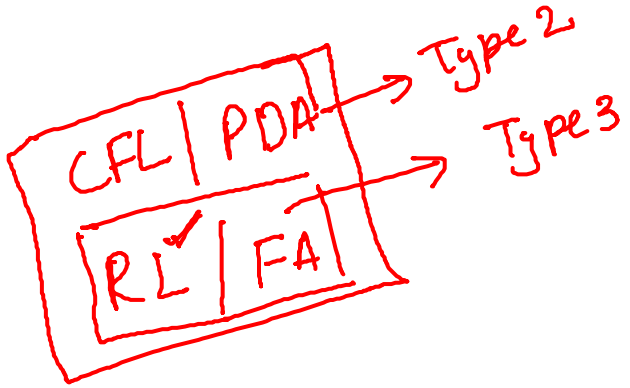
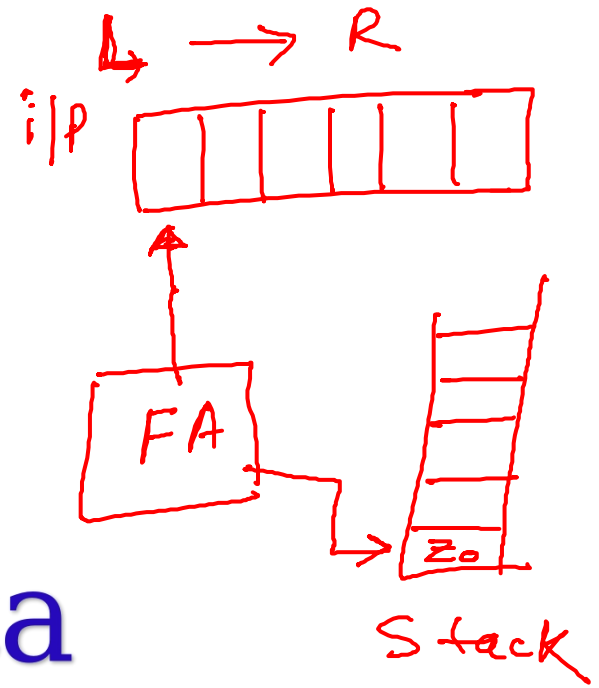




Pushdown Automata

PDA = FA + 1 memory (Stack)

$\Sigma = \{a, b\}$
$\Gamma = \{A, B\}$

PDA

for stack

$$M = (Q, \Sigma, \delta, q_0, Z_0, F, \Gamma)$$

Z_0 → Stack's top element

Γ → Stack alphabet

δ: Transition function.

→ $\delta: Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \to Q \times \Gamma^*$

(deterministic) DPDA

DCFL

→ $\delta: Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \to 2^{(Q \times \Gamma^*)}$

✓ Non deterministic ← By default

NDPDA

NDCFL

Top = Top + 1

Top = Top − 1

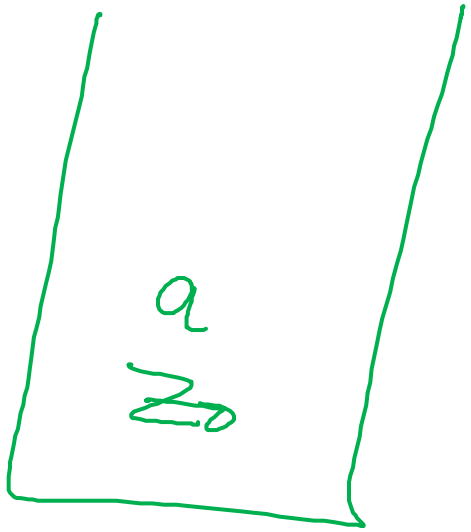
Z_0 → default stack element

$\delta(q_i, a, Z_0) \rightarrow \delta(q_j, aZ_0)$

stack updated top element

- q_i: current state
- a: current i/p alphabet
- Z_0: stack top element
- q_j: Next state

$Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$



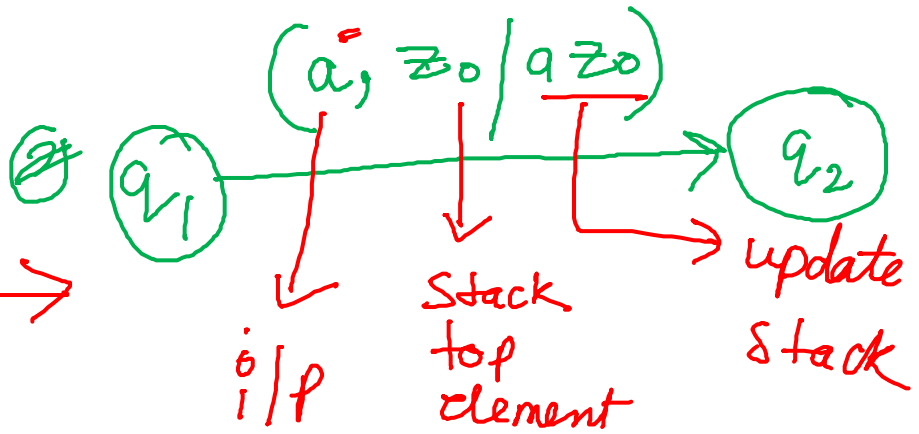
Operation on Stack →

① Push ⇒ $\delta(q_i, a, z_0) \longrightarrow \delta(q_j, az_0) \longrightarrow$

② Pop ⇒ $\delta(q_i, a, b) \longrightarrow \delta(q_j, \varepsilon) =$

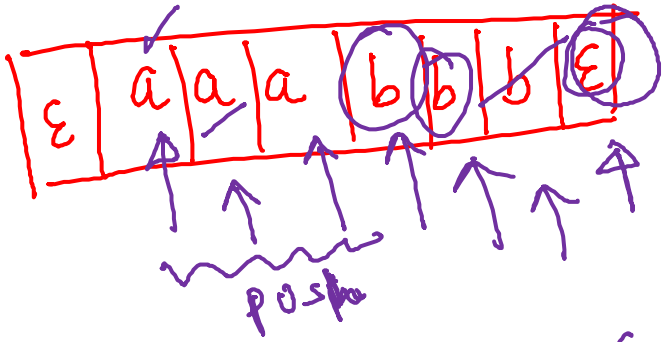
③ Skip –

$(q_1, a, z_0) = (q_2, az_0)$



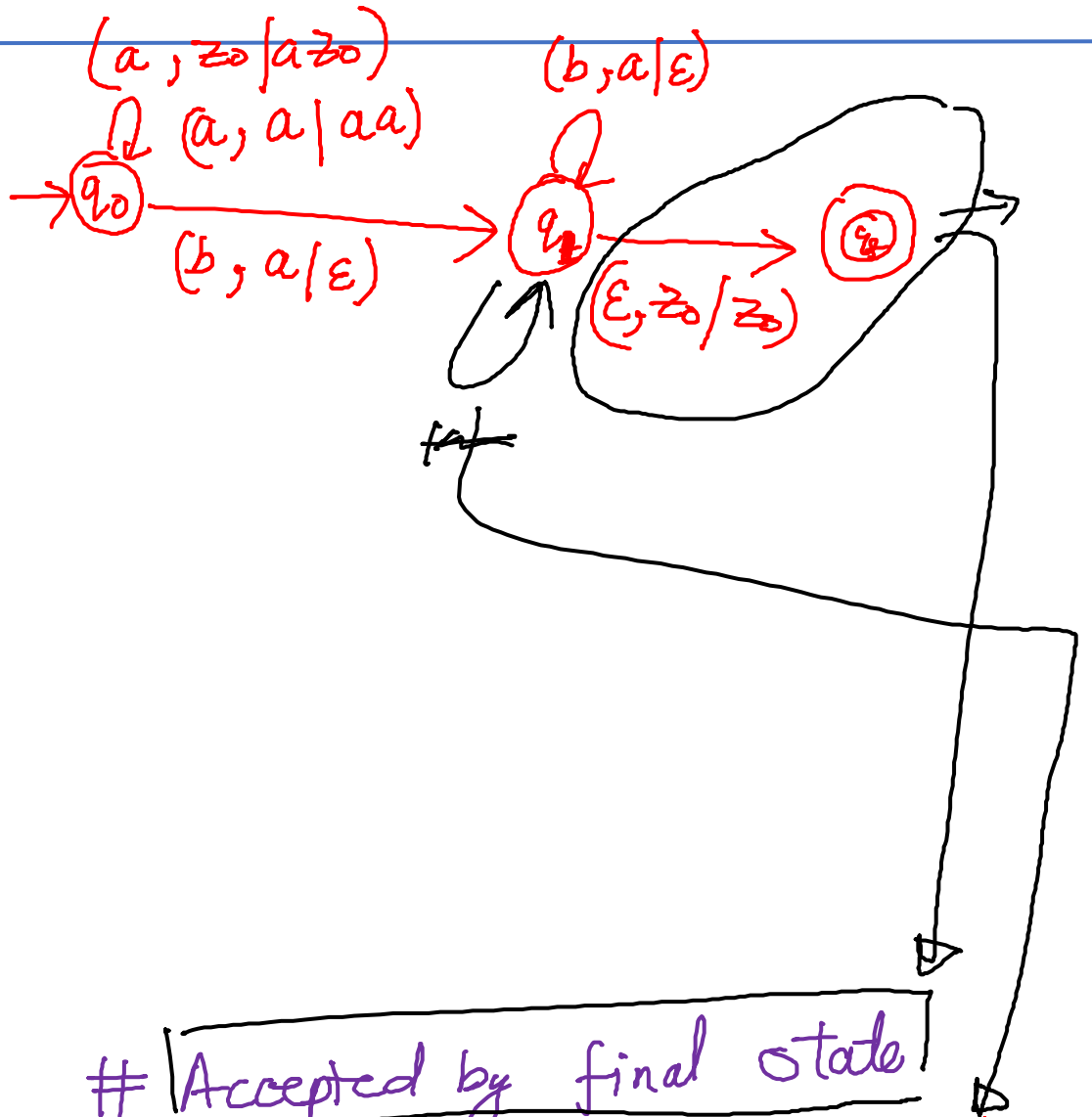
Ex → $L = \{a^n b^n : n \geq 1\}$ ⟹ $\Sigma = \{a, b\}$ $\Gamma = \{a\}$

no. of a = no. of b



Push a
- $(q_0, a, z_0) \rightarrow (q_0, az_0)$
- $(q_0, a, a) \rightarrow (q_0, aa)$

Pop a
- $(q_0, b, a) \rightarrow (q_1, \varepsilon)$
- $(q_1, b, a) \rightarrow (q_1, \varepsilon)$
- $(q_1, \varepsilon, z_0) \rightarrow (q_f, z_0)$ ✓ or



Accepted by final state

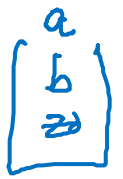
✓ $(q_1, \varepsilon, z_0) \rightarrow (q_1, \varepsilon)$ Accepted by empty stack

$a, b \mid a\dot{b}$ → Push operation

i/p, stack top element | update stack

CFL → PDA → DPDA, ~~NDPDA~~ NDPDA

CFL → CFG



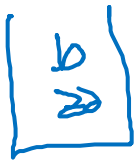
$M = (Q, \Sigma, q_0, F, \delta, \Gamma, Z_0)$

Z_0 → stack symbol

Γ → stack alphabet

δ → Transition f^n

Pop operation

$a, b \mid \varepsilon$ → 

i/p, stack top | ε

DPDA

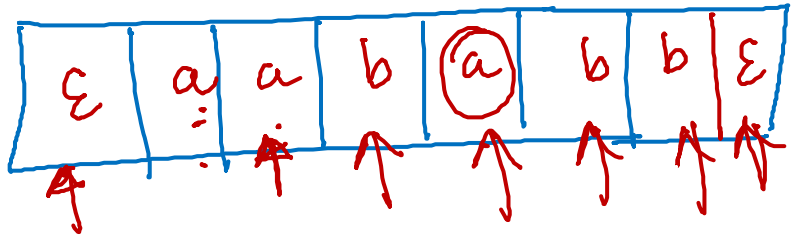
$\delta : Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$

