

CONTEXT-FREE GRAMMARS:

A more powerful way to specify languages.

- A CFG ~~has~~ consists of a (1) A set of symbols called terminals $(T \subseteq \Sigma)$
- (2) A set of symbols called non-terminals or variables V
- (3) A set of rules for generating sequences: productions.
- (4) A special variable called the start symbol: S .

Example $T \cup \Sigma = \{a, b\}$ $V = \{S\}$
 $S \xrightarrow{1} \epsilon$ $S \xrightarrow{2} a S b$

How does this produce a string in Σ^* ?

$S \xrightarrow{2} a S b \xrightarrow{2} a a S b b \xrightarrow{2} a a a S b b b \xrightarrow{1} a a a b b b$.

When you produce a string without variables you stop. The sequence is called a derivation.

This grammar produces the language $\{a^n b^n / n \geq 0\}$ which is not regular.

Used in linguistics to model sentence generation:

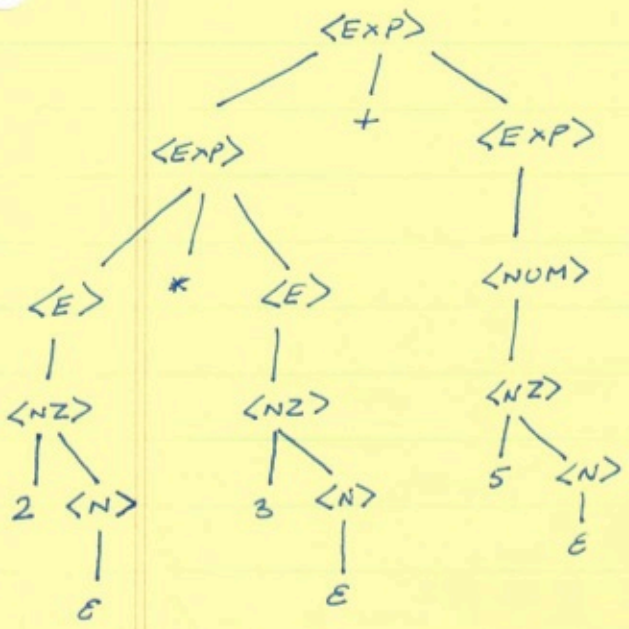
- $\langle \text{SENTENCE} \rangle \rightarrow \langle \text{N.P.} \rangle \langle \text{V.P.} \rangle$
- $\langle \text{N.P.} \rangle \rightarrow \langle \text{C.N.} \rangle | \langle \text{C.N.} \rangle \langle \text{PREP-PHRASE} \rangle$
- $\langle \text{V.P.} \rangle \rightarrow \langle \text{C.V.} \rangle | \langle \text{C.V.} \rangle \langle \text{PREP-P.} \rangle$
- $\langle \text{P. PHRASE} \rangle \rightarrow \langle \text{PREP} \rangle \langle \text{C.N.} \rangle$

ARITHMETIC EXPRESSIONS: $\Sigma = \{0, 1, 2, \dots, 9, +, x, (,)\}$.

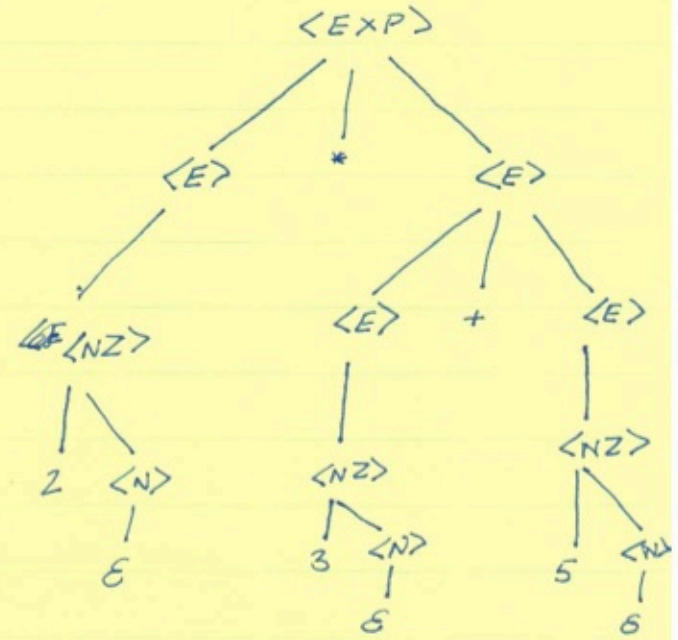
$V = \{ \langle \text{EXP} \rangle, \langle \text{NUM} \rangle, \langle \text{NZ} \rangle, \langle \text{N} \rangle \}$

