

# 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

$$S \longrightarrow aAbBa$$

or

$$S \xrightarrow{G_1} aAbBa$$

to emphasize the grammar  $G_1$ .

## Example CFG Continued

$G_1$  is compactly written:

$$S \longrightarrow aAbBa$$

$$A \longrightarrow aS$$

$$B \longrightarrow Ab$$

$$B \longrightarrow SbB$$

$$B \longrightarrow \lambda$$

## One-Step Derivation

Want to define  $\Longrightarrow$ , 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  $uwwv$ , written

$$uAv \xRightarrow[G]{} uwwv.$$

### EXAMPLE.

$$BaAb \xRightarrow[G_1]{} SbBaAb$$

$$BaAb \xRightarrow[G_1]{} BaaSb$$

# Derivation in Zero or More Steps

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

- **Basis:** If  $v \in (V \cup \Sigma)^*$ , then  $v \Longrightarrow^* v$ . Also,  $v \xrightarrow{0} v$ , read  $v$  derives  $v$  in zero steps.
- **Recursive step:** If  $u, v, w \in (V \cup \Sigma)^*$   $u \Longrightarrow^* v$ , and  $v \Longrightarrow w$ , then  $u \Longrightarrow^* w$ . Also, if  $u \xrightarrow{n} v$  and  $v \Longrightarrow w$ , then  $u \xrightarrow{n+1} w$ .

## Derivation Example

Here is a derivation in  $G_1$ :

$$\begin{aligned}
 BaAb &\Longrightarrow SbBaAb \\
 &\Longrightarrow SbBaaSb \\
 &\Longrightarrow SbBaaaAbBab \\
 &\Longrightarrow SbaaaAbBab \\
 &\Longrightarrow SbaaaAbAbab \\
 &\Longrightarrow SbaaaaSbAbab.
 \end{aligned}$$

We can conclude, for example, that

$$BaAb \xrightarrow[G_1]{4} SbaaaAbBab$$

and that

$$BaAb \xrightarrow[G_1]{*} SbaaaaSbAbab.$$



# Context-Free Language

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

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

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

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

## Continue Example of a CFL

So every

$$(B0)^n B$$

is a sentential form.

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

We also have sentential forms

$$0(B0)^n B = (0B)^{n+1}.$$

Putting it all together, we find that

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