Equivalence of CFG and PDA.
(From CFG to PDA)

Theorm: A language is context Free if some pushdown Automata recognizes it.

proof: part1: given CFG, show how to construct PDA That recognize it.

part2: given PDA, show how to construct CFG

given grammer $S \to BS \mid A$

$A \to 0A \mid \epsilon$

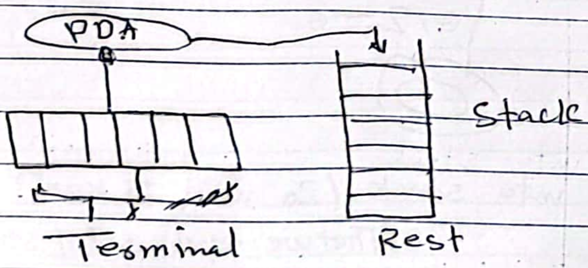
$B \to BB1 \mid 2$

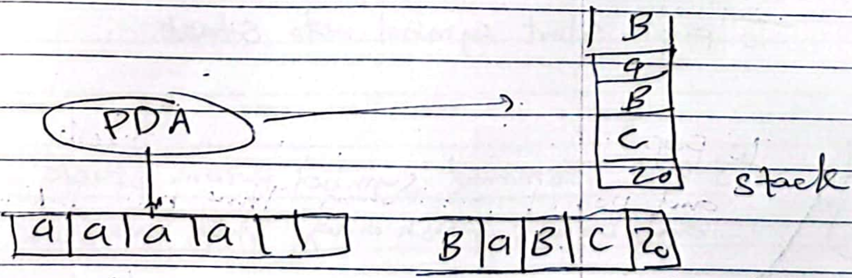
Find left most Derivation

$\to S \to BS \to BB1S \to 2B1S \to 221S$

$\to 221A \to 221\epsilon \to 221$

we can write general form : $\underbrace{aaa}_{\text{Termnl}}\ \underbrace{BaBC}_{\text{Rest}}$





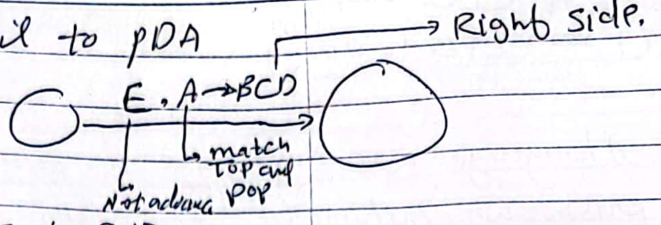
ex Rule $B \to ASAxBA \longrightarrow$ aaaa A | S | A | x | B | A | a | B | C

- match stack Top to rule
- pop stack
- push Right hand of rule onto stack

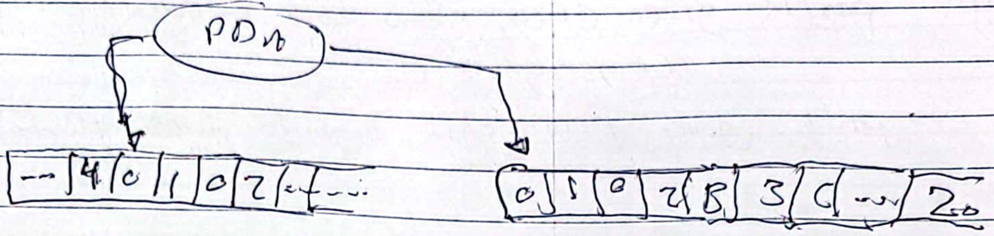
Z1

If You have rule of A → AB BCD

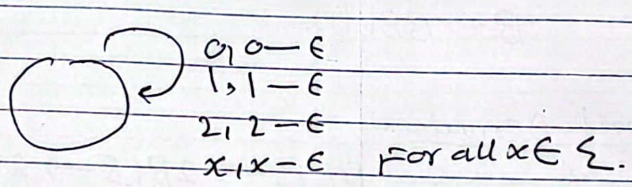
Then add it to PDA

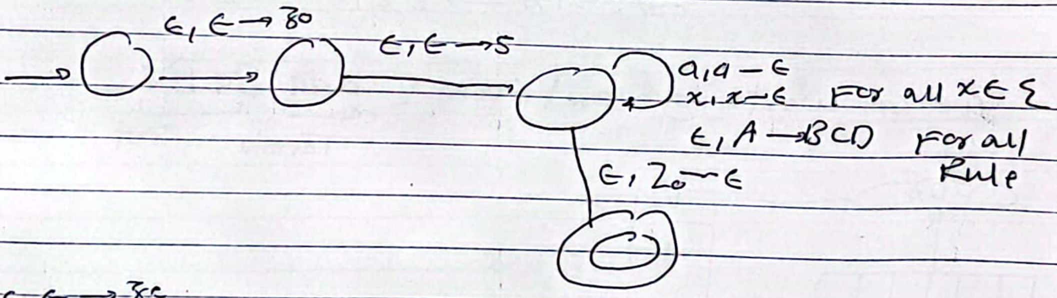


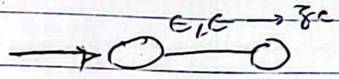
Rule → A → 0 1 0 2B 3 C



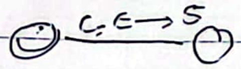
match Terminal symbols to stack



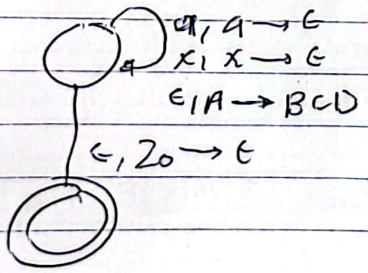




To push Z0 into stack (Z0 used To know That we reach end of stack



To push start symbol into stack.



To pop Terminal symbol From stack and Do not push any Thing into stack

You have To Divide it into multiple state Here we uses single Just to make it simple.

Convert PDA to CFG

given PDA → Construct/build CFG by Following :

1) Simplifed PDA

2) bulid CFG.