# Working of PDA

A pushdown automaton is a way to implement a context-free grammar in a similar way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.

Basically a pushdown automaton is −

**"Finite state machine" + "a stack"**

**A pushdown automaton has three components −**

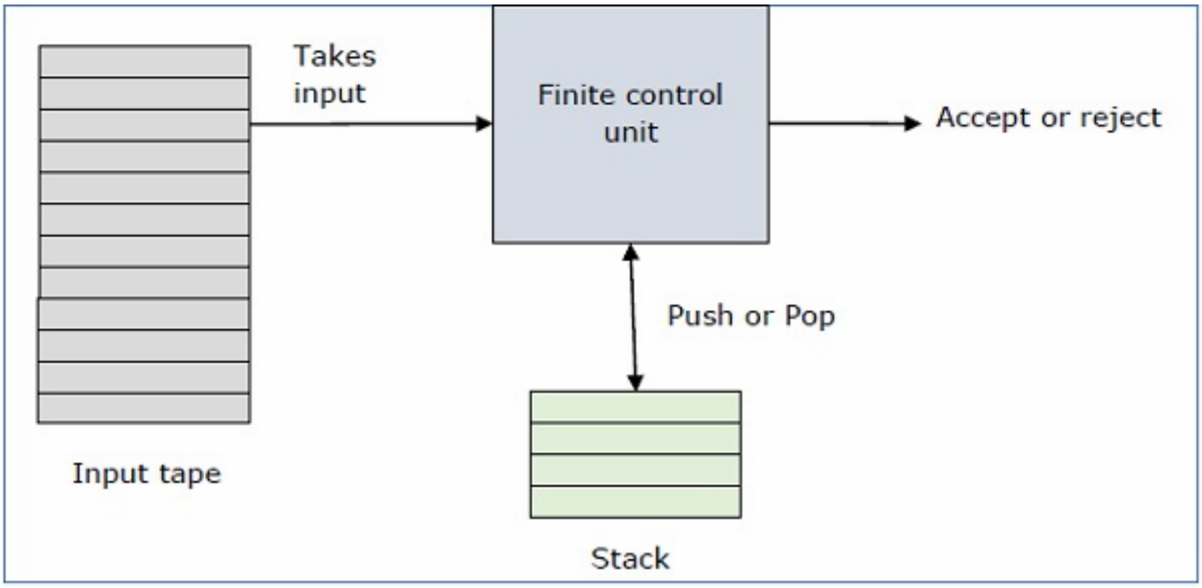
- **an input tape,**
- **a control unit, and**
- **a stack with infinite size.**

The stack head scans the top symbol of the stack.

A stack does two operations −

- **Push** − a new symbol is added at the top.
- **Pop** − the top symbol is read and removed.

A PDA may or may not read an input symbol, but it has to read the top of the stack in every transition.




## 1. Input Tape:

- The input tape is similar to the one used in a Turing machine or a finite automaton.
- It consists of a sequence of symbols (the input string), where the PDA reads one symbol at a time from the tape. The tape is finite, and the machine moves over it left to right as it processes the input.
- The input tape represents the data or symbols that the PDA processes in accordance with its states and transitions.

## 2. Control Unit:

- The control unit is responsible for managing the state transitions of the PDA.
- It functions like the finite control in a finite automaton. At any point, the control unit is in a particular state, and based on the current input symbol and the top symbol on

the stack, it determines which state to transition to next, whether it should read the next symbol, and if any symbols need to be pushed to or popped from the stack.

- The control unit processes the input symbol, determines the next state, and manages the stack as per the transition function.
- The control unit has a finite number of states and uses the stack to store additional information required for recognizing context-free languages.

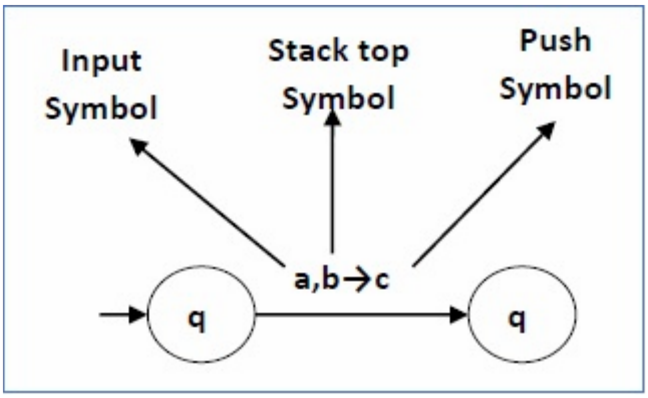
### 3. Stack with Infinite Size:

- The stack is a crucial feature that distinguishes PDAs from finite automata. It is used to store symbols that the PDA needs for its computation.
- The stack allows the PDA to "remember" information and can grow or shrink as needed during the computation. This is why PDAs can recognize context-free languages, which require more memory than what finite state machines (DFAs) can provide.
- The stack operates on a Last-In, First-Out (LIFO) principle. This means that the last symbol pushed onto the stack is the first one to be popped off.
- The stack has the theoretical ability to hold an infinite number of symbols, meaning the PDA can perform an unbounded number of push and pop operations, allowing it to handle complex language structures.
- The stack is used for managing recursive structures, such as matching parentheses or nested expressions, which are typical of context-free languages.

A PDA can be formally described as a 7-tuple (Q, ∑, S, δ, $q_0$, I, F) −

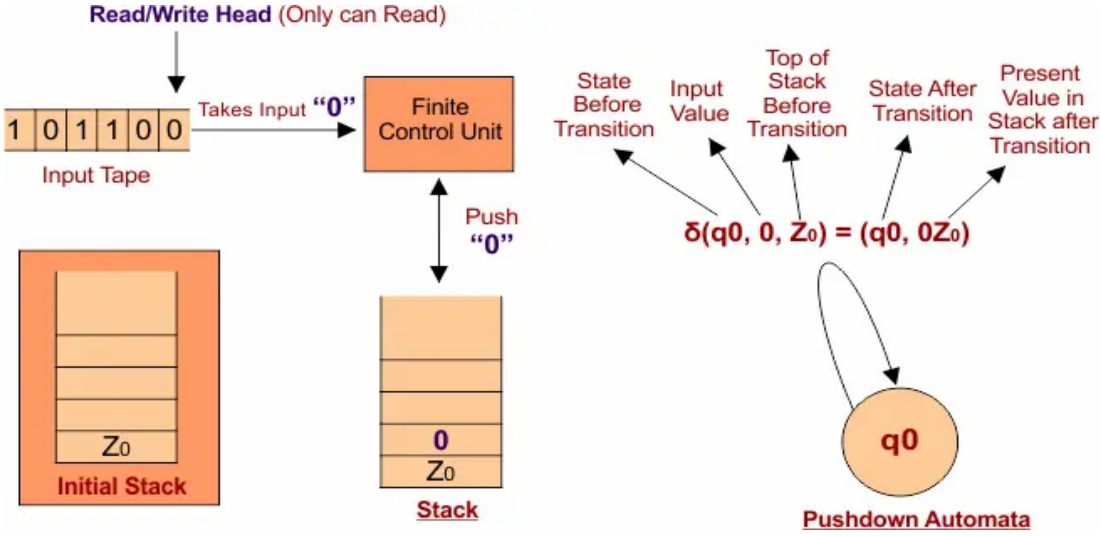
- **Q** is the finite number of states
- ∑ is input alphabet
- **S** is stack symbols
- **δ** is the transition function: Q × (∑ {ε}) × S × Q × S*
- **$q_0$** is the initial state ($q_0$ Q)
- **I** is the initial stack top symbol (I S)
- **F** is a set of accepting states (F Q)

The following diagram shows a transition in a PDA from a state $q_1$ to state $q_2$, labeled as a,b → c −




This means at state **$q_1$**, if we encounter an input string '**a'** and top symbol of the stack is '**b'**, then we pop '**b'**, push '**c'** on top of the stack and move to state **$q_2$**.

