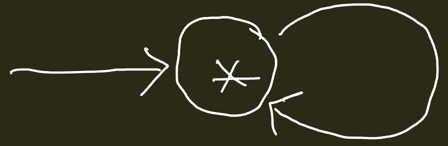
Pushdown Automata (PDA)

Example 2

$$L_2 = \{ w \in \{a, b\}^* \mid \#a(w) = \#b(w) \}$$

$$S \to \epsilon \mid aSb \mid bSa \mid SS$$

$$\delta(x) = \#a(x) - \#b(x)$$



$a, \perp / + \perp$

$a, + / + +$

$a, - / \epsilon$

$b, \perp / - \perp$

$b, - / - -$

$b, + / \epsilon$

$\epsilon, \perp / \epsilon$

$\epsilon, + / \epsilon$

$$\Gamma = \{ \perp, +, - \}$$

acceptance by empty stack

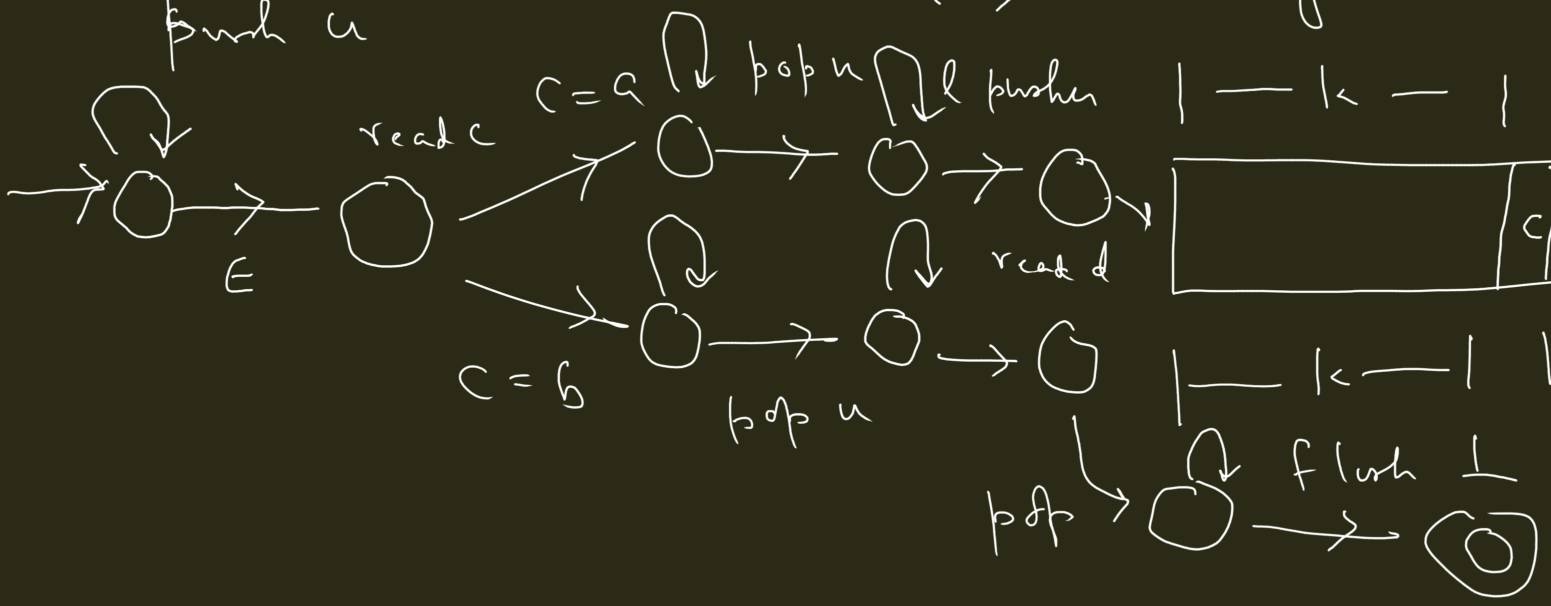
Examples 3 : $L_3 = \{ w \in \{a,b\}^* \mid \#a(w) \geq \#b(w) \}$

