Context-Free Languages

Context-Free Languages

A **context-free grammar** is a 4-tuple $G = (V, \Sigma, R, S)$. Here

1. V is the set of **variables** (or **nonterminals**),
2. Σ is the set of **terminals** with $V \cap \Sigma = \emptyset$,
3. R is the set of **rules**, each of which is of the form

   $$A \to w,$$

   where $A \in V$ and w is a string over $V \cup \Sigma$; and
4. S is the start **symbol.**

Derivation

The process of generating a string. Start with $u = S$, repeat the following until u is variable-free:

> Find a variable A in u, find a rule in R of the form $A \to w$, replace the A by w.

Use $A \Rightarrow u$ to denote that u is **derived** from A.

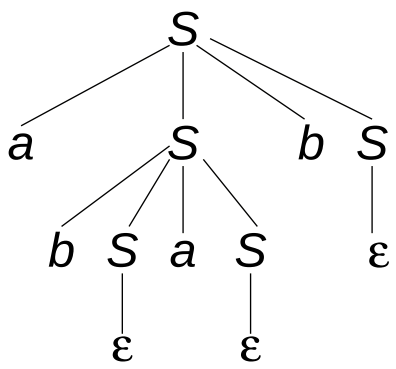
A **parse tree** (or **derivation tree**) is a tree that depicts the process of derivation.

Example: The strings over $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ consisting of an equal number of $\mathtt{a}$'s and $\mathtt{b}$'s

$V = \{S\}$ and the derivation rules are $S \to \epsilon \mid \mathtt{a}S\mathtt{b}S \mid \mathtt{b}S\mathtt{a}S$.

$\mathtt{abab}$ is derived as follows:

$$S \Rightarrow \mathtt{a}S\mathtt{b}S \Rightarrow \mathtt{ab}S\mathtt{a}S\mathtt{b}S \Rightarrow \mathtt{ab}S\mathtt{ab}S \Rightarrow \mathtt{abab}S \Rightarrow \mathtt{abab}.$$



Leftmost Derivation & Ambiguity

A **leftmost derivation** is the derivation in which each production rule is applied to the leftmost variable. The following derivation of abab

$$S \Rightarrow \mathtt{a}\underline{S}\mathtt{b}S \Rightarrow \mathtt{ab}\underline{S}\mathtt{a}S\mathtt{b}S \Rightarrow \mathtt{aba}\underline{S}\mathtt{b}S \Rightarrow \mathtt{abab}\underline{S} \Rightarrow \mathtt{abab}$$

is a leftmost derivation (giving the same parse tree as before) while

$$S \Rightarrow \mathtt{a}S\mathtt{b}\underline{S} \Rightarrow \mathtt{a}S\mathtt{ba}\underline{S}\mathtt{b}S \Rightarrow \mathtt{a}S\mathtt{bab}\underline{S} \Rightarrow \mathtt{a}\underline{S}\mathtt{bab} \Rightarrow \mathtt{abab}$$

is not (and the parse tree is different as well).

A context-free grammar is **unambiguous** if it has a unique leftmost derivation for every word (sentence) it generates.

There is an **inherently ambiguous** context-free language.

Chomsky Normal Form

A context-free grammar $G = (V, \Sigma, R, S)$ is in **Chomsky normal form** if each rule in R is either of the form $A \to BC$ for some $B, C \in V - \{S\}$ or of the form $A \to a$ with $a \in \Sigma$, except that $S \to \epsilon$ is permitted.

Theorem. *Each context-free language L is generated by a Chomsky normal form grammar.*

Proof of the Theorem

Step 1 Add new start symbol S_0 with a unique production rule $S_0 \rightarrow S$. If $S \rightarrow \epsilon \in R$ then add $S_0 \rightarrow \epsilon$.

Step 2 Elimination of ϵ rules

While there is a variable $A \neq S_0$ such that $A \rightarrow \epsilon \in R$

- for each rule r of the form $B \rightarrow y$ with an A in y, replace r with the collection of all rules of the form $B \rightarrow y'$ such that y' is constructed from y by eliminating some (possibly none) of the occurrences of A;
- eliminate $A \rightarrow \epsilon$.

Proof of the Theorem (cont'd)

Step 3 Elimination of Unit Rules

While there is a unit rule $A \rightarrow B$ with $B \in V$,

- eliminate the rule and
- if $B \neq A$, then for each rule $B \rightarrow w$, add $A \rightarrow w$ (provided that $A \rightarrow w$ wasn't previously eliminated)

Proof of the Theorem (cont'd)

Step 4 Normalization

For each terminal d

- add a new nonterminal D,
- add a new rule $D \to d$, and
- for each rule $A \to u, |u| > 1$, in which d occurs, replace each occurrence of d with a D.

For each rule $A \to w_1 \ldots w_m$, $m \geq 3$,

- add a new variable X and
- replace the rule with $A \to w_1 X$ and $X \to w_2 \cdots w_m$.

Proof of the Theorem (concluded)

It's pretty clear the transformation "works", but strictly, we should show this.

In particular, we can argue that

- the iterations in steps 2, 3 and 4 terminate (which is not completely obvious for step 3...); and
- each transformation step preserves the language generated.

Example

$V = \{S\}, \Sigma = \{\mathtt{a}, \mathtt{b}\}$, and R consists of $S \rightarrow \epsilon \mid \mathtt{a}S\mathtt{b}S \mid \mathtt{b}S\mathtt{a}S$

Step 1 Add $S_0 \rightarrow S \mid \epsilon$.

Step 2 Eliminate $S \rightarrow \epsilon$. The rules are

$$\begin{aligned} S_0 &\rightarrow S \mid \epsilon, \\ S &\rightarrow \mathtt{ab} \mid \mathtt{ab}S \mid \mathtt{a}S\mathtt{b}S \mid \mathtt{a}S\mathtt{b} \mid \\ &\quad\; \mathtt{ba} \mid \mathtt{ba}S \mid \mathtt{b}S\mathtt{a}S \mid \mathtt{b}S\mathtt{a}. \end{aligned}$$

STEP 3 Eliminate $S_0 \rightarrow S$ and add

$$\begin{aligned} S_0 &\rightarrow \mathtt{ab} \mid \mathtt{ab}S \mid \mathtt{a}S\mathtt{b}S \mid \mathtt{a}S\mathtt{b} \mid \\ &\quad\; \mathtt{ba} \mid \mathtt{ba}S \mid \mathtt{b}S\mathtt{a}S \mid \mathtt{b}S\mathtt{a} \end{aligned}$$

Example (concluded)

STEP 4 The rules are

$$S_0 \to \epsilon, \quad A \to \texttt{a}, \quad B \to \texttt{b},$$
$$S_0 \to AB \mid AX_1 \mid AX_2 \mid AX_3 \mid BA \mid BY_1 \mid BY_2 \mid BY_3,$$
$$S \to AB \mid AX_1 \mid AX_2 \mid AX_3 \mid BA \mid BY_1 \mid BY_2 \mid BY_3,$$
$$X_1 \to BS, \quad X_2 \to SX_4,$$
$$X_3 \to SB, \quad X_4 \to BS,$$
$$Y_1 \to AS, \quad Y_2 \to SY_4,$$
$$Y_3 \to SA, \quad Y_4 \to AS.$$