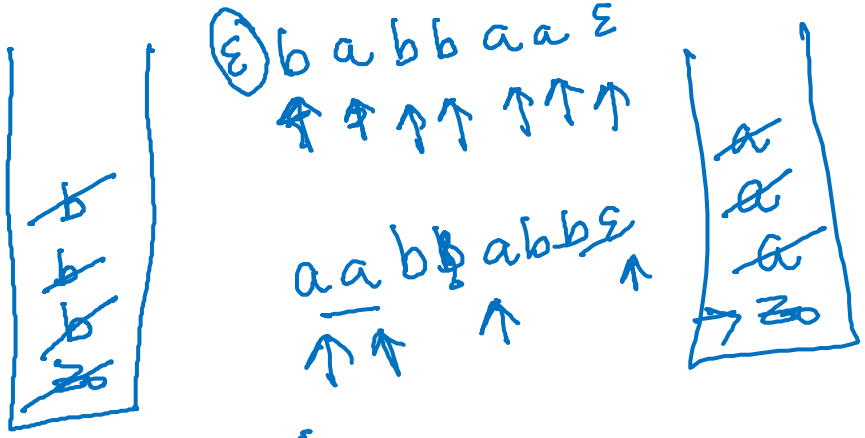
NDPDA

$Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$

PDA — Accepted by final state ✓ / Empty stack ✓ ⇒ By Nature NDPDA

Ex → $L = \{ w : n_a(w) = n_b(w) \}$

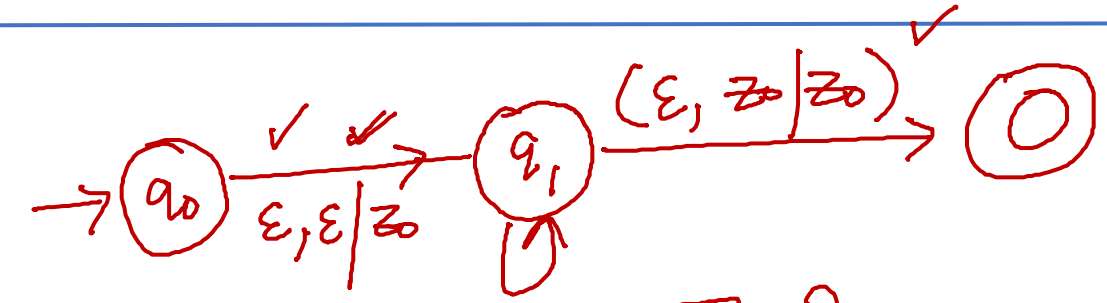




$\Gamma = \{ Z_0, a, b \}$

$\Sigma = \{ a, b \}$

$\Rightarrow (\varepsilon, \varepsilon | Z_0)$



$(a, Z_0 | a Z_0)$

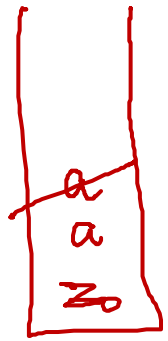
$(a, a | aa)$

$(b, a | \varepsilon)$

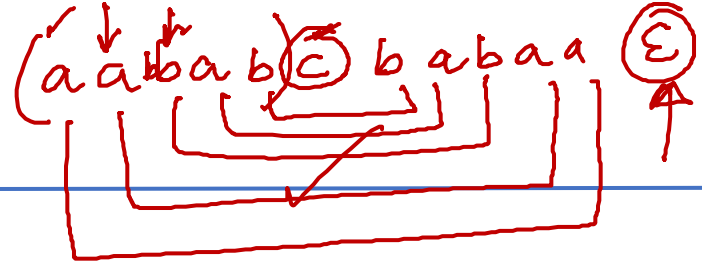
$(b, Z_0 | b Z_0)$

$(b, b | bb)$

$(a, b | \varepsilon)$

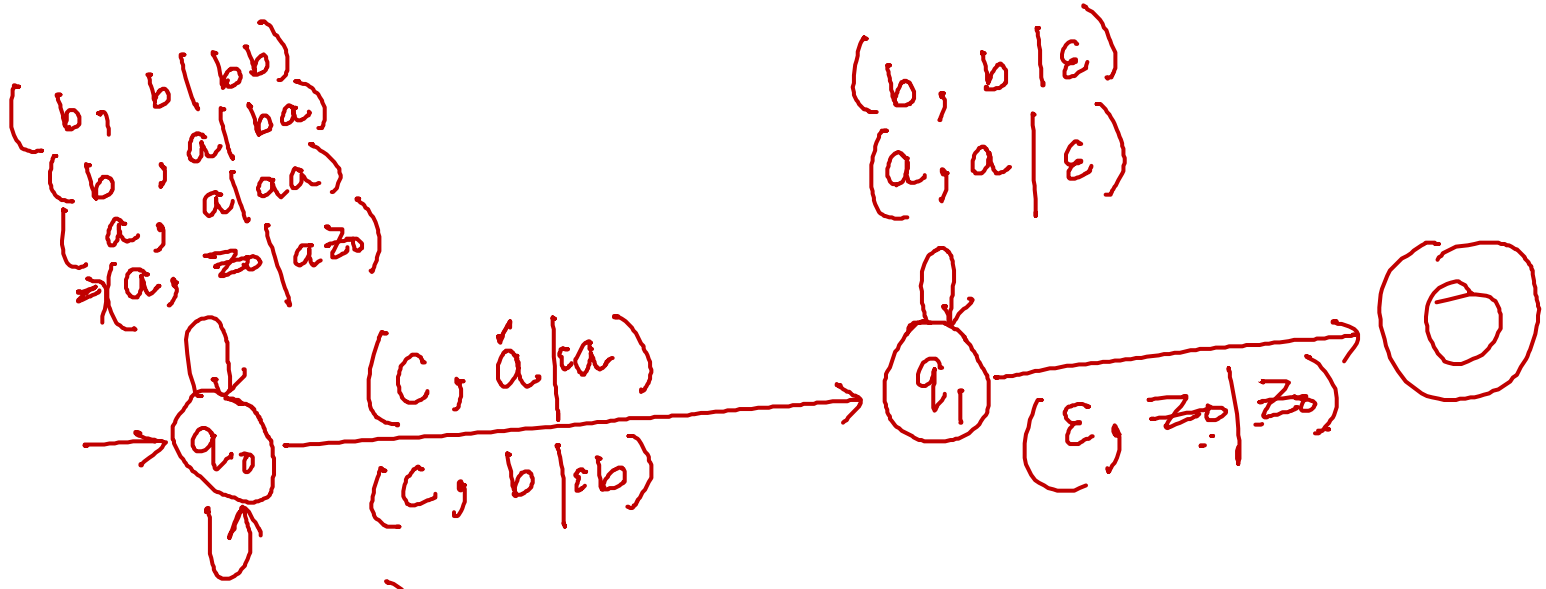


$L = \{ w c w^R : w \in \{0,1\}^* \}$



b a c a b

a b c b a



(i | p, stack | update)

