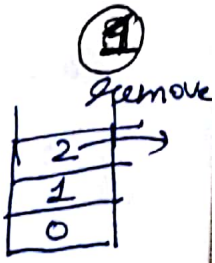
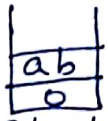
3/04/18

Operations on Stack in PDA:



- Top Element is removed (POP Operation)
- String of Stack symbols is pushed

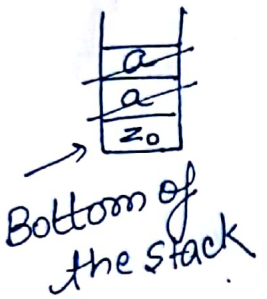


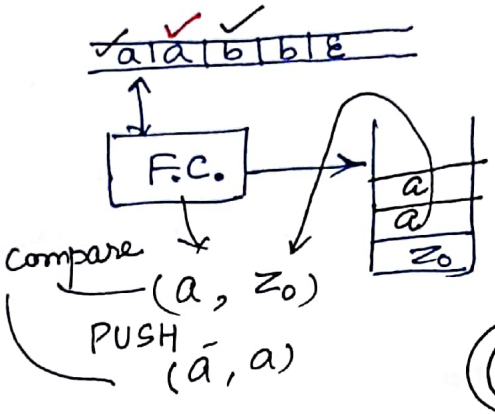
- Top element is replaced by String of Stack symbols.

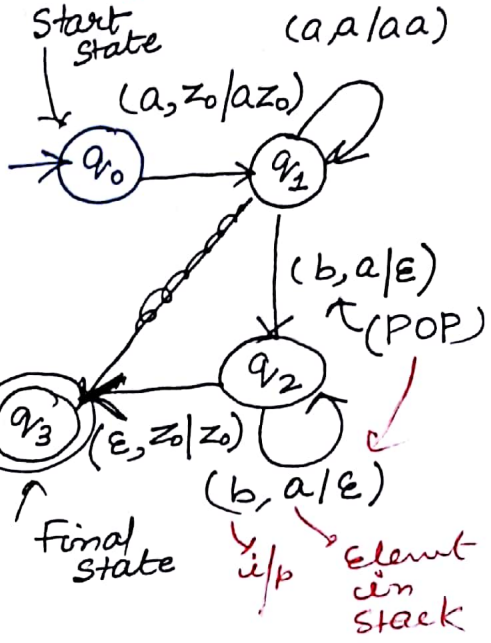
Ex: Construct PDA for Language, $L = \{a^n b^n \mid n >= 1\}$

$n=2$, $L = aabb\varepsilon$

$n=3$, $L = aaabbb\varepsilon$



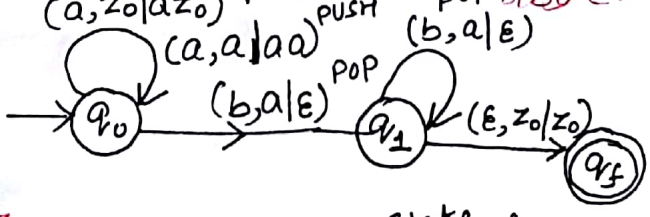




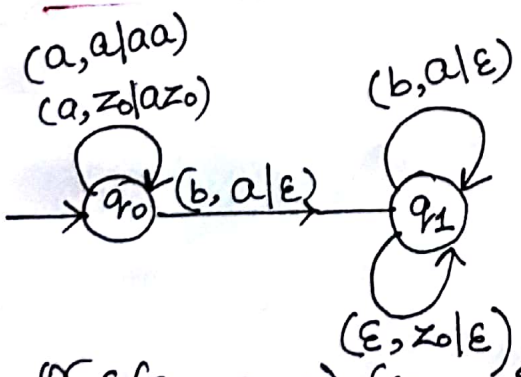
Transitions:

$$\delta(q_0, a, z_0) = (q_1, az_0)$$
$$\delta(q_1, a, a) = (q_1, aa)$$
$$\delta(q_1, b, a) = (q_2, \varepsilon)$$
$$\delta(q_2, b, a) = (q_2, \varepsilon)$$
$$\delta(q_2, \varepsilon, z_0) = (q_3, z_0)$$

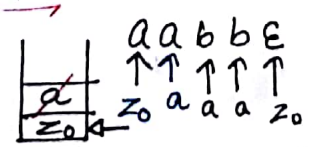
Another Solution for above.



State Transition Diagram



or $\delta(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\}$



$$\delta(q_0, a, z_0) = \{(q_0, az_0)\}$$
$$\delta(q_0, a, a) = \{(q_0, aa)\}$$
$$\delta(q_0, b, a) = \{(q_1, \varepsilon)\}$$
$$\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$$
$$\delta(q_1, \varepsilon, z_0) = \{(q_f, z_0)\}$$

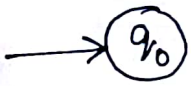
Acceptance by Empty Stack

(* Both have Same Power)

Acceptance by Final State

Construct a PDA to recognize $L = \{ wcw^R \mid w \in \{a, b\}^* \}$.

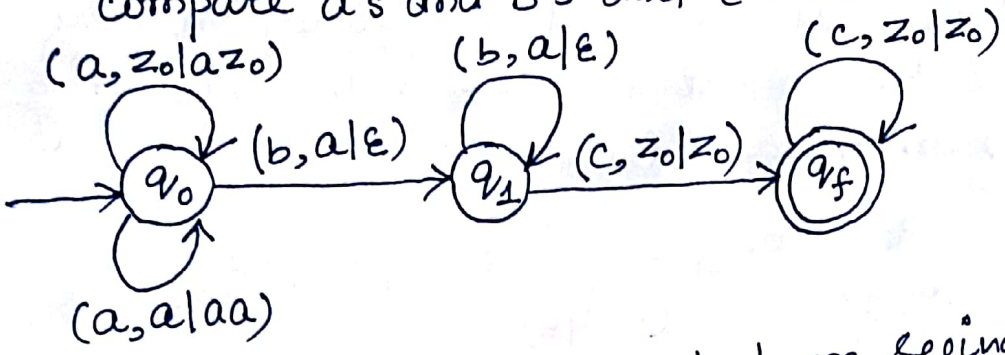
Soln:



7/04/18 Construct PDA for $L = \{a^n b^n c^m \mid n, m \geq 1\}$

Sol^n:

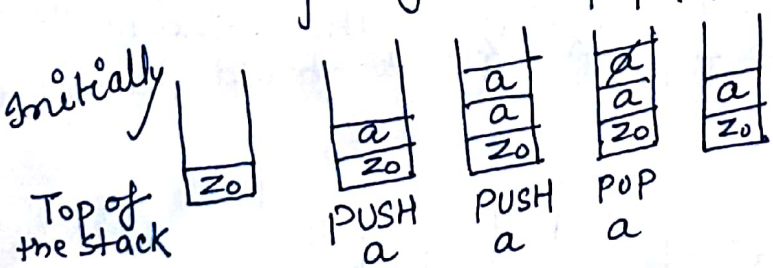
Note: Here, a's and b's should be equal and c's are independent of a's and b's. first we see all a's then all b's and then c's. So, we have to compare a's and b's and c's will not be matched.



Operation: a's will be pushed, on seeing b's, a's will be popped and on seeing c's, it will be accepted only or scanned only.