**Example:**

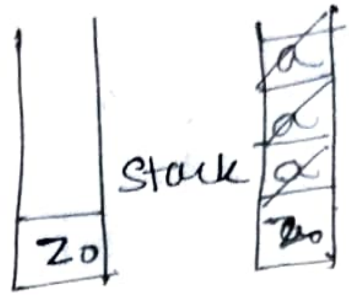

Read/Write Head (Only can Read)
1 0 1 1 0 0
Input Tape
Takes Input "0"
Finite Control Unit
Push "0"
Z0
Initial Stack
0
Z0
Stack
State Before Transition
Input Value
Top of Stack Before Transition
State After Transition
Present Value in Stack after Transition
δ(q0, 0, Z0) = (q0, 0Z0)
q0
Pushdown Automata

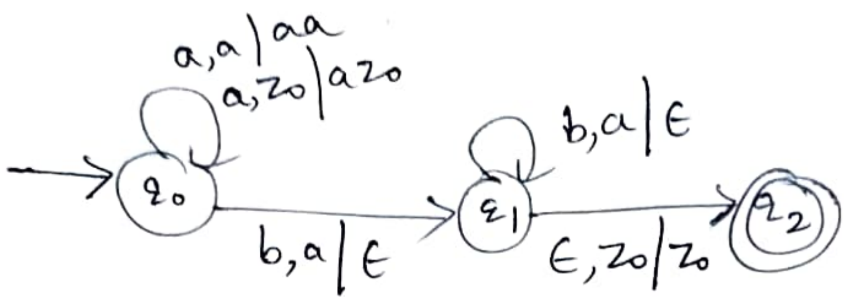
# PDA examples

1) Give a PDA to accept the following language $L = \{a^n b^n \mid n \geq 1\}$

Ex: $n = 3$,

$\underbrace{aaa}_{\text{push}}\underbrace{bbb}_{\text{pop}}\ \epsilon$




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

$Q = \{q_0, q_1, q_2\}$

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

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

$\delta$: $\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$

$q_0 \in Q$ is the start state

$Z_0 \in \Gamma$ is the initial symbol on the stack

$F = \{q_2\}$ — final state

Initial ID

$(q_0, aaabbb, Z_0) \vdash (q_0, aabbb, aZ_0)$
$\vdash (q_0, abbb, aaZ_0)$
$\vdash (q_0, bbb, aaaZ_0)$
$\vdash (q_1, bb, aaZ_0)$
$\vdash (q_1, b, aZ_0)$
$\vdash (q_1, \epsilon, Z_0)$
$\vdash (q_2, \epsilon, Z_0)$.

accepted

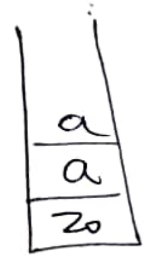
To reject the string

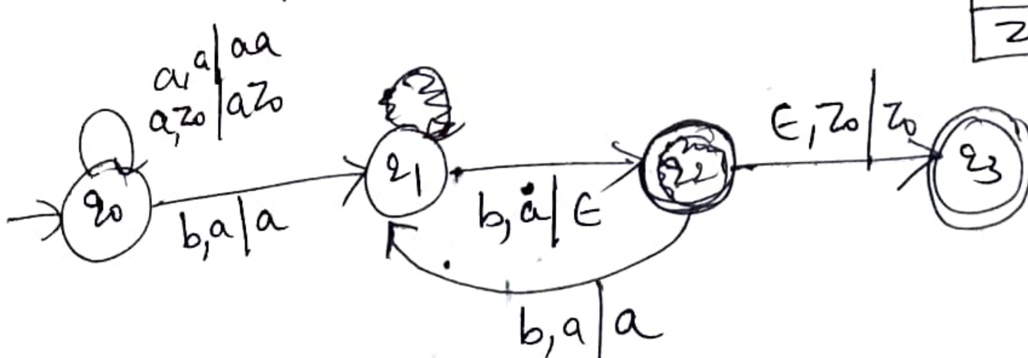
$(q_0, aabbb, Z_0) \vdash (q_0, abbb, aZ_0)$
$\vdash (q_0, bbb, aaZ_0)$
$\vdash (q_1, bb, aZ_0)$
$\vdash (q_1, b, Z_0)$

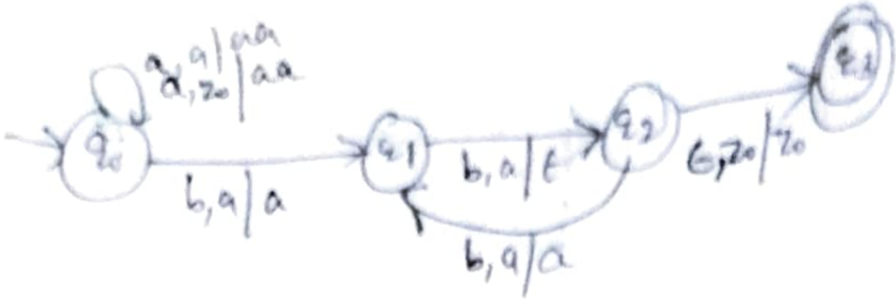
rejected

2] $L = \{a^n b^{2n} \mid n \geq 1\}$

Ex: aabbbb
push skip pop skip pop






$Q = \{q_0, q_1, q_2, q_3\}$

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

$\Gamma = \{z_0, a\}$

$\delta: \delta(q_0, a, z_0) = (q_0, az_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, z_0) = (q_3, z_0)$

$q_0$ : start state

$z_0$ : initial symbol on the stack

$F$ : $q_3$ — final state

<u>initial ID</u>

$(q_0, aabbbb, z_0) \vdash (q_0, abbbb, az_0)$

$\vdash (q_0, bbbb, aaz_0)$

$\vdash (q_1, bbb, aaz_0)$

$\vdash (q_2, bb, az_0)$

$\vdash (q_1, b, az_0)$

$\vdash (q_2, \epsilon, z_0)$ accepted.

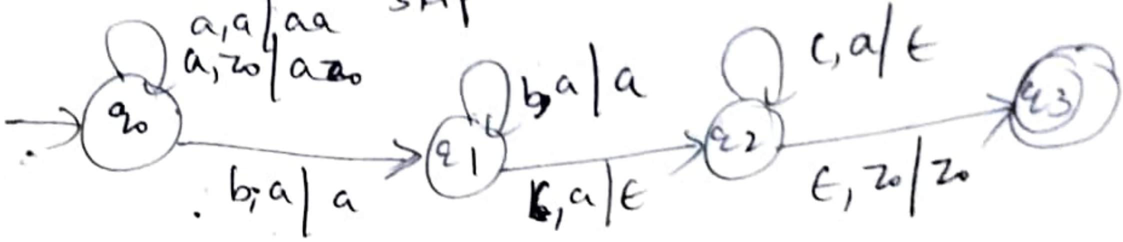
$\vdash (q_3, \epsilon, z_0)$

3) $L = \{a^n b^m c^n \mid n \geq 1, m \geq 1\}$

aabbbcc

push | pop

skip




$Q = \{q_0, q_1, q_2, q_3\}$

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

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

$\delta : \delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_1, a)$

$\delta(q_1, c, a) = (q_2, \epsilon)$

$\delta(q_2, c, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, Z_0) = (q_3, Z_0)$

$q_0 \rightarrow$ Start state

$Z_0 \rightarrow$ initial symbol on the stack

$F \rightarrow q_3 \rightarrow$ final state.

