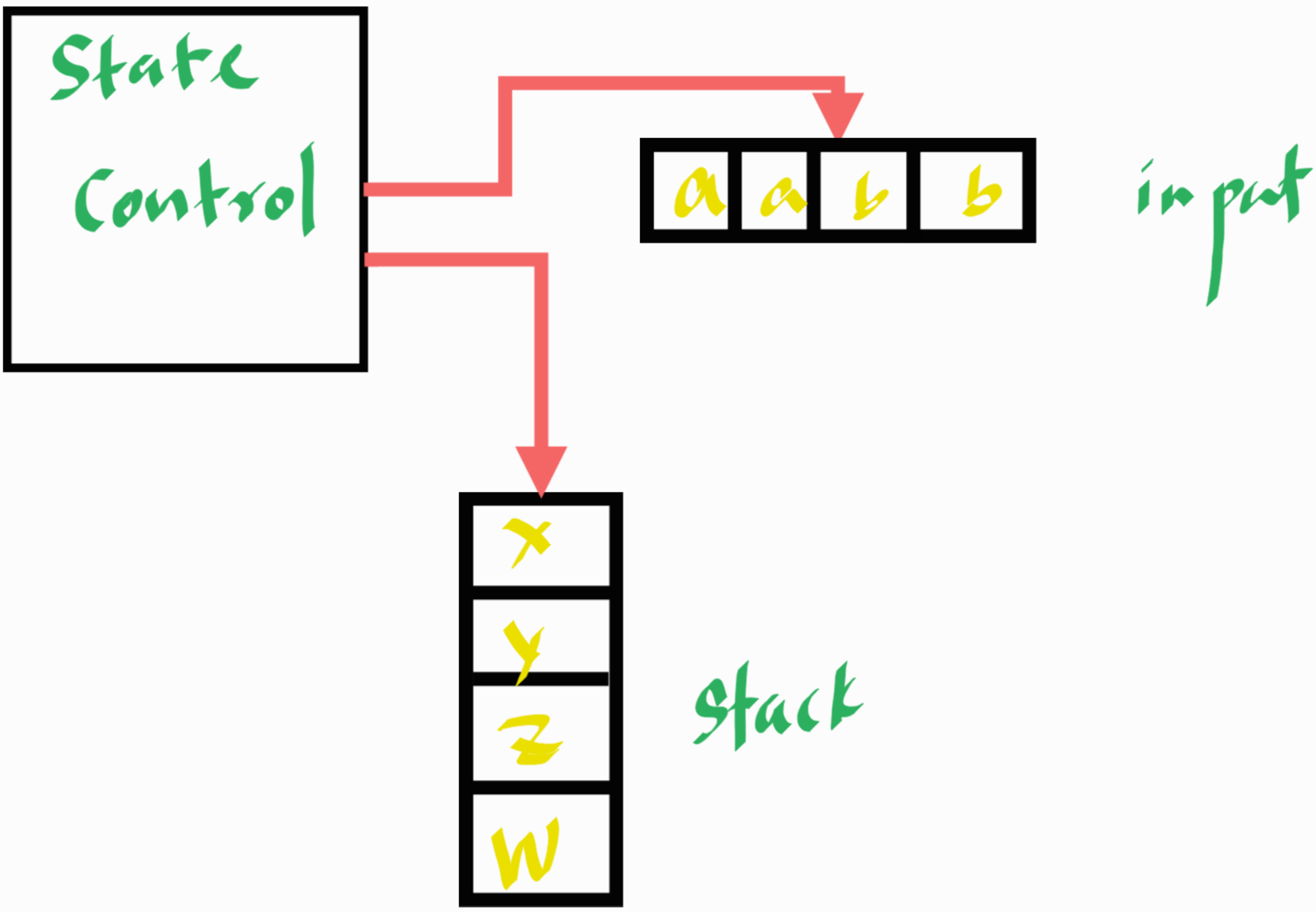
Push Down automaton (PDA)

- NFA + stack
- stack provides additional memory beyond the # of states (memory available).