Note: e.g. $\boxed{w = bc}$, here string is not started with a and in our PDA at state q_0, we have not mentioned the dead configuration, for such strings which can not be accepted. So, in such situation the PDA will hault and it will not read the string further and say the string is not accepted.

And if e.g. $\boxed{w = aabcc}$, this string $\notin L$.

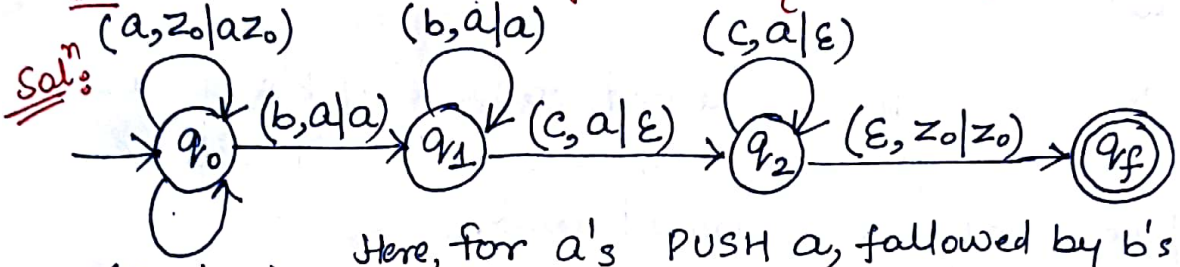


* Now, we can not PUSH 'c' here because on seeing 'c', z0 should be on the Top of the stack, but here a is on the Top, so the PDA will be hault and PDA will not accept the string.

It means if the string is available in the language (i.e. some $w \in L$), then we will stop at the final state. And, if the string is not available in the language (i.e. $w \notin L$), then in the PDA we will surely hault at the dead configuration and string will be rejected, because it is haulting at the non-final state.

Ex: Construct a PDA for $L = \{a^n b^m c^n \mid n, m \geq 1\}$

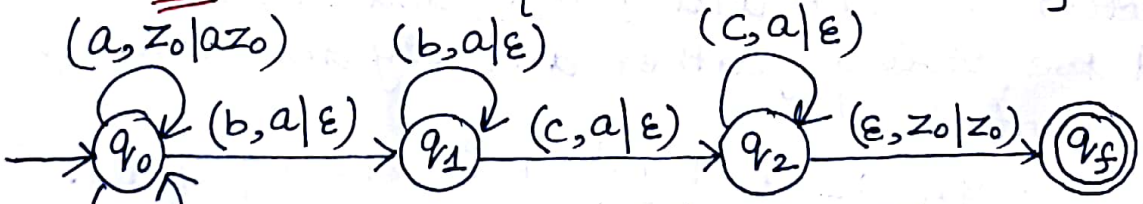
Sol^n:



Here, for a's PUSH a, followed by b's
for b's Neither PUSH nor POP, followed by c's
for c's POP a
No. of a's ≡ No. of c's $\boxed{n_a(w) = n_c(w)}$

Ex: Design PDA for $L = \{a^{m+n} b^m c^n \mid m, n \geq 1\}$

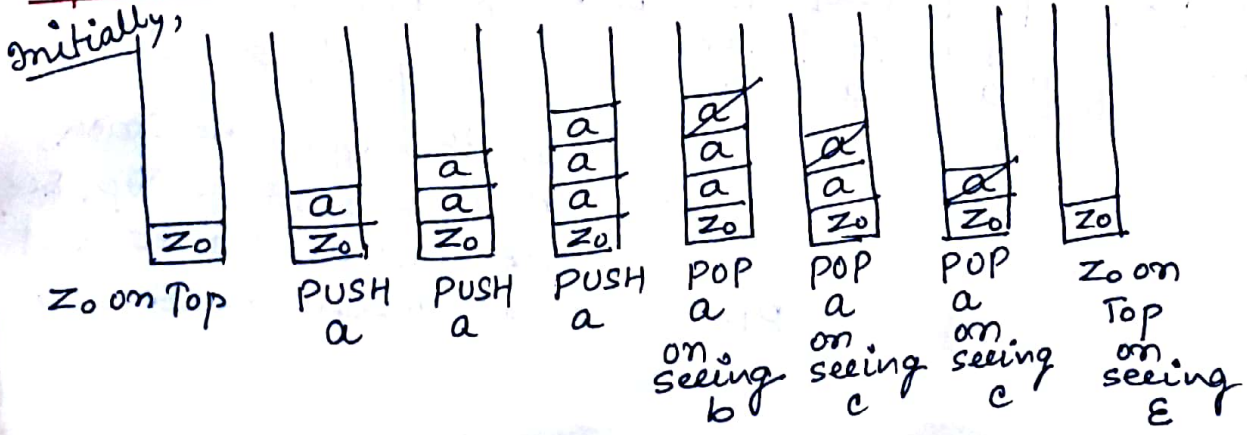
Sol^n: $L = \{a^m a^n b^m c^n \mid m, n \geq 1\}$ (Re-write)



Let $w =$ a a a b c c ε — The string $w \in L$. & should be accepted.

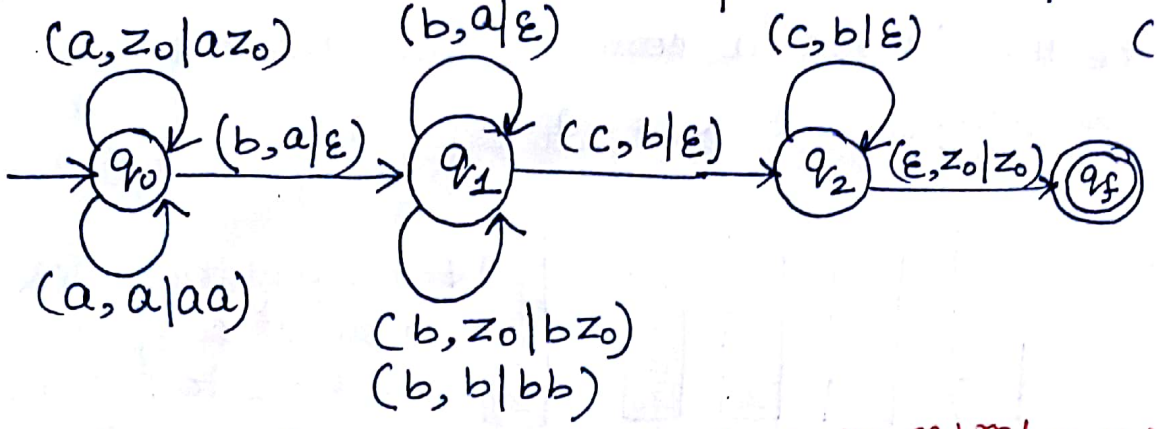
Operation:

Initially,



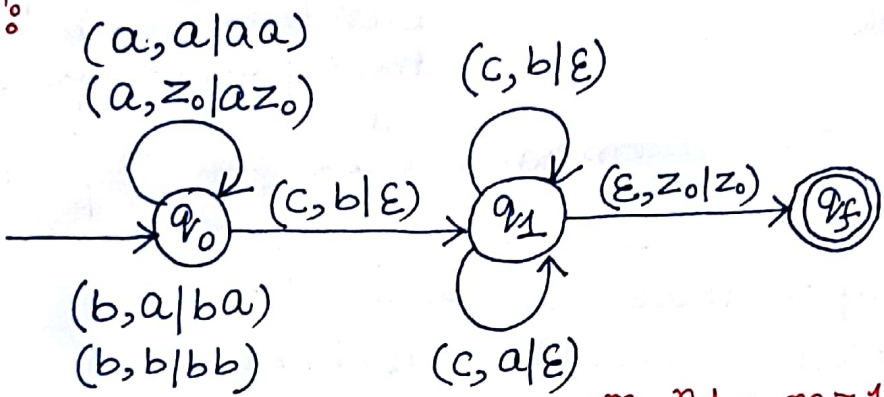
17/04/18 Ex: Design PDA for $L=\{a^n b^{m+n} c^m \mid n,m \geq 1\}$

