CONTEXT-FREE
=
PDA-RECOGNIZABLE
PART I

Recall: We drew PDAs for some context-free languages, and proved that they accepted exactly those specific sets

Today: PDAs are the machine model for context-free grammars.

For regular languages, we presented

- DFAs
- NFAs
- Regular expressions

and showed that the languages accepted by finite automata are exactly those that can be expressed during regexes.

We then showed that some languages are not regular.
Some of these can be generated using a context-free grammar.
Some of those are recognizable using a pushdown automaton.

Claim: The set of languages recognizable by pushdown automata is exactly the set of languages generatable by context-free grammars.

How do we show this?

① Given a CFG generating a language $\mathcal{L}$, construct a PDA which recognizes $\mathcal{L}$.

② Given a PDA recognizing $\mathcal{L}$, construct a CFG which generates $\mathcal{L}$.

Suppose we start with ①. Suppose we are given $G = (NT, T, R, S)$.

Do we have a general idea of the "shape" of the rules in R?

Without this, hard to "uniformly" translate G into a PDA!

Greibach Normal Form:

A CFG $G = (NT, T, R, S)$ is in Greibach Normal Form if all production rules in R are of the form

$A \to aB_1 \cdots B_k$, where $k \geq 0$, $A, B_1, \ldots, B_k \in NT$, and $a \in T$.

No rule for generating the empty string ε!

Thm: For any CFG G, there is a CFG G' in Greibach Normal Form s.t. $\mathcal{L}(G') = \mathcal{L}(G) \setminus \{\varepsilon\}$.

So, every CFG can be converted to a form where each rule is of the form $A \to aB_1 \ldots B_k$, where $k \geq 0$, $A, B_1, \ldots, B_k \in NT$, and $a \in T \cup \{\varepsilon\}$.

① Suppose we are given such a CFG $G = (NT, T, R, S)$.

We construct a PDA M which accepts $\mathcal{L}(G)$ by empty stack.

$M = (Q, \Sigma, \Gamma, \Delta, q_0, \emptyset)$, where

$Q = \{q_0, q_1\}$, $\Sigma = T$, $\Gamma = NT \cup \{\perp\}$ and

$(q_0, \omega, \perp) \xrightarrow[M]{*} (q, \varepsilon, \perp)$ for some $q \in Q$.

$$\Delta = \{((q_0, \varepsilon, \varepsilon), (q_1, S)),$$

$$\{((q_1, a, A), (q_1, B_1 \cdots B_k)) \mid A \to aB_1 \cdots B_k \in R\}$$

Thm: If a terminal string x is obtained by applying rules in R to the leftmost nonterminal symbol, then M accepts x.

Example: $L = \{ w \mid w$ has an equal number of 'a's and 'b's$\}$

$G = (\{S, A, B\}, \{a, b\}, R, S)$ generates L, where

$R = \{ S \to \varepsilon, S \to aBS, S \to bAS, B \to b, B \to aBB,$
$A \to a, A \to bAA \}$

$M = (\{q_0, q_1\}, \{a, b\}, \{A, B, S, \perp\}, \Delta, q_0, \phi)$

$\Delta = \{ ((q_0, \varepsilon, \varepsilon), (q_1, S)),$
$((q_1, \varepsilon, S), (q_1, \varepsilon)), ((q_1, a, S), (q_1, BS)),$
$((q_1, b, S), (q_1, AS)), ((q_1, b, B), (q_1, \varepsilon)),$
$((q_1, a, B), (q_1, BB)), ((q_1, a, A), (q_1, \varepsilon)), ((q_1, b, A), (q_1, AA)) \}$

$w = abaabbba$

Rules in G

$S \rightarrow aBS$	$\cancel{a}baabbba$
$B \rightarrow b$	$\cancel{b}aabbba$
$S \rightarrow aB\underline{S}$	$\cancel{a}abbba$
$B \rightarrow aBB$	$\cancel{a}bbba$
$B \rightarrow b$	$\cancel{b}bba$
$B \rightarrow b$	$\cancel{b}ba$
$S \rightarrow bAS$	$\cancel{b}a$
$A \rightarrow a$	$\cancel{a}$
$S \rightarrow \varepsilon$	

PDA configurations

$(q_0, abaabbba, \perp)$

$\downarrow$

$(q_0, abaabbba, S)$

$\downarrow$

$(q_0, baabbba, BS)$

$\downarrow$

$(q_0, aabbba, S)$

$\downarrow$

$(q_0, abbba, BS)$

$\vdots$