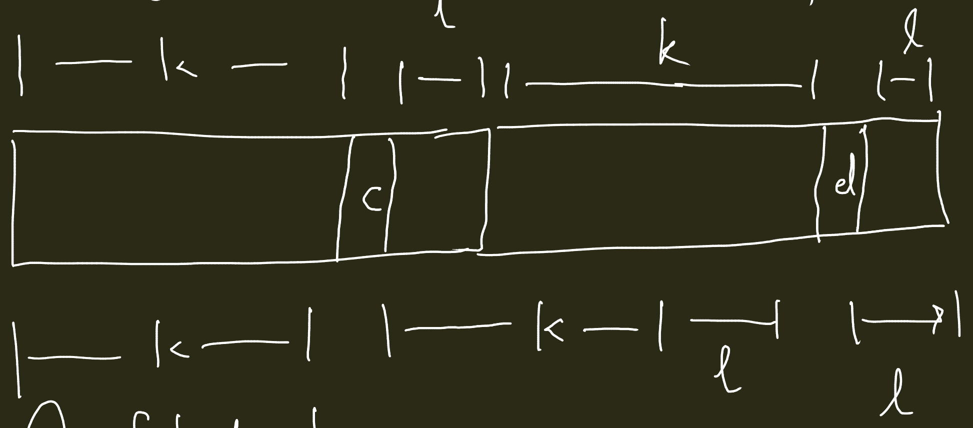
Example 4 : $L_4 = \{ ww \mid w \in \{a,b\}^* \}$ is not CF.

$\sim L_4$ is context-free

(1) strings of odd lengths

(2) strings of even lengths of this form





$c \neq d$

Configuration of a PDA M (current snapshot)

- the current state
- the part of the input yet to be read
- the current content of the stack

$$Q \times \Sigma^* \times \Gamma^*$$

Initial configuration for i/p w

$$(s, w, \perp)$$

acceptance configuration

final state (f, ϵ, γ)

$f \in F \quad \gamma \in \Gamma^*$

empty stack (q, ϵ, ϵ)

One-step change of configuration

$$(p, ax, A\beta) \xrightarrow[M]{1} (q, x, \gamma\beta)$$

$$((p, a, A), (q, \gamma)) \in \delta$$

$$a \in \Sigma \cup \{\epsilon\}$$

$$C \xrightarrow[M]{0} D \Rightarrow C = D$$

$C \xrightarrow[M]{n+1} D$ if there exists a configuration E

s.t. $C \xrightarrow[M]{n} E$ and $E \xrightarrow[M]{1} D$

$C \xrightarrow[M]{*} D$ if $C \xrightarrow[M]{n} D$ for some $n \geq 0$.

Language of a PDA M

Acceptance by final state

$$\mathcal{L}(M) = \{ w \in \Sigma^* \mid (s, w, \perp) \overset{*}{\underset{M}{\longmapsto}} (f, \epsilon, \gamma) \text{ for some } f \in F \text{ and } \gamma \in \Gamma^* \}$$

Acceptance by empty stack

$$\mathcal{L}(M) = \{ w \in \Sigma^* \mid (s, w, \perp) \overset{*}{\underset{M}{\longmapsto}} (q, \epsilon, \epsilon) \text{ for some } q \in Q \}$$

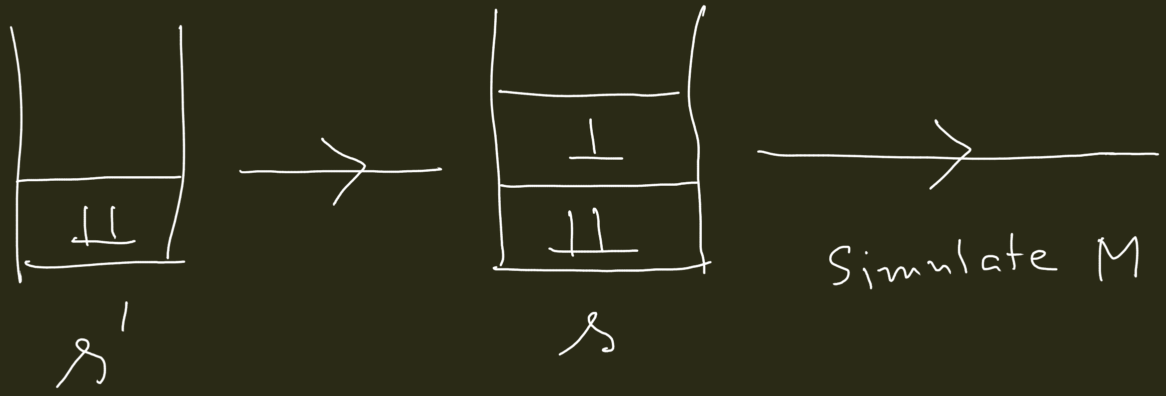
Given any $M = (Q, \Sigma, \Gamma, \perp, \delta, s, F)$, to construct