Sol^n: Solve it like $L=\{a^n b^n b^m c^m \mid n,m \geq 1\}$ (Re-write)

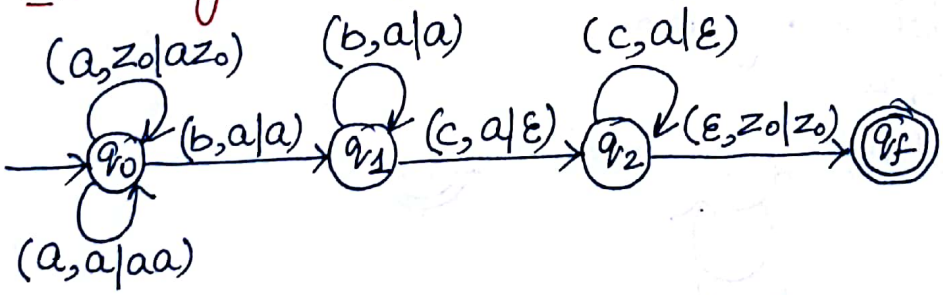


Ex: Design PDA for $L=\{a^n b^m c^{n+m} \mid n,m \geq 1\}$

Sol^n:



Ex: Design PDA for $L=\{a^n b^m c^n \mid n,m \geq 1\}$



Assignment:

1. Design PDA for $L=\{a^n b^n c^m d^m \mid n,m \geq 1\}$
2. Design PDA for $L=\{a^n b^m c^m d^n \mid n,m \geq 1\}$
3. " " " $L=\{a^m b^n \mid m > n\}$
4. " " " $L=\{a^i b^j \mid i \neq 2j+1\}$ & $(i=2j+1$ also)
5. " " " $L=\{a^i b^j c^k \mid j=i+k\}$
6. " " " $L=\{xcy \mid x,y \in \{0,1\}^*\}$
7. " " " $L=\{a^{2n+1} \mid n \geq 1\}$

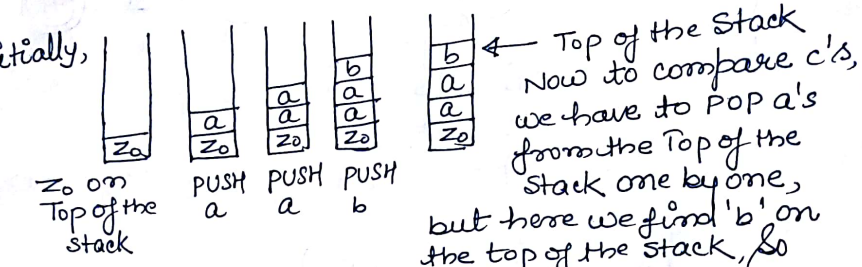
17/04/18

Ex: Design PDA for $L=\{a^n b^m c^n d^m \mid n,m \geq 1\}$

Sol^n: We can not be able to design PDA for this problem.
Hence, this is not a context-free language.

Counter Example: Let $w = a|a|b|c|c|d|\varepsilon$ i.e. $w \in L$

Logic:
Initially,



z_0 on Top of the stack — PUSH a — PUSH a — PUSH b

← Top of the Stack

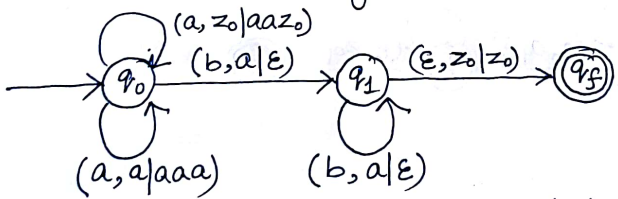
Now to compare c's, we have to POP a's from the Top of the stack one by one, but here we find 'b' on the top of the stack, so we can not POP 'a'.

Hence, the given problem can not be solved by PDA.

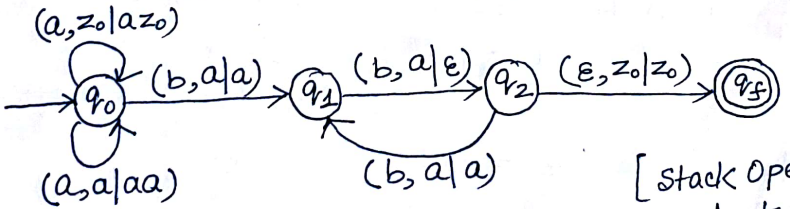
Ex: Design PDA for $L=\{a^n b^{2n} \mid n \geq 1\}$

Sol^n: for every 'a' there has to be two 'b's'.
i.e. $L=\{abb, aabbbb, aaabbbbbb, \ldots\}$

1st Sol^n: for single 'a', we push two b's, so that we can match it with no. of b's.



2nd Sol^n: Popping 2 b's for single 'a' or every alternating b will be used to pop one 'a' from the stack.



[Stack Operation Students should check yourself]

18/04/18

Equivalent PDA

18/04/18 Construct a PDA to recognize $L=\{wcw^R / w \in \{a,b\}^*\}$

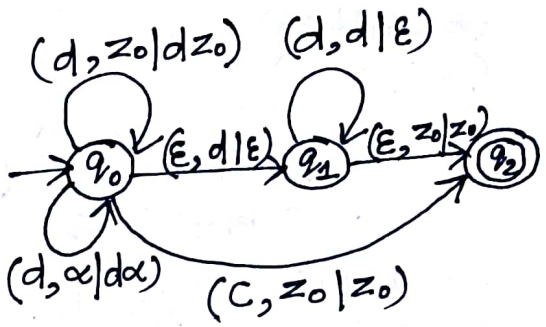
Sol^n:

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

$\Sigma=\{a,b,c\}$, $d \in \{a,b\}$, $\alpha \in S^*$

$Q=\{q_0,q_1,q_2\}$, $F=\{q_2\}$, $S=\{a,b,z_0\}$

1. $\delta(q_0,c,z_0)=\{(q_2,z_0)\}$; accept with $w=w^R=\varepsilon$
2. $\delta(q_0,d,z_0)=\{(q_0,dz_0)\}$
   $\delta(q_0,d,\alpha)=\{(q_0,d\alpha)\}$
   $\delta(q_1,d,d)=\{(q_1,\varepsilon)\}$
   $\delta(q_0,c,\alpha)=\{(q_1,\alpha)\}$
   $\delta(q_1,\varepsilon,z_0)=\{(q_2,z_0)\}$; accept



$\therefore (q_0,aabcbaa,z_0) \vdash^* (q_2,\varepsilon,z_0)$ [Instantaneous Description]

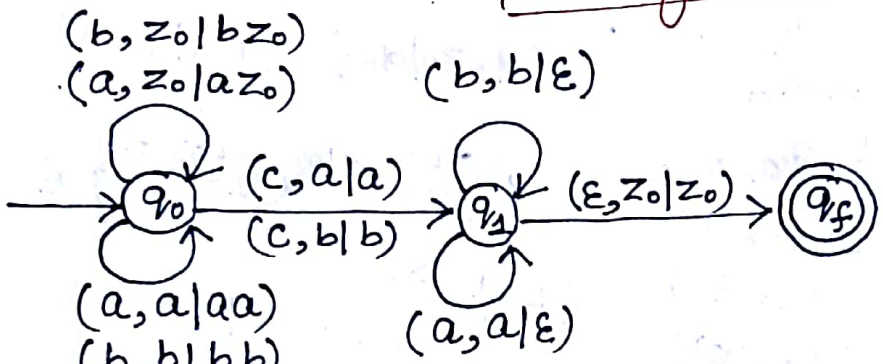
Design a PDA for $L = \{wcw^R \mid w \in \{a,b\}^+\}$

Sol^n: Given Language is an odd length palindrome.