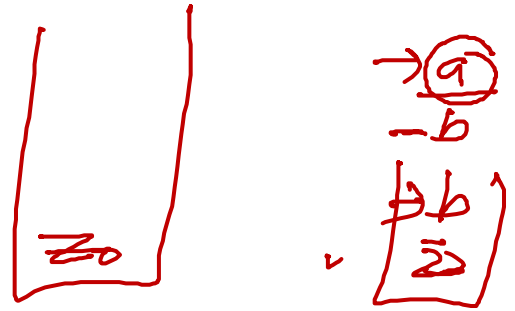
(b, Z0 | bZ0)

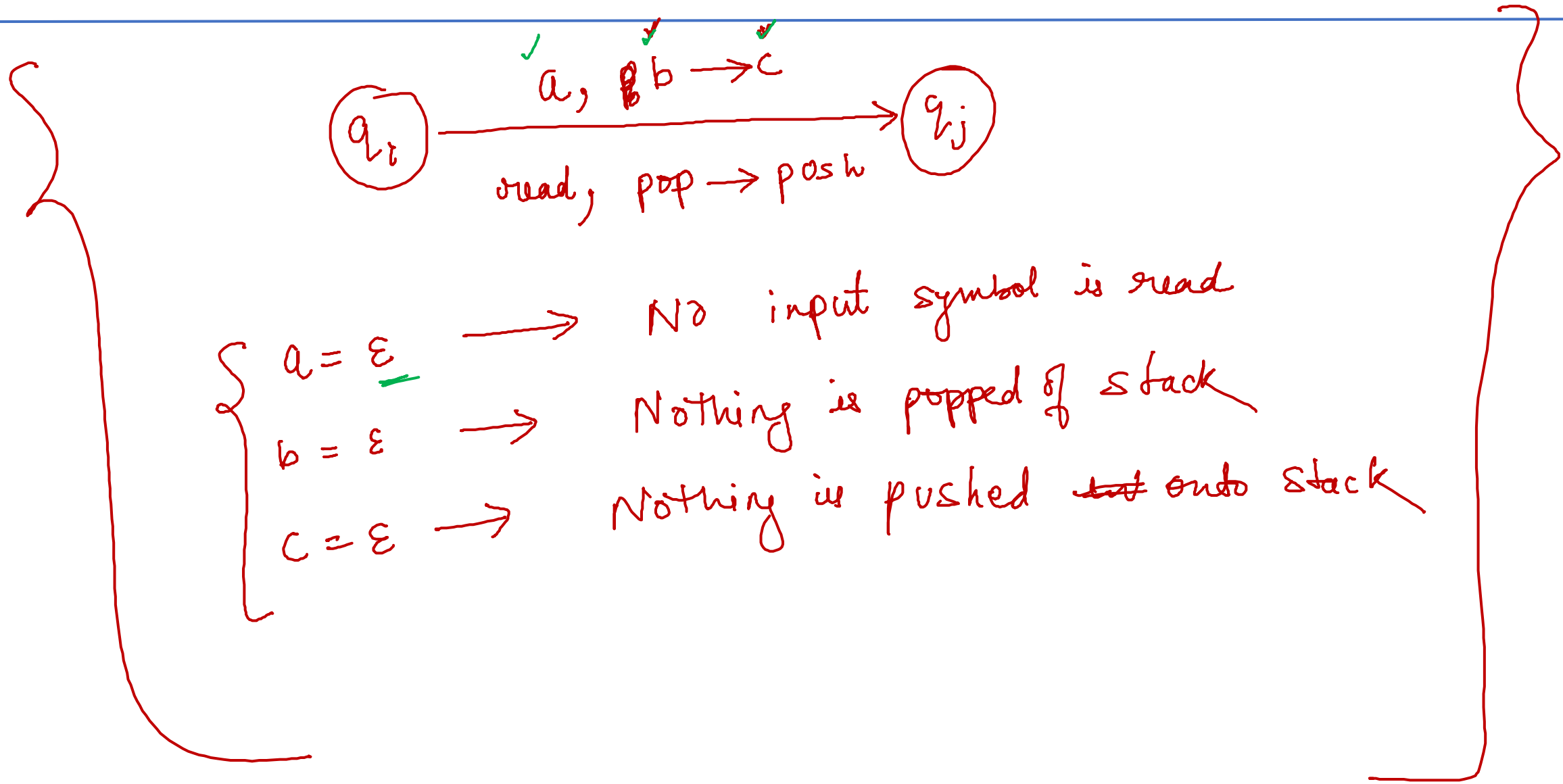
(b, b | bb)

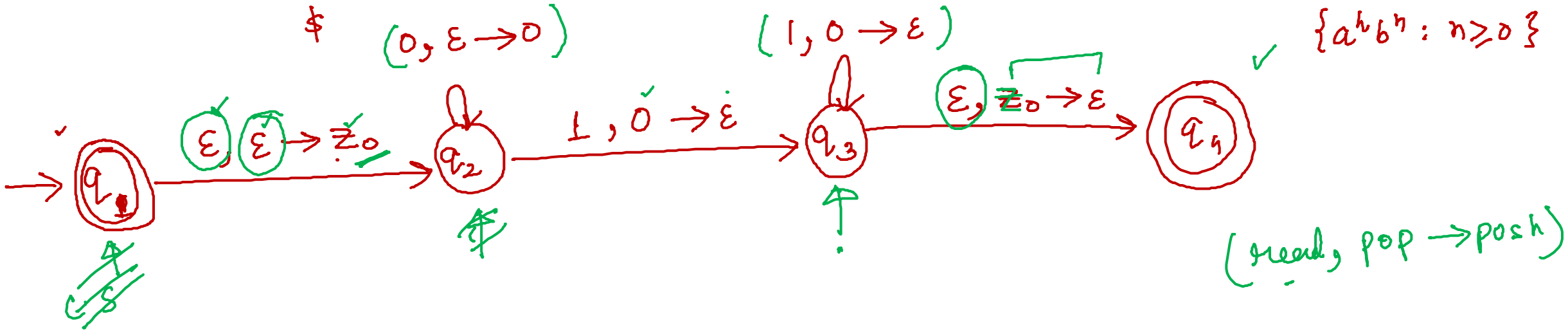
(a, b | ab)