$(q_0, aabbbcc, Z_0) \vdash (q_0, abbbcc, aZ_0)$

$\vdash (q_0, bbbcc, aaZ_0)$

$\vdash (q_1, bbcc, aaZ_0)$

$\vdash (q_1, bcc, aaZ_0)$

$\vdash (q_1, cc, aaZ_0)$

$\vdash (q_2, c, aZ_0)$

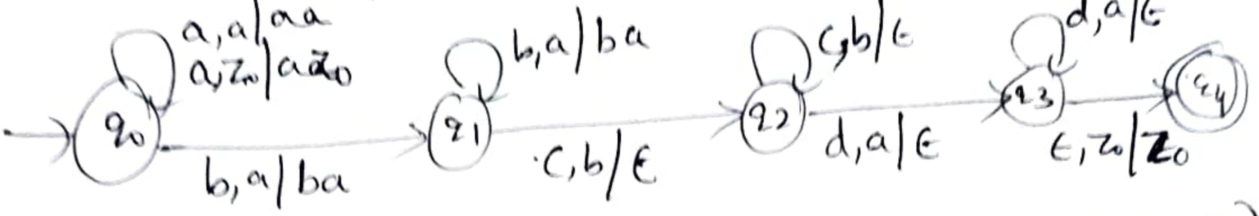
$\vdash (q_2, \epsilon, Z_0)$

$\vdash (q_3, \epsilon, Z_0)$

accepted.

4) $L = \{a^n b^m c^m d^n \mid n \geq 1, m \geq 1\}$

$\underbrace{aa}_{push}\underbrace{bbb}_{push}\underbrace{ccc}_{pop}\underbrace{dd}_{pop}$ $n=2, m=3$




$Q = \{q_0, q_1, q_2, q_3, q_4\}$ $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

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

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

$\delta :$ $\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, ba)$

$\delta(q_1, b, a) = (q_1, ba)$

$\delta(q_1, c, b) = (q_2, \epsilon)$

$\delta(q_2, c, b) = (q_2, \epsilon)$

$\delta(q_2, d, a) = (q_3, \epsilon)$

$\delta(q_3, d, a) = (q_3, \epsilon)$

$\delta(q_3, \epsilon, Z_0) = (q_4, Z_0)$

$q_0 \rightarrow$ start state

$z_0 \rightarrow$ initial symbol on the stack

$F \rightarrow q_4$ - final state

Initial ID

$(q_0, aabbbcccdd, z_0) \vdash (q_0, abbbcccdd, az_0)$

$\vdash (q_0, bbbcccdd, aaz_0)$

$\vdash (q_1, bbcccdd, baaz_0)$

$\vdash (q_1, bcccdd, bbaaz_0)$

$\vdash (q_1, cccdd, bbbaaz_0)$

$\vdash (q_2, ccdd, bbaaz_0)$

$\vdash (q_2, cdd, baaz_0)$

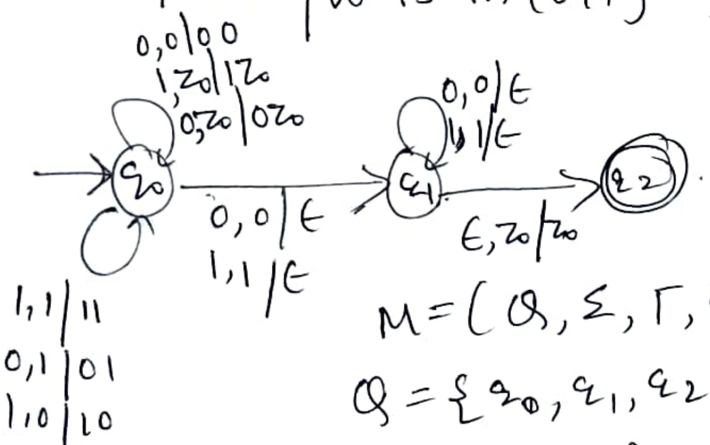
$\vdash (q_2, dd, aaz_0)$

$\vdash (q_3, d, az_0)$

$\vdash (q_3, \epsilon, z_0)$

$\vdash (q_4, \epsilon, z_0)$ accepted

5) $L = \{ww^R \mid w \text{ is in } (0+1)^+\}$




$M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{0, 1\}$

$\Gamma = \{0, 1, z_0\}$

$\delta : \delta(q_0, 0, z_0) = (q_0, 0z_0)$

$\delta(q_0, 1, z_0) = (q_0, 1z_0)$

$\delta(q_0, 0, 0) = (q_0, 00)$

$\delta(q_0, 1, 1) = (q_0, 11)$

$\delta(q_0, 0, 1) = (q_0, 01)$

$\delta(q_0, 1, 0) = (q_0, 10)$

$\delta(q_0, 1, 1) = (q_1, \epsilon)$

$\delta(q_0, 0, 0) = (q_1, \epsilon)$

$\delta(q_1, 1, 1) = (q_1, \epsilon)$

$\delta(q_1, 0, 0) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$ $q_0$ - start state

$z_0$ = initial symbol on the stack

$F = \{q_2\}$

<u>initial ID</u>

$\delta(q_0, abaaba, z_0) = (q_0, baaba, az_0)$

$= (q_0, aaba, baz_0)$

$= (q_0, aba, abaz_0)$

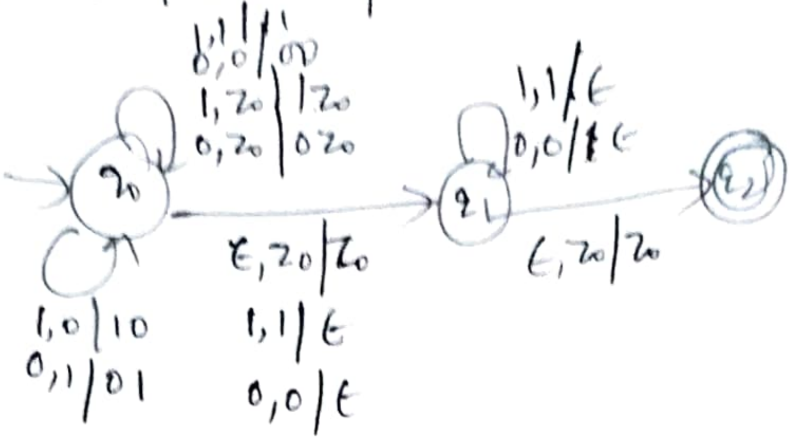
$= (q_0, ba, baz_0)$

$= (q_0, a, az_0)$

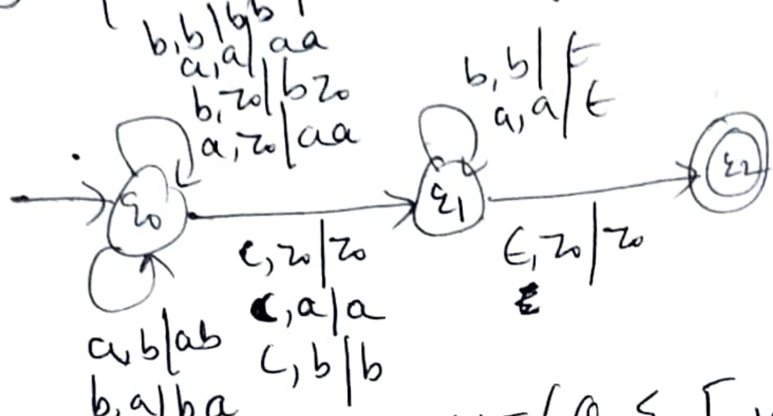
$= (q_1, \epsilon, z_0)$

$= (q_2, \epsilon, z_0)$ accepted.