Push down automata are equivalent in power to context-free grammar.



How can we use this idea?

$$A = \{0^n 1^n \mid n \geq 0\}$$

Rough Idea (not complet)

→ read 0, push to the stack

– read 1, match it up with a 0 on the top of the stack.

→ at the end, if there is nothing left on the stack and on the input tape then accept.

0 0 0 1 1 1 0 1

This is just the rough idea.

Definition

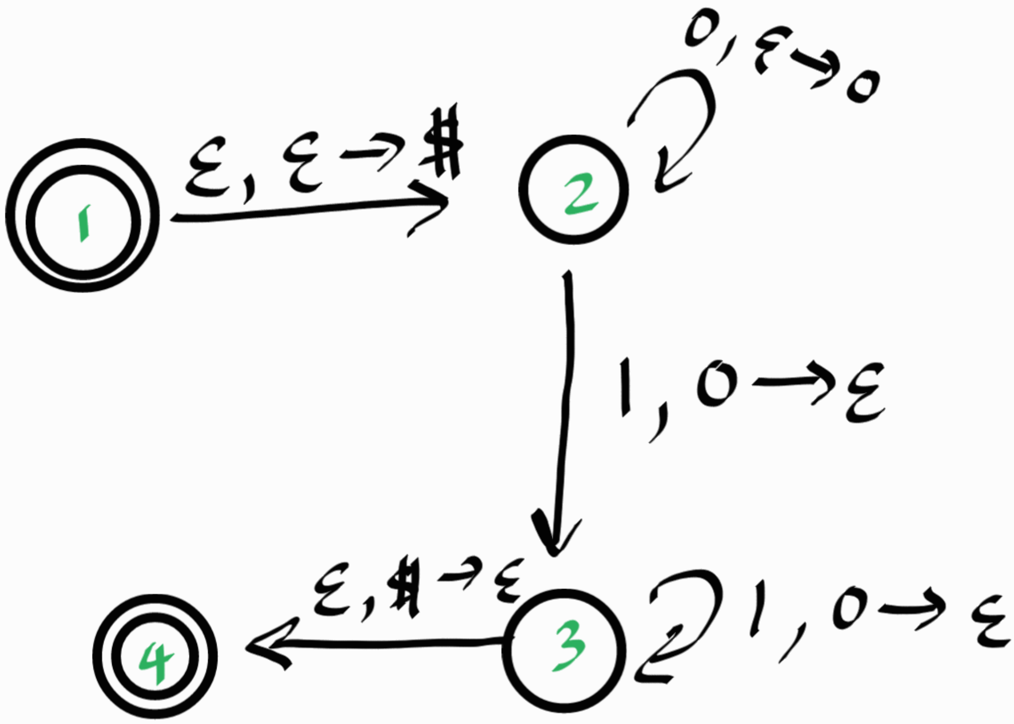
A pushdown automaton is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$

- Q is the set of states
- Σ is the input alphabet.
- Γ is the stack alphabet.

