More About Pushdown Automata

Recall: We proposed Pushdown Automata as a candidate machine model for recognizing context-free languages

A pushdown automaton (PDA) is essentially an NFA with access to a global stack. It can

→ read input letters

→ push and pop onto and from the stack

Operation pop x "gets stuck" if the symbol on the top of the stack is some y, where $y \neq x$.

Might need to check for emptiness of stack; do this by checking for a special end marker $\perp$.

Pushdown automata (PDA): A 6-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$

Q: set of states $\quad$ Σ: input alphabet $\quad$ Γ: stack alphabet, $\perp \in \Gamma$.

$\Delta \subseteq (Q \times (\Sigma \cup \{\varepsilon\}) \times (\Gamma \cup \{\varepsilon\})) \times (Q \times \Gamma^*)$: transition relation

$q_0 \in Q$: start state $\qquad$ $F \subseteq Q$: set of accepting states

$$((q, a, C), (q', D_1 D_2 \dots D_k)) \in \Delta$$

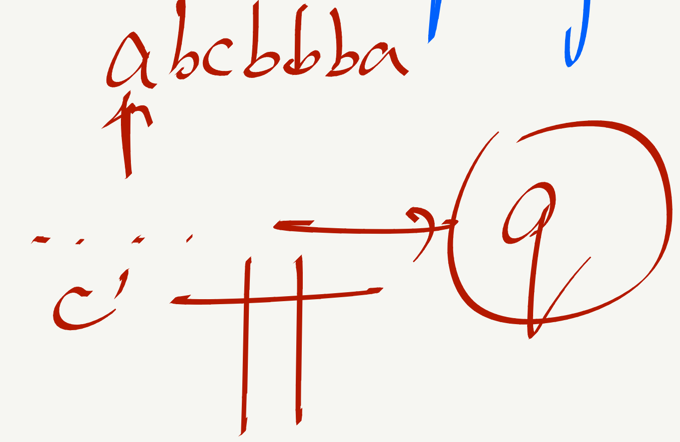
new state of M

new sequence of symbols pushed onto stack

current symbol on top of stack

current state of M

current input letter



What if this is the only transition in Δ out of q, and M gets to q but the stack contains C' as the top symbol?

not C

What all information does one need in order to fully specify the behaviour of a PDA?

→ Current state

→ Current stack contents

→ Whatever string the PDA is going to read

A Configuration of a PDA $M = (Q, \Sigma, \Gamma, \Delta, q_0, F)$ is

$c \in Q \times \Sigma^* \times \Gamma^*$, which fixes these three parameters.

What is the start configuration on an input word ω?

$(q_0, \omega, \perp)$

What can the machine M do in one step from a configuration?

Suppose $((q, a, A), (q', s)) \in \Delta$. Then, we say that

$w \in \Sigma^*$ $S \in \Gamma^*$

$$(q, aw, AS) \xrightarrow[M]{1} (q', w, sS),$$

for any $w \in \Sigma^*$, $S \in \Gamma^*$. If $a = \varepsilon$, $aw = w$.

Suppose $c, d \in Q \times \Sigma^* \times \Gamma^*$ are configurations of M. Then,

$c \xrightarrow[M]{0} d$ iff $c = d$

$c \xrightarrow[M]{n+1} d$ iff there is some c' s.t. $c \xrightarrow[M]{n} c'$ and $c' \xrightarrow[M]{1} d$.

$c \xrightarrow[M]{*} d$ iff there is some $n \geq 0$ s.t. $c \xrightarrow[M]{n} d$.

What strings does M accept?

One can specify acceptance in one of two (equivalent) ways

→ By final state: M accepts ω by final state if

$(q_0, \omega, \perp) \xrightarrow[M]{*} (f, \varepsilon, s)$ for some $s \in \Gamma^*$ and some $f \in F$.

→ By empty stack: M accepts ω by empty stack if

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

(F is irrelevant here!)

* These criteria are actually equivalent!
Either kind of machine can simulate the other.

$\mathcal{L} = \{a^n b^n \mid n \geq 0\}$ $\quad M = (\{q_0, q_1\}, \{a, b\}, \{\perp, C\}, \Delta, q_0, F)$

$$\Delta = \{((q_0, a, \varepsilon), (q_0, C)), ((q_0, b, C), (q_1, \varepsilon)), ((q_1, b, C), (q_1, \varepsilon))\}$$

What configurations does M go through on input word $a^4 b^4$?

$$(q_0, a^4 b^4, \perp) \xrightarrow[M]{1} (q_0, a^3 b^4, C\perp) \xrightarrow[M]{1} (q_0, a^2 b^4, CC\perp) \to (q_0, a^1 b^4, CCC\perp) \to (q_0, b^4, CCCC\perp) \to (q_1, b^3, CCC\perp) \to (q_1, b^2, CC\perp) \to (q_1, b, C\perp) \to (q_1, \varepsilon, \perp)$$

When does M accept? By empty stack

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