CSCE 355
3/5/2025

Parse trees $\longleftrightarrow$ left-most derivations

one-to-one

regexes $\longrightarrow$ CFGs (every reg. lang. is context-free)

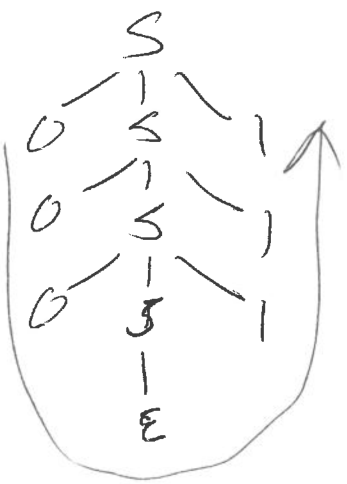
Some more examples of simple grammars:

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

CFG for L: $S \to 0S1$
$S \to \varepsilon$

shorthand: $S \to 0S1 \mid \varepsilon$

Parse tree ~~for~~ yielding 000111:



Derivation:

$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111$

$L := \{w \in \{\text{'('}, \text{')'}\}^* : w \text{ is balanced}\}$

$S \to (S)$
$S \to \varepsilon$

Gives you

ε, (), (()), ((())), ...

but not, say (())()

one more production:

$S \to SS$

ambiguous: 2 leftmost derivations of ()()()

$S \Rightarrow SS \Rightarrow SSS \Rightarrow (S)SS \Rightarrow ()SS \Rightarrow \cdots \Rightarrow ()()()$

$\underline{S} \Rightarrow \underline{S}S \Rightarrow (\underline{S})S \Rightarrow ()\underline{S} \Rightarrow ()SS$

Unambiguous: $S \to (\underline{S})\underline{S} \mid \varepsilon$

$\underline{(}\ \underbrace{\text{well-balanced}}_{S}\ \underline{)}\ \underbrace{\text{well-balanced}}_{S}$

$L := \{a^m b^n c^n : m, n \geq 0\}$ is a CFL:

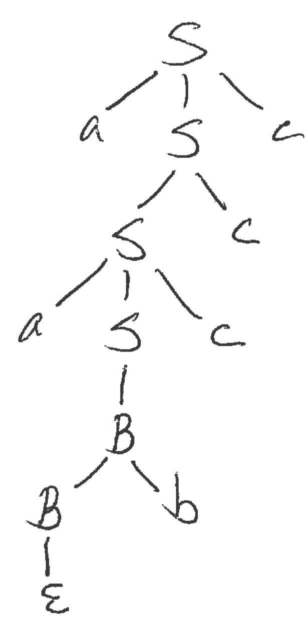
$S \to aS \mid A$

$A \to bAc \mid \varepsilon$

$L := \{a^r b^s c^t : r, s, t \geq 0 \text{ and } r \leq t\}$ is a CFL

ambiguous $\left\{\begin{array}{l} S \to aSc \mid Sc \mid B \\ B \to Bb \mid \varepsilon \end{array}\right.$

Parse tree for aabcccc:



---

unambiguous equiv grammar:

$S \to aSc \mid S'$

$S' \to S'c \mid B$

$B \to Bb \mid \varepsilon$

$S \to aSc \mid S' \mid B$

$S' \to S'c \mid B$

$B \to Bb \mid \varepsilon$

superfluous

---

what about $S \to aSc \mid Sc \mid Sb \mid \varepsilon$?

gets everything in L, but also strings not in L:

$$S \Rightarrow Sb \Rightarrow aScb \Rightarrow acb \notin L$$

There is a 1-1 correspondence between parse trees and leftmost derivations.

