

Context-Free Languages

- Grammars - rewrite rules to generate syntactically correct sentences

Natural languages - context-sensitive

Programming languages - restricted expressibility, more structured/rigid

Example:

$$\langle \text{stmt} \rangle := \langle \text{if-stmt} \rangle / \langle \text{while-stmt} \rangle / \langle \text{decl-stmt} \rangle / \langle \text{asgn-stmt} \rangle$$
$$\langle \text{if-stmt} \rangle := \text{if } \langle \text{bool-expr} \rangle \text{ then } \langle \text{stmt-list} \rangle \text{ else } \langle \text{stmt} \rangle$$
$$\langle \text{while-stmt} \rangle := \text{while } \langle \text{bool-expr} \rangle \text{ then } \langle \text{stmt-list} \rangle$$
$$\langle \text{asgn-stmt} \rangle := \langle \text{var} \rangle = \langle \text{arith-expr} \rangle$$
$$\langle \text{stmt-list} \rangle := \langle \text{stmt} \rangle ; \langle \text{stmt-list} \rangle$$
$$\langle \text{var} \rangle := \langle \text{init-char} \rangle \langle \text{char} \rangle \langle \text{var} \rangle$$
$$\langle \text{init-char} \rangle := - / a / b / \dots / z / A / B / \dots / Z$$
$$\langle \text{char} \rangle := \langle \text{init-char} \rangle / 0 / 1 / \dots / 9$$

Formal defn: A CFG is a 4-tuple

(N, Σ, P, S) , where

(i) N - set of non-terminals

↳ Symbols on the
rhs of productions

(ii) Σ - set of terminals

↳ alphabet of the language
generated by the grammar

(iii) P - set of production rules

$P \subseteq N \times (N \cup \Sigma)^*$

↳ rewrite rules to
generate sentences of
the language

(iv) S - start symbol

$S \in N$

↳ non-terminal from which
production starts

Eg: $L = \{0^n 1^n \mid n \geq 0\}$

$S \rightarrow 0S1 \mid \epsilon$