Grammars and Context-free Languages; Chomsky Hierarchy

H. Geuvers

Institute for Computing and Information Sciences
Radboud University Nijmegen

Version: fall 2015

Outline

Generating words

Example:

$$\begin{array}{lcl} S & \rightarrow & O \mid E \\ E & \rightarrow & \lambda \mid \mathtt{a}E\mathtt{a} \mid \mathtt{b}E\mathtt{b} \\ O & \rightarrow & \mathtt{a} \mid \mathtt{b} \mid \mathtt{a}O\mathtt{a} \mid \mathtt{b}O\mathtt{b} \end{array}$$

Productions, always start with S

$S \Rightarrow E \Rightarrow \mathtt{a}E\mathtt{a} \Rightarrow \mathtt{ab}E\mathtt{ba} \Rightarrow \mathtt{abba}$

$S \Rightarrow E \Rightarrow \mathtt{b}E\mathtt{b} \Rightarrow \mathtt{ba}E\mathtt{ab} \Rightarrow \mathtt{bab}E\mathtt{bab} \Rightarrow \mathtt{baba}E\mathtt{abab} \Rightarrow \mathtt{babaabab}$

$S \Rightarrow O \Rightarrow \mathtt{b}O\mathtt{b} \Rightarrow \mathtt{bab}$

$S \Rightarrow O \Rightarrow \mathtt{b}O\mathtt{b} \Rightarrow \mathtt{ba}O\mathtt{ab} \Rightarrow \mathtt{babab}$

We can generate exactly the set of words w with $w = w^R$
(w is a palindrome)

Context-free grammar

Let Σ be a finite alphabet.

Def. A context-free grammar (CFG) $G = \langle V, S, P \rangle$ over Σ consists of

- V, a set of non-terminal symbols,
- $S \in V$, a start symbol,
- P, a set of production rules of the form

$$X \to w,$$

where $X \in V$ and $w \in (V \cup \Sigma)^*$.

Notation:

$$E \to \lambda \mid \mathtt{a}E\mathtt{a} \mid \mathtt{b}E\mathtt{b}$$

is shorthand for three rules:

$$E \to \lambda, \quad E \to \mathtt{a}E\mathtt{a}, \quad E \to \mathtt{b}E\mathtt{b}.$$

Context-free language

Using G, a language is generated using a relation $\Rightarrow$ ('produces') defined as follows (where u, v, w are arbitrary elements of $(\Sigma \cup V)^*$)

$$X \to w \quad \text{implies} \quad uXv \Rightarrow uwv$$
$$u \Rightarrow v, v \Rightarrow w \quad \text{implies} \quad u \Rightarrow w$$

The language generated by G is

$$\mathcal{L}(G) = \{w \in \Sigma^* \mid S \Rightarrow w\}.$$

A language L is context-free if $L = \mathcal{L}(G)$ for some context-free G.

Leftmost derivations

There can be many ways to produce a word $w \in \mathcal{L}(G)$.

G: $S \to \mathtt{a}SB \mid \lambda, \quad B \to \mathtt{b}$

Derivations of $\mathtt{aabb}$:

$$d_1 : S \Rightarrow \mathtt{a}SB \Rightarrow \mathtt{aa}SBB \Rightarrow \mathtt{aa}BB \Rightarrow \mathtt{aab}B \Rightarrow \mathtt{aabb}$$

$$d_2 : S \Rightarrow \mathtt{a}SB \Rightarrow \mathtt{a}S\mathtt{b} \Rightarrow \mathtt{aa}SB\mathtt{b} \Rightarrow \mathtt{aa}S\mathtt{bb} \Rightarrow \mathtt{aabb}$$

$$d_3 : S \Rightarrow \mathtt{a}SB \Rightarrow \mathtt{a}S\mathtt{b} \Rightarrow aaSB\mathtt{b} \Rightarrow \mathtt{aa}B\mathtt{b} \Rightarrow \mathtt{aabb}$$

Derivation d_1 is leftmost, d_2 is rightmost, d_3 is neither.

Def. A derivation is leftmost if in each step a rule is applied to the leftmost non-terminal.

Lemma. For all grammars G and words w, $w \in \mathcal{L}(G)$ iff there is a leftmost derivation of w.

Derivation/parse trees (on blackboard).

