Context-free languages

Tuesday, February 16, 2021 11:38 AM

$L = \{a^n b^n \mid n \geq 0\}$

A more powerful way to specify languages.

NOAM CHOMSKY

Regular grammars $\longleftrightarrow$ Reg. langs

$\Rightarrow$ <u>Context free grammars</u> (CFG) $\longleftrightarrow$ Context-free languages (CFL)

$\rightarrow$ Push down automata

× Context sensitive $\rightarrow$ Linear bounded automata

Phrase structure $\rightarrow$ Turing machine

<u>ALL</u> programming languages are CONTEXT FREE.

A CONTEXT-FREE GRAMMAR consists of (i) a set of symbols called the <u>terminal</u> <u>symbols</u> or <u>alphabet</u>

Σ or T : <u>finite</u>.

(ii) The <u>non-terminal</u> or <u>variable</u> symbols

V: <u>finite</u> set

$$V \cap T = \phi$$

(iii) a special symbol, usually $S \in V$ called the start symbol.

(iv) a finite set of <u>rules</u> or <u>productions</u> of the form

$$A \longrightarrow \alpha \in (V \cup T)^*$$

$\uparrow$ one variable

<u>EXAMPLE</u> T or $\Sigma = \{a, b\}$

$$V = \{S\}$$

$$S \xrightarrow{1} \varepsilon$$

$$S \xrightarrow{2} a\,S\,b$$

We produce strings by starting with S and using rules <u>as</u> <u>we</u> <u>please</u> until all symbols from V are gone.

$$S \longrightarrow a\,\underline{S}\,b \longrightarrow a\;a\,S\,b\;b \longrightarrow \underline{aa}a\,S\,b\,\underline{bb}$$ ← using ②

$\rightarrow$ aaabbb

$\longrightarrow$ aaabbb

aaabbb is in the language defined by this grammar.

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

ARITHMETIC EXPRESSIONS

$\Sigma = \{0,1,2,3,4,5,6,7,8,9,+,\times,(,)\}$

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

$\langle EXP \rangle \longrightarrow \langle EXP \rangle + \langle EXP \rangle \mid \langle EXP \rangle \times \langle EXP \rangle \mid (\langle EXP \rangle)$

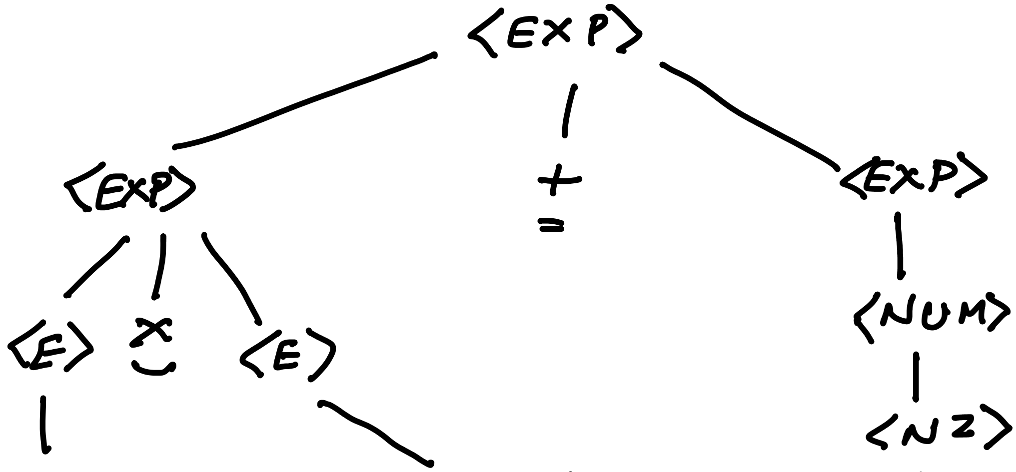
$\langle EXP \rangle \longrightarrow \langle NUM \rangle$

