Context-Free Grammars

Sunday, November 1, 2020 4:57 PM

Motivation.

How does computer know sequence of characters as an expression and determine syntactic validity?

Arithmetic Expressions.

Expr → int
Expr → Expr Op Expr
Expr → (Expr)
Op → +
Op → −
Op → ×
Op → /

Expr
Expr Op Expr
Expr Op int
int Op int
int / int

gray - terminals, do not get replaced

Context-Free Grammars

Defn. Collection of 4 items:

→ nonterminal symbols (variables)
→ terminal symbols (alphabet)
→ Production rules (how to replace nonterminals w/ terminals)
→ Start symbol (terminal) that begins derivation
Conventionally: LHS of first production

Notation. Uppercase - NT's

lowercase - T's

greek letters - arbitrary strings

Vertical bar (|) - separator

Derivation. Sequence of 0 or more steps where NT's are replaced by RHS of production

i.e. α derives w

$$\alpha \Rightarrow^* w$$

Language If G is CFG w/ alphabet Σ, start symbol S, then language of G is

$$\mathcal{L}(G) = \{ w \in \Sigma^* \mid S \Rightarrow^* w \}$$

A context-free language L is defined if there is a CFG G where $\mathcal{L}(G) = L$

CFG's and regexes.

Theorem. Every regular language is context-free.

(i.e., can generate CFG from regex)

BUT not all CFG's are regular.

ex. $S \rightarrow aSb \mid \varepsilon$

$$\{ a^n b^n \mid n \in \mathbb{N} \}$$

Designing CFG's

Ex. Let $\Sigma = \{a, b\}$

Let $L = \{ w \in \Sigma^* \mid w \text{ is a palindrome} \}$

→ Base case: a, b are palindromes

→ Recursive case: if w is a palindrome, then awa and bwb are palindrome

→ No other strings are palindromes

$$S \to \varepsilon \mid a \mid b \mid aSa \mid bSb$$

Note: Need ε in a rule for finite strings

Each NT expands independently of the others