Ambiguity

Def. A context-free grammar G is unambiguous if for each $w \in \mathcal{L}(G)$ there exists a unique leftmost derivation of w in G. Otherwise G is ambiguous.

Equivalently: A context-free grammar is ambiguous if there is a word with two different parse trees.

Example: "dangling else"

$$
\begin{array}{lcl}
S & \to & \texttt{print}\ E \\
 & | & \texttt{if}\ E\ \texttt{then}\ S\ \texttt{else}\ S \\
 & | & \texttt{if}\ E\ \texttt{then}\ S \\
E & \to & \texttt{a} \mid \texttt{b} \mid \cdots
\end{array}
$$

Natural language

$$
\begin{array}{rcl}
S = \langle \text{sentence} \rangle & \to & \langle \text{noun-phrase} \rangle \langle \text{verb-phrase} \rangle \texttt{.} \\
\langle \text{sentence} \rangle & \to & \langle \text{noun-phrase} \rangle \langle \text{verb-phrase} \rangle \langle \text{object-phrase} \rangle \texttt{.} \\
\langle \text{noun-phrase} \rangle & \to & \langle \text{name} \rangle \mid \langle \text{article} \rangle \langle \text{noun} \rangle \\
\langle \text{name} \rangle & \to & \texttt{John} \mid \texttt{Jill} \\
\langle \text{noun} \rangle & \to & \texttt{bicycle} \mid \texttt{mango} \\
\langle \text{article} \rangle & \to & \texttt{a} \mid \texttt{the} \\
\langle \text{verb-phrase} \rangle & \to & \langle \text{verb} \rangle \mid \langle \text{adverb} \rangle \langle \text{verb} \rangle \\
\langle \text{verb} \rangle & \to & \texttt{eats} \mid \texttt{rides} \\
\langle \text{adverb} \rangle & \to & \texttt{slowly} \mid \texttt{frequently} \\
\langle \text{adjective-list} \rangle & \to & \langle \text{adjective} \rangle \langle \text{adjective-list} \rangle \mid \lambda \\
\langle \text{adjective} \rangle & \to & \texttt{big} \mid \texttt{juicy} \mid \texttt{yellow} \\
\langle \text{object-phrase} \rangle & \to & \langle \text{adjective-list} \rangle \langle \text{name} \rangle \\
\langle \text{object-phrase} \rangle & \to & \langle \text{article} \rangle \langle \text{adjective-list} \rangle \langle \text{noun} \rangle
\end{array}
$$

Jill frequently eats a juicy yellow mango. belongs to this language

Some context-free languages

Is there a grammar G_2 such that

$$\mathcal{L}(G_2) = \{\mathtt{a}^n\mathtt{b}^n \mid n \geq 0\}?$$

G_2: $\boxed{S \rightarrow \lambda \mid \mathtt{a}S\mathtt{b}}$

All possible derivations

$$\begin{array}{ccccccccccc}
S & \Rightarrow & \mathtt{a}S\mathtt{b} & \Rightarrow & \mathtt{aa}S\mathtt{bb} & \Rightarrow & \cdots & \Rightarrow & \mathtt{a}^nS\mathtt{b}^n & \Rightarrow & \cdots \\
\Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \Downarrow & & \\
\lambda & & \mathtt{ab} & & \mathtt{aabb} & & \cdots & & \mathtt{a}^n\mathtt{b}^n & &
\end{array}$$

What is a grammar G_3 such that

$$\mathcal{L}(G_3) = \{\mathtt{a}^n\mathtt{b}^n \mid n > 0\}?$$

G_3: $\boxed{S \rightarrow \mathtt{ab} \mid \mathtt{a}S\mathtt{b}}$

More context-free languages

Let $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$

Claim: $L = \{\mathtt{a}^n\mathtt{b}^m\mathtt{c}^{2n+1} \mid m, n \geq 0\}$ is context-free.

Let us first show that $L' = \{\mathtt{a}^n\mathtt{c}^{2n+1} \mid m, n \geq 0\}$ is context-free.

For L' use

$$\mathrm{S} \rightarrow \mathtt{c} \mid \mathtt{a}S\mathtt{cc}$$

For L use

$$\begin{array}{lcl} \mathrm{S} & \rightarrow & B\mathtt{c} \mid \mathtt{a}S\mathtt{cc} \\ \mathrm{B} & \rightarrow & \lambda \mid \mathtt{b}B \end{array}$$

Fact: $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n \geq 0\}$ is *not* context-free.