We will PUSH 'w' on to the stack, and whenever we see 'c' then we will start popping w^R from the stack.

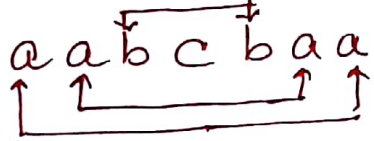
Odd Length Palindrome

we will start popping the string on seeing 'c' i.e. center of the string.

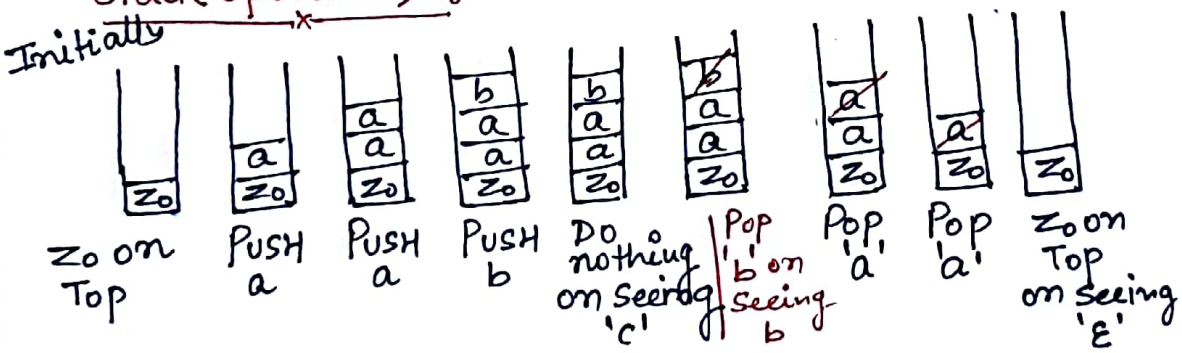


At state q_0, we representing pushing of 'w' with all possible combinations.

let $w = aab$, So string = $\underbrace{aab}_{w}\underbrace{cbaa}_{w^R}$



Stack operations:



18/04/18.

Equivalent PDA by Given C.F.G.

Theorem: If L is a context-free language, then we can construct a PDA 'P' accepting L by empty stack (store), i.e.

$$L = N(A).$$

Proof: We construct 'A' by making use of productions in C.F.G. or Grammar 'G'.

Step 1: Construction of PDA 'P':

Let $L = L(G)$, where $G = (N, T, P, S)$ is a context-free grammar.

We construct a PDA 'P' as:

$$P = (\{q\}, T, \{N \cup T\}, \delta, q, S, \phi)$$

where δ is defined by the following rules:

R_1: $\delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ is in } P\}$ (Productions)

R_2: $\delta(q, a, a) = \{(q, \varepsilon)\} \quad \forall a \in T.$

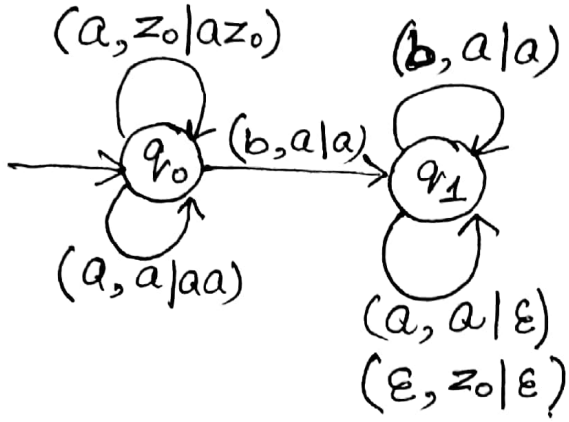
Explanation: The pushdown symbols in P are variables (Non Terminals) and Terminals. If PDA reads a Non-Terminal (variable) 'A' on the top of PDS (Pushdown Stack), it makes a ε-move by placing the R.H.S. of any A-production (after erasing A). If the PDA reads a Terminal 'a' on PDS and if it matches with the current input symbol, then the PDA erases a. In other cases the PDA halts.

If $w \in L(G_1)$ is obtained by a leftmost derivation

$$S \Rightarrow u_1 A_1 \alpha_1 \Rightarrow u_1 u_2 A_2 \alpha_2 \alpha_1 \Rightarrow \dots \Rightarrow w,$$

then PDA (A) can empty the PDS on application of input string w. The first move of PDA is by a ε-move corresponding to $S \to u_1 A_1 \alpha_1$. The PDA erases S and stores $u_1 A_1 \alpha_1$. Then using R_2, the PDA erases the symbols in u_1 by processing a prefix of w. Now, the topmost symbol in PDS is A_1. Once again by applying the ε-move corresponding to $A_1 \to u_2 A_2 \alpha_2$, the PDA erases A_2 and stores $u_2 A_2 \alpha_2$ above α_1. Processing in this way, the PDA empties the PDS by processing the entire string w.

18/04/18 **Ex:** Construct a pda accepting $\{a^n b^m a^n \mid m, n \geq 1\}$ by null store.



Transitions δ

$\delta(q_0, a, z_0) = \{(q_0, az_0)\}$
$\delta(q_0, a, a) = \{(q_0, aa)\}$
$\delta(q_0, b, a) = \{(q_1, a)\}$
$\delta(q_1, b, a) = \{(q_1, a)\}$
$\delta(q_1, a, a) = \{(q_1, \varepsilon)\}$
$\delta(q_1, \varepsilon, z_0) = \{(q_1, \varepsilon)\}$

or in ID's notation

$$(q_0, a^n b^m a^n, z_0) \overset{*}{\underset{M}{\vdash}} (q_1, \varepsilon, z_0) \underset{M}{\vdash} (q_1, \varepsilon, \varepsilon)$$

Acceptance by Null Store or empty stack.

PDA $M = (\{q_0, q_1\}, \{a, b\}, \{a, z_0\}, \delta, q_0, z_0, \phi)$

Q Σ S — Start State — Start Top Sym. — Null Store.

Construction of CFG Equivalent to Given PDA:

Theorem: If $A=(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ is a pda, then there exists a context-free grammar G_1 such that $L(G_1) = N(A)$.

Proof: Construction of C.F.G. 'G_1':

We define $G_1 = (V_N, \Sigma, P, S)$, where

Set of variables $V_N = \{S\} \cup \{[q, z, q'] \mid q, q' \in Q, z \in \Gamma\}$

i.e. any element of V_N is either the new symbol S acting as the start symbol for G_1 or an ordered triplet whose first and third elements are states and the second element is a pushdown symbol.

The productions in P are induced by moves of pda as follows:

R_1: S-productions are given by $S \to [q_0, z_0, q]$ for every q in Q.

