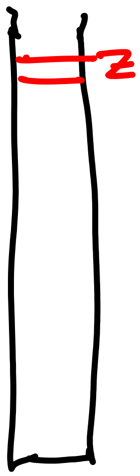
Lecture 21 COL 352 Mar 9

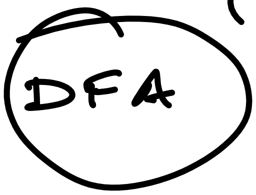
A stack based machine (Push Down Automaton)

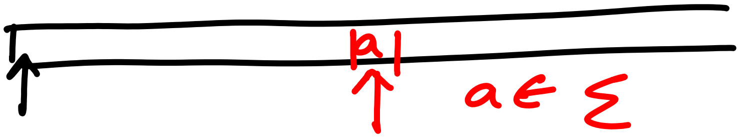
In addition to a finite state transition system, we have an (infinite capacity) stack.

$$(Q, \Sigma, \delta, \overset{\phi}{\boxed{F}}, q_0, \overset{\text{stack alphabet}}{\Gamma}, Z_0)$$

bottom stack marker







$a \in \Sigma$

$\delta(q, a, z) = (p, \gamma)$

$z \in \Gamma$

$p, q \in Q$

$\gamma \in \Gamma^*$

ϵ is used for popping stack

Two separate terminating conditions

① The stack is empty when input string is exhausted

② We are in a final state when input is exhausted

$L = \{ w \textcircled{c} w^R \mid w \in (0+1)^* \}$

$M = (\{q_1, q_2\}, \{0, 1, c\} \{R, B, G\}, \delta, q_1, R, \phi)$

1. $\delta(q_1, 0, R) = \{(q_1, BR), \_\}$ $\delta(q_1, 1, R) = (q_1, GR)$
2. $\delta(q_1, 0, B) = (q_1, BB)$ $\delta(q_1, 1, B) = (q_1, GB)$
3. $\delta(q_1, 0, G) = (q_1, BG)$ $\delta(q_1, 1, G) = (q_1, GG)$
4. $\delta(q_2, 0, B) = (q_2, \epsilon)$ $\delta(q_2, 1, G) = (q_2, \epsilon)$
5. $\delta(q_2, \epsilon, R) = (q_2, \epsilon)$
6 $\delta(q_1, c, R) = (q_2, R)$ $\delta(q_1, c, G) = (q_2, G)$
7 $\delta(q_1, c, B) = (q_2, B)$

Instantaneous Description (ID) of a PDA

The complete information about a PDA can be obtained from

① Current state

(2) the current symbol it is scanning

(3) the stack contents

ID: $(q, a \cdot \omega, \alpha)$

$q \in Q$

$a \in \Sigma, w \in \Sigma^*$

$\alpha \in \Gamma^*$

input string

$I_0 : (q_0, \delta, Z_0)$

$I_0 \vdash I_1 \vdash I_2 \vdash \cdots \quad I_f$

$I_0 \overset{*}{\vdash} I_f$

$I_j \vdash_M I_{j+1}$

$(p, a\omega, A\alpha) \vdash (q, \omega, \beta\alpha)$

$\delta(p, a, A)$ must contain (q, β)

PDAs that accept by empty stack

w is accepted by the PDA iff

$(q_0, w, Z_0) \vdash^* (p, \epsilon, \epsilon)$ $p \in Q$

PDAs that accept by final state

$(q_0, w, Z_0) \vdash^* (q_f, \epsilon, \alpha)$

$q_f \in F$ $\alpha \in \Gamma^*$

Thm Let L be a language accepted by a PDA using empty stack. Then L is also accepted by some PDA that accepts using final state.

and vice versa

Thm Suppose L is a CFL generated by a CFG $G = (V, T, S, P)$. Then we can design a PDA M (accepts using empty stack) s.t.

$$L(M) = L$$

The proof uses Greibach Normal form

Thm Given a PDA M that accepts a language L. Then we can design a CFG G s.t.

$$L(G) = L$$

The deterministic version of PDA is not equivalent to PDA (non-det)

↗ $L = \{ ww \mid w \in (0+1)^* \}$

011011

$\overline{L}$: all strings not of the form ww

↖ 001011