6) $L = \{ww^R \mid w \text{ is in } (0+1)^*\}$




7) $L = \{wcw^R \mid w \in \{a,b\}^*\}$




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

$Q = \{q_0, q_1, q_2\}$

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

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

$\delta$ : $\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, ab)$
$\delta(q_0, b, a) = (q_0, bb)$
$\delta(q_0, c, a) = (q_1, a)$
$\delta(q_0, c, b) = (q_1, b)$
$\delta(q_0, c, Z_0) = (q_1, Z_0)$

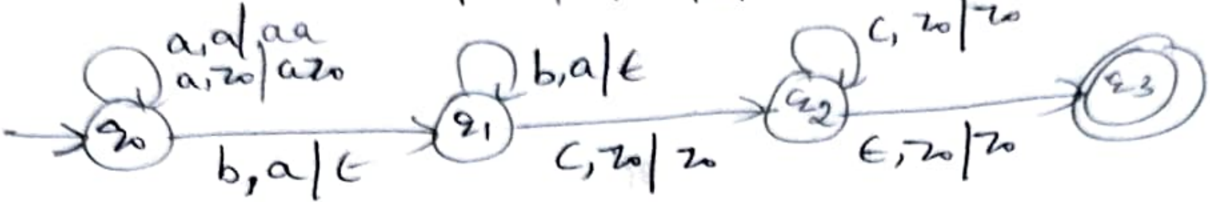
$\delta(q_1, a, a) = (q_1, \epsilon)$
$\delta(q_1, b, b) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$.

