philadelphia]
NP PP_{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]

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

Given formal grammar $G = (N, \Sigma, R, S)$:

Def: Given strings a and $b \in (\Sigma \cup N)^*$, $a \Rightarrow_G b$ iff there exist $p, q, r, s \in (\Sigma \cup N)^*$ such that

- $a = p + q + s$,
- $b = p + r + s$, and
- $(q \rightarrow r) \in R$

Def: A string $a \in \Sigma^*$ is $\mathrm{grammatical}$ in grammar G iff $S \Rightarrow^* a$.

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?

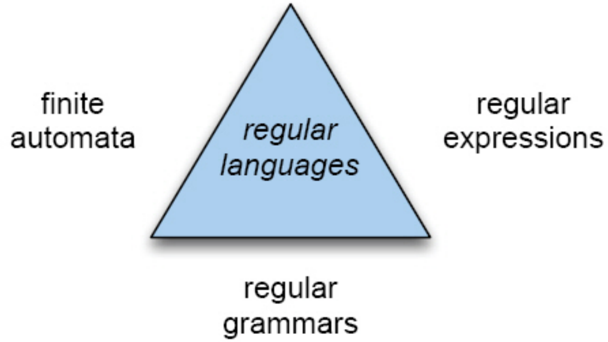
Goal: generate exactly the attested human languages.

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:

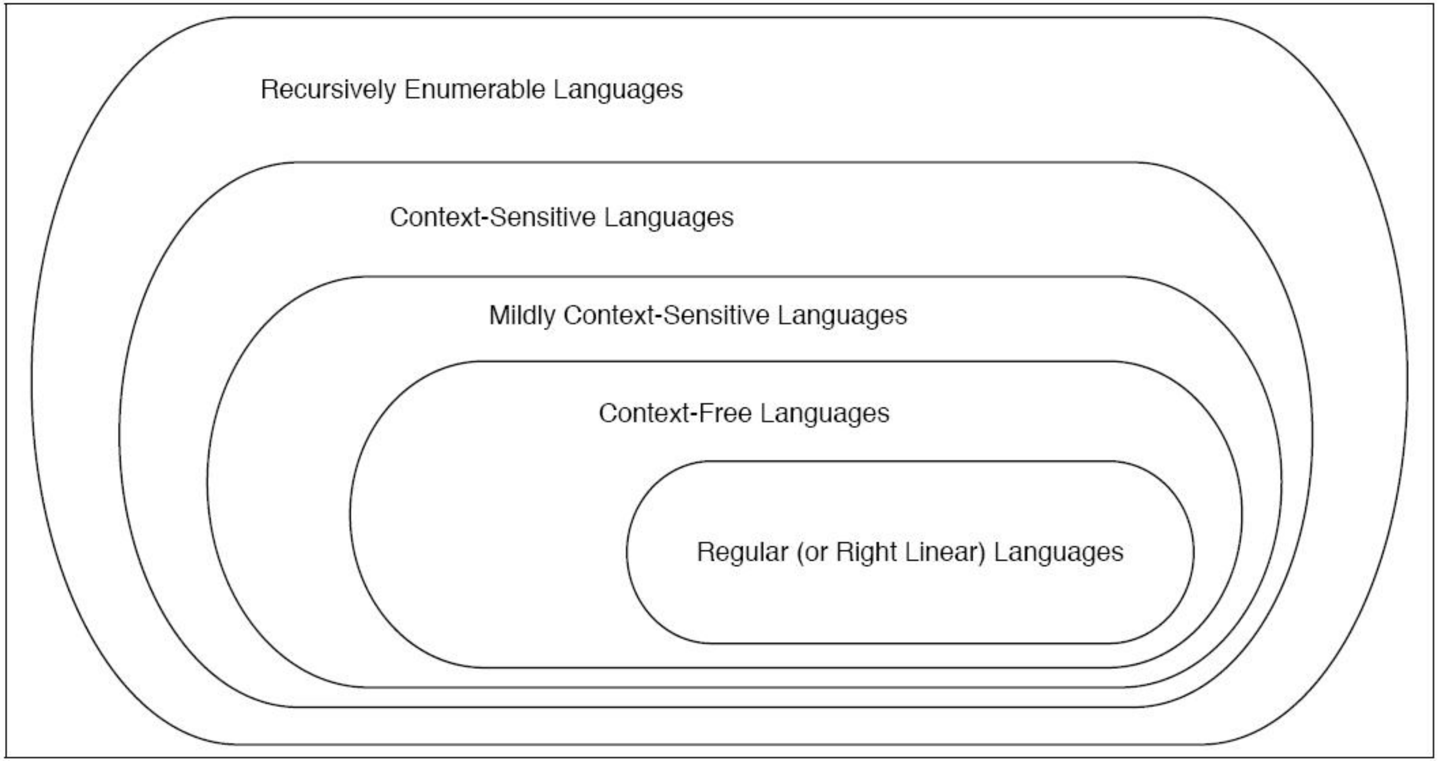
0: Turing machine

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: $S \rightarrow A\ B$, $A \rightarrow A\ A$, $B \rightarrow B\ B$, $A \rightarrow a$, $B \rightarrow b$

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 $N \geq 0$.

Example: $\{a^n b^n\}$ is not regular.

The Pumping Lemma

Example: $L = \{xx^{rev} | x \in \{a, b\}^* \}$ is not regular.

where x^{rev} is the string x reversed. The strings in L are symmetrical.

Proof:

1. Intersect L with regular language aa^*bbaa^*.
   If L were regular, then intersection would also be regular, because regular languages are closed for intersection (J&M 2.3).

2. Resulting language is $a^n b^2 a^n$, which is not regular, due to the pumping lemma. Therefore L cannot be regular, either.

