

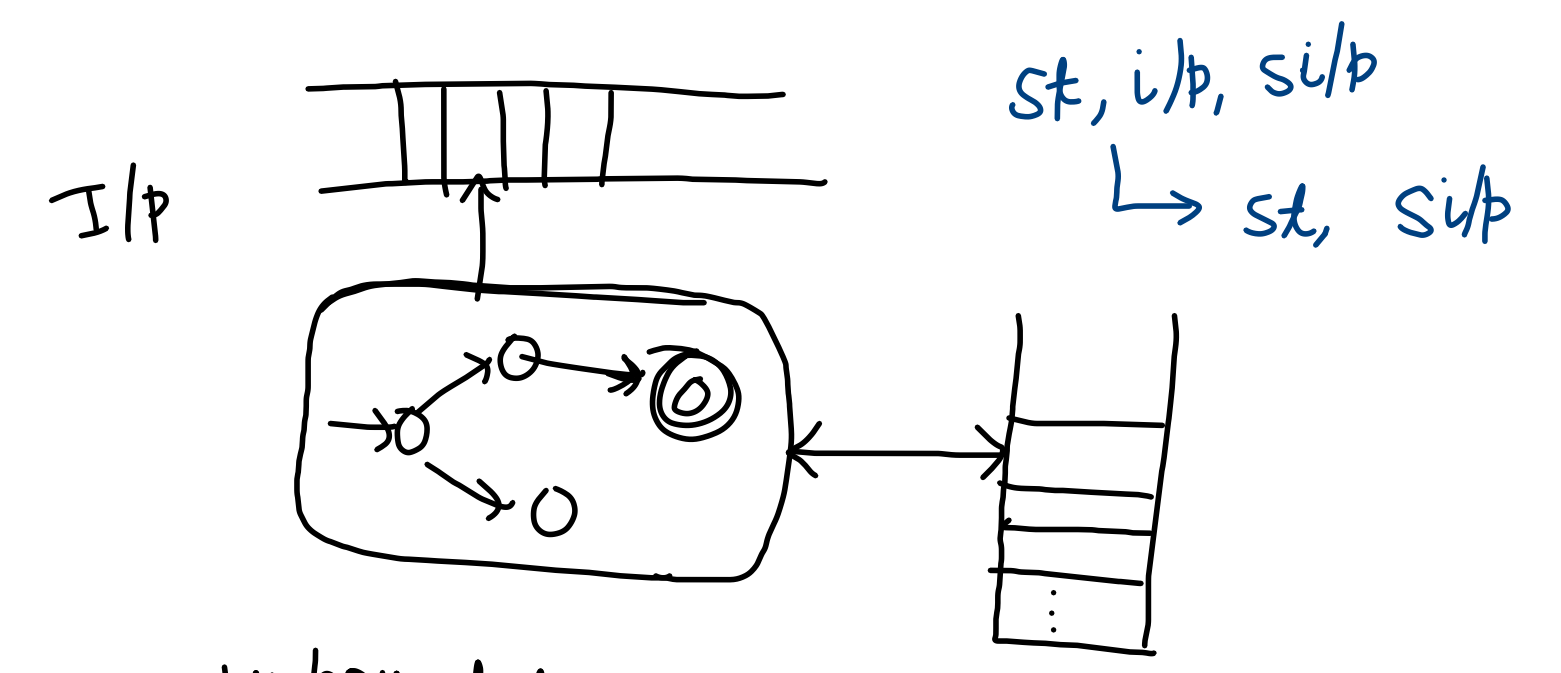
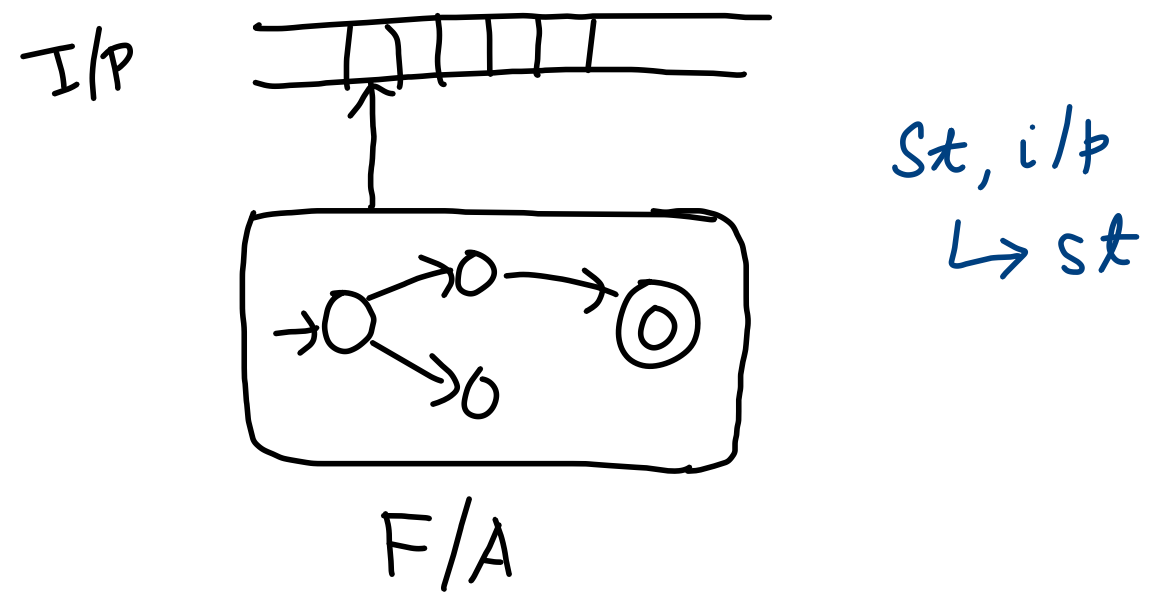
## Regular Languages