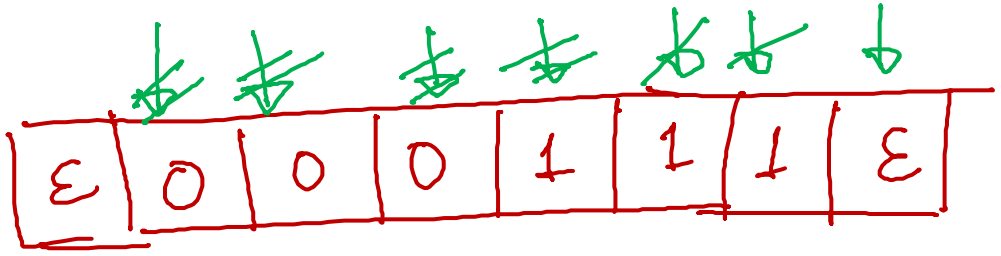
(a, a | aa)

b b a a c a a b b

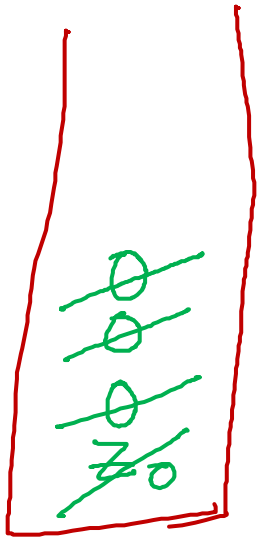



a, b → c
q_i
q_j
read, pop → push
a = ε → No input symbol is read
b = ε → Nothing is popped of stack
c = ε → Nothing is pushed onto stack





1. q_1 read nothing & pushed Z_0 onto the stack
   ↳ start $q_1 \rightarrow q_2$
2. q_2 read 0 then push 0 onto the stack
   q_2 read 1 then pop 0 from stack
   $q_2 \rightarrow q_3$



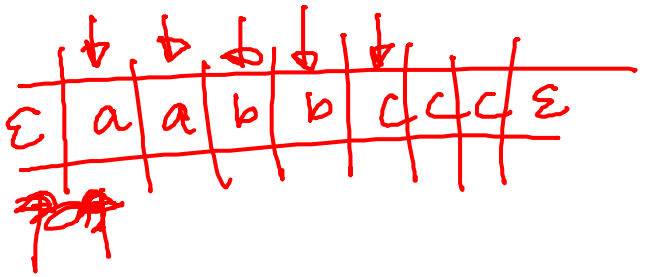
3. q_3 read 1 then pop 0 from stack
   q_3 read ε then Z_0 is poped from stack
4. $q_3 \rightarrow q_4$
   q_4 → final state

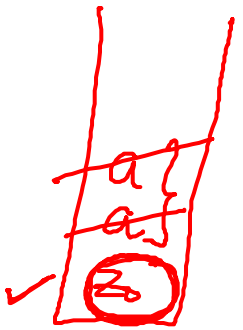
$L = \{ a^n b^n c^m : n \geq 1, m \geq 1 \}$

read every a, push a

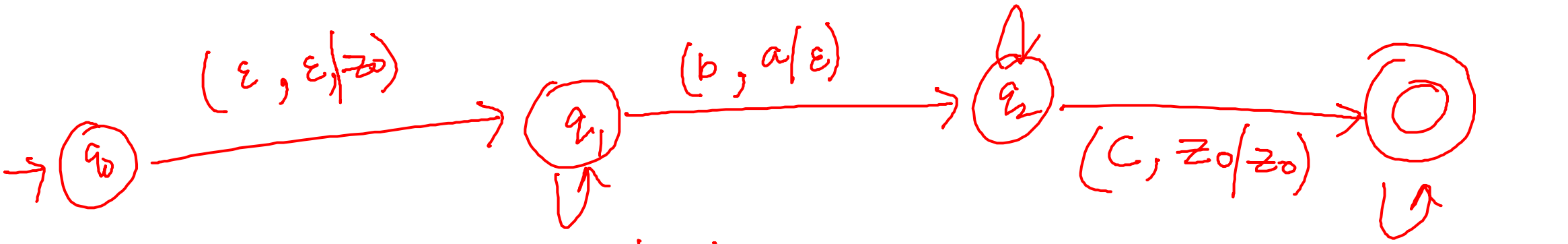
read b pop a

read c, no operation (no updation)





(b, a | ε)



(a, Z0 | a Z0)

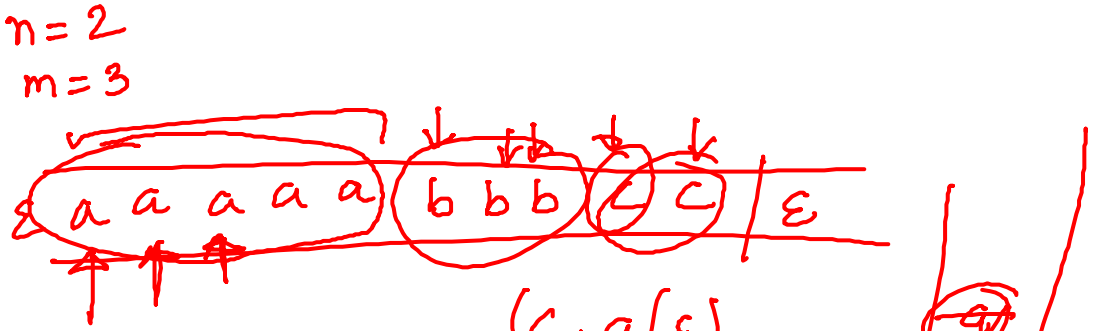
(a, a | a a)

