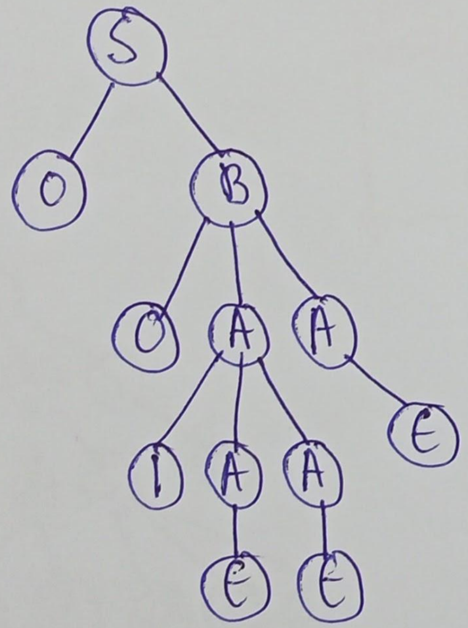
Derivation Tree, Ambiguous Grammar and PDA

Derivation Tree

⇒ A Derivation Tree or Parse Tree is an ordered rooted tree that graphically represents the semantic information of strings derived from a CFG.

Example: For the Grammar $G = \{V, T, P, S\}$ where

$S \rightarrow 0B,\ A \rightarrow 1AA \mid \epsilon,\ B \rightarrow 0AA$



Rootvertex: Must be labelled by the Start symbol

vertex: Labelled by Non-Terminal Symbols

Leaves: Labelled by Terminal Symbols or ϵ

Derivation Tree

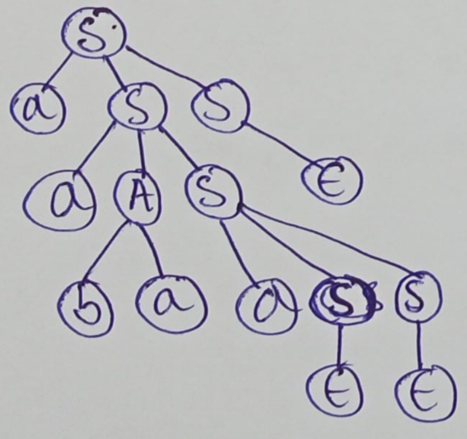
Left Derivation Tree

A Left Derivation Tree is obtained by applying production to the leftmost variable in each step.

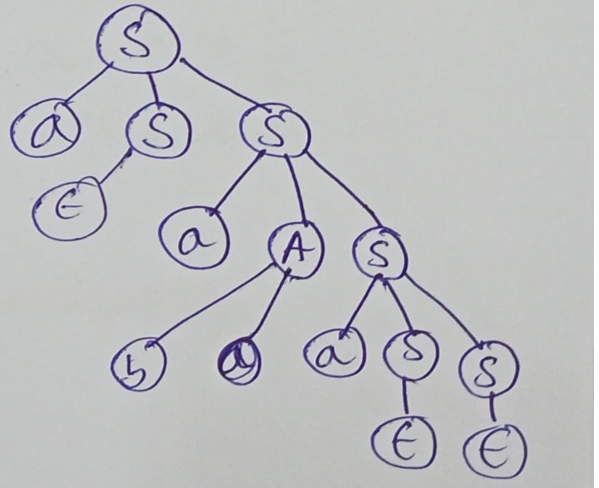
Right Derivation Tree

A Right Derivation Tree is obtained by applying production to the rightmost variable in each step.

Eg. For generating the string aabaa from the Grammar
$S \to aAS \mid aSS \mid \epsilon, A \to SbA \mid ba$



aabaa



aabaa

Ambiguous Grammar

⇒ A Grammar is said to be Ambiguous if there exists two more derivation tree for a string w (that means two or more left derivation trees)

Example: $G = (\{S\}, \{a+b, +, *\}, P, S\}$ where P consists of $S \rightarrow S+S \mid S*S \mid a \mid b$. The String $a+a*b$ can be generated as:

$$
\begin{aligned}
S &\rightarrow \underline{S}+S \\
&\rightarrow a+\underline{S} \\
&\rightarrow a+\underline{S}*S \\
&\rightarrow a+a*\underline{S} \\
&\rightarrow a+a*b
\end{aligned}
$$

$$
\begin{aligned}
S &\rightarrow \underline{S}*S \\
&\rightarrow \underline{S}+S*S \\
&\rightarrow a+\underline{S}*S \\
&\rightarrow a+a*\underline{S} \\
&\rightarrow a+a*b
\end{aligned}
$$

