1 Introduction

- Thus far we have mainly dealt with **language recognizers**, which are devices to recognize whether a given string belongs to a language or not.

- There are also **language generators**, which generate all and only the grammatical sentences of a language.

- Let us look at the regular expression $a(a^* \cup b^*)b$ from the generation perspective:

  First output an a, then
  either output a number of a's or a number of b's, then
  output a b, and stop.

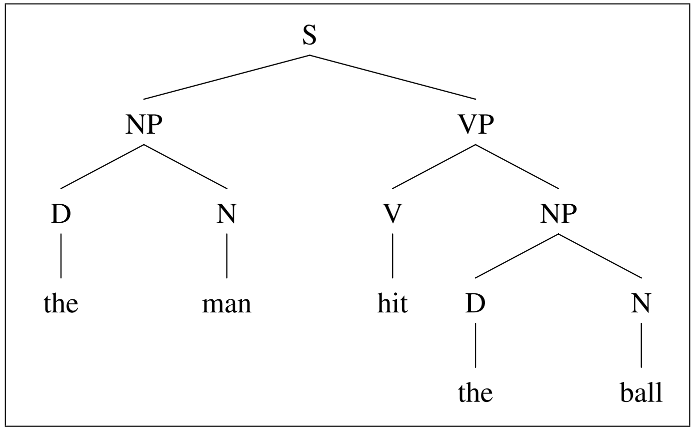
- Let's see a generator – a context-free grammar – for a tiny fragment of English:

  i. $S \rightarrow NP + VP$
  ii. $NP \rightarrow D + N$
  iii. $VP \rightarrow V + NP$
  iv. $D \rightarrow the$
  v. $N \rightarrow man, ball,$ etc.
  vi. $V \rightarrow hit, took,$ etc.

- A **leftmost**[1] derivation of the string $the + man + hit + the + ball$:

Derivation	*Rules*
S	$S \rightarrow NP + VP$
$NP + VP$	$NP \rightarrow D + N$
$D + N + VP$	$VP \rightarrow V + NP$
$D + N + V + NP$	$D \rightarrow the$
$the + N + V + NP$	$N \rightarrow man$
$the + man + V + NP$	$V \rightarrow hit$
$the + man + hit + NP$	$NP \rightarrow D + N$
$the + man + hit + D + N$	$D \rightarrow the$
$the + man + hit + the + N$	$N \rightarrow ball$
$the + man + hit + the + ball$	

[1] In every step of the derivation, you expand the leftmost nonterminal in the current string. The notion of **rightmost** derivation is defined similarly.



- The parse tree (constituent structure) contains less information than the derivation (Why?).

- A terminal string σ covered by a single node X is a constituent of type X.

- The terminal string covered by the root node is the **yield** of the parse tree.

  Exercise 1.1.
  Introduce new rules to the grammar so that it can handle modification of NPs and VPs by PPs.

2 Formal Definition

Definition 2.1.
A **context-free grammar** G is a quadruple $\langle V_N, V_T, R, S\rangle$ where

V_N is the set of **non-terminal** symbols,

V_T is the set of **terminal** symbols,

R (the set of **rules**) is a finite subset of $V_N \times (V_N \cup V_T)^*$,

$S \in V_N$ is the **start** symbol.

For any $A \in V_N$ and $u \in (V_N \cup V_T)^*$, we write $A \to u$ whenever $\langle A, u\rangle \in R$.

For any $u, v \in (V_N \cup V_T)^*$, we write $u \Rightarrow v$ if and only if there are strings $x, y \in (V_N \cup V_T)^*$ such that $u = xAy$, $v = xty$, and $A \to t$.

We say that a string t is **generated** by a grammar $G = \langle V_N, V_T, R, S\rangle$ if and only if there exists a sequence of the form:

$$S \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n \Rightarrow t$$

for $n \geq 0$ and $w_i \in (V_T \cup V_N)^*$.

The language generated by a grammar G, denoted as $L(G)$, is the set $\{w \in V_T^* \mid w \text{ is generated by } G\}$.

A language A is a context-free language if and only if there exists some context-free grammar G such that $A = L(G)$.

Example 2.2.
Write context-free grammars for the languages:

(a) $a(a^* \cup b^*)b$

(b) $\{ww^R \mid w \in \{a,b\}^*\}$

(c) $\{w \in \{a,b\}^* \mid w = w^R\}$

(d) $\{a^n b^m a^n \mid n, m \geq 1\}$

(e) $\{a^n b^n a^m b^m \mid n, m \geq 1\}$

(f) $\{a^n b^m c^m d^{2n} \mid n, m \geq 1\}$

(g) $\{a^n b^m \mid 0 \leq n \leq m \leq 2n\}$

(h) Set of strings with exactly two b's.

(i) ... with at least two b's.

(j) ... with an even length.

(k) ... with an even number of b's.

(l) ... with a positive even number of b's.

3 CFLs and Regular Languages (FALs)

- Not every context-free language is a regular language. (We have already seen this through counterexamples.)
- Every regular language is a context-free language (proof by *direct construction*).
- **Direct Construction:** For any deterministic finite state machine $M = \langle K, \Sigma, \delta, q_1, F\rangle$, you can construct a context-free grammar $G_M = \langle K, \Sigma, R, q_0\rangle$ with

$$R = \{q \to ap \mid \delta(q,a) = p\} \cup \{q \to \varepsilon \mid q \in F\}$$

such that $L(M) = L(G_M)$.
- Such grammars are called **regular grammars**.
- A regular language can be generated by a non-regular but context-free grammar.

4 Closure Properties of CFLs

Union:

Given two context-free grammars $G_1 = \langle V_{N_1}, V_{T_1}, R_1, S_1\rangle$ and $G_2 = \langle V_{N_2}, V_{T_2}, R_2, S_2\rangle$ form the grammar $G = \langle V_N, V_T, R, S\rangle$ in the following way,

i. If the non-terminals of G_1 and G_2 are not disjoint, make them so (e.g. by putting primes to those of G_2).

ii. Let $R = \{S \to S_1, S \to S_2\} \cup R_1 \cup R_2$.

G will generate all and only the strings that are generated by G_1 or G_2, or both, namely $L(G) = L(G_1) \cup L(G_2)$. Therefore, CFL's are closed under union.

Concatenation:

The method of constructing G from G_1 and G_2 is the same except that $R = \{S \to S_1S_2\} \cup R_1 \cup R_2$.

Kleene star:

Given a context-free grammar $G = \langle V_N, V_T, R, S\rangle$ one can construct a grammar G' that generates $L(G)^*$ in the following way:

i. Let S' be the start symbol of G'.

ii. Let $R' = \{S' \to S'S, S' \to \varepsilon\} \cup R$.

- CFLs are *not* closed under *intersection* and *complementation*.
- Intersection of a CFL with a FAL is a CFL.