$q_0$ - start state
$Z_0$ - initial symbol on stack
F - $q_2$ final state

8) $L = \{a^n b^n c^m \mid n, m \geq 1\}$

$\underline{aa}\,\underline{bb}\,\underline{ccc}$ $n = 2,\ m = 3$

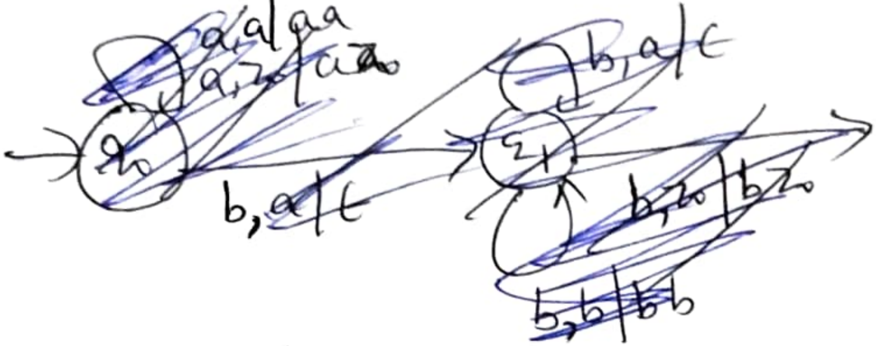
push pop ~~pop~~ skip

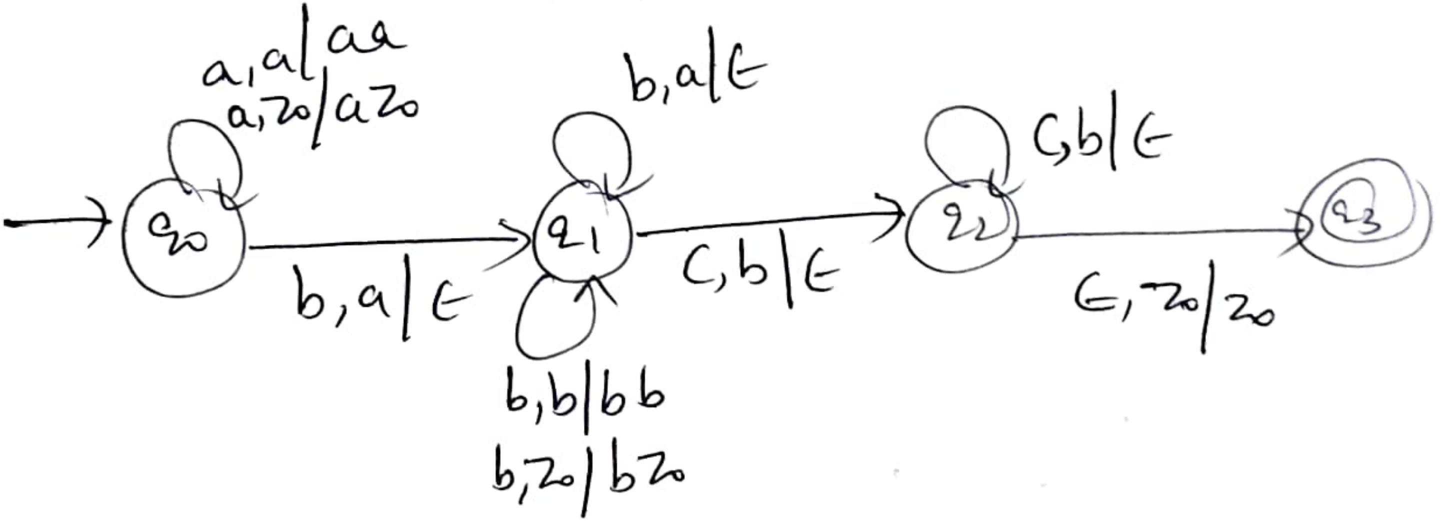


(9) $L = \{a^n b^{m+n} c^m \mid n, m \geq 1\}$

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

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

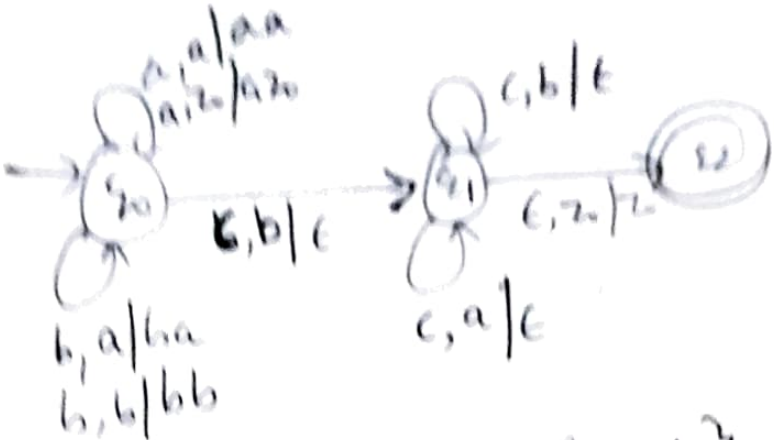




10) $L = \{a^n b^m c^{n+m} \mid n, m \geq 1\}$

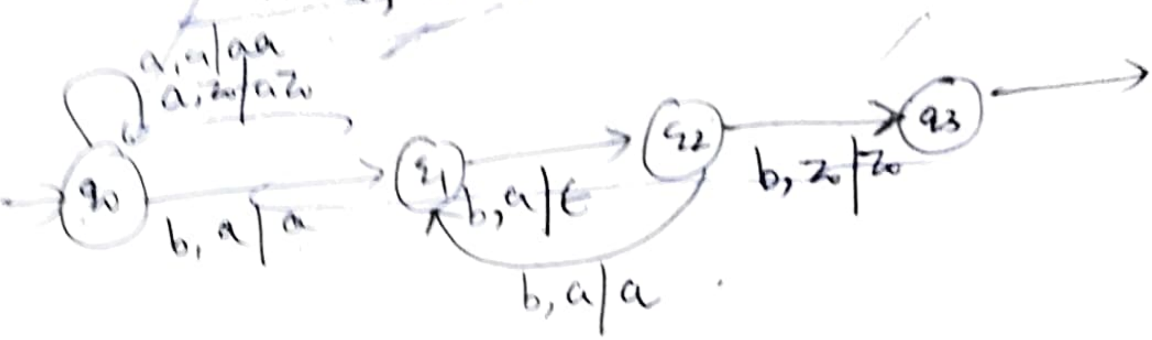
$a^n b^m c^n c^m$

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




10) $L\{a^n b^{n+1} \mid \text{where } n \geq 1\}$

$n = 2$, aabbbb




11) $L = \{a^i b^j c^k \mid i+j=k,\ i, j \geq 0\}$

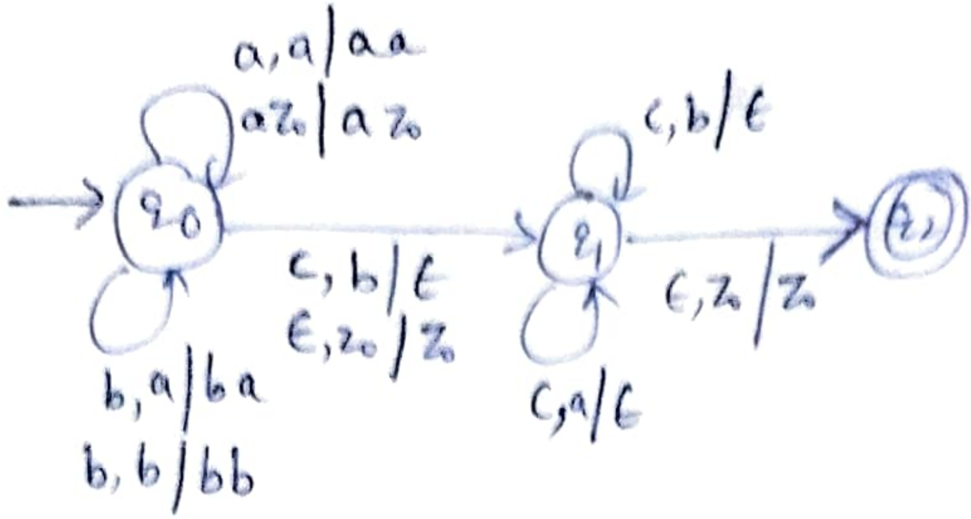
$a^i b^j c^{i+j}$ $\quad (\because k = i+j)$

$a^i b^j c^i c^j$

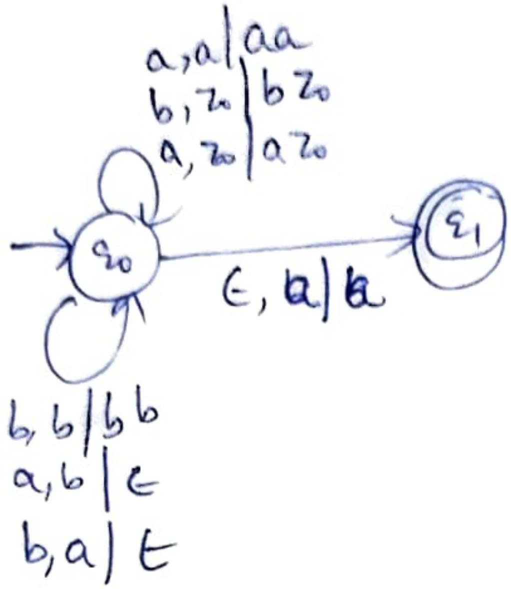
$a^i b^j c^j c^i$

Push Push Pop Pop

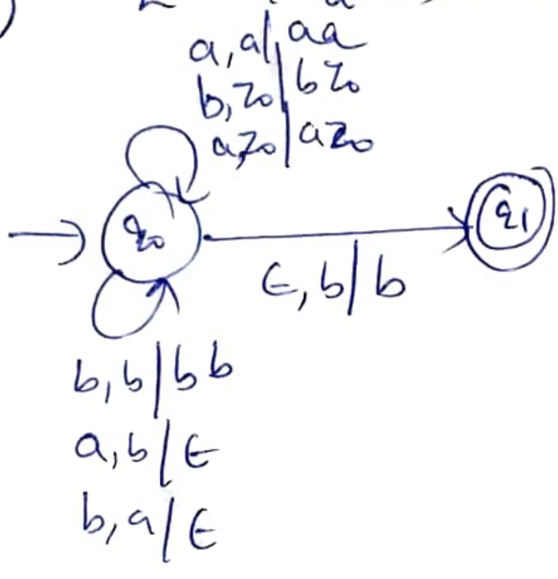
$L = \{\epsilon, abcc, aabccc, \ldots\}$