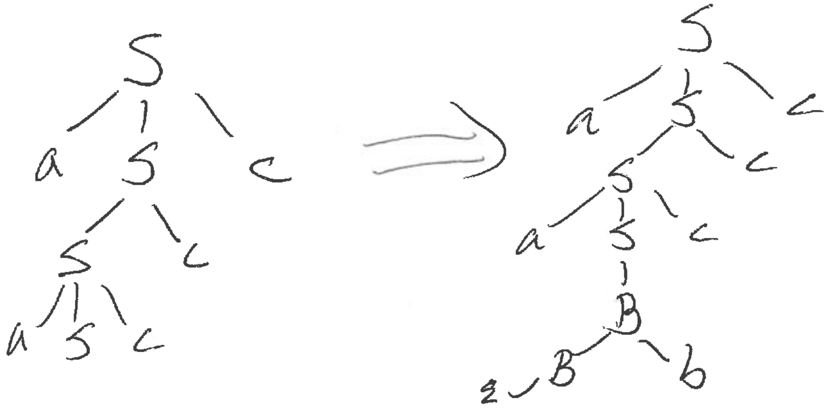
Ex: $S \to aSc \mid Sc \mid B$

$B \to Bb \mid \varepsilon$

Derivation for aabccc:

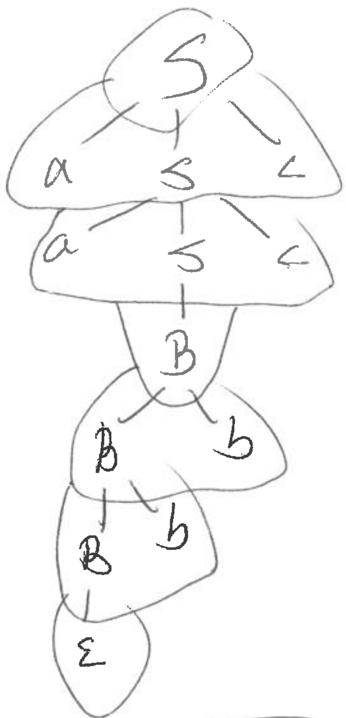
$$\underline{S} \Rightarrow a\underline{S}c \Rightarrow a\underline{S}cc \Rightarrow aa\underline{S}ccc \Rightarrow aa\underline{B}ccc$$

$$\Rightarrow aa\underline{B}bccc \Rightarrow aabccc$$



corresponding parse tree

Conversely, given a parse tree, convert it into a left-most derivation:



$\Longrightarrow$

leftmost deriv:

$S \Rightarrow aSc \Rightarrow aaScc$

$\Rightarrow aaBcc \Rightarrow aaBbcc$

$\Rightarrow aaBbbcc$

$\Rightarrow aabbcc$

Ex: $S \to (S)S \mid \varepsilon$

Parse tree yielding (())():

leftmost derivation

$S \Rightarrow (S)S \Rightarrow ((S)S)S$

$\Rightarrow (()S)S \Rightarrow (())S$

$\Rightarrow (())(S)S$

$\Rightarrow (())()S$

$\Rightarrow (())()$

regexes $\longrightarrow$ CFGs [Theorem: Every regular language is context-free]

Construct a CFG from a regex by recursion on regex syntax. Fix an alphabet Σ, let r be a regex over Σ.

r	
$\emptyset$	$\langle V, \Sigma, S, P \rangle$ where $V = \{S\}$ and $P = \{\}$ or $P = \{S \to S\}$
$(a \in \Sigma)$ a	$S \to a$
$s + t$ (s, t regexes over Σ)	Given grammars for s, t where G_s grammar for s start symbol A G_t grammar for t start symbol B $S \to A \mid B$ G_s — G_t — Must rename nonterminals so that G_s, G_t have no nonterminals in common.
st	Same as above for $s+t$, except instead of $S \to A \mid B$ have $S \to AB$
s^*	Given G_s with start symbol A, add new start symbol S and these two productions: $S \to SA \mid \varepsilon$

Example: $(a + bc)^*$

A derives $(a+bc)$

$S \to SA \mid \varepsilon$

$A \to F \mid G$

$F \to a$

$G \to BC$

$B \to b$

$C \to c$

or

$G \to bc$

$a^* + b^*$

$S \to A \mid B$

$A \to Aa \mid \varepsilon$

$B \to Bb \mid \varepsilon$