CONTEXT-FREE LANGUAGES

We will define CFGs as forms of grammars introduced.

Define a generative grammar.

RULES for producing words.

def A CONTEXT-FREE GRAMMAR consists of a set of symbols called variables or nonterminals and a set of symbols called non-terminals or terminals $V, V \cap \Sigma = \emptyset$, and $S \in V$ as finite.

(1) A finite set of rules called productions

$$A \to x \quad \begin{array}{l} A \in V \\ x \in (V \cup \Sigma)^* \end{array}$$

EXAMPLE $\Sigma = \{a, b\}$ terminal

$V = \{S\}$

$S \to a \quad S \to SS$

Apply these rules you find like a replacement terminal with the terminal.

$S \to a \Rightarrow \Sigma \to a \Rightarrow \Sigma \Sigma \Rightarrow$

$\to aa \Rightarrow \Sigma \Sigma \Rightarrow$

$\to aaa \Rightarrow \Sigma \Sigma \Rightarrow$

$\to aaaa$ You notice a string word we have non-terminals you have separated in string in the CFG.

$L(G) = \{aa^n \mid n \geq 0\}$

CFGs are already more powerful than regular languages.

EX2 Arithmetic expressions

$\Sigma = \{a, \ldots, z, +, *, (,)\}$

$V = \{\langle \text{EXP}\rangle, \langle \text{TERM}\rangle, \langle \text{FACTOR}\rangle\}$

$\langle \text{EXP}\rangle \to \langle \text{EXP}\rangle + \langle \text{TERM}\rangle \mid \langle \text{TERM}\rangle$

$\langle \text{TERM}\rangle \to \langle \text{TERM}\rangle * \langle \text{FACTOR}\rangle \mid \langle \text{FACTOR}\rangle$

$\langle \text{FACTOR}\rangle \to (\langle \text{EXP}\rangle) \mid a \mid b \mid \ldots \mid z$

$\langle \text{EXP}\rangle \to \langle \text{TERM}\rangle \mid \langle \text{EXP}\rangle + \langle \text{TERM}\rangle$

$\langle \text{TERM}\rangle \to \langle \text{FACTOR}\rangle \mid \langle \text{TERM}\rangle * \langle \text{FACTOR}\rangle$

$\langle \text{FACTOR}\rangle \to (\langle \text{EXP}\rangle) \mid a \mid b \mid \ldots \mid z$

To properly display productions we will use tree. called parse trees

$Z \Rightarrow S+T$

(EXP)
(EXP)
(TERM)
(TERM)
(FACTOR)
(FACTOR)
(FACTOR)
(EXP)
(TERM)
(FACTOR)
(EXP)
(TERM)
(FACTOR)
(EXP)
(TERM