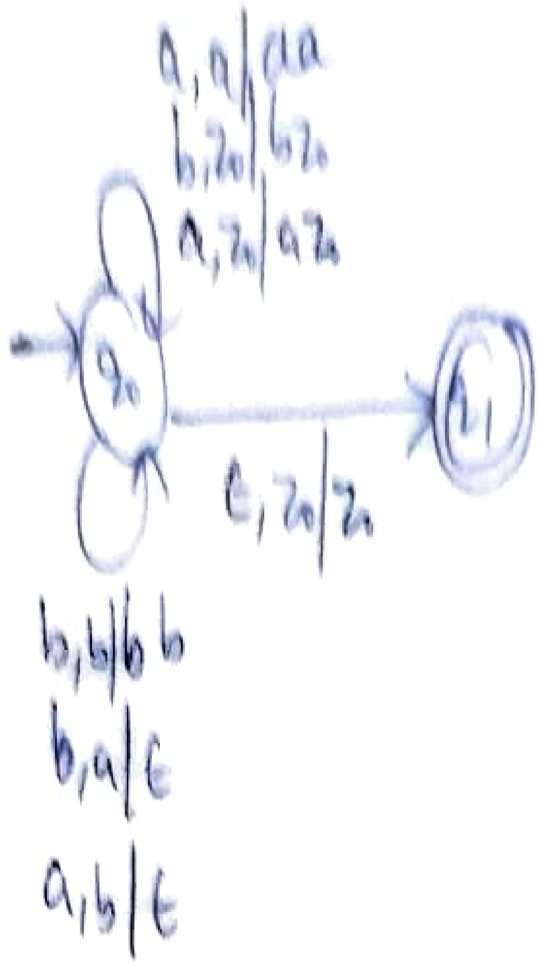

12) $L = \{ n_a(w) > n_b(w) \}$




13) $L = \{ n_a(w) < n_b(w) \}$



19) $L = \{ n_a(w) = n_b(w) \}$



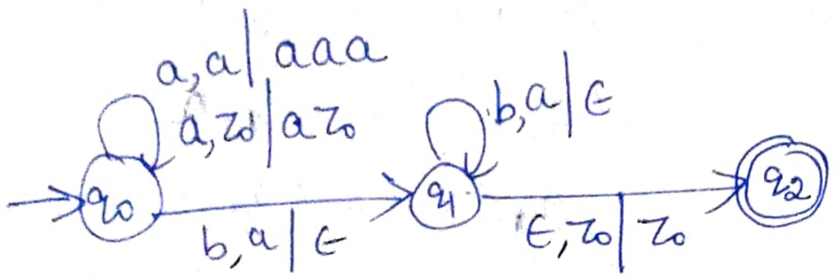
1) Construct PDA for the language

$L = \{a^n b^{2n} \mid n \geq 1\}$

$L = \{abb, aabbbb, \ldots\}$ for 1 a → push 2 a's

| a | a | b | b | b | b | ε |
|---|---|---|---|---|---|---|

~~pushba~~





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

$Q = \{q_0, q_1, q_2\}$

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

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

$q_0 = \{q_0\}$

$Z_0 = \{Z_0\}$

$F = \{q_2\}$

$\delta$: transition function

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aaa)$

$\delta(q_0, b, a) = (q_1, \epsilon)$

$\delta(q_1, b, a) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$

2) Construct PDA for the language

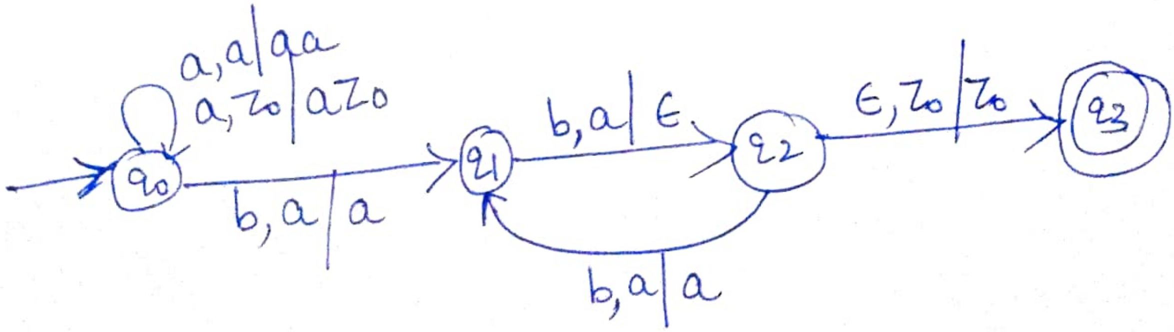
$L = \{a^n b^{2n} \mid n \geq 1\}$

aabbbbε

push skip skip pop pop

$L = \{abb, aabbbb, \ldots\}$

| a | a | b | b | b | b | ε |
|---|---|---|---|---|---|---|



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

$Q = \{q_0, q_1, q_2, q_3\}$

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

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

$q_0 = \{q_0\}$

$Z_0 = \{Z_0\}$

$F = \{q_3\}$

$\delta$ : transition function

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

$\delta(q_0, a, a) = (q_0, aa)$

$\delta(q_0, b, a) = (q_1, a)$

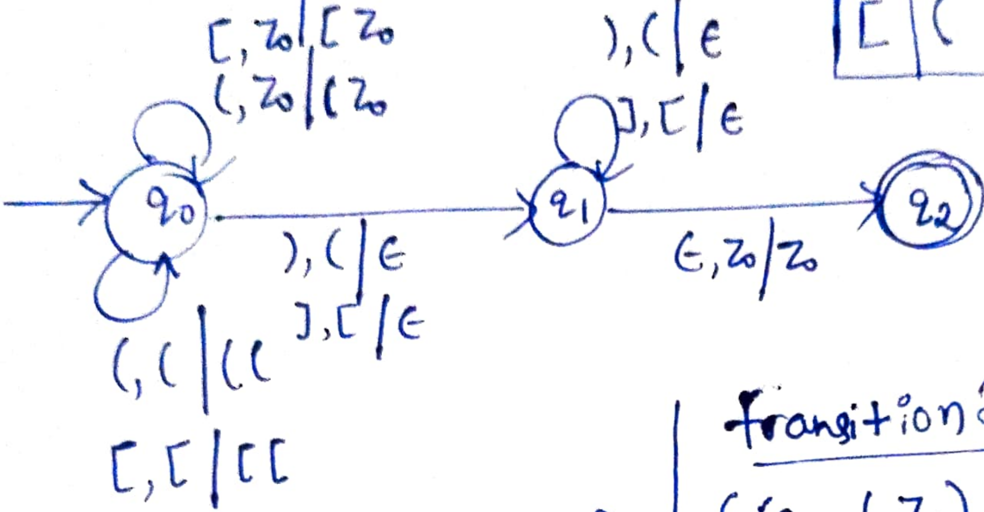
$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, b, a) = (q_1, a)$

$\delta(q_1, b, a) = (q_2, \epsilon)$

$\delta(q_2, \epsilon, Z_0) = (q_3, \epsilon)$

# Design PDA for Balanced parenthesis




| | | | | | | |
|---|---|---|---|---|---|---|
| [ | ( | ( | ) | ) | ] | ϵ |

$(Q, \Sigma, \cancel{\delta}, \delta, q_0, Z_0, F)$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{ (, ), [, ] \}$

$\Gamma = \{ (, [, Z_0 \}$

$q_0 = \{q_0\}$

$Z_0 = \{Z_0\}$

$F = \{q_2\}$

## Transition function

$\delta(q_0, (, Z_0) = (q_0, (Z_0)$

$\delta(q_0, [, Z_0) = (q_0, [Z_0)$

$\delta(q_0, (, () = (q_0, (()$

$\delta(q_0, [, [) = (q_0, [[)$

$\delta(q_0, ), () = (q_1, \epsilon)$

$\delta(q_0, ], [) = (q_1, \epsilon)$

$\delta(q_1, ), () = (q_1, \epsilon)$

$\delta(q_1, ], [) = (q_1, \epsilon)$

$\cancel{\delta}$

$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$.

## CFG to PDA

1)

$S \to aABC$
$A \to aB \mid a$
$B \to bA \mid b$
$C \to a$

first we have to check, whether the given grammar is in GNF.

Sol:

① $\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$ This grammar is in GNF

② 

| | |
|---|---|
| $S \to aABC$ | $\delta(q_1, a, S) = (q_1, ABC)$ |
| $A \to aB$ | $\delta(q_1, a, A) = (q_1, B)$ |
| $A \to a$ | $\delta(q_1, a, A) = (q_1, \epsilon)$ |
| $B \to bA$ | $\delta(q_1, b, B) = (q_1, A)$ |
| $C \to a$ | $\delta(q_1, a, C) = (q_1, \epsilon)$ |
| $B \to b$ | $\delta(q_1, b, B) = (q_1, \epsilon)$ |
| | $\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$ |

final state

$S \to aABC$

$\delta(q_1, a, S) = (q_1, ABC$

Derive the String,

$S \Rightarrow aABC$
$\Rightarrow aaBC \quad (\because A \to a)$
$\Rightarrow aabC \quad (\because B \to b)$
$\Rightarrow aaba \quad (\because C \to a)$

Instantaneous description

$\delta(q_0, aaba, Z_0) \vdash (q_1, aaba, SZ_0)$ $(\because \delta(q_0, \epsilon, Z_0) = (q_1, SZ_0))$

$\epsilon aaba, Z_0$

$\vdash (q_1, aba, ABCZ_0)$ $(\because \delta(q_1, a, S) = (q_1, ABC))$

'a' is removed 'S' is removed and replaced with ABC

$\vdash (q_1, ba, BCZ_0)$ $(\because \delta(q_1, a, A) = (q_1, \epsilon))$

'a' is removed 'A' is removed and replaced with $\epsilon$.