- Finite Automata
  - Verifying whether strings belong to the lang
- Regular Expression
  - listing all the strings belong to the lang.

## Context-Free Language

- Pushdown Automata

- Context-free Grammar



- unbounded memory
- access is restricted (only top is visible, follows LIFO structure)

PDA

$$L = \{ 0^n 1^n \mid n \in \mathbb{N} \}$$

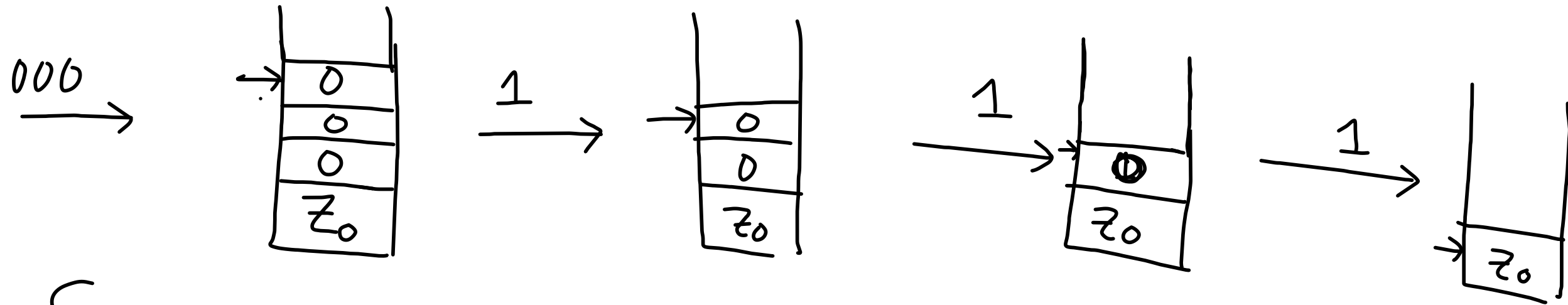
↳ Is this a regular lang?

No (formal proof → pumping lemma)