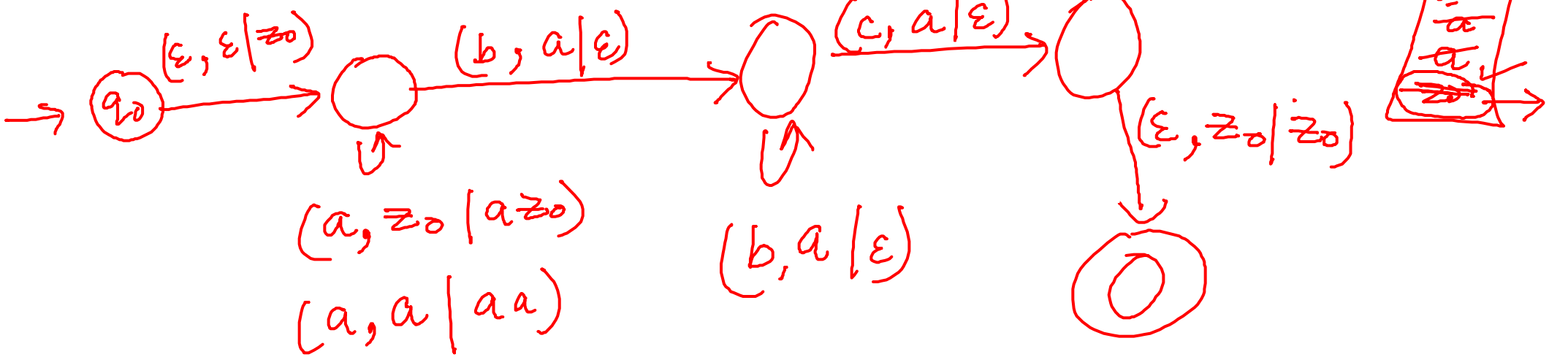
(c, Z0 | Z0)

PDA

$L = \{ a^{m+n} b^m c^n : m, n \geq 1 \}$

$L = \{ a^n a^m b^m c^n : n, m \geq 1 \}$



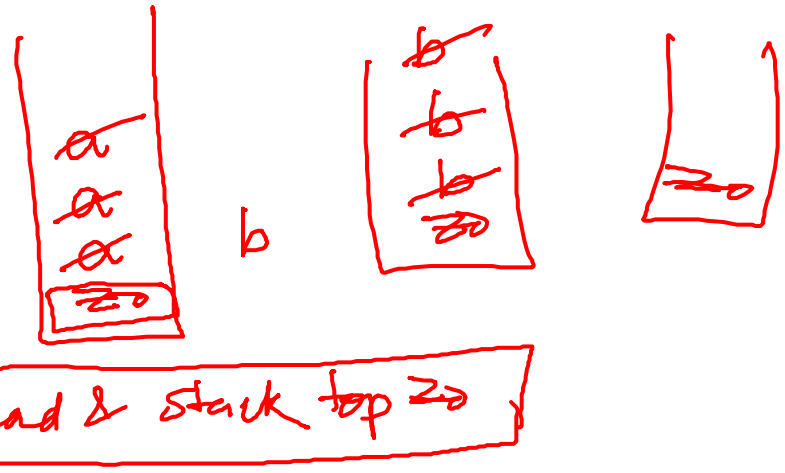


Q → $L = \{ a^n b^{m+n} c^m : n, m \geq 1 \}$

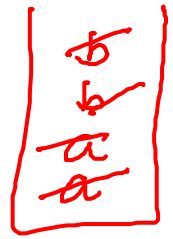
$a^n b^n b^m c^m$

Push

if a read & stack top is z0 or if b read & stack top z0



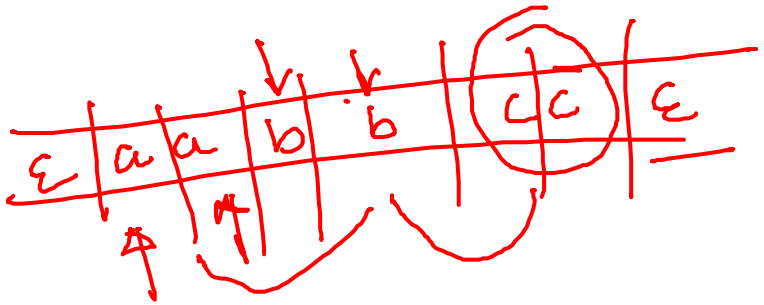
Q → $L = \{ a^n b^m c^{n+m} \mid n, m \geq 1 \}$ ⇒ $a^n b^m c^m a c^n$



Q → $L = \{ a^n b^n c^n : n \geq 1 \}$ ⟶ Not CFL

≤ 0

we can not draw PDA for that language



⇒ $n_a(w) = n_b(w) = n_c(w)$

& first read a then b then c

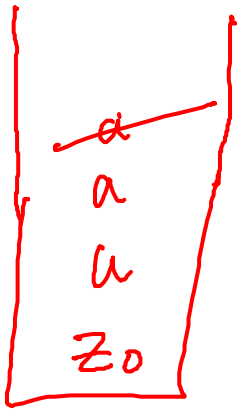
CSL

CSL $L = \{a^n b^n c^n : n \geq 1\}$ → solve by 2 stack PDA

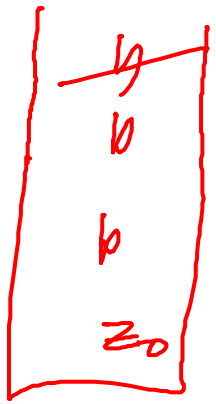
a a a b b b c c c

FA + 1 stack = PDA

FA + 2 stack = TM ✓



push a onto the stack 1



push b onto the stack

c read then pop a from first stack & pop b from second stack

PUMPING LEMMA for CFL → always give negative result

If Language

↓

CFL

→ PDA

→ CFG

Pumping Lemma

if L is CFL then

$w \in L$, $|w| \geq n$ → Pumping length

$\Rightarrow w = pqrst$

such that

1. $|qs| \geq 1$ or always positive
2. $|qrs| \leq n$
3. $pq^i r s^i t \in L$

CFL

$\forall i \geq 1$

$L = \{ a^n b^n c^n : n \geq 1 \}$ — Not CFL

Let n (pumping length) $= 7$ ✓

$|w| = 9$

$|w| \geq 7$ → true

$w = \underbrace{a}_{p}\underbrace{a}_{q}\underbrace{ab}_{q}\underbrace{b}_{r}\underbrace{b}_{s}\underbrace{c}_{s}\underbrace{c}_{t}\underbrace{c}_{t}$