$\vdash (q_1, a, CZ_0)$ $(\because \delta(q_1, b, B) = (q_1, \epsilon))$

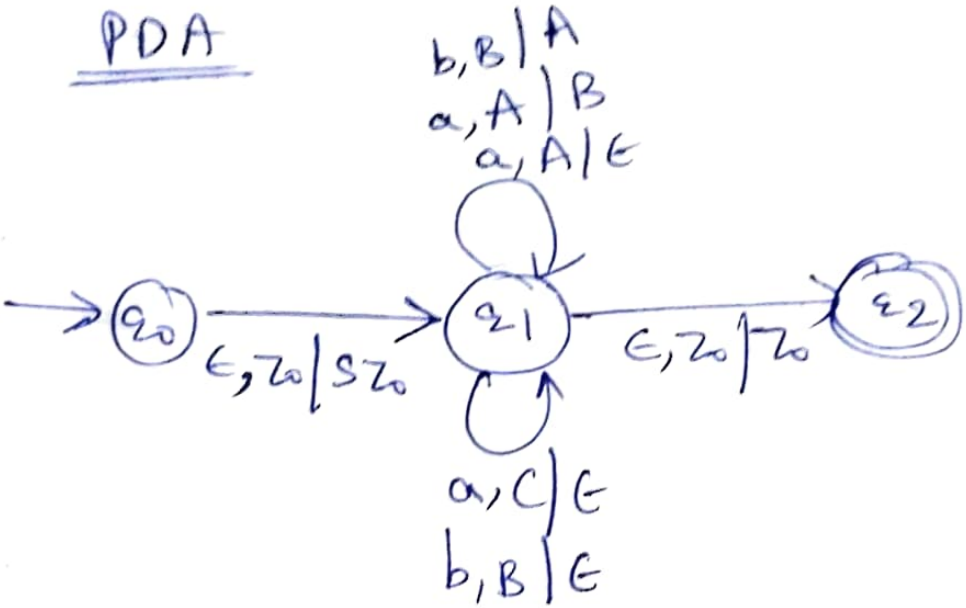
'b' is removed 'B' is removed and replaced with $\epsilon$

$\vdash (q_1, \epsilon, Z_0)$ $(\because \delta(q_1, a, C) = (q_1, \epsilon))$

'a' is removed 'C' is removed and replaced with $\epsilon$

$\vdash (q_2, Z_0)$ $(\because \delta(q_1, \epsilon, Z_0) = (q_2, Z_0))$

# PDA




$Q = \{q_0, q_1, q_2\}$

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

$\Gamma = \{S, A, B, C, Z_0\}$

$q_0$ = Start State

$F = q_2$ - final state

$Z_0$ - initial symbol on stack.

2) $S \to aABB \mid aAA$

$A \to aBB \mid a$

$B \to bBB \mid A$

$C \to a$

Soln: The grammar should be in GNF.

$B \to \underline{A}$ not in GNF

$$B \to A \begin{cases} aBB \checkmark \\ a \checkmark \end{cases}$$

$B \to aBB \mid a$

Now,

$S \to aABB \mid aAA$

$A \to aBB \mid a$

$B \to bBB \mid \underbrace{aBB \mid a}$

$C \to a$

(2) $S \to aABB$

$S \to aAA$

$A \to aBB | a$

$A \to a$

$B \to bBB$

$B \to ABB$

$B \to a$

$C \to a$

(1) $\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$

$\delta(q_1, a, S) = (q_1, ABB)$

$\delta(q_1, a, S) = (q_1, AA)$

$\delta(q_1, a, A) = (q_1, BB)$

$\delta(q_1, a, A) = (q_1, \epsilon)$

$\delta(q_1, b, B) = (q_1, BB)$

$\delta(q_1, a, B) = (q_1, BB)$

$\delta(q_1, a, B) = (q_1, \epsilon)$

$\delta(q_1, a, C) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, Z_0) = (q_2, Z_0)$

final state

derive string

$S \Rightarrow a\underline{A}BB$

$\Rightarrow aa\underline{B}B$ $\quad (\because A \to a)$

$\Rightarrow aaa\underline{B}$ $\quad (\because B \to a)$

$\Rightarrow aaaa$ $\quad (\because B \to a)$

$\delta(q_0, aaaa, Z_0) \vdash (q_1, \underline{a}aaa, \underline{S}Z_0)$

$\vdash (q_1, \underline{a}aa, \underline{A}BBZ_0)$ $\quad (\because \delta(q_1, a, S) = (q_1, ABB))$

$\vdash (q_1, \underline{a}a, \underline{B}BZ_0)$ $\quad (\because \delta(q_1, a, A) = (q_1, \epsilon))$

$\vdash (q_1, a, BZ_0)$ $\quad (\because \delta(q_1, a, B) = (q_1, \epsilon))$

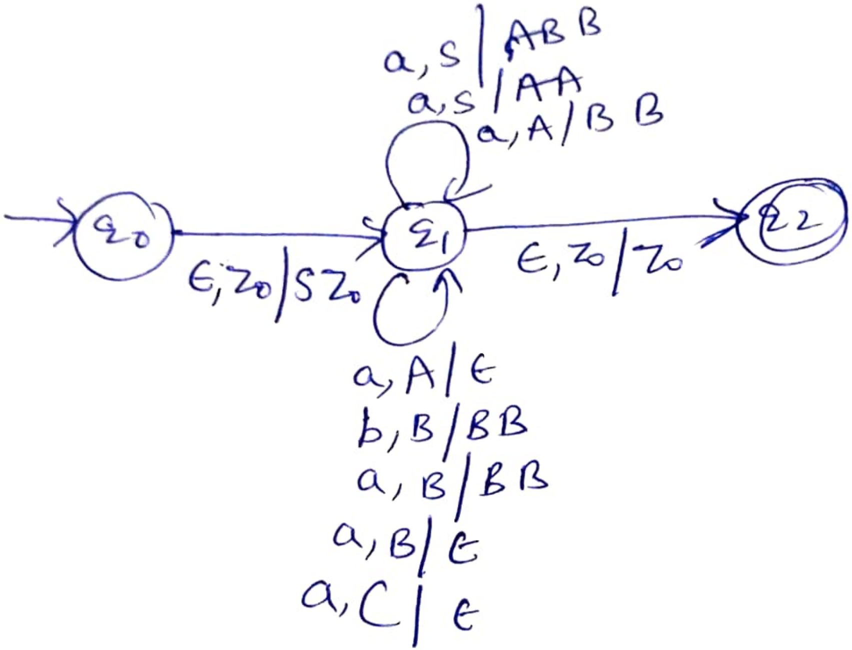
$\vdash (q_1, \epsilon, Z_0)$ $\quad (\because \delta(q_1, a, B) = (q_1, \epsilon))$

$\vdash (\underline{q_2}, Z_0)$ $\quad$ accepted

final state

# PDA

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




$Q = \{q_0, q_1, q_2\}$

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

$\Gamma = \{S, A, B, C, Z_0\}$

$q_0$ = start state

$Z_0 \rightarrow$ initial symbol on stack

$F \rightarrow q_2$ – final state

## 7.9 PDA to CFG

As we have converted CFG to PDA, we can convert a given PDA to CFG. The general procedure for this conversion is shown below:

1. The input symbols of PDA will be the terminals of CFG
2. If the PDA moves from state $q_i$ to state $q_j$ on consuming the input $a \in \Sigma$ when Z is the top of the stack, then the non-terminals of CFG are the triplets of the form $(q_i Z q_j)$
3. If $q_0$ is the start state and $q_f$ is the final state then $(q_0 Z q_f)$ is the start symbol of CFG.
4. The productions of CFG can be obtained from the transitions of PDA as shown below:

    a. For each transition of the form

    $$\delta(q_i, a, Z) = (q_j, AB)$$

    introduce the productions of the form

    $$(q_i Z q_k) \rightarrow a\,(q_j A q_l)(q_l B q_k)$$

    where $q_k$ and $q_l$ will take all possible values from Q.

    b. For each transition of the form

    $$\delta(q_i, a, Z) = (q_j, \varepsilon)$$

    introduce the production

    $$(q_i Z q_j) \rightarrow a$$

eliminated ... of useless symbols. ...