Probabilistic Context Free Grammars

Probabilistic Context Free 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 $A \rightarrow \beta\ [p]$, where A is a non-terminal, β is a string of symbols from the infinite set of strings $(\Sigma \cup N)*$, and p is a number between 0 and 1 expressing $P(\beta\|A)$
S	a designated **start symbol**

$$\sum_{\beta \in (N \cup \Sigma)^*} P(A \rightarrow \beta) = 1$$

Probabilistic Context Free Grammars

Probability of tree T (which yields sentence S):

$$P(T,S) = \prod_{i=1}^{n} P(RHS_i|LHS_i)$$

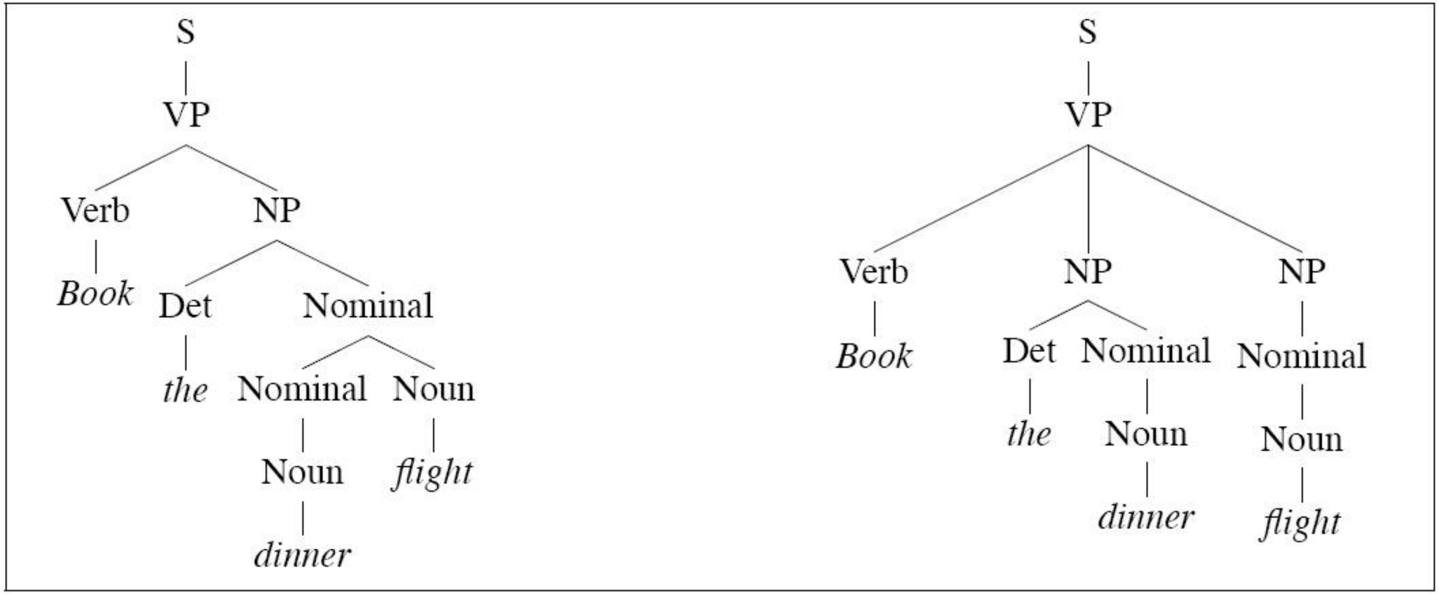
the product of the probabilities of the n rules used to expand each of the n non-terminal nodes in parse tree T (J&M 14.1.1).

Probabilistic CFG: an example

Grammar		Lexicon
$S \rightarrow NP\ VP$	[.80]	$Det \rightarrow that$ [.10] | a [.30] | the [.60]
$S \rightarrow Aux\ NP\ VP$	[.15]	$Noun \rightarrow book$ [.10] | $flight$ [.30]
$S \rightarrow VP$	[.05]	| $meal$ [.15] | $money$ [.05]
$NP \rightarrow Pronoun$	[.35]	| $flights$ [.40] | $dinner$ [.10]
$NP \rightarrow Proper\text{-}Noun$	[.30]	$Verb \rightarrow book$ [.30] | $include$ [.30]
$NP \rightarrow Det\ Nominal$	[.20]	| $prefer$; [.40]
$NP \rightarrow Nominal$	[.15]	$Pronoun \rightarrow I$ [.40] | she [.05]
$Nominal \rightarrow Noun$	[.75]	| me [.15] | you [.40]
$Nominal \rightarrow Nominal\ Noun$	[.20]	$Proper\text{-}Noun \rightarrow Houston$ [.60]
$Nominal \rightarrow Nominal\ PP$	[.05]	| NWA [.40]
$VP \rightarrow Verb$	[.35]	$Aux \rightarrow does$ [.60] | can [40]
$VP \rightarrow Verb\ NP$	[.20]	$Preposition \rightarrow from$ [.30] | to [.30]
$VP \rightarrow Verb\ NP\ PP$	[.10]	| on [.20] | $near$ [.15]
$VP \rightarrow Verb\ PP$	[.15]	| $through$ [.05]
$VP \rightarrow Verb\ NP\ NP$	[.05]	
$VP \rightarrow VP\ PP$	[.15]	
$PP \rightarrow Preposition\ NP$	[1.0]	

Probabilistic Context Free Grammars

An example:



(booking a flight serving dinner vs. booking a flight on behalf of ‘dinner’.)

Parsing and grammar learning

- **Parsing:** Given (probabilistic) CFG G, given sentence s, find (possible/most probable) parse tree for s in G.

- **Learning:** Given set of sentences, build a (probabilistic) context free grammar.

Have a nice break, and

see you after the spring recess!