Thus, this Grammar is Ambiguous

Pushdown Automata (Introduction)

⇒ A pushdown Automata (PDA) is a way to implement a CFG is a similar way we design FA for Regular Grammar

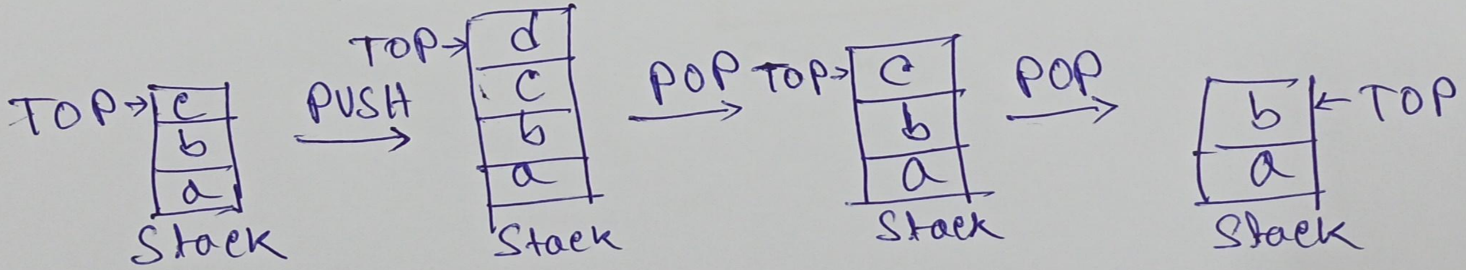
- → It is more powerful tha FSM
- → FSM has a very limited memory but PDA has more memory
- → PDA = Finite State Machine + A Stack

⇒ A Stack is a way we arrange elements one on top of another.

⇒ A stack does two basic operations:

PUSH: A new element is added at the Top of the stack.

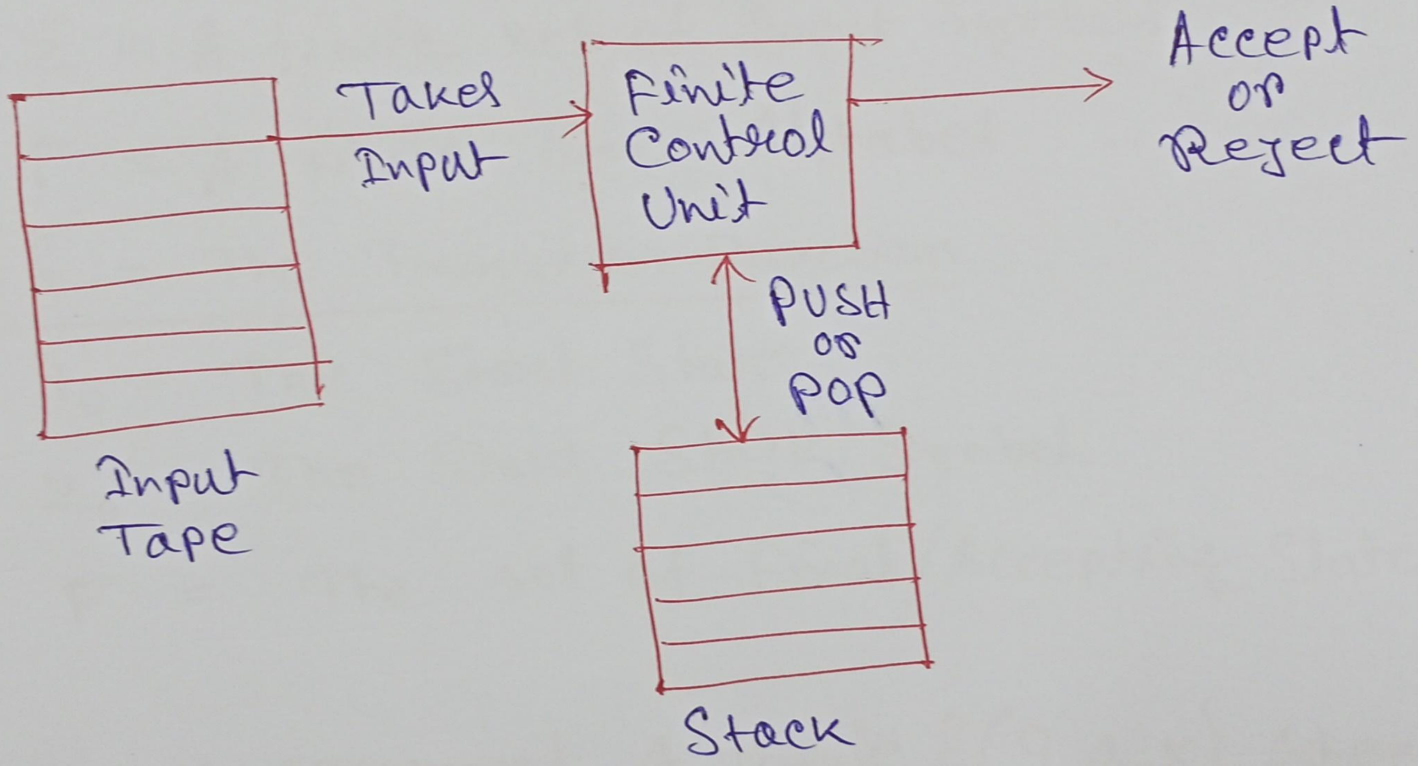
POP: The Top element of the stack is read or removed



