

Chapter 2, Part 1

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**,
3. R is the set of **rules**, each of which is of the form

$$A \rightarrow w,$$

where A is a nonterminal and w is a string over $V \cup \Sigma$;
and

4. S is a nonterminal called the **start symbol**.

Derivation

Derivation refers to the process of generating strings with a context-free grammar. Derivation may start from any word u over $V \cup \Sigma$. We write $u \Rightarrow v$ to mean that v can be produced by repeatedly applying the following:

- Pick any occurrence of any nonterminal in u .
- Let $u = xAy$ where A is the identified occurrence of the nonterminal.
- Pick any rule in R of the form $A \rightarrow w$.
- Replace the identified A in u by w to turn u into xwy .

We say that G produces $w \in \Sigma^*$ if $S \Rightarrow w$, i.e., the process produces w starting from S .

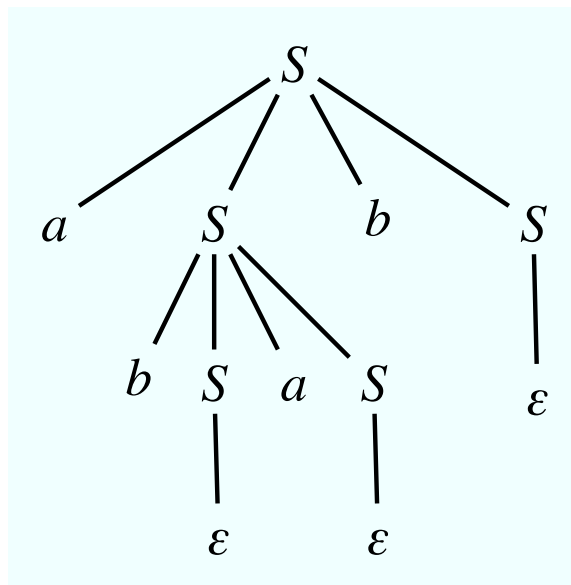
A **parse tree** (or **derivation tree**) is a tree that depicts the process of derivation.

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

$V = \{S\}$ and the derivation rules are $S \rightarrow \epsilon \mid aSbS \mid bSaS$.

abab is derived as follows:

$$S \Rightarrow aSbS \Rightarrow abSaSbS \Rightarrow abSabS \Rightarrow ababS \Rightarrow abab.$$



Two More Examples:

$$\{0^n 1^n \mid n \geq 0\}$$

$$S \rightarrow \epsilon \mid 0S1$$

$$\{0^m 1^n \mid n \geq m \geq 0\}$$

$$S \rightarrow \epsilon \mid 0S1 \mid S1$$

Leftmost Derivation

A **leftmost derivation** is the derivation in which each production rule is applied to the leftmost nonterminal at the moment. For abab in the previous example,

$$S \Rightarrow a\underline{S}bS \Rightarrow ab\underline{S}aSbS \Rightarrow aba\underline{S}bS \Rightarrow abab\underline{S} \Rightarrow abab$$

is a leftmost derivation while

$$S \Rightarrow aSb\underline{S} \Rightarrow aSba\underline{S}bS \Rightarrow aSbab\underline{S} \Rightarrow a\underline{S}bab \Rightarrow abab$$

isn't one.

Ambiguity

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

There is a context-free language that is **inherently ambiguous**, in the sense that not a single context-free grammar that produces the language is unambiguous.

One example of inherently ambiguous languages is $\{0^i 1^j 2^k \mid \text{either } (i = j) \text{ or } (j = k)\}$.

Chomsky Normal Form

A context grammar $G = (V, \Sigma, R, S)$ is in **Chomsky normal form** if, except for a permissible rule $S \rightarrow \epsilon$, every rule in R is one of the following two forms:

- $A \rightarrow BC$ for some $B, C \in V - \{S\}$ and
- $A \rightarrow a$ for some $a \in \Sigma$.

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

Proof of the Theorem

Let $G = (V, \Sigma, R, S)$ be an arbitrary CFG and let $L = L(G)$.

We will convert G to an equivalent Chomsky normal form grammar.

A general idea:

- Remove rules that produce ϵ .
- Remove rules that transform a nonterminal to a nonterminal.

Now, for every rule $A \rightarrow w$, we have

- $|w| \geq 1$ and
 - if $|w| = 1$ then w is a terminal.
- Express each rule $A \rightarrow w$ such that $|w| \geq 3$ as a collection of rules such that the right-hand side of each rule consists of either two nonterminals or one terminal.

Proof of the Theorem (cont'd)

Step 1 Initialization

- Add a new start symbol S_0 .
- Introduce a unique production rule from S_0 : $S_0 \rightarrow S$.
- If $\epsilon \in L$ then add $S_0 \rightarrow \epsilon$. (There is an algorithm for finding whether $\epsilon \in L$.)

Proof of the Theorem (cont'd)

Step 2 Elimination of ϵ rules

While there is a variable $A \neq S_0$ such that $A \rightarrow \epsilon \in R$, pick such an A and do the following:

- For each rule r of the form $B \rightarrow y$ such that A appears 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 $r : A \rightarrow B$ with $B \in V$, pick such a rule r and do the following:

- Eliminate the rule r .
- If $B \neq A$, then for each rule $B \rightarrow w$, add $A \rightarrow w$.

Proof of the Theorem (cont'd)

Step 4 Normalization

For each terminal d

- add a new nonterminal D ,
- add a new rule $D \rightarrow d$, and
- for each rule $A \rightarrow u$ such that $|u| \geq 2$ and d appears in u , replace each occurrence of d with a D .

For each rule $A \rightarrow w_1 \dots w_m$ such that $m \geq 3$, do the following:

- Add a new variable X .
- Replace $A \rightarrow w$ by two rules: $A \rightarrow w_1 X$ and $X \rightarrow w_2 \dots w_m$.

Example

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

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

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

$$S_0 \rightarrow S \mid \epsilon,$$

$$S \rightarrow ab \mid abS \mid aSbS \mid aSb \mid$$

$$ba \mid baS \mid bSaS \mid bSa.$$

STEP 3 Eliminate $S \rightarrow S_0$ and add

$$S_0 \rightarrow ab \mid abS \mid aSbS \mid aSb \mid$$

$$ba \mid baS \mid bSaS \mid bSa$$

Example (cont'd)

STEP 4 The rules are

$$S_0 \rightarrow \epsilon, \quad A \rightarrow \mathbf{a}, \quad B \rightarrow \mathbf{b},$$

$$S_0 \rightarrow AB \mid AX_1 \mid AX_2 \mid AX_3 \mid BA \mid BY_1 \mid BY_2 \mid BY_3,$$

$$S \rightarrow AB \mid AX_1 \mid AX_2 \mid AX_3 \mid BA \mid BY_1 \mid BY_2 \mid BY_3,$$

$$X_1 \rightarrow BS, \quad X_2 \rightarrow SX_4,$$

$$X_3 \rightarrow SB, \quad X_4 \rightarrow BS,$$

$$Y_1 \rightarrow AS, \quad Y_2 \rightarrow SY_4,$$

$$Y_3 \rightarrow SA, \quad Y_4 \rightarrow AS.$$

Here we are using the same variables X_1, \dots, X_4 and Y_1, \dots, Y_3 for S_0 and S .