R_2: Each move erasing a pushdown symbol given by $(q', \varepsilon) \in \delta(q, a, z)$ induces the production $[q, z, q'] \to a$.

R_3: Each move not erasing a pushdown symbol given by $(q_1, z_1 z_2 \ldots z_m) \in \delta(q, a, z)$ induces many productions of the form

$$[q, z, q'] \to a[q_1, z_1, q_2][q_2, z_2, q_3] \ldots [q_m, z_m, q']$$

where each of the states $q', q_2, \ldots, q_m$ can be any state in Q. Each move yields many productions because of R_3.

18/04/18

Ex: construct a C.F.G G', where pda, $A=(\{q_0,q_1\},\{a,b\},\{z_0,z\},\delta,q_0,z_0,\phi)$ and δ is given by

$$\delta(q_0,b,z_0)=\{(q_0,zz_0)\}$$
$$\delta(q_0,\varepsilon,z_0)=\{(q_0,\varepsilon)\}$$
$$\delta(q_0,b,z)=\{(q_0,zz)\}$$
$$\delta(q_0,a,z)=\{(q_1,z)\}$$
$$\delta(q_1,b,z)=\{(q_1,\varepsilon)\}$$
$$\delta(q_1,a,z_0)=\{(q_0,z_0)\}$$

Sol^n: Let CFG. be $G=(V_N,\{a,b\},P,S)$.

where $V_N=\{S,[q_0,z_0,q_0],[q_0,z_0,q_1],[q_0,z,q_0],[q_0,z,q_1],[q_1,z_0,q_0],[q_1,z_0,q_1],[q_1,z,q_0],[q_1,z,q_1]\}$.

The productions are

$P_1: S\longrightarrow[q_0,z_0,q_0]$ [Applying R_1]

$P_2: S\longrightarrow[q_0,z_0,q_1]$ [Applying R_1]

$\delta(q_0,b,z_0)=\{(q_0,zz_0)\}$ yields

$P_3:[q_0,z_0,q_0]\longrightarrow b[q_0,z,q_0][q_0,z_0,q_0]$

$P_4:[q_0,z_0,q_0]\longrightarrow b[q_0,z,q_1][q_1,z_0,q_0]$

$P_5:[q_0,z_0,q_1]\longrightarrow b[q_0,z,q_0][q_0,z_0,q_1]$

$P_6:[q_0,z_0,q_1]\longrightarrow b[q_0,z,q_1][q_1,z_0,q_1]$

[By R_3]

$\delta(q_0,\varepsilon,z_0)=\{(q_0,\varepsilon)\}$ gives

$P_7:[q_0,z_0,q_0]\longrightarrow\varepsilon$ [By, R_2]

18/04/18

$\delta(q_0, b, z) = \{(q_0, z)\}$ yields

$P_8 : [q_0, z, q_0] \longrightarrow b[q_0, z, q_0][q_0, z, q_0]$

$P_9 : [q_0, z, q_0] \longrightarrow b[q_0, z, q_1][q_1, z, q_0]$ [By R_3]

$P_{10} : [q_0, z, q_1] \longrightarrow b[q_0, z, q_0][q_0, z, q_1]$

$P_{11} : [q_0, z, q_1] \longrightarrow b[q_0, z, q_1][q_1, z, q_1]$

$\delta(q_0, a, z) = \{(q_1, z)\}$ yields

$P_{12} : [q_0, z, q_0] \longrightarrow a[q_1, z, q_0]$ [By R_3]

$P_{13} : [q_0, z, q_1] \longrightarrow a[q_1, z, q_1]$

$\delta(q_1, b, z) = \{(q_1, \varepsilon)\}$ gives

$P_{14} : [q_1, z, q_1] \longrightarrow b$

$\delta(q_1, a, z_0) = \{(q_0, z_0)\}$ gives

$P_{15} : [q_1, z_0, q_0] \longrightarrow a[q_0, z_0, q_0]$

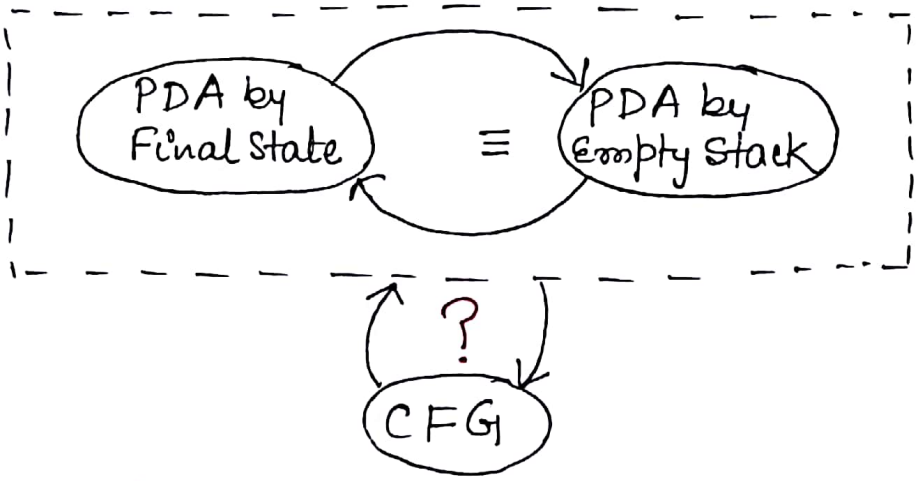
$P_{16} : [q_1, z_0, q_1] \longrightarrow a[q_0, z_0, q_1]$

$P_1 - P_{16}$ give the productions in P.

20/04/18

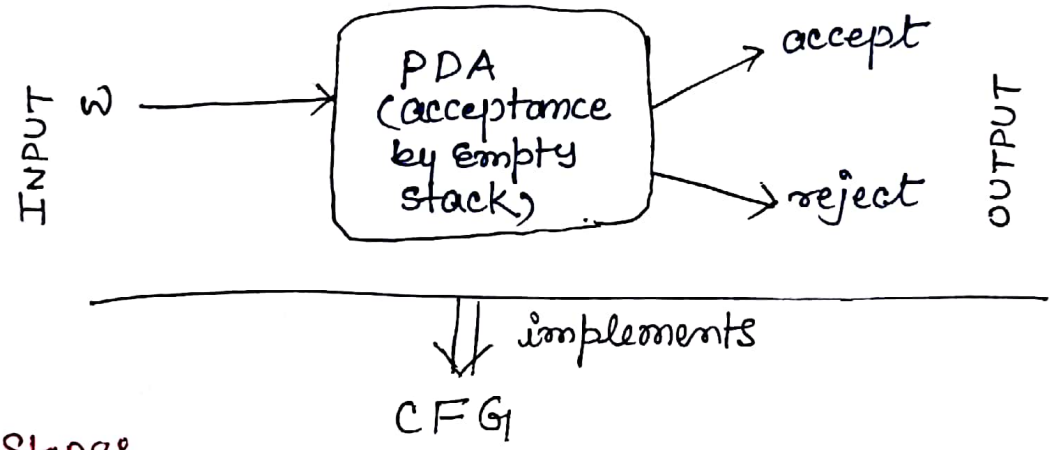
Equivalence of PDAs and CFGs

$CFG_s == PDAs \Rightarrow CFLs$



Converting CFG to PDA:

Main Idea: The PDA simulates the left most derivation on a given w, and upon consuming it fully it either arrives at acceptance (by empty stack) or non-acceptance.



