Chapter 3 Context-Free Languages (CFL)

Human languages (speech) obey rules of a grammar.

noun verb preposition

noun phrases verb phrases prepositional phrases

- ways they can be put together into grammatical sentences are governed by a grammar.

You can use a grammar to generate strings in a language, or to check a string to determine if it is grammatical.

<sentence> → <noun-phrase> <verb-phrase>

<noun-phrase> → <cmplx-noun> | <cmplx-noun> <prep-phrase>

<verb-phrase> → <cmplx-verb> | <cmplx-verb> <prep-phrase>

<cmplx-noun> → <article> <noun>

<cmplx-verb> → <verb> | <verb> <noun-phrase>

<article> → a | the | her

<noun> → cat | dog | paw

<verb> → swats | likes | sees

<prep> → with

<prep-phrase> → <prep> <noun-phrase>

<sentence> ⇒ <noun p> <verb p>

⇒ <complex n> <verb p>

⇒ <art> <noun> <verb phrase>

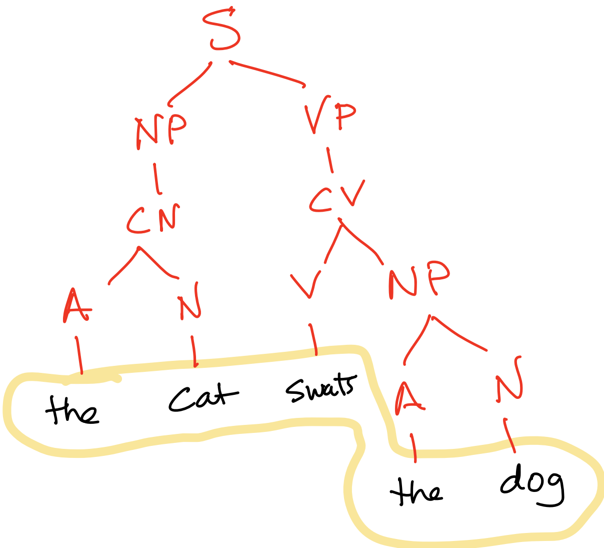
⇒ the <noun> <verb phrase>

$\Rightarrow$ the cat <verb phrase>

$\Rightarrow$ the cat

$\vdots$

$\Rightarrow$ the cat swats the dog.



derived string is on the leaves L − R + consist of terminals.

Definition 2.2 A Context-Free Grammar (CFG) is a 4-tuple (V, Σ, R, S) where

1. V is a finite set of variables
2. Σ is a finite set of terminals $\Sigma \cap V = \emptyset$.
3. R is a finite set of rules, each being a variable and a string from $(\Sigma \cup V)^*$
4. $S \in V$ is start variable.

If u, v, w are strings $\in (\Sigma \cup V)^*$ and $A \to w$ is a rule in the grammar, then we say

uAv "yields (in one step)" uwv

$$uAv \Rightarrow uwv$$

We say "u derives v" write $u \overset{*}{\Rightarrow} v$

if : $u = v$, or

$\exists$ $u \Rightarrow u_1 \Rightarrow u_2 \Rightarrow \dots \Rightarrow v$

The language $L(G)$ of grammar G is

$$\{ w \in \Sigma^* \mid S \overset{*}{\Rightarrow} w \}$$

Eg $G = (\{S\}, \{a,b\}, R, S)$

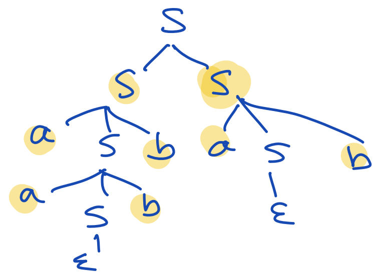
and R is $S \to aSb \mid SS \mid \varepsilon$ "or"

(that is the same as $S \to aSb$
$S \to SS$
$S \to \varepsilon$)

$aabb$

$S \Rightarrow$

aabbab ?



* Leftmost derivation

$$S \Rightarrow SS \Rightarrow aSbS \Rightarrow aaSbbS \Rightarrow aabbS$$

$$\Rightarrow aabbaSb \Rightarrow aabbab.$$

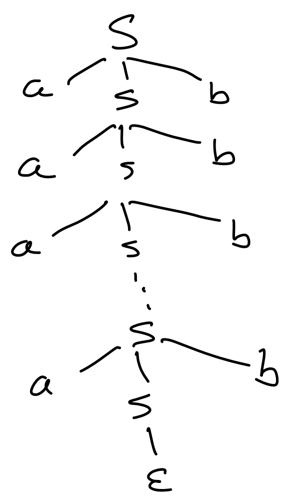
– always apply production rule to the leftmost variable.

Rightmost derivation:

$$S \Rightarrow SS \Rightarrow SaSb \Rightarrow Sab \Rightarrow aSbab \Rightarrow aaSbbab$$

$$\Rightarrow aabbab$$

Fun with Grammars.



$A^nB^n = \{a^n b^n \mid n \geq 0\}.$

$G: \quad S \to aSb \mid \varepsilon$

$\{\varepsilon, ab, aabb, \ldots, \}.$

A string w is $\in L(G)$ iff .

1. $w = \varepsilon$
2. $w = aw'b$ where $w' \in L(G)$

Bal is context-free:

$$S \to SS \mid (S) \mid \varepsilon$$

Bal: 1. $\#_(= \#_)$

2. In all prefixes $\#_(\geq \#_)$

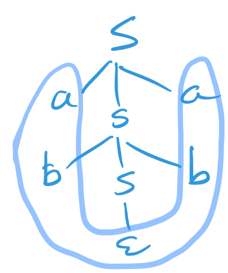
To argue that a CFG G is correct for a language X, (i.e. that $L(G) = X$), we need to show:

1. $w \in L(G) \Rightarrow w \in X$
2. $w \in X \Rightarrow w \in L(G)$

1. $\{w \in \{a,b\}^* \mid \#_a(w) = \#_b(w)\}$.
2. EvenPal $= \{w \in \{a,b\}^* \mid |w|$ is even, w is palindrome$\}$

EvenPal $= \{ww^R \mid w \in \{a,b\}^*\}$

$$S \to aSa \mid bSb \mid \varepsilon$$



abba

$$\text{OddPal} = \{ w \in \{a,b\}^* \mid |w| \text{ is odd, and } w = w^R \}$$

$$S \to aSa \mid bSb \mid a \mid b$$

Pal: $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

$$L_1 = \{ a^{2k} b^k : k \in \mathbb{W} \}$$

$$L_2 = \{ a^i b^j : i \text{ is odd, } j \text{ is even} \}$$