Pushdown Automata (Introduction)

⇒ A pushdown Automata has 3 Components

1) An input tape
2) A finite Control Unit
3) A Stack with infinite size



Pushdown Automata (Formal Definition)

⇒ A pushdown Automata is formally defined by 7 Tuples as shown below:

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

where

Q = A finite set of States

Σ = A finite set of Input Symbols

Γ = A finite Stack Alphabet

δ = The Transition Function

q_0 = The Start State

Z_0 = The Start Stack Symbol

F = The set of Final/Accepting States

→ δ takes as argument a triple $\delta(q, a, X)$ where:

i) q is a State in Q

ii) a is either an Input Symbol in Σ or $a = \epsilon$

iii) X is a Stack Symbol, that is a member of Γ

⇒ The output of δ is finite set of pairs (P, Y) where

→ P is a new State

→ Y is a string of stack Symbols that replaces X at the Top of the Stack.

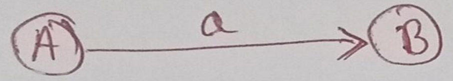
Eg. If $Y = \epsilon$ then the stack is popped.

If $Y = X$ then the stack is unchanged.

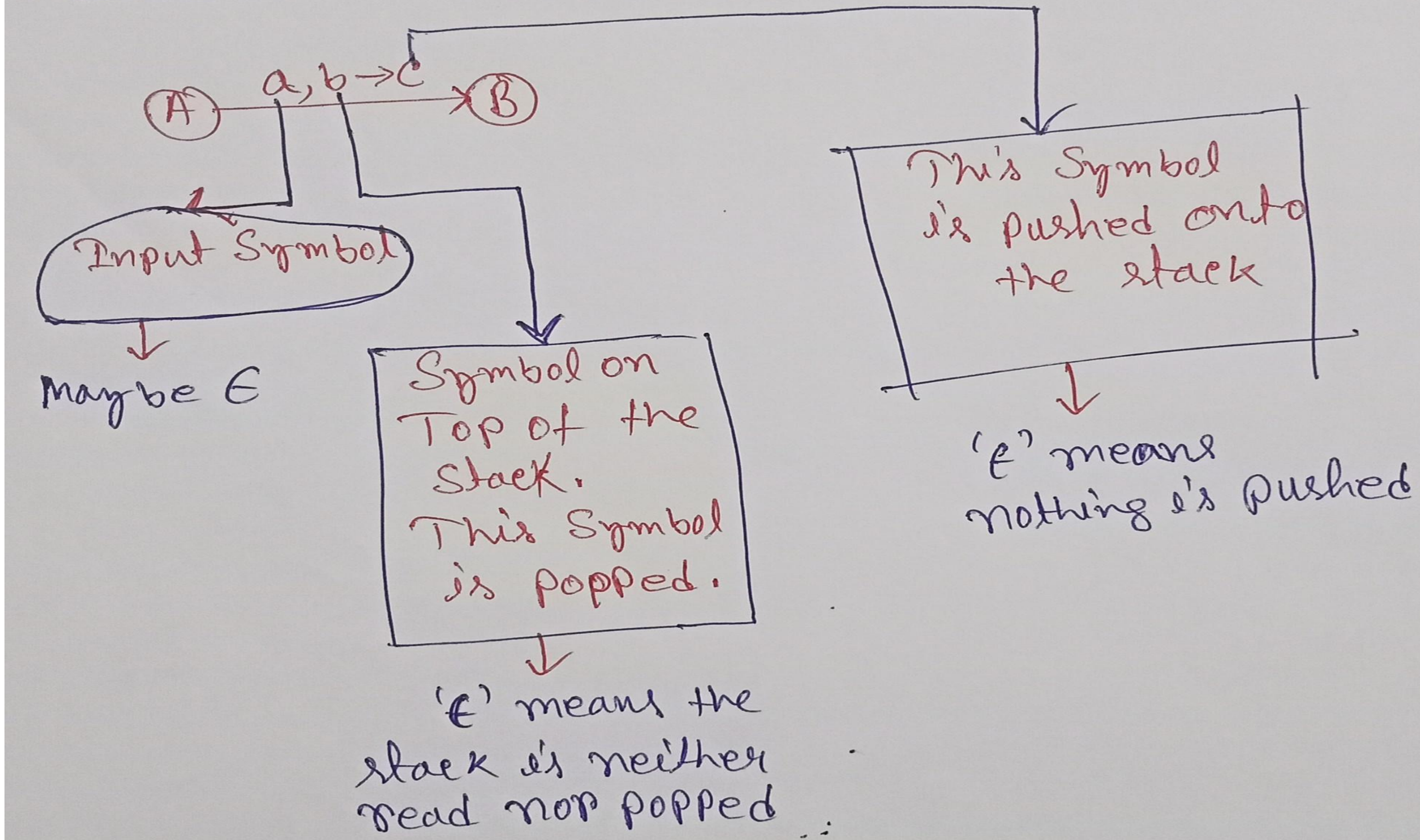
If $Y = YZ$ then X is replaced by Z and Y is pushed onto the stack.

Pushdown Automata (Graphical Notation)

Finite State Machine



Pushdown Automata



Input Symbol

Maybe ∈

Symbol on Top of the Stack. This Symbol is popped.

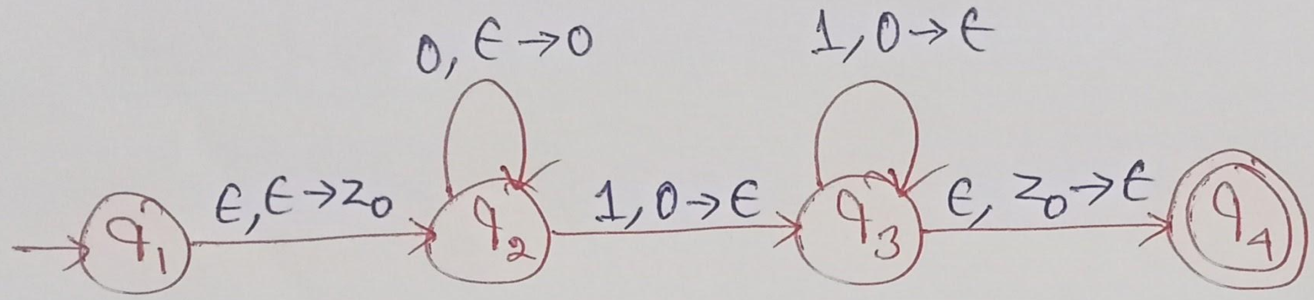
'∈' means the stack is neither read nor popped

