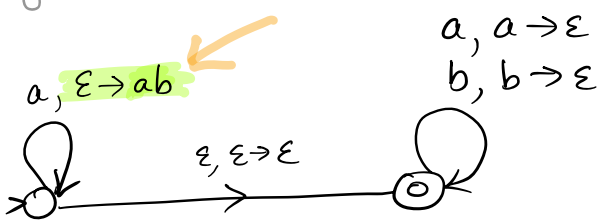
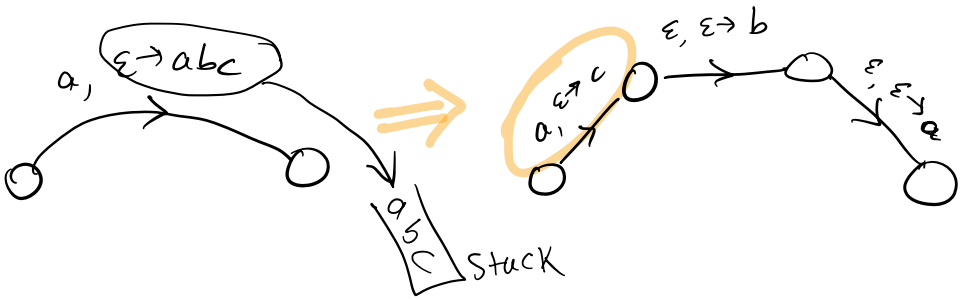
PDA's, continued.

A note about PDAs, and some warm-ups.

For nice, compact PDAs, we will allow ourselves the following notation:



What does $\varepsilon \to ab$ mean, and how do we get away with it?

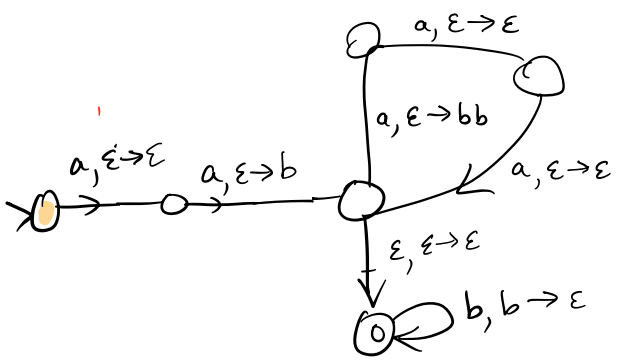


We "get away with it" because it does not change the underlying model of PDA: for any PDA with strings on the stack part of the transition, we can easily convert it to one that just uses single symbols for the stack transitions.

Note: the string slides onto the stack "back end first".