Intuition: No unbounded memory to keep track of  $n$

$$L = \{0^n 1^n \mid n \in \mathbb{N}\}$$

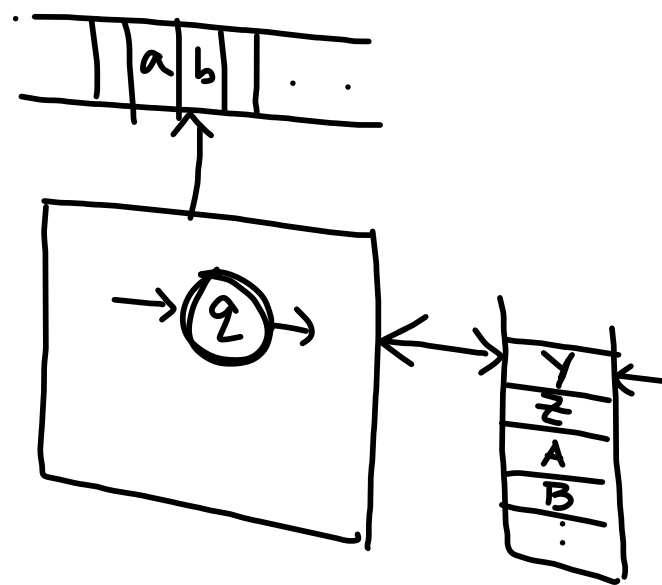
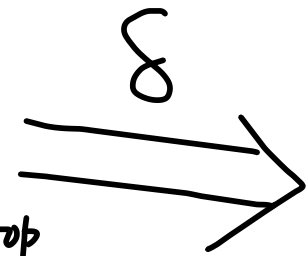
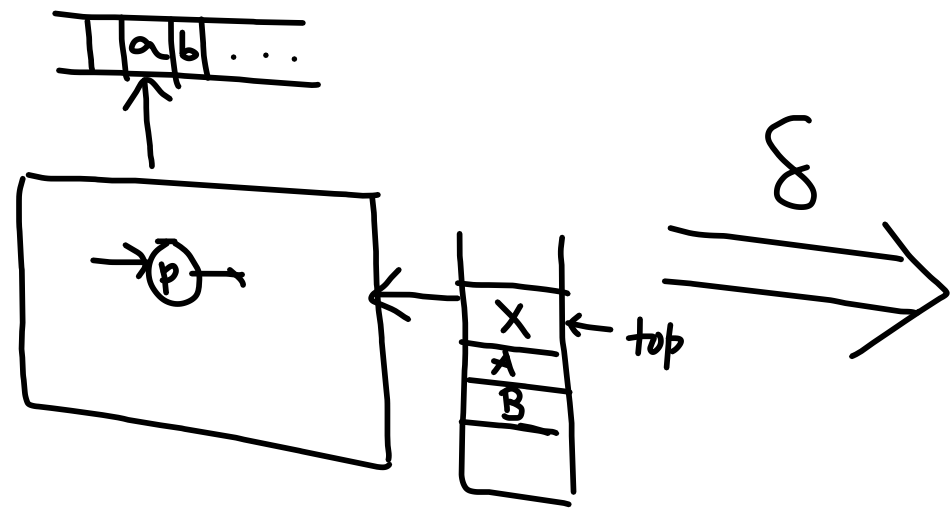
00111



- Whenever you encounter a '0' → push '0' in the stack
- - - - - a '1' → if top is '0' then pop from the stack
- At the end if you have empty stack ⇒ the string should be accepted.  
(When i/p is finished)

# PDA for $L = \{0^n 1^n \mid n \in \mathbb{N}\}$

$$\delta: (p, a, X) \rightarrow (q, YZ)$$



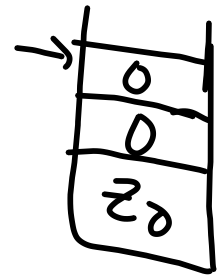
Transitions

- pop(x)
- push(z)
- push(y)

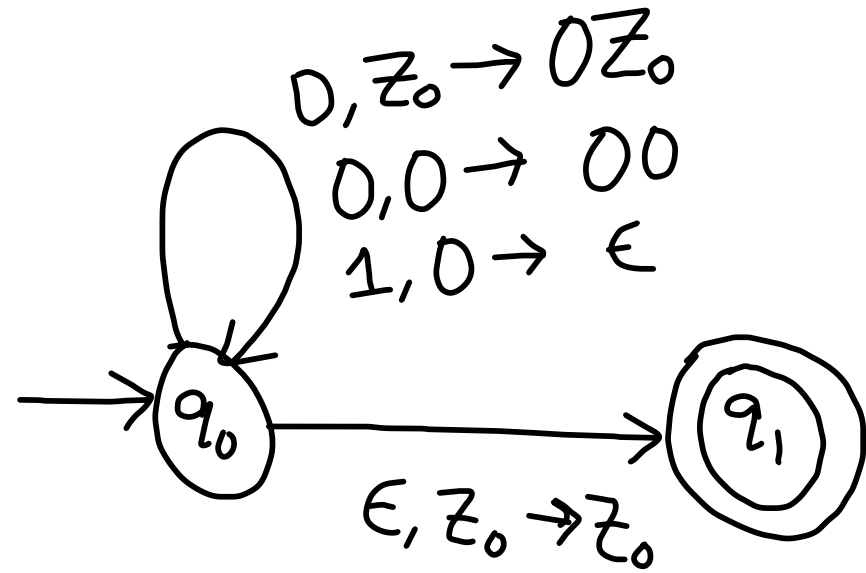
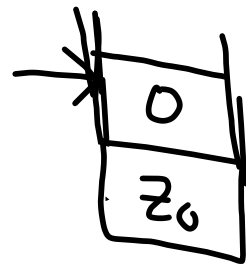
# PDA for $L = \{0^n 1^n \mid n \in \mathbb{N}\}$

0110-...  
 0010111...