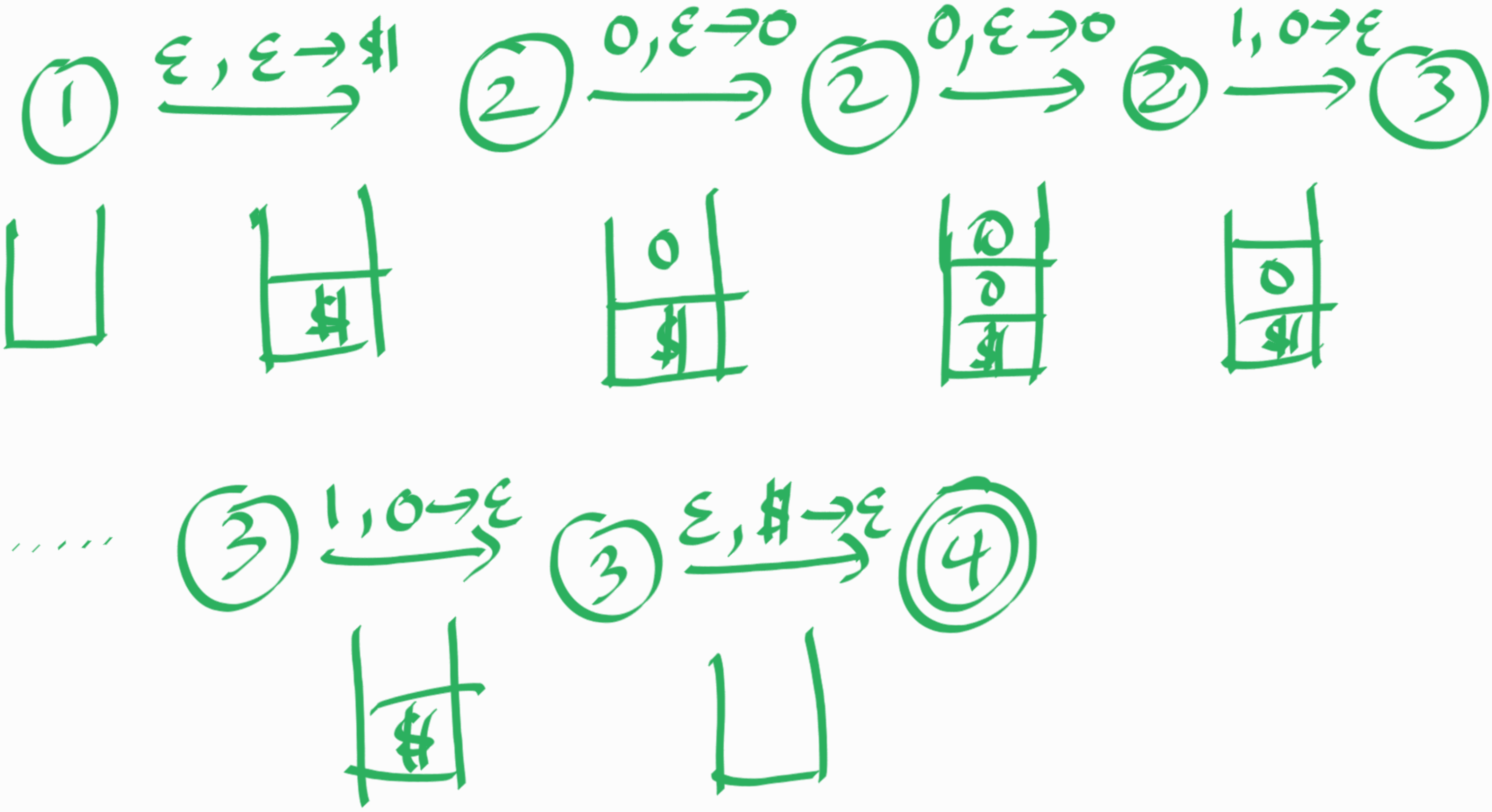
  (Γ does not need to be Σ)
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \to \mathcal{P}(Q \times \Gamma_\varepsilon)$

  $\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$ $\Gamma_\varepsilon = \Gamma \cup \{\varepsilon\}$
- q_0 is the start state
- $F \subseteq Q$ is the set of accept states