$$N = (Q', \Sigma, \Gamma', \perp\!\!\perp, \delta', s', \{t'\}) \text{ s.t.}$$

N accepts both by final state and empty stack

$$Q' = Q \cup \{s', t'\} \qquad \Gamma' = \Gamma \cup \{\perp\!\!\perp\}$$

$((s', \epsilon, \perp\!\!\perp), (s, \perp \perp\!\!\perp))$

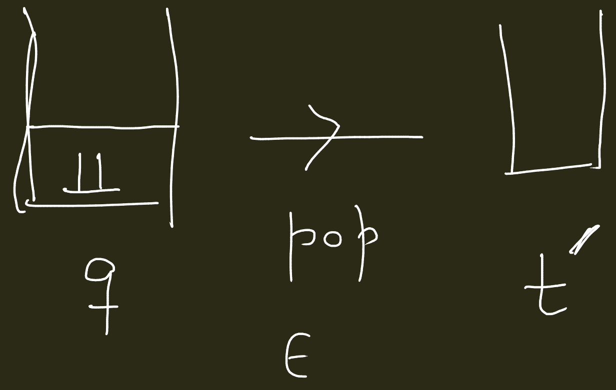


M accepts by empty stack

$w \in \mathcal{L}(M)$

at the end of the simulation

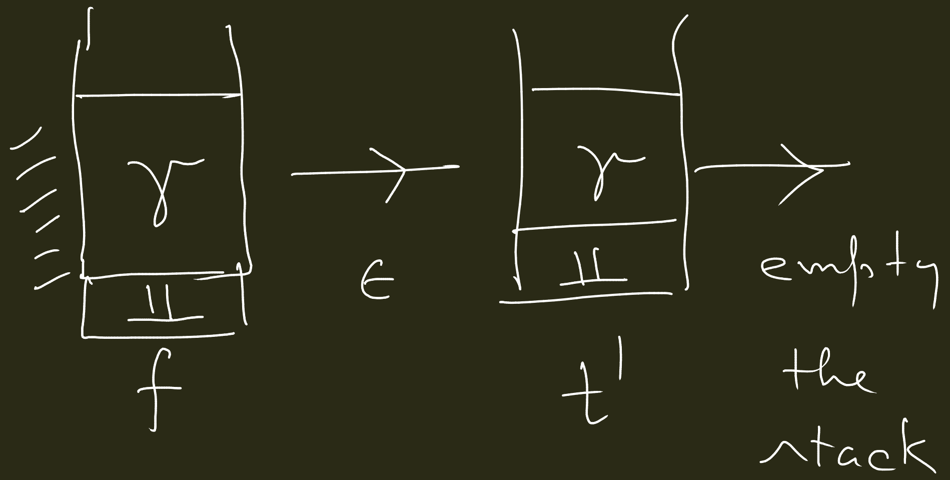
$$((q, \epsilon, \perp), (t', \epsilon))$$



M accepts by final state

$$((f, \epsilon, A), (t', A))$$

$$((t', \epsilon, A), (t', \epsilon))$$



Exercise: $\mathcal{L}(M) = \mathcal{L}(N)$

Equivalence of CFG and PDA

CFG to PDA

$$G = (N, \Sigma, P, S)$$

To design $M = (Q, \Sigma, \Gamma, \perp, \delta, s, F)$

s.t. $\mathcal{L}(G) = \mathcal{L}(M)$

$Q = \{*\}$, $s = *$, acceptance by empty stack

$F = \emptyset$

$\Gamma = N \cup \Sigma$, $\perp = S$

$\boxed{\delta}$ $A \rightarrow \gamma$ a rule in G

$((*, \epsilon, A), (*, \gamma))$

$((*, a, a), (*, \epsilon))$

M simulates leftmost derivations of G.