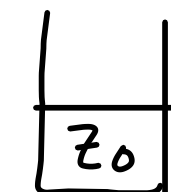
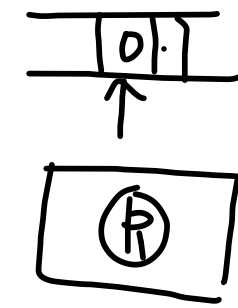
001011



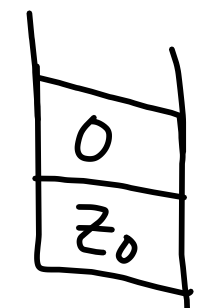
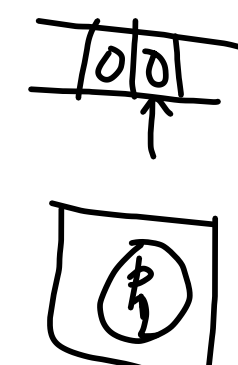
$\Rightarrow$



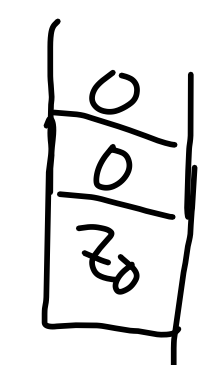
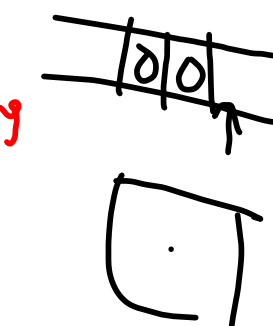
$\Downarrow$   
 All strings with  
 - #0 = #1  
 - all substring from begining  
 should have #0  $\geq$  #1



$\Uparrow$



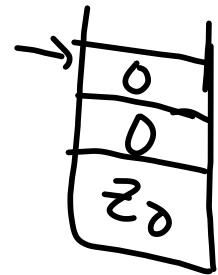
$\Uparrow$



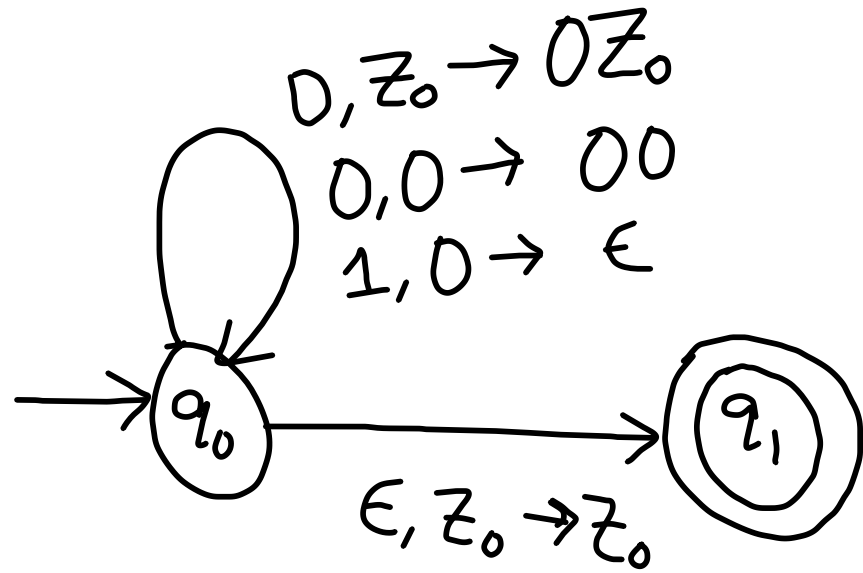
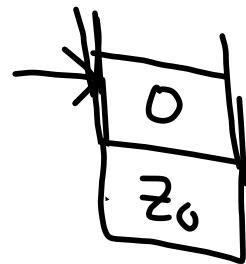
# PDA for $L = \{0^n 1^n \mid n \in \mathbb{N}\}$

0110-...  
0010111...