This Symbol is pushed onto the stack

'∈' means nothing is pushed

Pushdown Automata - Example

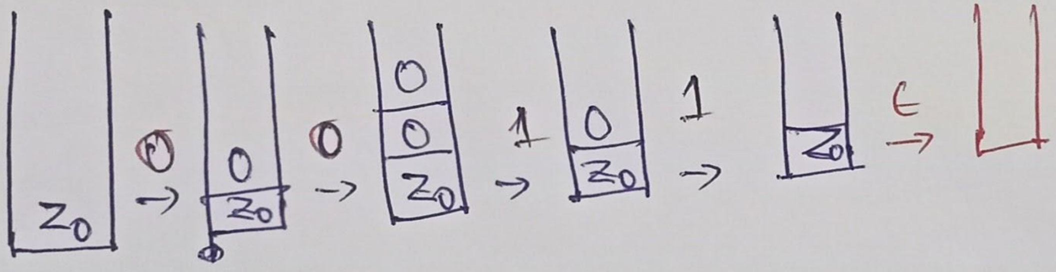
Example: Construct a PDA that accepts $L = \{0^n 1^n \mid n \geq 0\}$



← NPDA

0011 - ✓

$/



Accepted cases → ① Reach the Final state
→ ② Stack is empty.

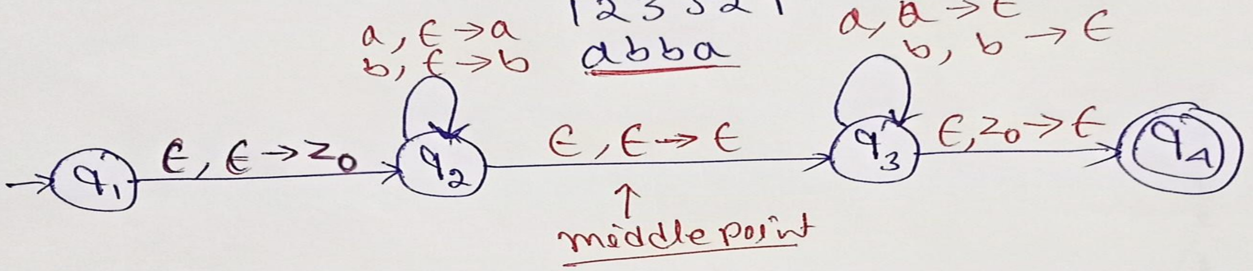
Pushdown Automata - Example (Even palindrome)

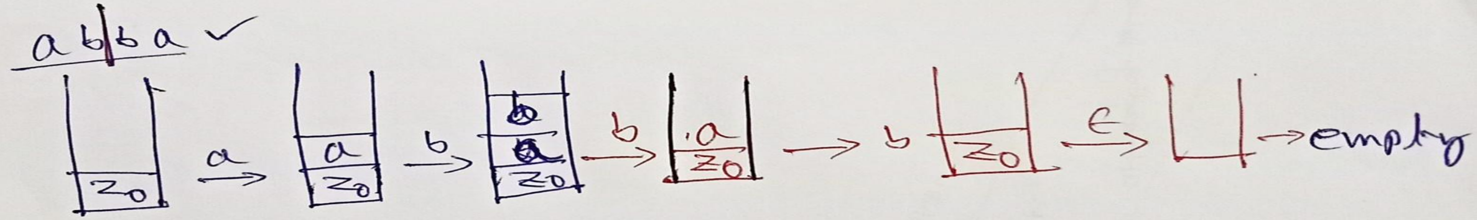
Construct a PDA that accepts Even Palindromes of the form