Warm up: a PDA for $\{a^n b^m \mid 2n = 3m+1\}$

Option 1: a natural PDA



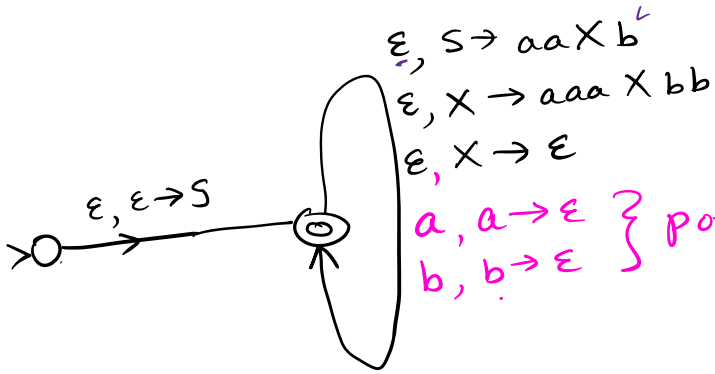
$S \to aaXb$

$X \to aaaXbb \mid \varepsilon$

Option 2: a top-down parser PDA

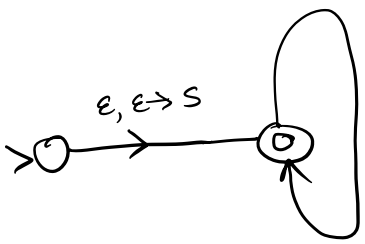
$S \to aaXb$

$X \to aaaXbb \mid \varepsilon$



Grammar for $\{ w \in \{a,b\}^* \mid \#_a(w) = \#_b(w) \}$

Top-Down Parser for $\{ w \in \{a,b\}^* \mid \#_a(w) = \#_b(w) \}$



Grammar for $\{ a^n b^m \mid m \leq n \leq 2m \}$

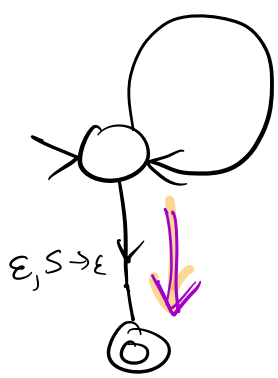
Top-Down Parser for $\{ a^n b^m \mid m \leq n \leq 2m \}$

Top down parsing answers the question, "is string w derivable in G?" by starting with S and applying rules of G ("lucky guessing" what rule to apply next) until it has a string of terminals on top of stack, which it "pops + matches".

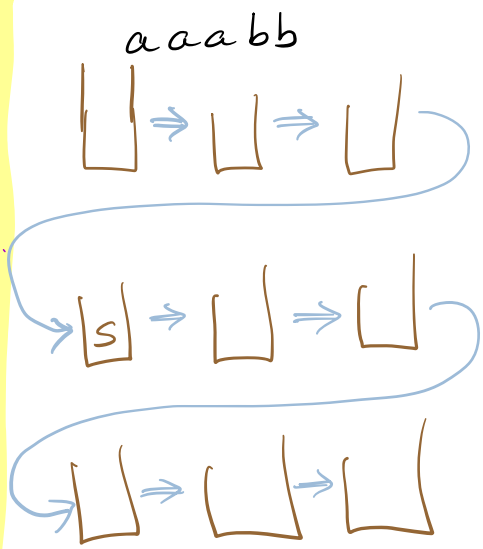
Bottom-up Parsing

- start with w
- apply rules of G "upwards" (and backwards) on w
  - see whether can get to S.
  - also called a "shift-Reduce" parser.

$S \to aSb \mid aaSb \mid \varepsilon$



$\varepsilon, bSa \to S$
$\varepsilon, bSaa \to S$
$\varepsilon, \varepsilon \to S$
$a, \varepsilon \to a$
$b, \varepsilon \to b$

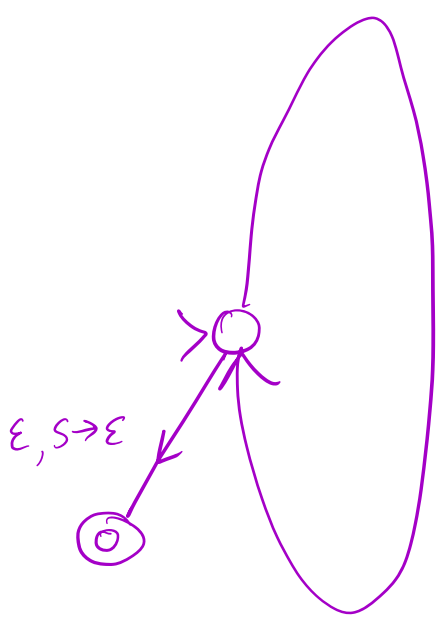


Idea: Want the backwards version of some RHS of a rule on top of stack ... then can pop and replace with the LHS variable.
(Goes up the derivation tree)

$S \to AC$

$A \to aAb \mid \varepsilon$

$C \to bCc \mid \varepsilon$



$$L = \{ a^i b^i c^j d^j \mid i, j \geq 0 \}$$

Give a CFG, Natural PDA, Top-Down Parser, Bottom-Up Parser for L.