**Example 7.22: Obtain a CFG for the PDA shown below:**

$$\begin{aligned}
\delta(q_0, a, Z) &= (q_0, AZ)\\
\delta(q_0, a, A) &= (q_0, A)\\
\delta(q_0, b, A) &= (q_1, \varepsilon)\\
\delta(q_1, \varepsilon, Z) &= (q_2, \varepsilon)
\end{aligned}$$

**Note:** To obtain a CFG from the PDA, all the transitions should be of the form

$$\delta(q_i, a, Z) = (q_j, AB)$$

or

$$\delta(q_i, a, Z) = (q_j, \varepsilon)$$

In the given transitions except the second transition, all transitions are in the required form. So, let us take the second transition

$$\delta(q_0, a, A) = (q_0, A)$$

and convert it into the required form. This can be achieved if we have understood what the transition indicates. It is clear from the transition that when input symbol $a$ is encountered and top of the stack is A, the PDA remains in state $q_0$ and contents of the stack are not altered. This can be interpreted as delete A from the stack and insert A onto the stack. So, once A is deleted from the stack we enter into new state $q_3$. But, in state $q_3$ without consuming any input we add A on to the stack. The corresponding transitions are:

$$\begin{aligned}
\delta(q_0, a, A) &= (q_3, \varepsilon)\\
\delta(q_3, \varepsilon, Z) &= (q_0, AZ)
\end{aligned}$$

So, the given PDA can be written using the following transitions

$$\begin{aligned}
\delta(q_0, a, Z) &= (q_0, AZ)\\
\delta(q_0, a, A) &= (q_3, \varepsilon)\\
\delta(q_3, \varepsilon, Z) &= (q_0, AZ)\\
\delta(q_0, b, A) &= (q_1, \varepsilon)\\
\delta(q_1, \varepsilon, Z) &= (q_2, \varepsilon)
\end{aligned}$$

Now, the transitions

$$\begin{aligned}
\delta(q_0, a, A) &= (q_3, \varepsilon)\\
\delta(q_0, b, A) &= (q_1, \varepsilon)\\
\delta(q_1, \varepsilon, Z) &= (q_2, \varepsilon)
\end{aligned}$$

can be converted into productions as shown below:

| For $\delta$ of the form | Resulting Productions |
|---|---|
| $\delta(q_i, a, Z) = (q_j, \varepsilon)$ | $(q_i Z q_j) \rightarrow a$ |
| $\delta(q_0, a, A) = (q_1, \varepsilon)$ | $(q_0 A q_1) \rightarrow a$ |
| $\delta(q_0, b, A) = (q_1, \varepsilon)$ | $(q_0 A q_1) \rightarrow b$ |
| $\delta(q_1, \varepsilon, Z) = (q_2, \varepsilon)$ | $(q_1 Z q_2) \rightarrow \varepsilon$ |

Now, the transitions

$$\delta(q_0, a, Z) \quad = \quad (q_0, AZ)$$
$$\delta(q_1, \varepsilon, Z) \quad = \quad (q_0, AZ)$$

can be converted into productions using rule 4.a as shown below:

| For $\delta$ of the form | Resulting Productions |
|---|---|
| $\delta(q_i, a, Z) = (q_j, AB)$ | $(q_i Z q_k) \rightarrow a\,(q_j A q_l)(q_l B q_k)$ |
| $\delta(q_0, a, Z) = (q_0, AZ)$ | $(q_0Zq_0) \rightarrow a\,(q_0Aq_0)(q_0Zq_0) \mid a\,(q_0Aq_1)(q_1Zq_0) \mid$<br>$a\,(q_0Aq_2)(q_2Zq_0) \mid a\,(q_0Aq_3)(q_3Zq_0)$<br>$(q_0Zq_1) \rightarrow a\,(q_0Aq_0)(q_0Zq_1) \mid a\,(q_0Aq_1)(q_1Zq_1) \mid$<br>$a\,(q_0Aq_2)(q_2Zq_1) \mid a\,(q_0Aq_3)(q_3Zq_1)$<br>$(q_0Zq_2) \rightarrow a\,(q_0Aq_0)(q_0Zq_2) \mid a\,(q_0Aq_1)(q_1Zq_2) \mid$<br>$a\,(q_0Aq_2)(q_2Zq_2) \mid a\,(q_0Aq_3)(q_3Zq_2)$<br>$(q_0Zq_3) \rightarrow a\,(q_0Aq_0)(q_0Zq_3) \mid a\,(q_0Aq_1)(q_1Zq_3) \mid$<br>$a\,(q_0Aq_2)(q_2Zq_3) \mid a\,(q_0Aq_3)(q_3Zq_3)$ |
| $\delta(q_3, \varepsilon, Z) = (q_0, AZ)$ | $(q_3Zq_0) \rightarrow (q_0Aq_0)(q_0Zq_0) \mid (q_0Aq_1)(q_1Zq_0) \mid$<br>$(q_0Aq_2)(q_2Zq_0) \mid (q_0Aq_3)(q_3Zq_0)$<br>$(q_3Zq_1) \rightarrow (q_0Aq_0)(q_0Zq_1) \mid (q_0Aq_1)(q_1Zq_1) \mid$<br>$(q_0Aq_2)(q_2Zq_1) \mid (q_0Aq_3)(q_3Zq_1)$<br>$(q_3Zq_2) \rightarrow (q_0Aq_0)(q_0Zq_2) \mid (q_0Aq_1)(q_1Zq_2) \mid$<br>$(q_0Aq_2)(q_2Zq_2) \mid (q_0Aq_3)(q_3Zq_2)$<br>$(q_3Zq_3) \rightarrow (q_0Aq_0)(q_0Zq_3) \mid (q_0Aq_1)(q_1Zq_3) \mid$<br>$(q_0Aq_2)(q_2Zq_3) \mid (q_0Aq_3)(q_3Zq_3)$ |

The start symbol of the grammar will be $q_0Zq_2$.

**Example 7.23: Obtain a CFG that generates the language accepted by PDA M = ( $\{q_0,q_1\}$, $\{a, b\}$, $\{A, Z\}$, $\delta$, $q_0$, Z, $\{q_1\}$), with the transitions**

$$\delta(q_0, a, Z) \quad = \quad (q_0, AZ)$$
$$\delta(q_0, b, A) \quad = \quad (q_0, AA)$$
$$\delta(q_0, a, A) \quad = \quad (q_1, \varepsilon)$$

Now, the transition

$$\delta(q_0, a, A) \quad = \quad (q_1, \varepsilon)$$

can be converted into production as shown below:

| For $\delta$ of the form | Resulting Productions |
|---|---|
| $\delta(q_i, a, Z) = (q_j, \varepsilon)$ | $(q_i Z q_j) \rightarrow a$ |
| $\delta(q_0, a, A) = (q_1, \varepsilon)$ | $(q_0 A q_1) \rightarrow a$ |

Now, the transitions

$$\delta(q_0, a, Z) = (q_0, AZ)$$
$$\delta(q_0, b, A) = (q_0, AA)$$

can be converted into productions using rule 4.a as shown below:

| For $\delta$ of the form | Resulting Productions |
|---|---|
| $\delta(q_i, a, Z) = (q_j, AB)$ | $(q_i Z q_k) \rightarrow a\,(q_j A q_l)(q_l B q_k)$ |
| $\delta(q_0, a, Z) = (q_0, AZ)$ | $(q_0 Z q_0) \rightarrow a\,(q_0 A q_0)(q_0 Z q_0) \mid a\,(q_0 A q_1)(q_1 Z q_0)$<br>$(q_0 Z q_1) \rightarrow a\,(q_0 A q_0)(q_0 Z q_1) \mid a\,(q_0 A q_1)(q_1 Z q_1)$ |
| $\delta(q_0, b, A) = (q_0, AA)$ | $(q_0 Z q_0) \rightarrow b(q_0 A q_0)(q_0 A q_0) \mid b(q_0 A q_1)(q_1 A q_0)$<br>$(q_0 Z q_1) \rightarrow b(q_0 A q_0)(q_0 A q_1) \mid b(q_0 A q_1)(q_1 A q_1)$ |

The start symbol of the grammar will be $q_0 Z q_1$.