① $|qs| \geq 1$

$5 \geq 1$ true

② $|qrs| \leq n$

$5 \leq 7$ true

③ $p q^i r s^i t \in L$

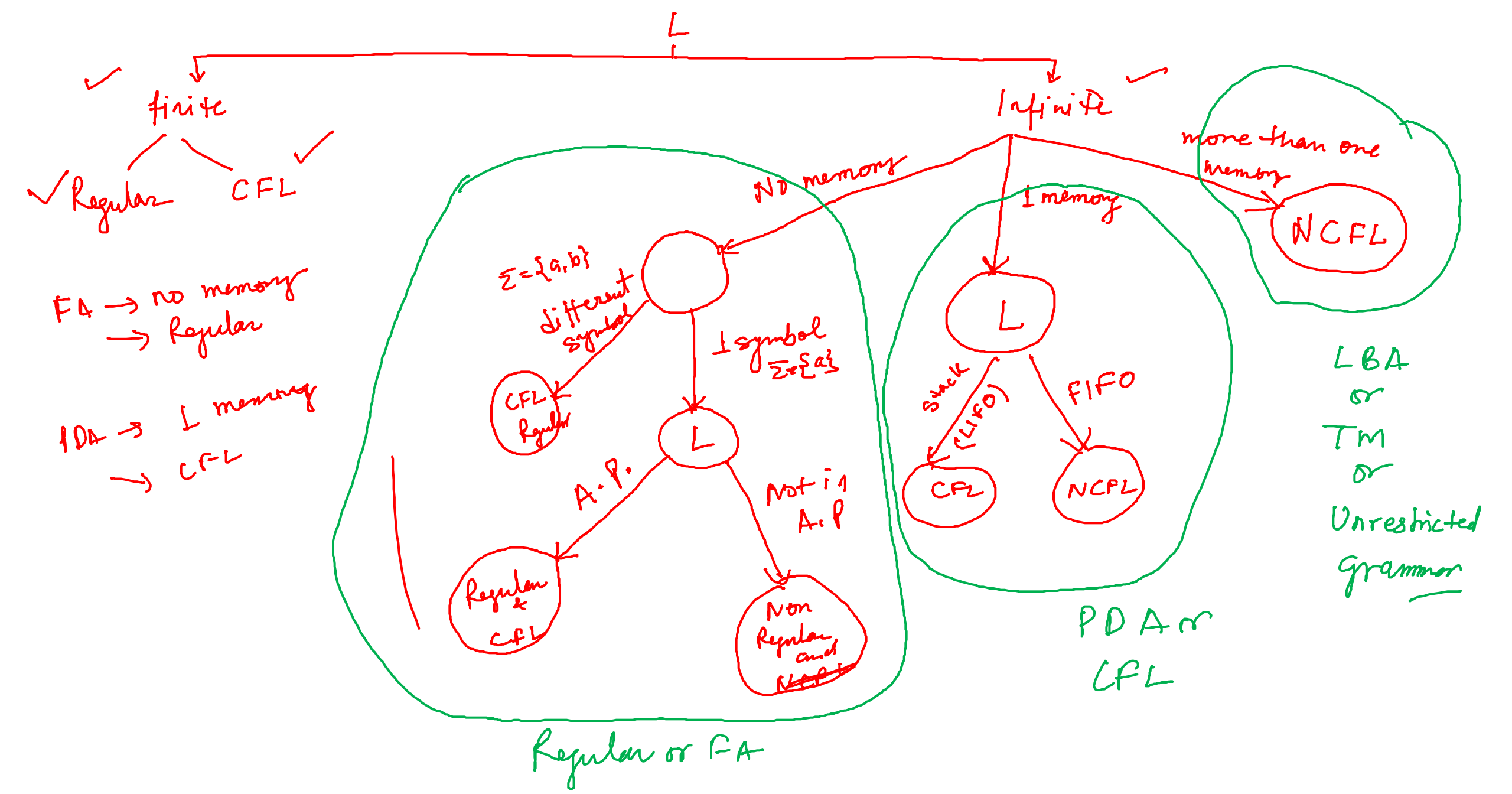
$\forall\ i \geq 1$

$i = 1$ $pqrst \in L$ true ⟹

$i = 2$ $pqqrsst \in L$

$aa\ abb\ ab\ bcc\ ccc \notin L$

L is not CFL

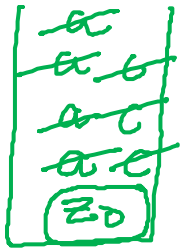

L
finite
Infinite
Regular
CFL
FA → no memory
→ Regular
PDA → 1 memory
→ CFL
NO memory
1 memory
more than one memory
NCFL
Σ={a,b}
different symbol
1 symbol Σ={a}
CFL
Regular
L
A.P.
Not in A.P
Regular & CFL
Non Regular and NCFL
L
stack (LIFO)
FIFO
CFL
NCFL
LBA or TM or Unrestricted grammar
PDA or CFL
Regular or FA

① $L = \{ a^n b^n c^m d^m : n, m \geq 1 \}$ CFL

CFL · CFL

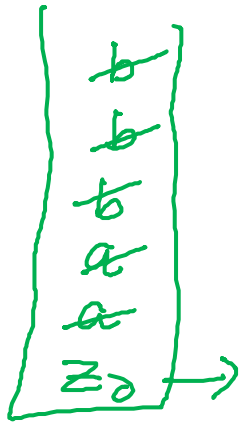
CFL

$\{a^n b^n : n \geq 1\}$ — CFL



$a^n b^n$

② $L = \{ a^n b^m c^m d^n : n, m \geq 1 \}$ CFL



a a b b b c c c d d

push

(3) $L = \{ a^n b^m c^n d^m : n, m \geq 1 \}$ Not CFL or PDA (No)

$m = 3$
$n = 2$

a a b b b c c d d d

b → stack top element b

b
b
a
a
Z_0

(4) $L = \{ w c w^R : w \in (a+b)^+ \}$ ✓ odd length palindrome.

