Context-Free Grammars

Recall: Regular languages had regexes as a representation

We saw that for some non-regular languages, we could write grammars to represent them.

A grammar is a 4-tuple of the form $(\underbrace{NT}_{\text{set of non-terminals}}, \underbrace{T}_{\text{set of terminals}}, \underbrace{R}_{\text{set of production rules}}, \underbrace{S}_{\in NT \text{ start symbol}})$

Each production rule in a context-free grammar has a non-terminal on the left, followed by ::= or $\longrightarrow$, and some sequence of terminals & non-terminals on the right

We gave a grammar for

$$\mathcal{L} = \{ \omega \mid \omega \text{ contains as many `a's as `b's} \} \subseteq \{a, b\}^*$$

as follows

$$S ::= \varepsilon \mid aSb \mid bSa \mid SS$$

$$G = (NT, T, R, S)$$

where $NT = \{S\}$

$T = \{a, b\}$

$R = \{ S ::= \varepsilon,\ S ::= aSb,\ S ::= bSa,\ S ::= SS \}$

The main use of CFGs is in parsing.

For example, I might want to recognize the language of all strings with balanced parentheses

But the same CFG from above does not work for this!

$$\mathcal{L}_{()} = \left\{ \omega \;\middle|\; \begin{array}{l} \omega \text{ contains an equal number of `(' and `)', and} \\ \text{every prefix } s \text{ of } \omega \text{ contains at least} \\ \text{as many `('s as `)'s} \end{array} \right\}$$

This is a very verbose description.

What is a CFG for $\mathcal{L}$?

$$S ::= \varepsilon \mid (S) \mid SS$$

$$G = \left(\{S\}, \{(,)\}, \{S ::= \varepsilon, S ::= (S), S ::= SS\}, S\right)$$

How do we prove that $\mathcal{L}(G) = \mathcal{L}_{()}$?

① Show that every string $w \in \mathcal{L}(G)$ is s.t. $w \in \mathcal{L}_{()}$.

② Show that every string $w \in \mathcal{L}_{()}$ is s.t. $w \in \mathcal{L}(G)$.

Possible cases for w:

$w = \varepsilon$: $\quad S ::= \varepsilon$ ✓

$\omega = (\omega'($: ✗

$\omega =)\omega'($: ✗

$\omega =)\omega')$: ✗

$\omega = (\omega')$: Let ω_p be the leftmost prefix of (ω') s.t.

$$\#\text{'('}(\omega_p) = \#\text{')'}(\omega_p)$$

Let $\omega = \omega_p \cdot \omega_s$

(a) $\omega_p = \omega$: $S ::= (S)$ $\quad$ $S ::= \omega'$

(b) $\omega_p \neq \omega$: $S ::= \omega_p$ $\quad$ $S ::= \omega_s$ $\quad$ $S ::= SS$