Context Free Grammars

We think of context free grammars as generating strings.

1. Start from the start symbol S.
2. Choose a nonterminal in the string, and a production rule $A \rightarrow w_1|w_2|\ldots|w_k$ replace that copy of the nonterminal with w_i.
3. If no nonterminals remain, you're done! Otherwise, goto step 2.

A string is in the language of the CFG iff it can be generated starting from S.

Examples

$S \rightarrow 0S0|1S1|0|1|\varepsilon$

$S \rightarrow 0S|S1|\varepsilon$

$S \rightarrow (S)|SS|\varepsilon$

The alphabet here is {(,)} i.e. parentheses are the characters.

$S \rightarrow AB$

$A \rightarrow 0A1|\varepsilon$

$B \rightarrow 1B0|\varepsilon$

Arithmetic

$E \rightarrow E + E|E * E|(E)|x|y|z|0|1|2|3|4|5|6|7|8|9$

Generate $(2 * x) + y$

Generate $2 + 3 * 4$ in two different ways

Parse Trees—remember where parentheses go

Suppose a context free grammar G generates a string x

A parse tree of x for G has

Rooted at S (start symbol)

Children of every A node are labeled with the characters of w for some $A \rightarrow w$

Reading the leaves from left to right gives x.

$S \rightarrow 0S0|1S1|0|1|\varepsilon$