CFL

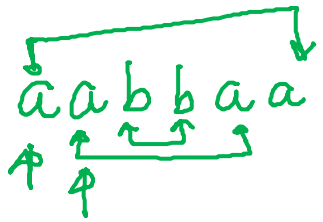
a b a b c b a b a

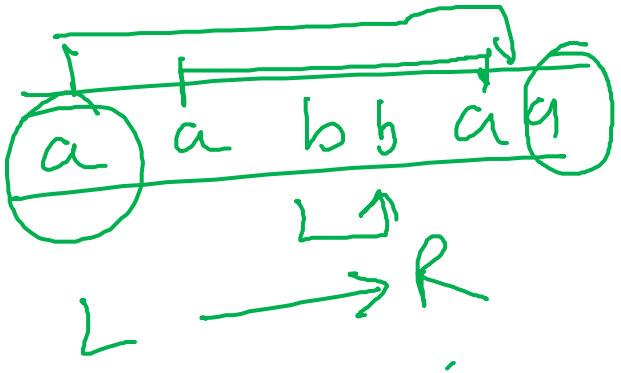
b
a
b
a
Z_0

Z_0

(5) $L = \{ w w^R : w \in (a+b)^+ \}$

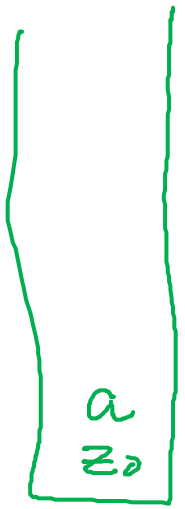
Q → $L = \{ w w^R : w \in (a+b)^+ \}$ Not CFL





aaaabbbb

abaaba

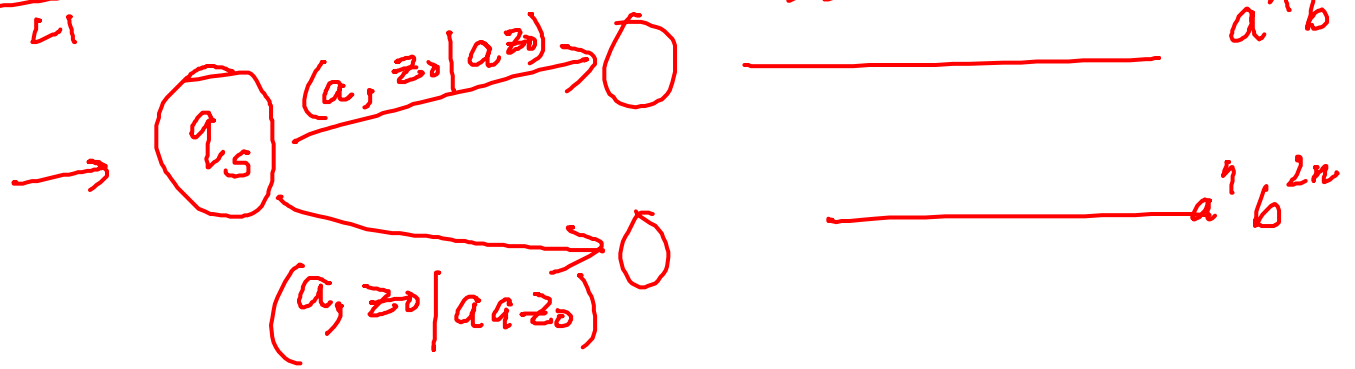


PDA for CFG

CFL ∪ CFL = CFL

Q. $L = \{ a^n b^n : n \geq 1 \} \cup \{ a^n b^{2n} : n \geq 1 \}$ ✓ ✓ ✓

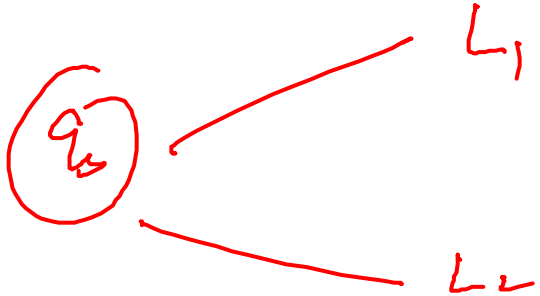
L_1 L_2



Q = $L = \{ a^i b^j c^k d^l : i = k \text{ or } j = l \}$ ⟹ CFL

L_1 L_2

either (a & c are equal) ∪ Or (b & d are equal)



CLOSURE PROPERTIES OF DCFL & CFL

L_1 & L_2 → CFL

1. Union → $L_1 \cup L_2 = CFL$ (Not in DCFL)
2. Concatenation — $L_1 \cdot L_2 = CFL$ (DCFL X)
3. Kleen Star — $L_1^* = CFL$ (& DCFL X)
4. Intersection → $L_1 \cap L_2 \Rightarrow$ Not CFL
5. Intersection with ~~R~~ Regular language

L_3 → Regular lang

$L_1 \cap L_3 \Rightarrow$ CFL ✓

(6) Complement →

CFL are not closed under complement

DCFL " " " "

(7) Different — CFL are not closed under difference

(8) Difference with Regular language

Demorgan's law

$(L_1 - L_3) \Rightarrow$ CFL $\Leftrightarrow$ $L_1 \cap \overline{L_2}$

L_1 → CFL, L_3 → Regular

(9) CFL are closed under reverse

$L_1^R \Rightarrow$ CFL

$(L_2 - L_4) - (\overline{L_3} - \overline{L_4})$

CFL − Regular ⇒ CFL

⇒ L_1 and L_2 ⇒ CFL

L_3 & L_4 ⇒ Regular Language

CFL Regular

$(L_1 - L_2) - (L_3 - L_4)$

CFL − CFL

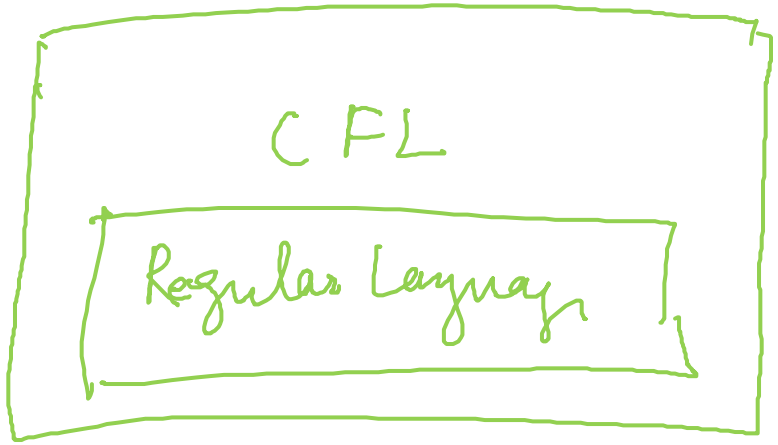
⇒ May or may not be CFL

may or may not be CFL

⇒ $(L_1 \cap L_3) \cup (\overline{L_2} \cup L_4)$

CFL ∪ (RL or ~~CFL~~ or ReL or Recu) ⇒ May or may not be CFL

Hierarchy of Languages (as so far)



Every Regular language is CFL but CFL is not Regular language