$L_2 = \{ w \in \{a, b\}^* \mid \#a(w) = \#b(w) \}$

$S \to \epsilon \mid aSb \mid bSa \mid SS$

<u>δ</u>

$((*, a, a), (*, \epsilon)),$

$((*, b, b), (*, \epsilon)),$

$((*, \epsilon, S), (*, \epsilon)),$

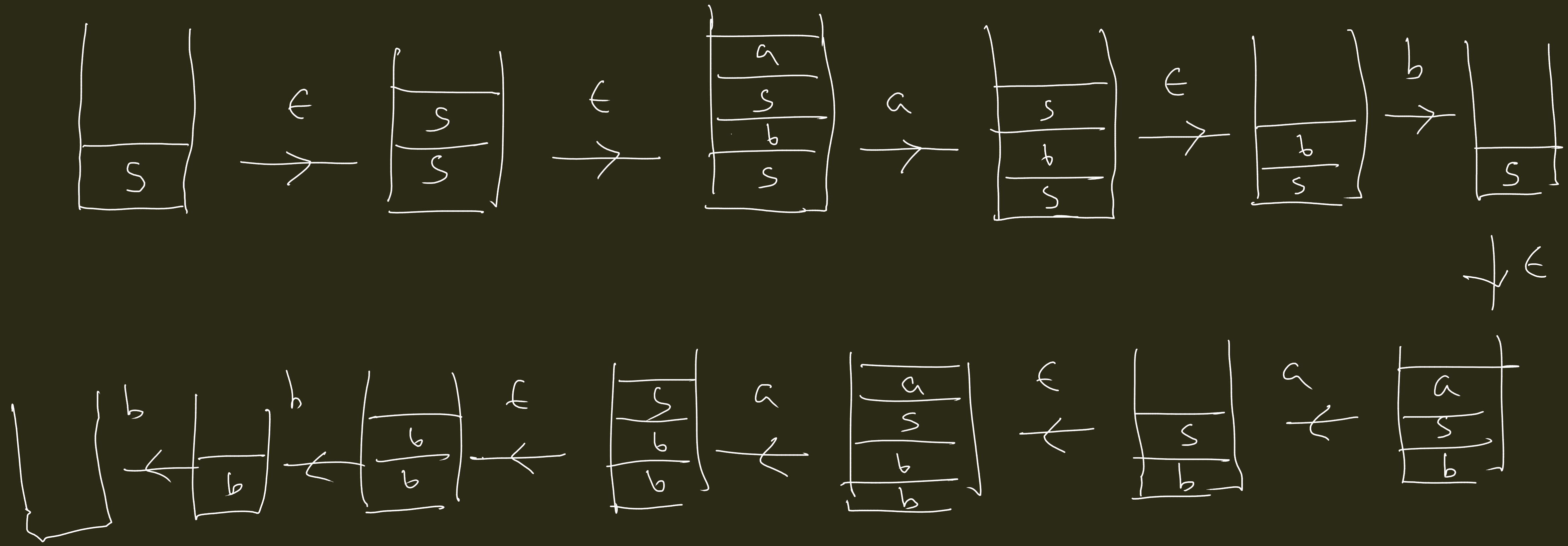
$((*, \epsilon, S), (*, aSb)),$

$((*, \epsilon, S), (*, bSa)), ((*, \epsilon, S), (*, SS))$

ab aabb

$S \rightarrow SS \rightarrow aSbS \rightarrow abS \rightarrow abaSb$

$\rightarrow abaaSbb \rightarrow abaabb$



To convert a PDA (N) to a CFG (G)

N accepts by both final state (t) and empty stack

Two-step construction

Step 1 : Convert N to an eqvt N' s.t.

(a) N' has only state

(b) N' accepts by empty stack

Step 2 : Convert N' to G

For transition $((*, a, A), (*, \gamma))$,

introduce the production $A \rightarrow a\gamma$

N' "stores" the state info of N in its stack.