Steps:

1. PUSH the right hand side (R.H.S.) of the production onto the stack, with left-most symbol at the stack top.
2. If stack top is the leftmost variable, then replace it by all its productions (each possible substitution will represent a distinct path taken by NDPDA)
3. If stack top has a terminal symbol, and if it matches with the next symbol in the input string, then POP it.

State is inconsequential (only one state is needed)

Formal construction of PDA from CFG

Given (or Input): $G_1 = (N, T, P, S)$

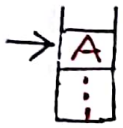
Note: Initial Stack symbol ($) same as the Start variable in the Grammer

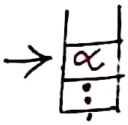
Output: $P_N = (\{q\}, T, (N \cup T), \delta, q, S, \phi)$

Transitions are as follows:

R_1: for all $A \in N$, add the following transition(s) in the PDA:

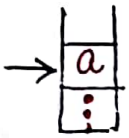
$$\delta(q, \varepsilon, A) = \{(q, \alpha) \mid "A \to \alpha" \in P\}$$

Before: 

After: 

R_2: for all $a \in T$, add the following transition(s) in the PDA:

$$\delta(q, a, a) = \{(q, \varepsilon)\}$$

Before: 

After: a... 

Ex: CFG to PDA

Given, $G_1 = (\{S, A\}, \{0, 1\}, P, S)$

P: $S \to AS \mid \varepsilon$

$A \to 0A1 \mid A1 \mid 01$

Leftmost derivation:

$S \Rightarrow AS$
$\Rightarrow 0A1S$
$\Rightarrow 0011S$
$\Rightarrow 0011$

Sol^n:

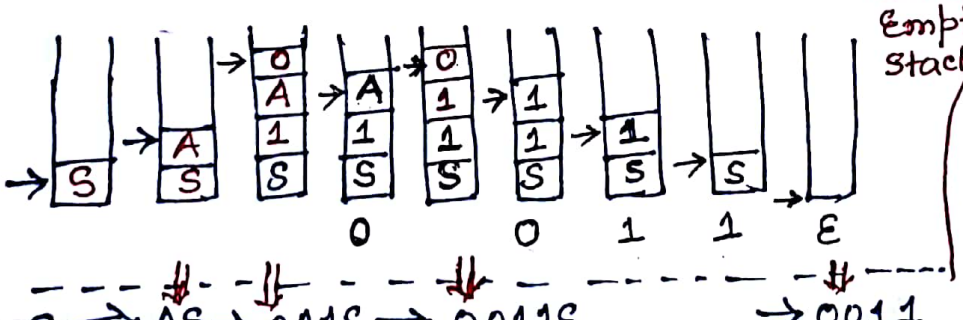
PDA: $P_N = (\{q\}, \{0, 1\}, \{0, 1, A, S\}, \delta, q, S, \phi)$

δ:

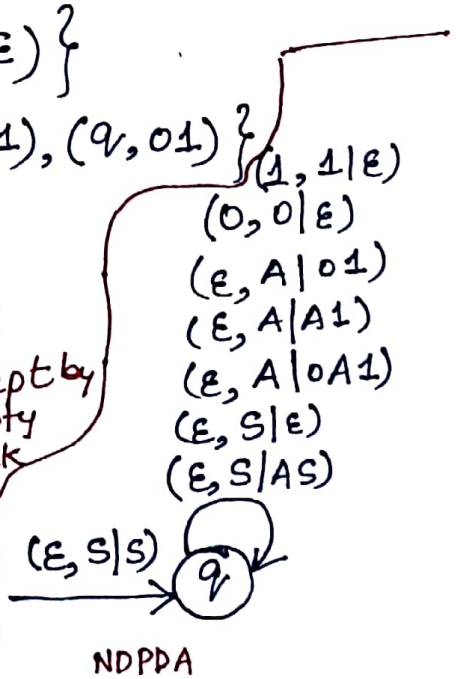
$$\delta(q, \varepsilon, S) = \{(q, AS), (q, \varepsilon)\}$$
$$\delta(q, \varepsilon, A) = \{(q, 0A1), (q, A1), (q, 01)\}$$
$$\delta(q, 0, 0) = \{(q, \varepsilon)\}$$
$$\delta(q, 1, 1) = \{(q, \varepsilon)\}$$

Simulating string 0011 on the new PDA

Stack moves (shows only the successful Path):



Accept by Empty stack



NDPDA

$S \Rightarrow AS \Rightarrow 0A1S \Rightarrow 0011S \Rightarrow 0011$

20/04/18

Converting a PDA into a CFG

Main Idea: Reverse engineer the productions from transitions

If $\delta(q, a, z) \Rightarrow (p, Y_1 Y_2 Y_3 \ldots Y_k)$:

1. State is changed from q to p;
2. Terminal a is consumed;
3. Stack top symbol z is popped and replaced with a sequence of k variables.

Action: Create a grammar variable called "$[q, z, p]$" which includes the following production:

- $[q, z, p] \longrightarrow a[p, Y_1, q_1][q_1, Y_2, q_2][q_2, Y_3, q_3] \ldots [q_{k-1}, Y_k, q_k]$

I^st • $S \longrightarrow$ productions are given by $[q_0, z_0, q]$ $\forall q \in Q$.

Ex: Bracket Matching: To avoid confusion we will use $b =$ "(" and $e =$ ")".

Given:

$$P_N = (\{q_0\}, \{b, e\}, \{z_0, z_1\}, \delta, q_0, z_0, \phi)$$

PDA Transitions:

1. $\delta(q_0, b, z_0) = \{(q_0, z_1 z_0)\}$
2. $\delta(q_0, b, z_1) = \{(q_0, z_1 z_1)\}$
3. $\delta(q_0, e, z_1) = \{(q_0, \varepsilon)\}$
4. $\delta(q_0, \varepsilon, z_0) = \{(q_0, \varepsilon)\}$

Grammar Productions:

0. $S \longrightarrow [q_0, z_0, q_0]$
1. $[q_0, z_0, q_0] \longrightarrow b\,[q_0, z_1, q_0]\,[q_0, z_0, q_0]$
2. $[q_0, z_1, q_0] \longrightarrow b\,[q_0, z_1, q_0]\,[q_0, z_1, q_0]$
3. $[q_0, z_1, q_0] \longrightarrow e$
4. $[q_0, z_0, q_0] \longrightarrow \varepsilon$

P: Grammar

$$S \longrightarrow bSeS \mid \varepsilon$$

$G = (S, \{b, e\}, P, S)$

Ans

Let $A = [q_0, z_0, q_0]$

$B = [q_0, z_1, q_0]$

0. $S \longrightarrow A$
1. $A \longrightarrow bBA$
2. $B \longrightarrow bBB$
3. $B \longrightarrow e$
4. $A \longrightarrow \varepsilon$

Removing unit, null productions & useless we get ⇒

Simplifying

0. $S \longrightarrow bBS \mid \varepsilon$
1. $B \longrightarrow bBB \mid e$

Equivalence of CFG to PDA

Ex: Given a Grammar

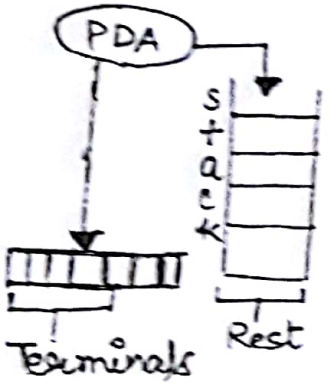
P:
$S \rightarrow BS \mid A$
$A \rightarrow 0A \mid \varepsilon$
$B \rightarrow BB1 \mid 2$