Regular languages

Let Σ be a finite alphabet.

Def. A right-linear grammar $G = \langle V, S, P \rangle$ over Σ is a context-free grammar where all rules are of the following shape

$$\boxed{\begin{array}{lcl} X & \to & \lambda \\ X & \to & wY \end{array}}$$

where $X, Y \in V$ and $w \in \Sigma^*$

Theorem. A language is regular language iff it can be generated by a right-linear grammar.

Corollary. All regular languages are context-free.

Context-sensitive languages

A context-free grammar has rules like:

$$X \quad \rightarrow \quad w$$

With $X \in V$, $w \in (\Sigma \cup V)^*$.

A context-sensitive grammar has rules like:

$$\alpha X \beta \quad \rightarrow \quad \alpha w \beta$$

With $X \in V$, $\alpha, \beta, w \in (\Sigma \cup V)^*$, $w \neq \lambda$.

Context-sensitive grammars generate context-sensitive languages.

Context-sensitive languages

Example: $\{\mathtt{a}^n\mathtt{b}^n\mathtt{c}^n \mid n > 0\}$

$$
\begin{array}{lcl}
S & \to & \mathtt{a}BC \mid \mathtt{a}SBC \\
CB & \to & XB \\
XB & \to & XC \\
XC & \to & BC \\
\mathtt{a}B & \to & \mathtt{ab} \\
\mathtt{b}B & \to & \mathtt{bb} \\
\mathtt{b}C & \to & \mathtt{bc} \\
\mathtt{c}C & \to & \mathtt{cc}
\end{array}
$$

Note that this is a context-sensitive language.
A crucial point is that $CB \Rightarrow BC$ and thereby, e.g.

$$S \Rightarrow aaBCBC \Rightarrow aaBBCC \Rightarrow aabbcc$$

Enumerable and computable languages

An unrestricted grammar has rules like:

$$u \quad \rightarrow \quad v$$

With $u, v \in (\Sigma \cup V)^*$.

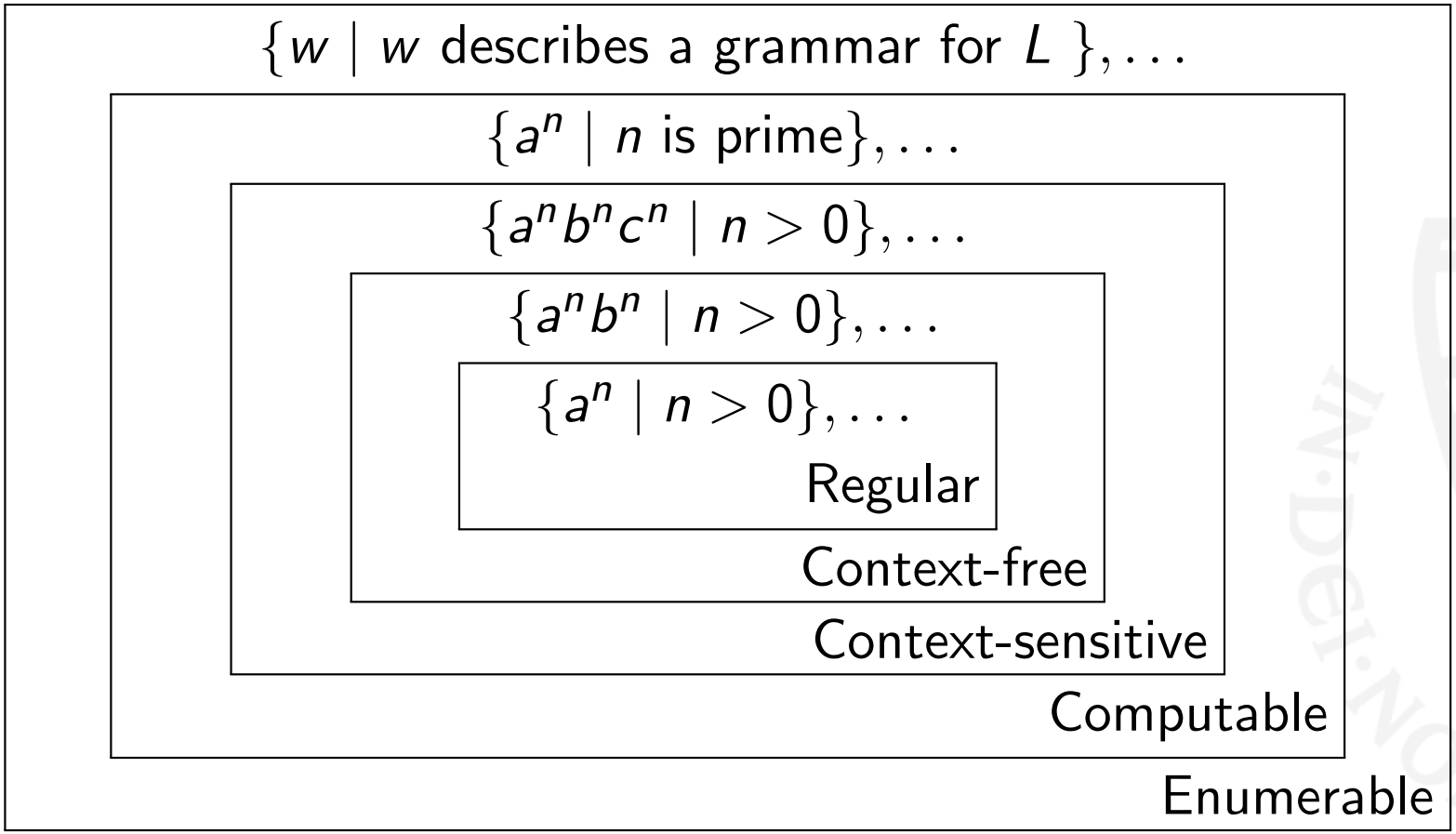
Recognized by Turing machines.