Ex 01: PDA that recognizes $\{0^n 1^n \mid n \geq 0\}$

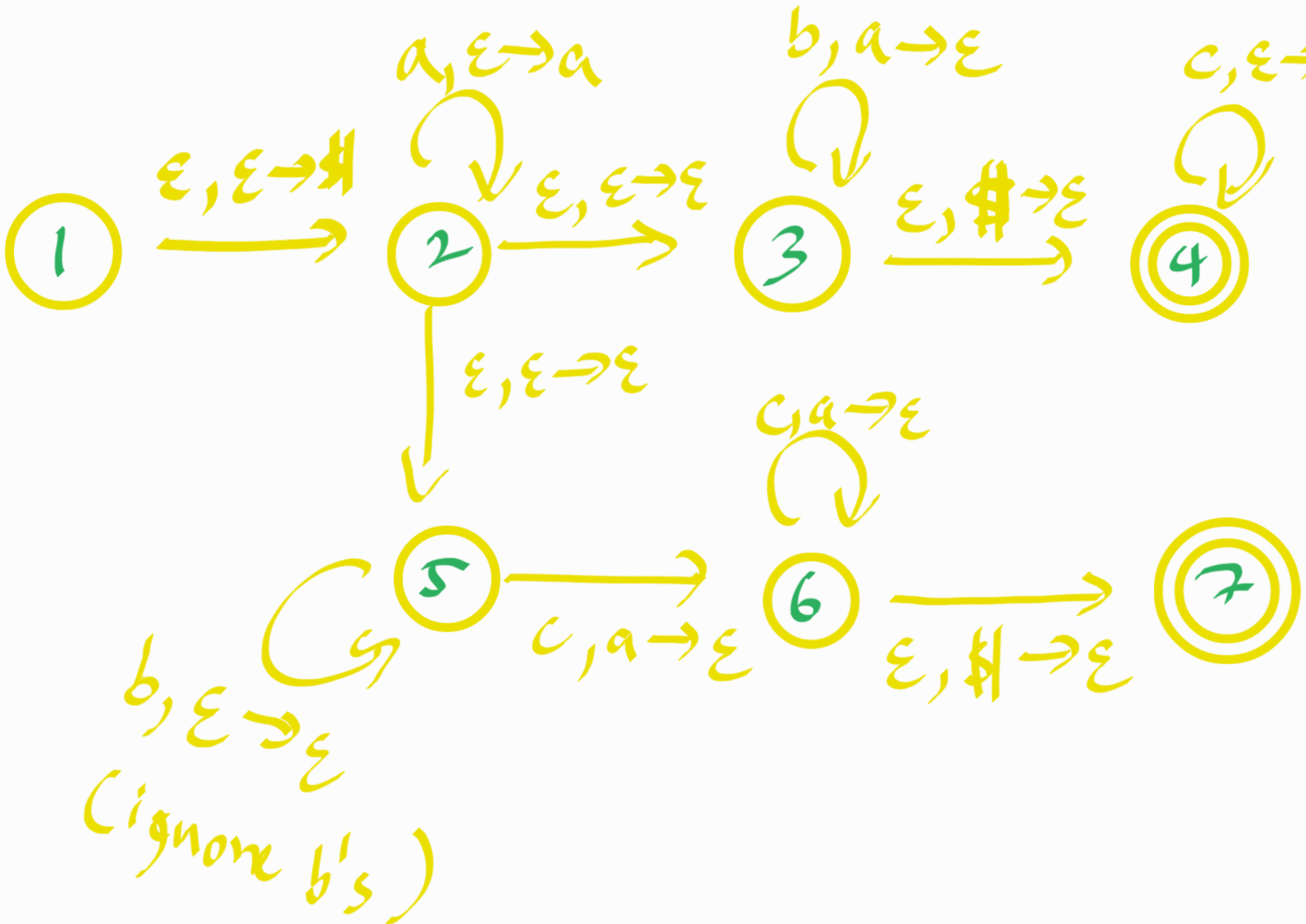


$S = 0^2 1^2$



PDA for $A = \{a^i b^j c^k \mid i,j,k \geq 0,\ i=j \text{ or } i=k\}$

$$\{a^i b^i c^k\} \cup \{a^i b^j c^i\}$$