Leftmost derivation:

$S \Rightarrow BS$
$\Rightarrow BB1S$
$\Rightarrow 2B1S$
$\Rightarrow 221S$
$\Rightarrow 221A$
$\Rightarrow 221\varepsilon$
$\Rightarrow 221$

Required (Say) String

General Form:



aaaa BaBC

Terminals Rest

Let PDA

$P_N = (\{q\}, \{0,1,2\}, \{0,1,2,A,B,S\}, \delta, q, S, \phi)$

Transitions:

δ:

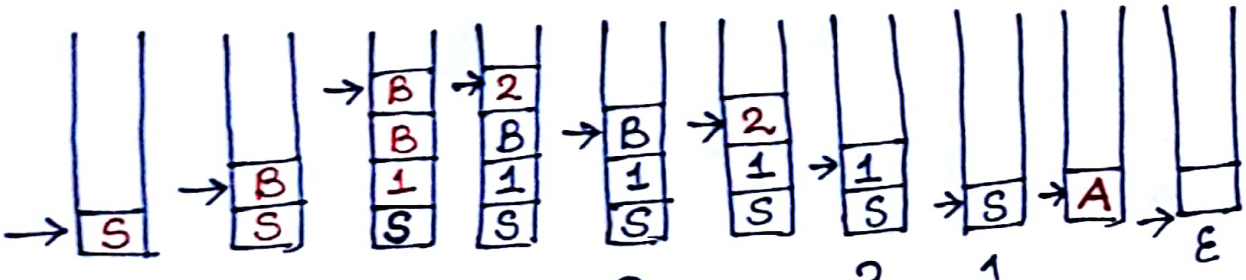
$\delta(q, \varepsilon, S) = \{(q, BS), (q, A)\}$
$\delta(q, \varepsilon, A) = \{(q, 0A), (q, \varepsilon)\}$
$\delta(q, \varepsilon, B) = \{(q, BB1), (q, 2)\}$
$\delta(q, 0, 0) = \{(q, \varepsilon)\}$
$\delta(q, 1, 1) = \{(q, \varepsilon)\}$
$\delta(q, 2, 2) = \{(q, \varepsilon)\}$

PDA Diagram

(1,1|ε) (2,2|ε)
(0,0|ε)
(ε, B|2)
(ε, B|BB1)
(ε, A|ε)
(ε, A|0A)
(ε, S|A)
(ε, S|BS)

* Simulating string 221 on the new PDA ...

Stack moves (shows only the successful path):



$S \Rightarrow BS \Rightarrow BB1S \Rightarrow 2B1S \Rightarrow 221S \Rightarrow 221A \Rightarrow 221$

(Leftmost derivation)

20/04/18 Convert the following CFG to a PDA

$S \rightarrow aAA$

$A \rightarrow aS \mid bS \mid a$

Soln: The PDA $P_N = (Q, \Sigma, V \cup \Sigma, \delta, q_0, Z_0, F)$ is defined as

$Q = \{q\}$, $\Sigma = \{a, b\}$, $V \cup \Sigma = \{a, b, S, A\}$

$q_0 = q$, $Z_0 = S$, $F = \{\}$ or ϕ.

And the transition function is defined as:

δ:

$\delta(q, \varepsilon, S) = \{(q, aAA)\}$

$\delta(q, \varepsilon, A) = \{(q, aS), (q, bS), (q, a)\}$

$\delta(q, a, a) = \{(q, \varepsilon)\}$

$\delta(q, b, b) = \{(q, \varepsilon)\}$

13/04/18

$I_1 \vdash_M^n I_2$ → exactly 'n' steps (definite sequence)

In general $I_1 \vdash_M^* I_2$

Note: If we want zero or more moves of a PDA. ← use this notation.

Note is Above Reflexive-transitive Closure: $I \vdash_M^* I$ zero number of moves.

- Acceptability of PDA or Acceptance of Language of PDA

Two types

Finite State Acceptability

for a PDA P, the Language accepted by P, denoted by L(P) by final state, is:

$\{w \mid (q_0, w, z_0) \vdash^* (q_f, \varepsilon, A)\}$

Empty Stack Acceptability (NULL or store)

for a PDA P, the language accepted by P, denoted by N(P) by empty stack, is:

$\{w \mid (q_0, w, z_0) \vdash^* (q, \varepsilon, \varepsilon)\}$ for any $q \in Q$.

Check list: - Input exhausted? → Same
- In a final state? - Is the stack empty?

Q: Does a PDA that accepts by Empty Stack need any final state specified in the design?

13/04/18

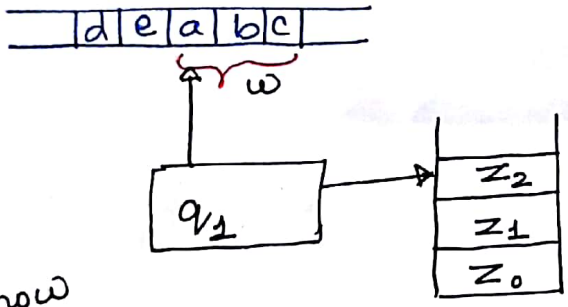
Instantaneous Description (ID)

Configuration of a Pushdown automata at a particular instant is described by Instantaneous description.

It has 3 components:

- Present state of Pushdown automata. (i.e. q)
- String of input symbols yet to be read i.e. (ω) i.e. unconsumed string
- content of stack. (S)

The Instantaneous Description (ID) of a PDA is represented by a triplet (q, ω, S), where $q \in Q$, $\omega \in \Sigma^*$ and $S \in S^*$ (S is the content of stack in which left most symbol is top of the stack)



Leftmost symbol
Top of the stack

ID :

$(q_1, \underbrace{abc}_{\omega}, \underbrace{Z_2 Z_1 Z_0}_{S})$

We know that, Transition rule according to PDA is

$$\delta(p, a, z) = (q, \alpha)$$

Turnstile Notation: The "turnstile" notation is used for connecting pairs of ID's that represent one or many moves of a PDA. The process of transition is denoted by the turnstile symbol "$\vdash$".

The transition can be mathematically represented by the following turnstile notation as:

$$(p, a\omega, T\beta) \vdash (q, \omega, \alpha\beta)$$

or

$$(p, \underbrace{a_0 a_1 a_2 \ldots a_n}_{\omega}, \underbrace{Z_n}_{T}, Z_{n-1}, \ldots Z_1, Z_0) \vdash (q, a_1 a_2 \ldots a_n, \underbrace{\alpha}_{\beta}, Z_{n-1} \ldots Z_1 Z_0)$$

This implies while taking a transition from state p to q, the input symbol 'a' is consumed, and Top of the stack 'T' is replaced by a new string 'α'.

13/04/18

Ex 2: Construct PDA for Language, such that Number of 'a' is equal to number of 'b'.

i.e. $L = \{ w \in \{a,b\}^* \mid n_a(w) = n_b(w) \}$.