Unrestricted grammars generate enumerable languages.

A languages is computable if both L and $\overline{L} = \Sigma^* - L$ are enumerable.

Example: $\{\mathtt{a}^n \mid n \text{ is not prime}\}$.

Chomsky hierarchy



Trade-offs

Bigger classes of languages:

- More languages can be described.
- But, you can say less about them.

	$w \in L$?	time	memory	$L_1 = L_2$?
Regular	yes	$\lvert w \rvert$	const.	yes
Context-free	yes	$\lvert w \rvert^3$	$\lvert w \rvert^2$	no
Context-sensitive	yes	$2^{\lvert w \rvert}$	$\lvert w \rvert^k$	no
Computable	yes	∞	∞	no
Enumerable	if $w \in L$	∞	∞	no

Programming languages

Most programming languages are context-free.

There are tools to automatically build:

- a lexical analyzer ('lexer') from regular expressions.
- a parser from a CFG.

Lindenmayer systems

Model of organism growth.

Example:

$$\begin{array}{lcl} A & \rightarrow & AB \\ B & \rightarrow & A \end{array}$$

Start symbol A.

Expand once per iteration

0	A
1	AB
2	ABA
3	$ABAAB$
4	$ABAABABA$
	$\ldots$

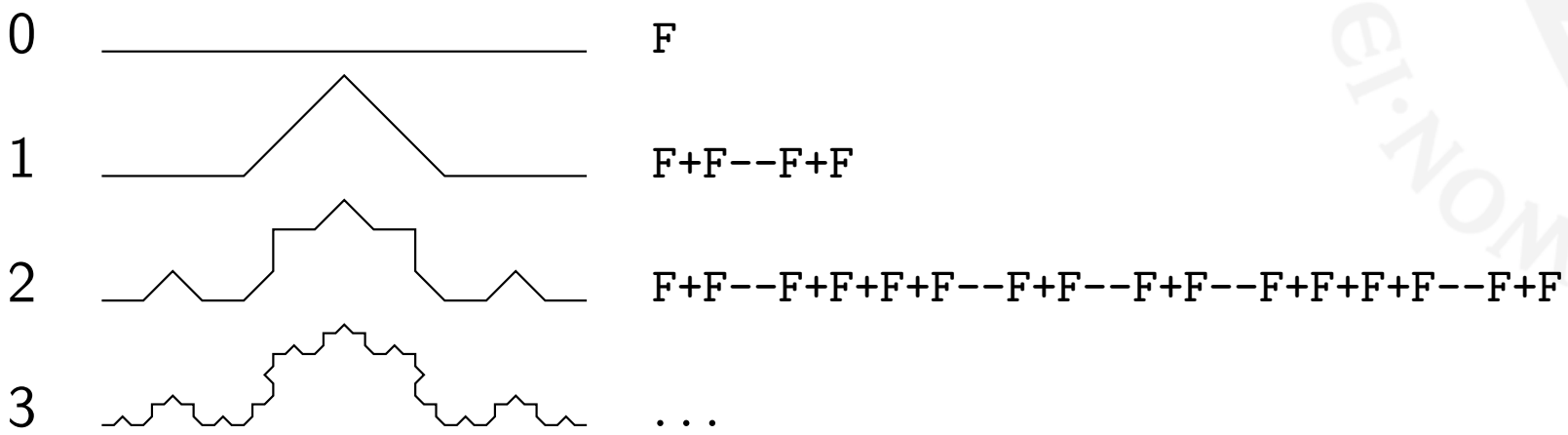
Lindenmayer systems

Drawing a Lindenmayer system:

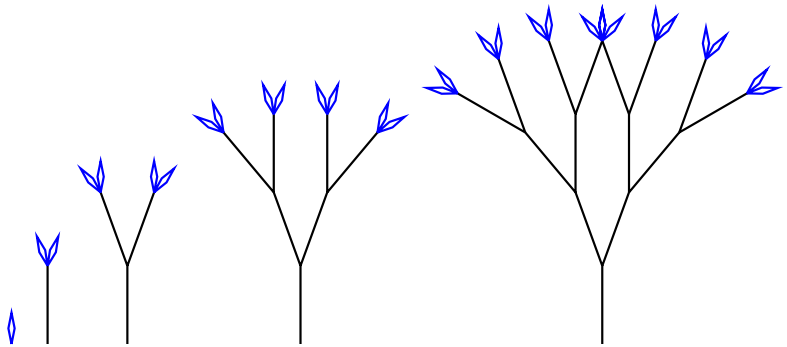
- `F`: move forward
- `+`: rotate clockwise
- `-`: rotate counter clockwise
- `[···]`: branch

Example:

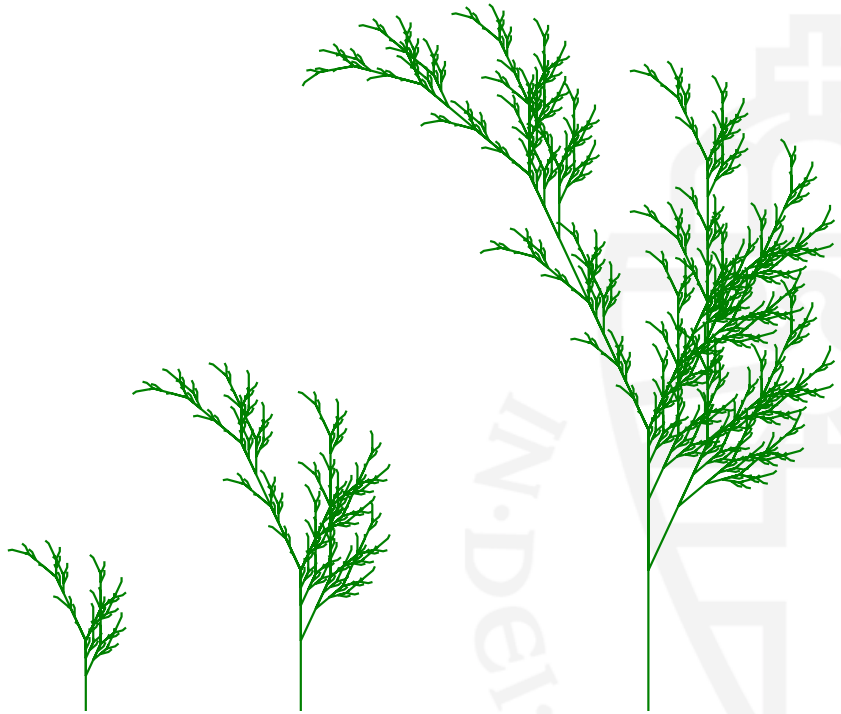
$$F \rightarrow F + F - - F + F$$



Lindenmayer systems



$$S \rightarrow F[+S][-S]$$



$$\begin{aligned} S &\rightarrow F-[[S]+S]+F[+FS]-S \\ F &\rightarrow FF \end{aligned}$$