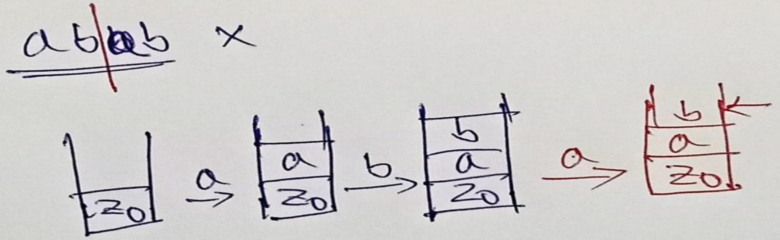
$$L = \{ WW^R \mid w = (a+b)^+ \}$$

PALINDROMES: A word or sequence that reads the same backwards as forwards

Example : NOON
123321
abba





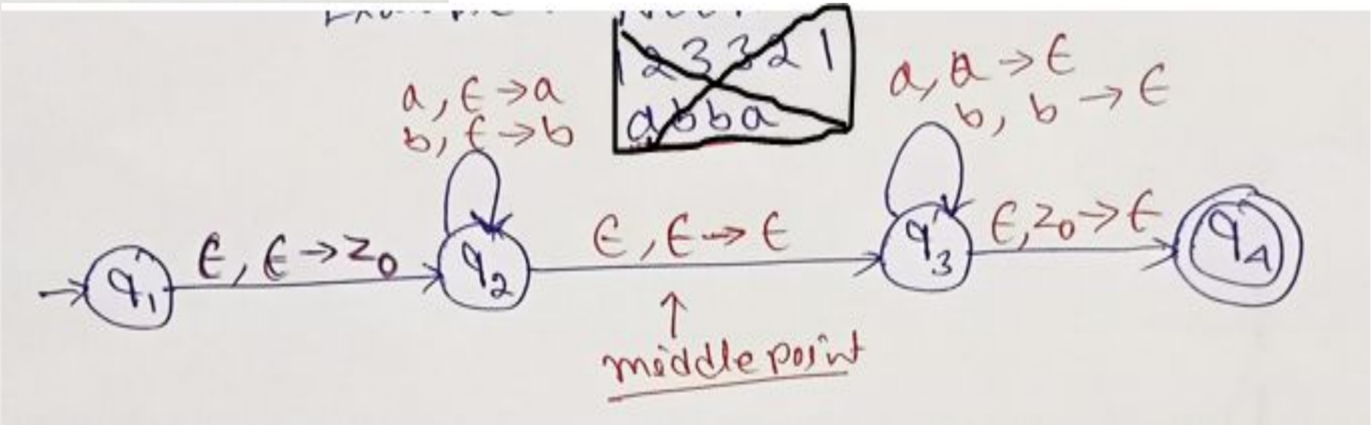


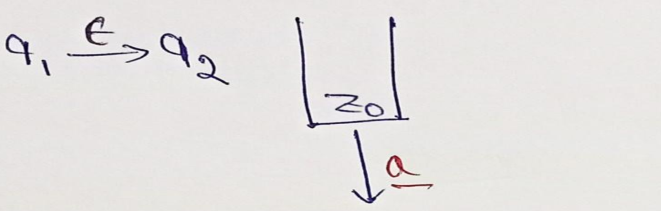
doubt?

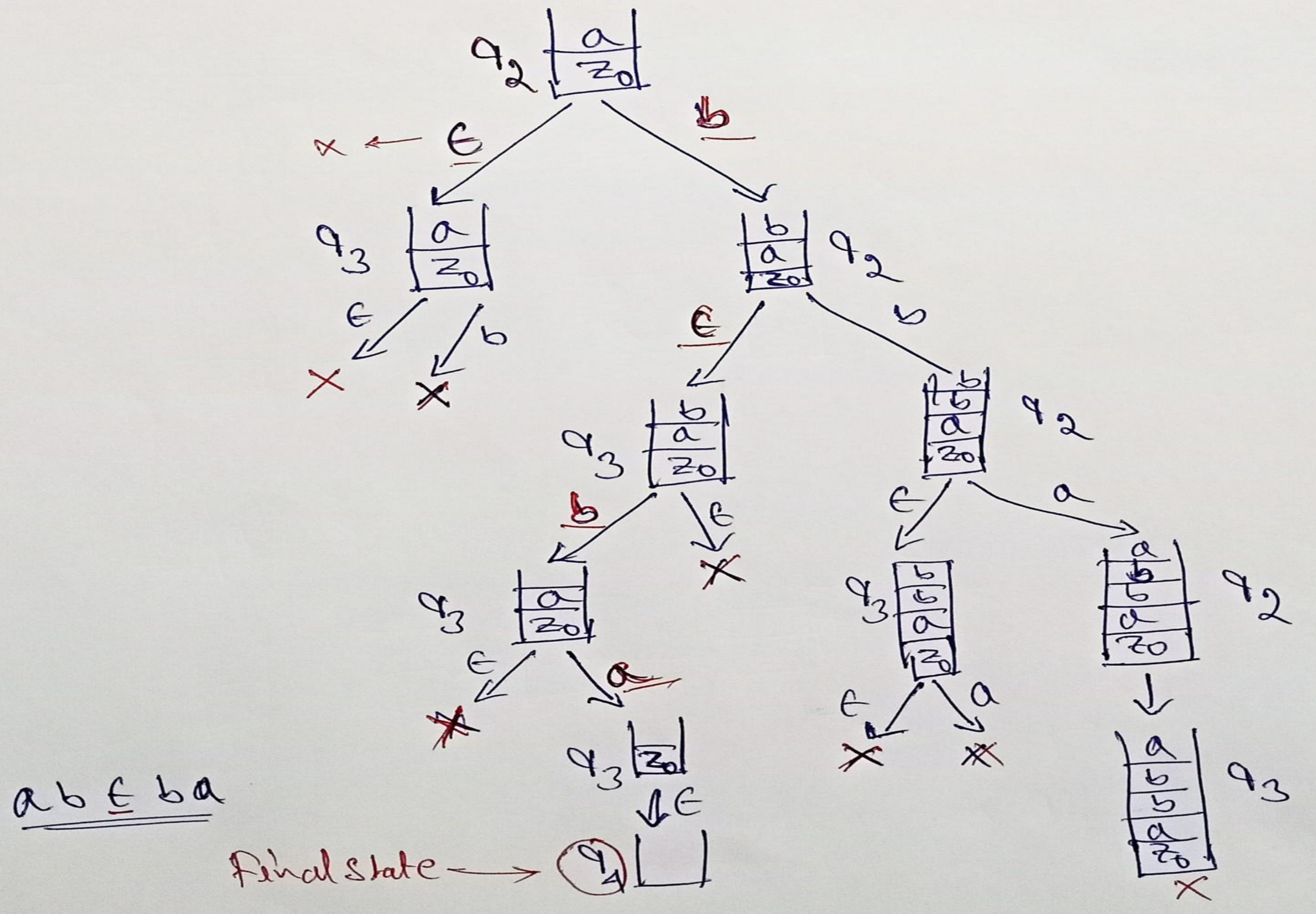
⇒ How do we know the the mid point of the string?

