Grammars and Context Free Languages

Twan van Laarhoven

Institute for Computing and Information Sciences – Intelligent Systems
Radboud University Nijmegen

Version: fall 2014

Midterm recap

View graded tests in the brake.

Common mistakes:

- A language *contains* words.
- L^* is $\{w_1 w_2 \cdots w_n \mid w_i \in L, n \geq 0\}$, not $\{w^n \mid w \in L, n \geq 0\}$.
- $\lambda \in L^*$ regardless of L.
- To show that $L = L'$: show $L \subseteq L'$ and $L \supseteq L'$.
- Converting a DFA to a regular expression: use the algorithm.
- Pumping lemma.

Outline

Generating words

Example:

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

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

Derivations of 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.

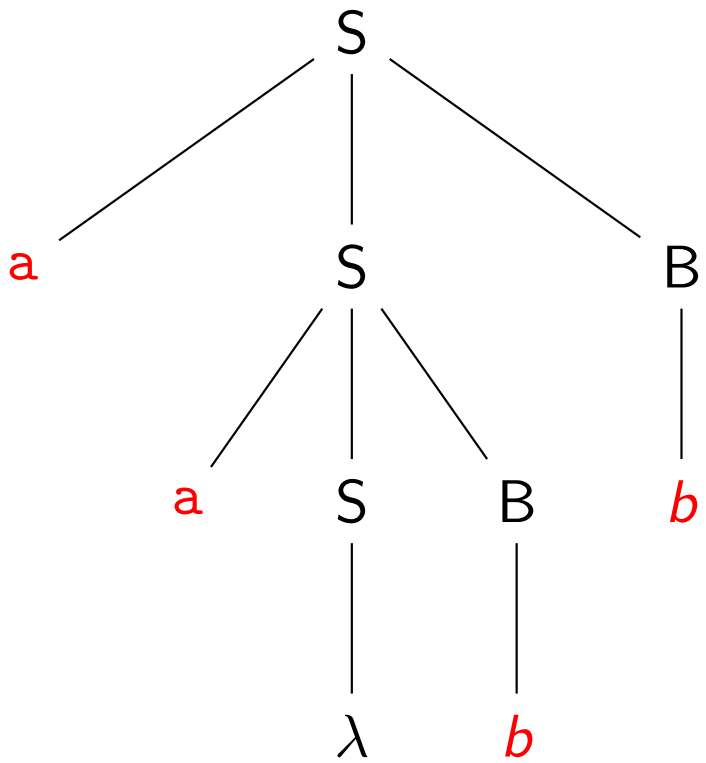
Def. A derivation is leftmost if in each step a rule is applied to the leftmost non-terminal.

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).

Parse trees

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



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"

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

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

Theorem. Every regular language is context-free.

Proof: Let $M = \langle Q, q_0, \delta, F \rangle$ be a NFA that accepts $L \subseteq \Sigma^*$.
Define a context-free grammar G_M as follows:

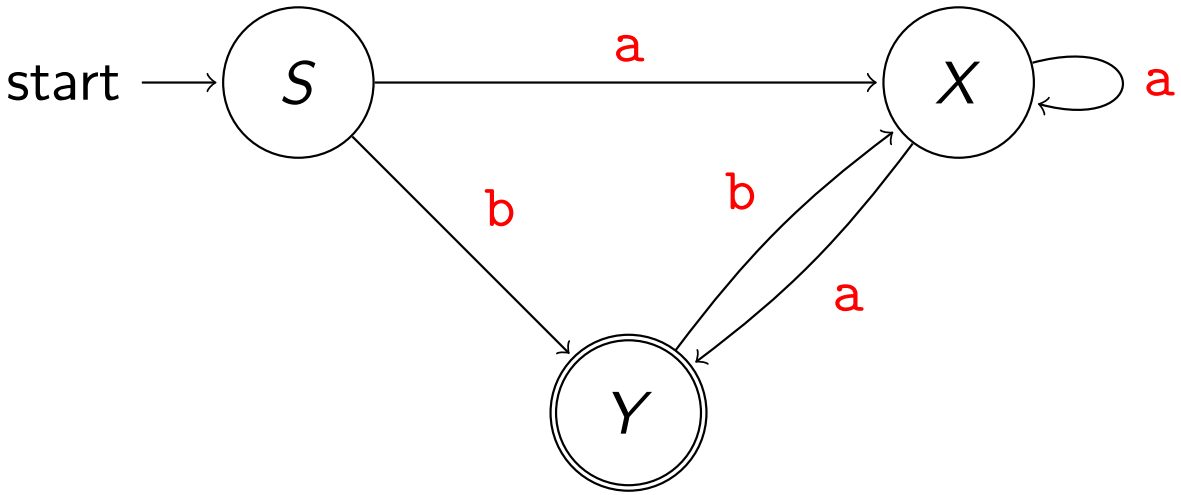
$$V = Q \qquad \text{non-terminals are states}$$
$$S = q_0$$
$$P = \{q \to \mathtt{a}q' \mid q' \in \delta(q, \mathtt{a})\} \cup \{q \to \lambda \mid q \in F\}$$

$\mathcal{L}(G_M) = L(M)$, since computations correspond to derivations:

$$M : q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} q_n \in F$$

$$G_M : q_0 \Rightarrow a_1 q_1 \Rightarrow a_1 a_2 q_2 \Rightarrow \cdots \Rightarrow a_1 \cdots a_n q_n \Rightarrow a_1 \cdots a_n \lambda$$

Regular languages: example



Corresponding grammar:

$$\begin{array}{lcl} S & \to & \mathtt{a}X \mid \mathtt{b}Y \\ X & \to & \mathtt{a}X \mid \mathtt{a}Y \\ Y & \to & \mathtt{b}X \mid \lambda \end{array}$$

Regular Grammars

A regular grammar is a context-free grammar $G = \langle V, S, P \rangle$ in which all rules have the form

$$X \to uY \quad \text{or} \quad X \to \lambda$$

where $X, Y \in V$ and $u \in \Sigma$.

Theorem. If G is a regular grammar, then $\mathcal{L}(G)$ is a regular language.

Regular Grammars

Theorem. If G is a regular grammar, then $\mathcal{L}(G)$ is a regular language.

Proof: Build an NFA $M_G = \langle Q, \delta, q_0, F \rangle$ as follows:

- State set $Q = V$, $q_0 = S$,
- $F = \{X \in V \mid (X \to \lambda) \in P\}$.
- $\delta(X, u) = \{Y \in V \mid (X \to uY) \in P\}$

$\mathcal{L}(M_G) = L(G)$, since derivations correspond to computations:

$$G : S \Rightarrow a_1 X_1 \Rightarrow a_1 a_2 X_2 \Rightarrow \cdots \Rightarrow a_1 \cdots a_n X_n \Rightarrow a_1 \cdots a_n \lambda$$

$$M_G : S \xrightarrow{a_1} X_1 \xrightarrow{a_2} \cdots \xrightarrow{a_n} X_n \in F$$