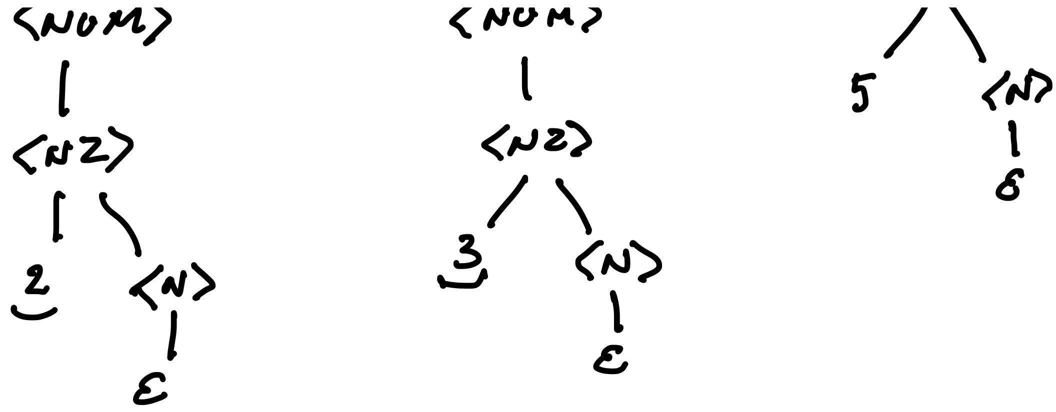
$\langle NUM \rangle \longrightarrow 0 \mid \langle NZ \rangle$

$\langle NZ \rangle \longrightarrow 1\langle N \rangle \mid 2\langle N \rangle \mid \cdots \mid 9\langle N \rangle$

$\langle N \rangle \longrightarrow 0\langle N \rangle \mid 1\langle N \rangle \mid \cdots \mid 9\langle N \rangle \mid \varepsilon$

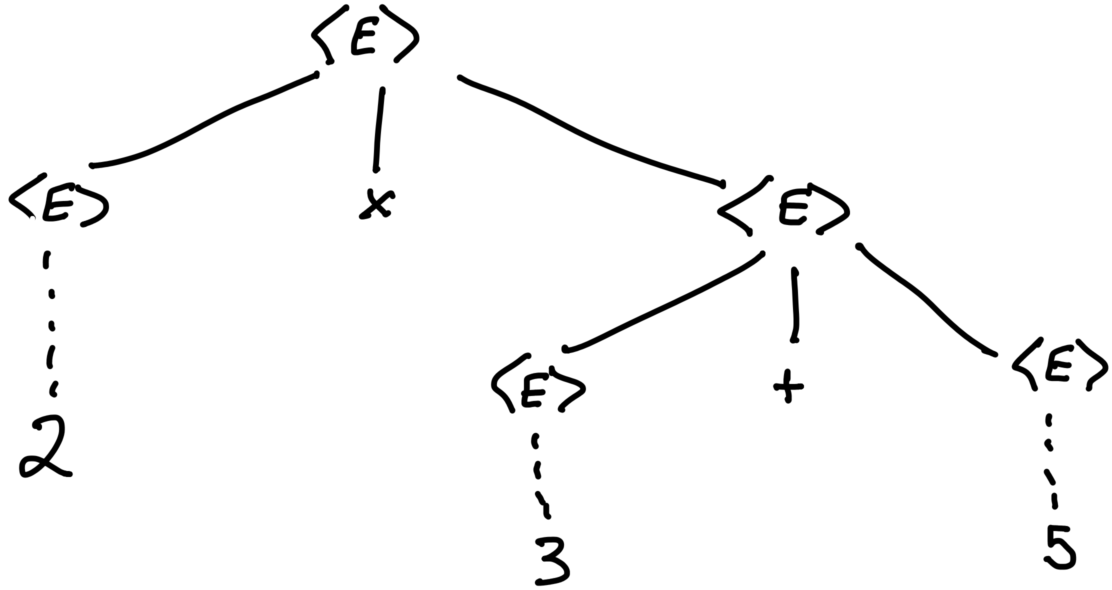
To display derivations we use a <u>tree</u> rather than a string





string generated is $2 \times 3 + 5$

The 2,3 are grouped more tightly



$2 \times 3 + 5$ same string but the tree is different and the grouping is <u>wrong</u>.

<u>PROBLEM:</u> The same string is (parsed) generated in 2 ways with different grouping.

SUCH A grammar is said to be

<u>AMBIGUOUS</u>.

This is a badly designed grammar.

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

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

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

$$\langle T \rangle \longrightarrow \langle T \rangle \times \langle F \rangle \mid \langle F \rangle$$

$$\langle F \rangle \longrightarrow (\langle E$$