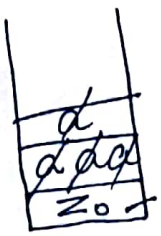
$L = \{ \varepsilon, ab, ba, aabb, abab, \ldots \}$

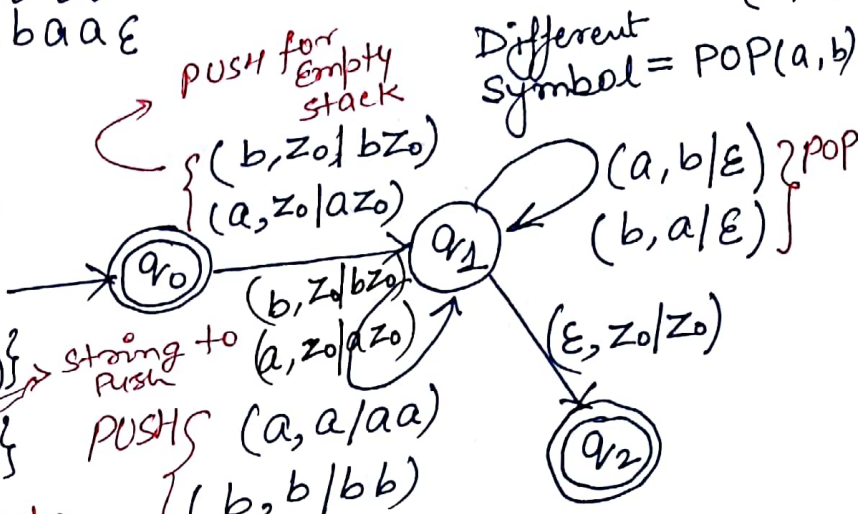
Let the string abababbaaε

babbaaε

Logic: Same Symbol = PUSH (a,a)

Different Symbol = POP (a,b)





Transition functions:

(i) $\delta(q_0, a, Z_0) = \{(q_1, aZ_0)\}$ → string to Push

or $\delta(q_0, b, Z_0) = \{(q_1, bZ_0)\}$

- q_0 → Current state
- q_1 → Next state
- a / b → Input symbol
- Z_0 → Top of the stack

Another Solution:

$\delta(q_1, a, a) = \{(q_1, aa)\}$

or $\delta(q_1, b, b) = \{(q_1, bb)\}$

$\delta(q_1, a, b) = \{(q_1, \varepsilon)\}$

or $\delta(q_1, b, a) = \{(q_1, \varepsilon)\}$

$\delta(q_1, \varepsilon, Z_0) = \{(q_2, Z_0)\}$

12/04/18

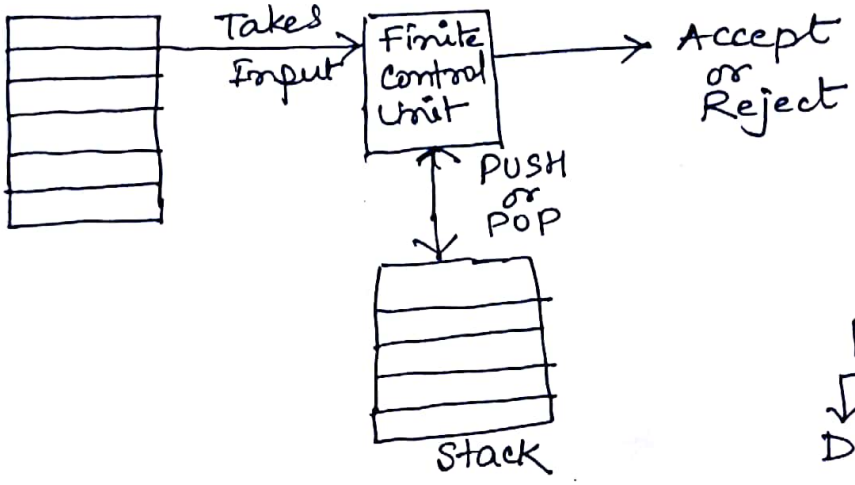
Pushdown Automata

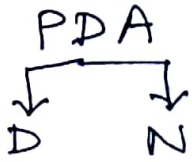
A pushdown Automata (PDA) is a way to implement a context free Grammar in a similar way we design finite Automata for Regular Grammar

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

A Pushdown Automata has 3 components:

1. An input tape
2. A Finite control unit
3. A stack with infinite size





Def^n: 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 = Start State ($q_0 \in Q$)
Z_0 = The start Stack Symbol ($z_0 \in \Gamma$)
F = The set of final / Accepting States. ($F \in Q$)

12/04/18

δ: The Transition function

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 $a = \varepsilon$

(iii) X is a stack symbol, that is a member of Γ.

The output of δ is finite set of pairs (p, γ) where:

p is a new state

γ is a string of stack symbols that replaces X at the top of the stack

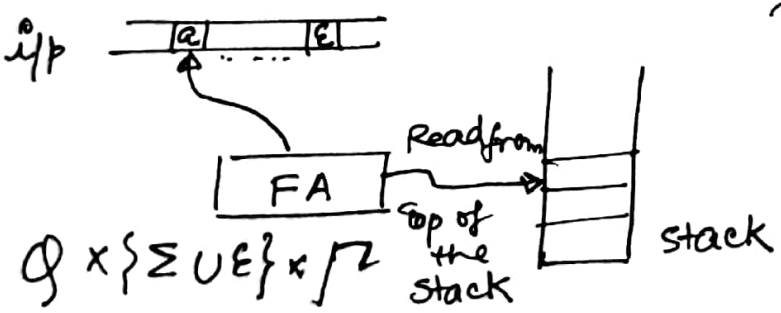
e.g. If $\gamma = \varepsilon$ then, stack is popped

If $\gamma = X$ then the stack is unchanged

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

PDA → D : $\delta : Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \longrightarrow Q \times \Gamma^*$

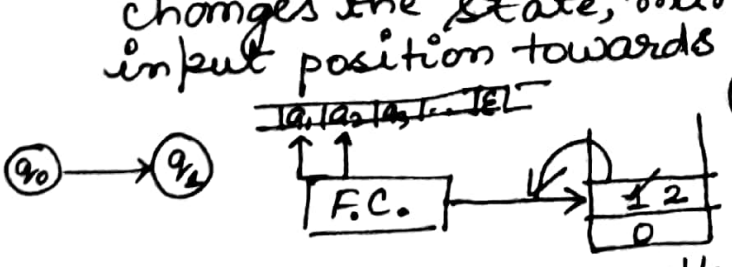
PDA → N : $\delta : Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \longrightarrow 2^{(Q \times \Gamma^*)}$



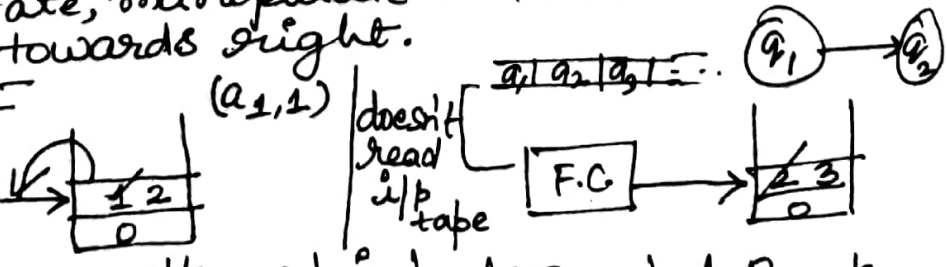
Transition function describes moves in PDA.

Types of Moves: In PDA

- Reads input symbol and top element of the stack, changes the state, manipulate the stack and move input position towards right.



$(a_1, 1)$



- Empty Moves: Doesn't read input symbol. Reads, Manipulate Stack and changes the State.