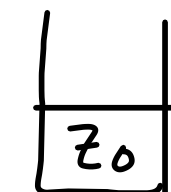
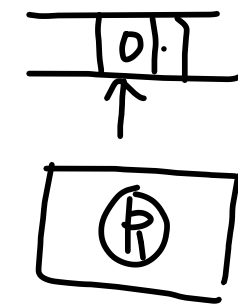
001011



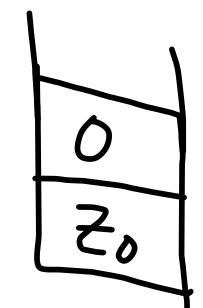
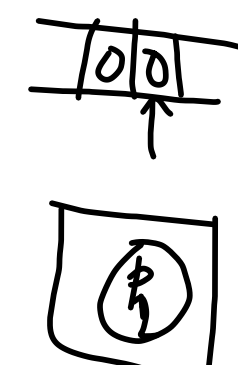
$\Rightarrow$



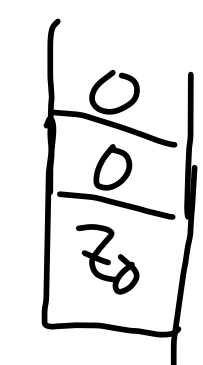
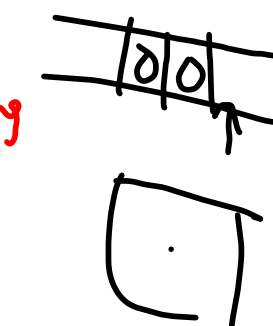
$\Downarrow$   
 All strings with  
 - #0 = #1  
 - all substring from begining  
 should have #0  $\geq$  #1



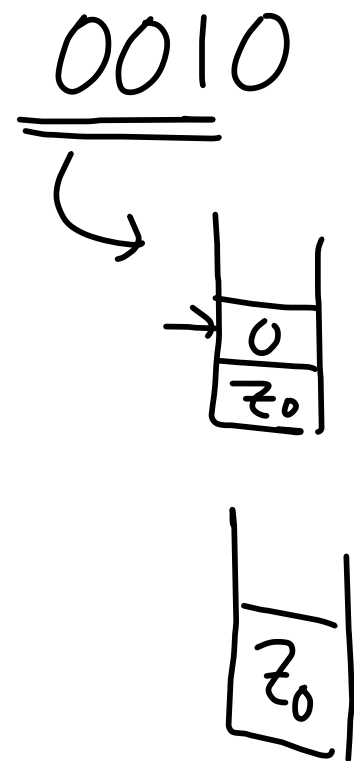
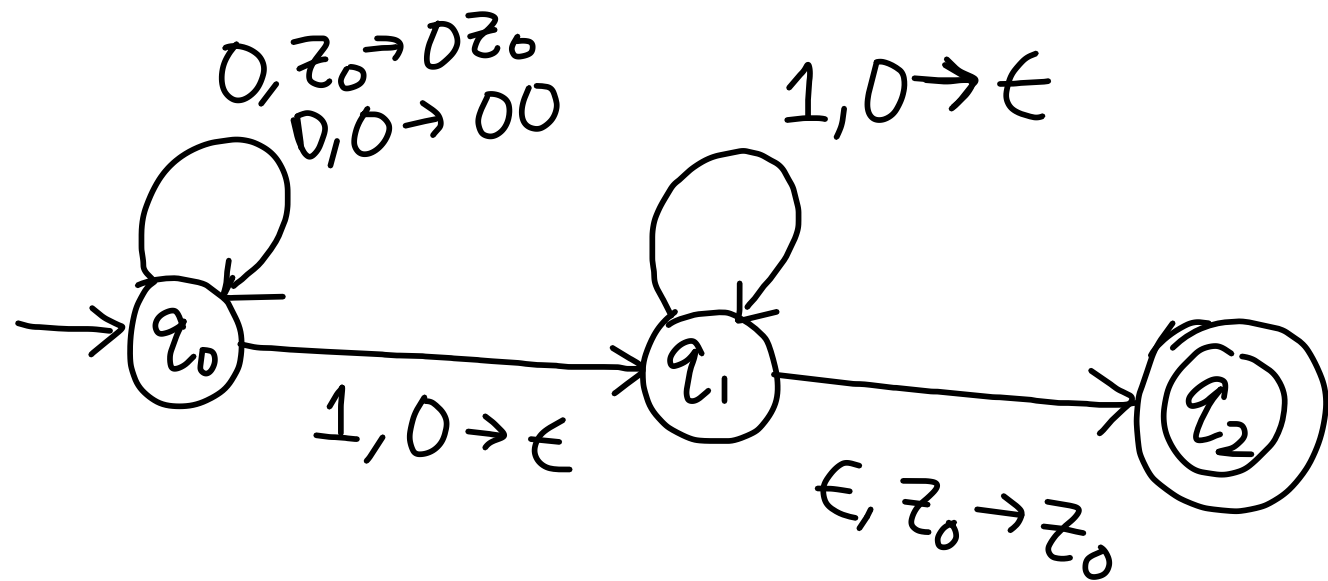
$\Uparrow$