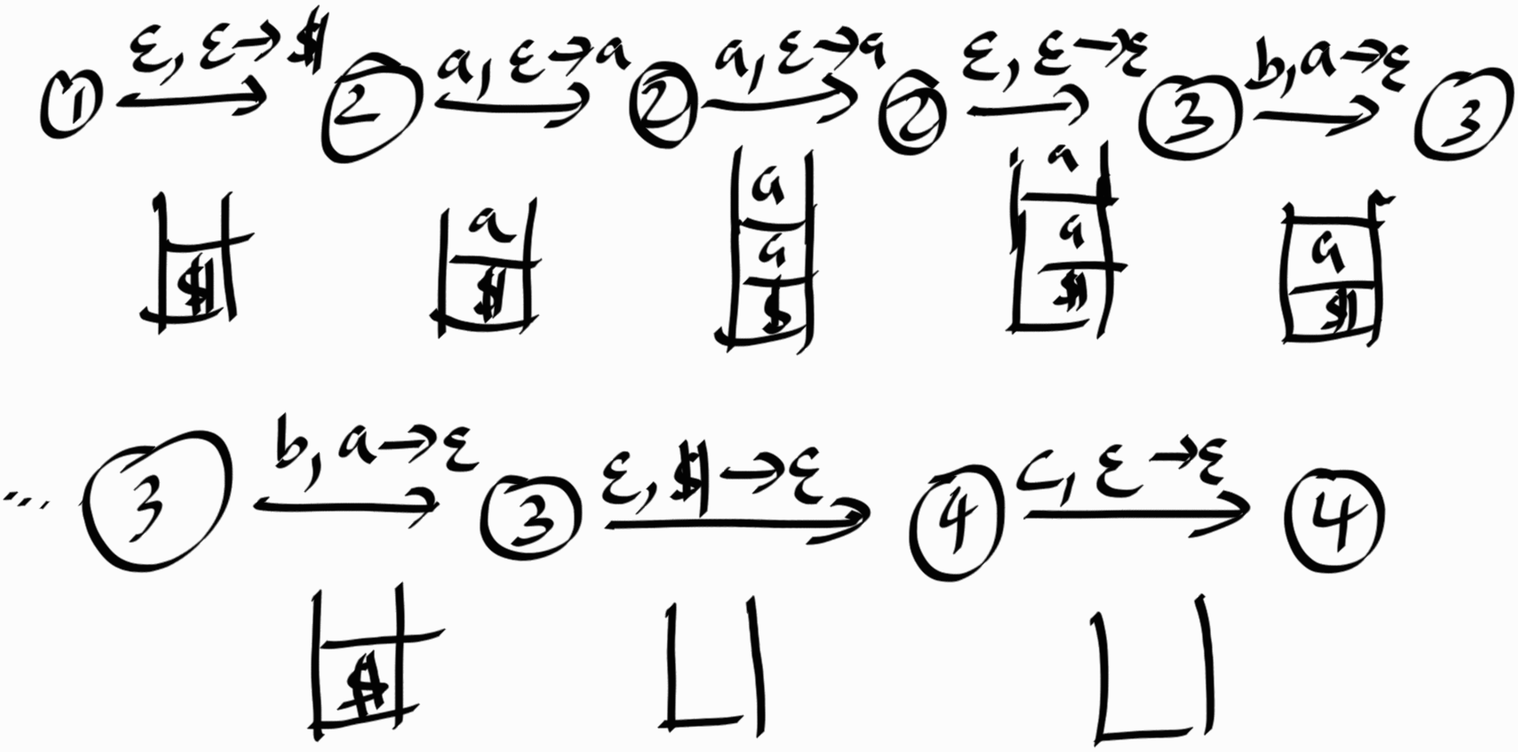
$a, b \longrightarrow c$

if a is ε : reading nothing

if b is ε : pop nothing from the stack

if c is ε : push nothing to stack

How about $a^2 b^2 c^1$?



PDA for $B = \{ww^R \mid w \in \{0,1\}^*\}$