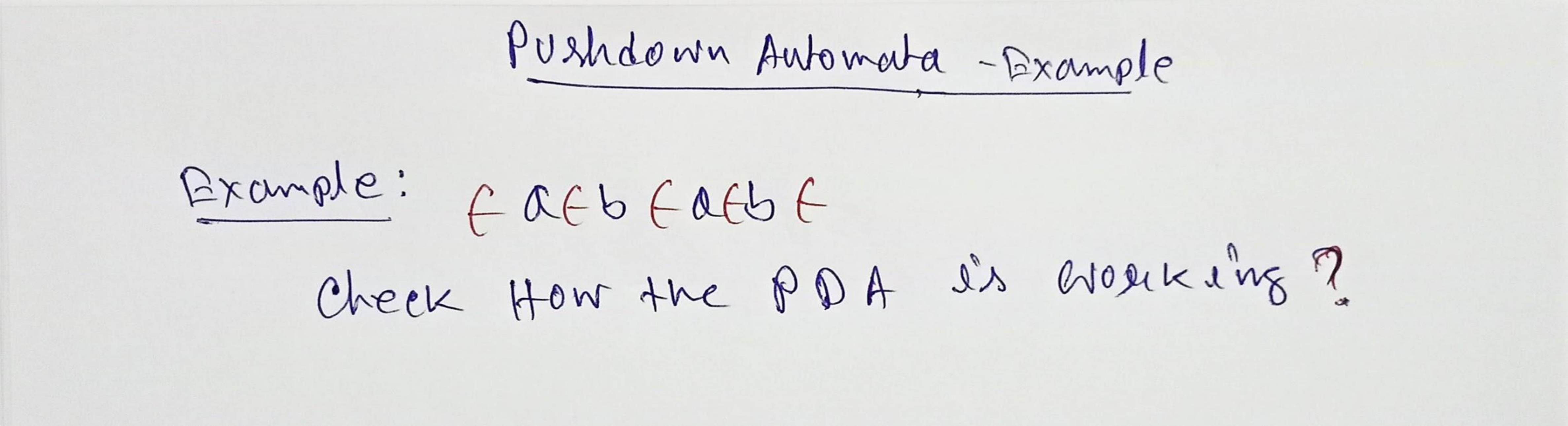
PDA - Example (Even palindrome)

Example: ε a ε b ε b ε a ε









Pushdown Automata - Example

Example: $\epsilon a \epsilon b \epsilon a \epsilon b \epsilon$

Check How the PDA is working?