$\Uparrow$

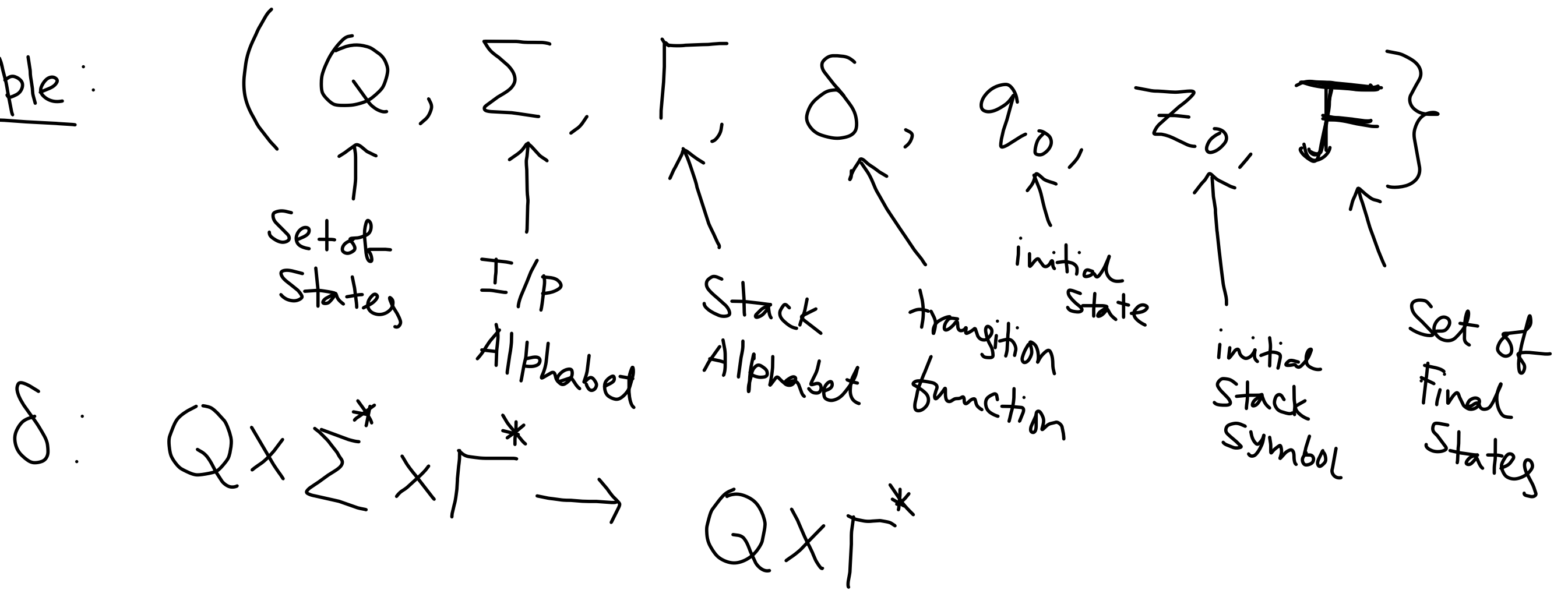


# PDA for $L = \{0^n 1^n / n \in \mathbb{N}\}$



# Pushdown Automata

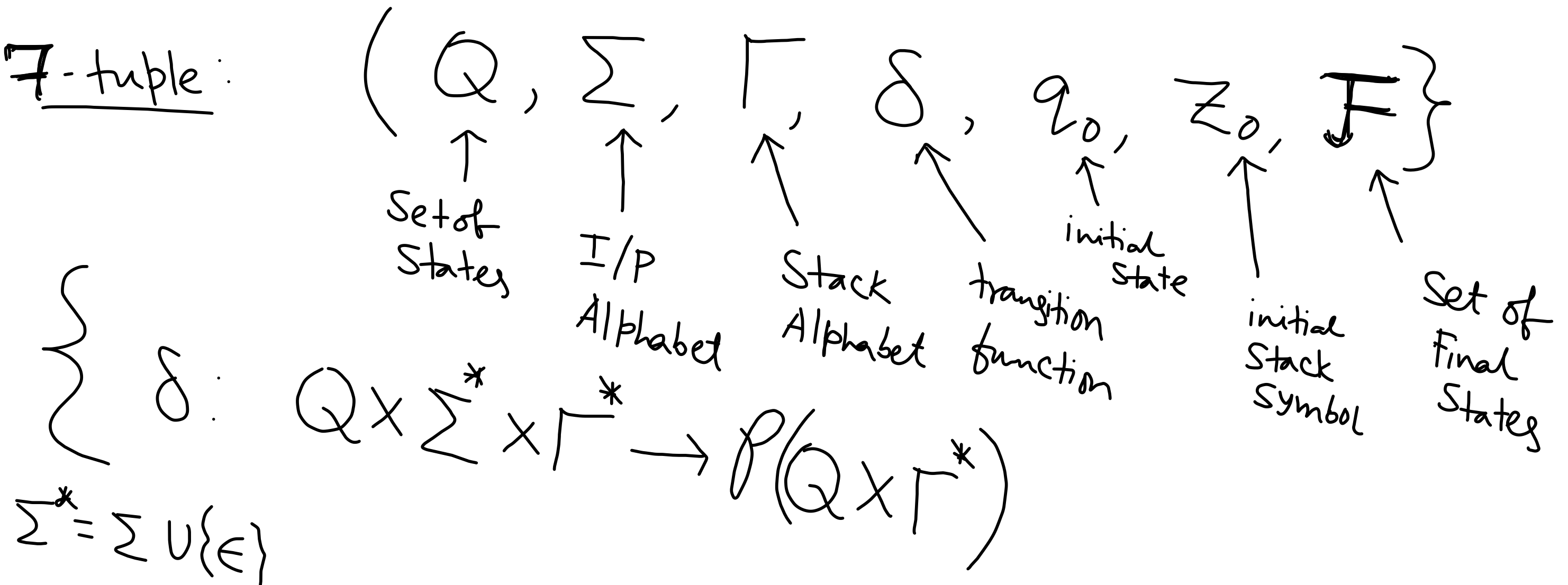
7-tuple:





# Pushdown Automata

7-tuple:



Design a PDA for  $L = \{ wew^r \mid w \in \{0,1\}^* \}$

- Before  $\epsilon$

↳ Push accordingly

- i/p  $\epsilon$

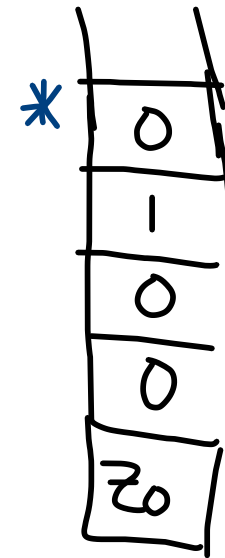
↳ go to the next state

- After  $\epsilon$

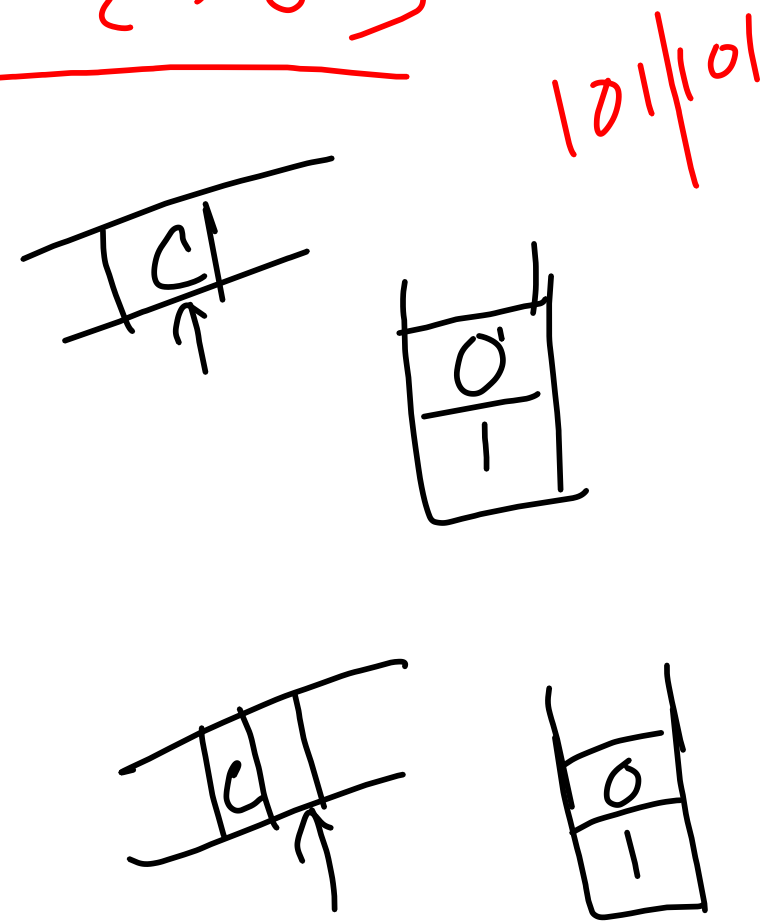
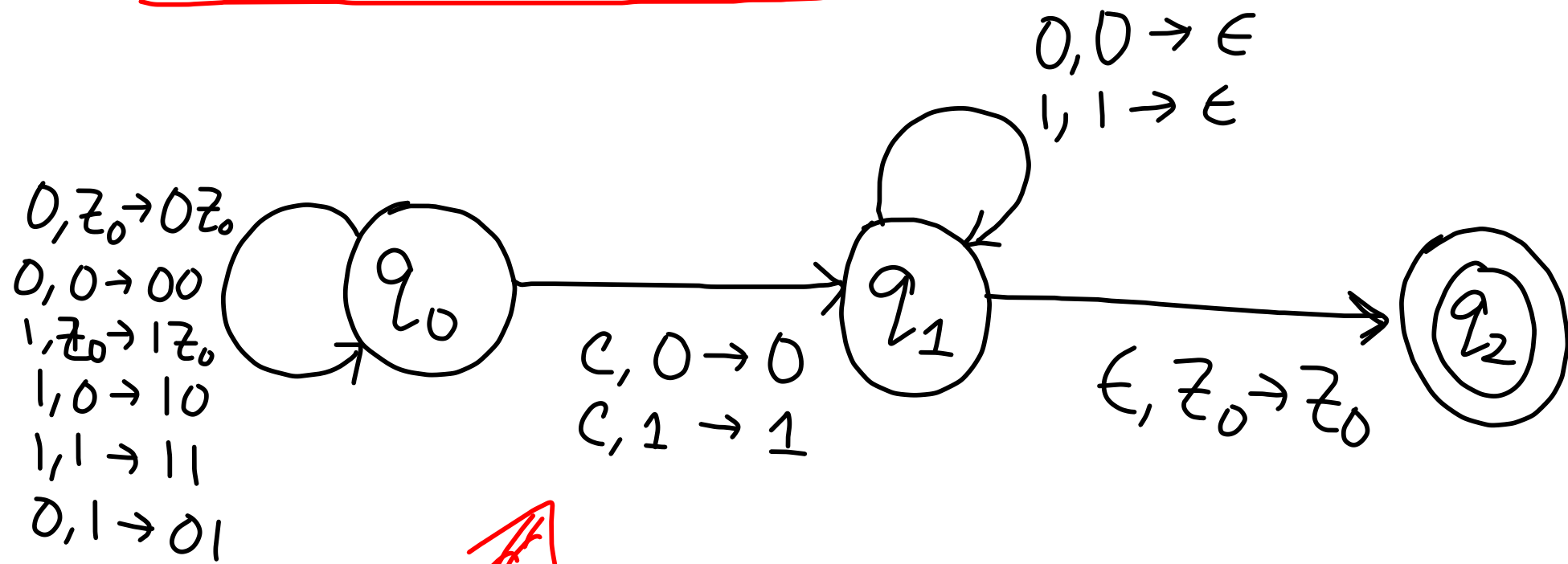
↳ check if top of stack & next i/p matches. If yes → push  $\epsilon$

↳ if any i/p is consumed & stack  $\rightarrow z_0 \Rightarrow$  Done (Accept)

0010  $\epsilon$  | 0100  
 ↑



Design a PDA for  $L = \{w c w^r \mid w \in \{0,1\}^*\}$



How about  $L = \{w w^r \mid w \in \{0,1\}^*\}$ ?

$\epsilon, 0 \rightarrow 0$   
 $\epsilon, 1 \rightarrow 1$

Exercise:

$L = \{w \in \{0,1\}^* \mid w \text{ has equal no. of } 01 \text{ \& } 10 \text{ as substring}\}$

(Counter-intuitive)

↳ Is this regular? Is this context-free?

↓

Actually do this even  
w/o counting the #01 or  
#10

Try to construct the DFA

$$L = \{0^n 1^n 2^n \mid n \in \mathbb{N}\}$$

$$L = \{0^n 1^n 2^n 3^n 4^n \mid n \in \mathbb{N}\}$$

Is this context-free?

No

2 stacks  $\rightarrow$   $\checkmark$   
PDA with 2 stacks  $\equiv$  TM.