- $\langle \text{EXP} \rangle \rightarrow \langle \text{EXP} \rangle + \langle \text{EXP} \rangle | \langle \text{EXP} \rangle * \langle \text{EXP} \rangle | (\langle \text{EXP} \rangle) | \langle \text{NUM} \rangle$
- $\langle \text{NUM} \rangle \rightarrow 0 | \langle \text{NZ} \rangle$
- $\langle \text{NZ} \rangle \rightarrow 1 \langle \text{N} \rangle | 2 \langle \text{N} \rangle | \dots | 9 \langle \text{N} \rangle$
- $\langle \text{N} \rangle \rightarrow 0 \langle \text{N} \rangle | 1 \langle \text{N} \rangle | \dots | 9 \langle \text{N} \rangle | \epsilon$.

To display derivations it is far better to use a tree: parse tree of a string. CFG's capture tree structure



2 * 3 + 5



2 * 3 + 5

AMBIGUOUS!

Def A grammar is said to be ambiguous if there are two distinct parse trees for the same string.

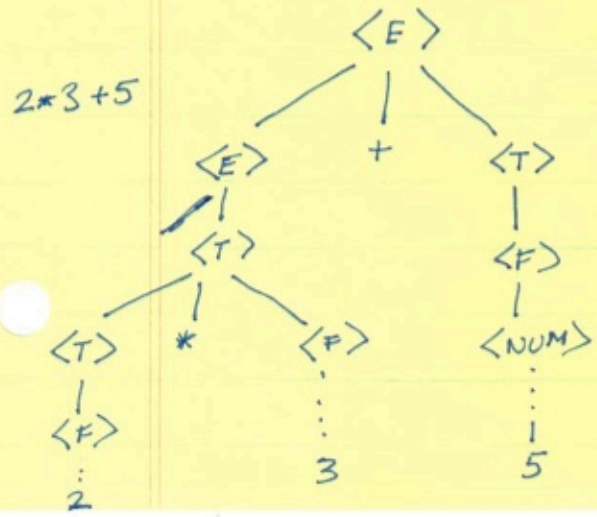
NOTE This is a property of the grammar not of the language.

$$V = \{ \langle E \rangle, \langle T \rangle, \langle F \rangle \}$$

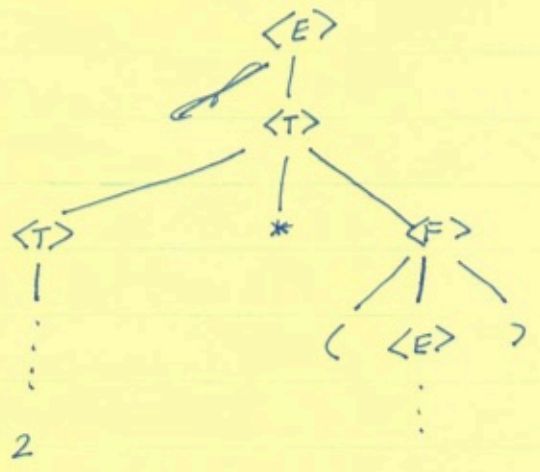
$$\langle E \rangle \rightarrow \langle E \rangle + \langle T \rangle \mid \langle T \rangle$$

$$\langle T \rangle \rightarrow \langle T \rangle * \langle F \rangle \mid \langle F \rangle$$

$$\langle F \rangle \rightarrow (\langle E \rangle) \mid \langle \text{NUM} \rangle$$



2 * 3 + 5



2 * (3 + 5) PARENS ARE FORCED