Applications of Grammars

- Specifying syntax of programming languages
- Representing syntactic structures in natural languages
- Models of computation

Context-free grammars are the most commonly used kind of grammar in computer science.
Progression of Concepts

- **Context-free grammar:** Variables, terminals, rules, start symbol

- **Derivation:** How one string derives another

- **Context-free language:** Generated by a context-free grammar

- **Derivation tree:** Graphically representing a derivation
Context-Free Grammar

A context-free grammar (CFG) is a 4-tuple $G = (V, \Sigma, P, S)$ where

- $V$ is an alphabet of **variables** (or **nonterminals**);

- $\Sigma$ is an alphabet of **terminals**, disjoint from $V$;

- $P$ is a finite subset of $V \times (V \cup \Sigma)^*$, called the set of **rules** (or **productions**); and

- $S \in V$ is the **start symbol**.
Example CFG

\[ G_1 = (\{S, A, B\}, \{a, b\}, \]
\[ \{(S, aAbBa), (A, aS), (B, Ab), \]
\[ (B, SbB), (B, \lambda)\}, S) \]

Variables are capitalized. The start symbol is (almost) always \( S \).

Rules are usually written

\[ S \rightarrow aAbBa \]

or

\[ S \xrightarrow{G_1} aAbBa \]

to emphasize the grammar \( G_1 \).
Example CFG Continued

$G_1$ is compactly written:

$S \rightarrow aAbBa$

$A \rightarrow aS$

$B \rightarrow Ab$

$B \rightarrow SbB$

$B \rightarrow \lambda$
One-Step Derivation

Want to define $\Rightarrow$, a binary relation on $(V \cup \Sigma)^*).

Let $G = (V, \Sigma, P, S)$ be a CFG. Suppose that $u, v \in (V \cup \Sigma)^*$, that $A \in V$, and that

$$A \xrightarrow{G} w$$

is a rule. Then the string $uAv$ derives (in one step) the string $uvw$, written

$$uAv \xrightarrow{G} uvw.$$ 

**EXAMPLE.**

$$BaAb \xrightarrow{G_1} SbBaAb$$

$$BaAb \xrightarrow{G_1} BaaSb$$
Derivation in Zero or More Steps

We give a recursive definition for $\Rightarrow^*$, a binary relation on $(V \cup \Sigma)^*$.

- **Basis:** If $v \in (V \cup \Sigma)^*$, then $v \Rightarrow^* v$. Also, $v \Rightarrow^0 v$, read $v$ derives $v$ in zero steps.

- **Recursive step:** If $u, v, w \in (V \cup \Sigma)^*$
  $u \Rightarrow^* v$, and $v \Rightarrow w$, then $u \Rightarrow^* w$. Also, if $u \Rightarrow^n v$ and $v \Rightarrow w$, then $u \Rightarrow^{n+1} w$. 

Derivation Example

Here is a derivation in $G_1$:

$$BaAb \implies SbBaAb$$
$$\implies SbBaaSb$$
$$\implies SbBaaaAbBab$$
$$\implies SbaaaAbBab$$
$$\implies SbaaaAbAbab$$
$$\implies SbaaaSbAbab.$$

We can conclude, for example, that

$$BaAb \xrightarrow{4} SbaaaAbBab$$

and that

$$BaAb \xrightarrow{*} SbaaaaSbAbab.$$
Context-Free Language

Let $G = (V, \Sigma, P, S)$ be a context-free grammar.

- If $S \xrightarrow[\text{G}]{*} w$, then $w$ is a **sentential form** of $G$.

- If $S \xrightarrow[\text{G}]{*} w$ and $w \in \Sigma^*$, then $w$ is a **sentence** of $G$.

- The **language generated by** $G$, written $L(G)$, is the set of all sentences of $G$:

$$L(G) = \left\{ w \in \Sigma^* \mid S \xrightarrow[\text{G}]{*} w \right\}.$$  

Any language generated by a CFG is a **context-free language (CFL)**.
Continue Example of a CFL

So every

\[(B0)^nB\]

is a sentential form.

We note that \(B\) generates \(1^+\) and that \(\lambda \in L(G_2)\).

We also have sentential forms

\[0(B0)^nB = (0B)^{n+1}.\]

Putting it all together, we find that

\[L(G_2) = \{w \in \{0, 1\}^* | \text{each 0 in } w \text{ is followed immediately by a 1}\}.\]