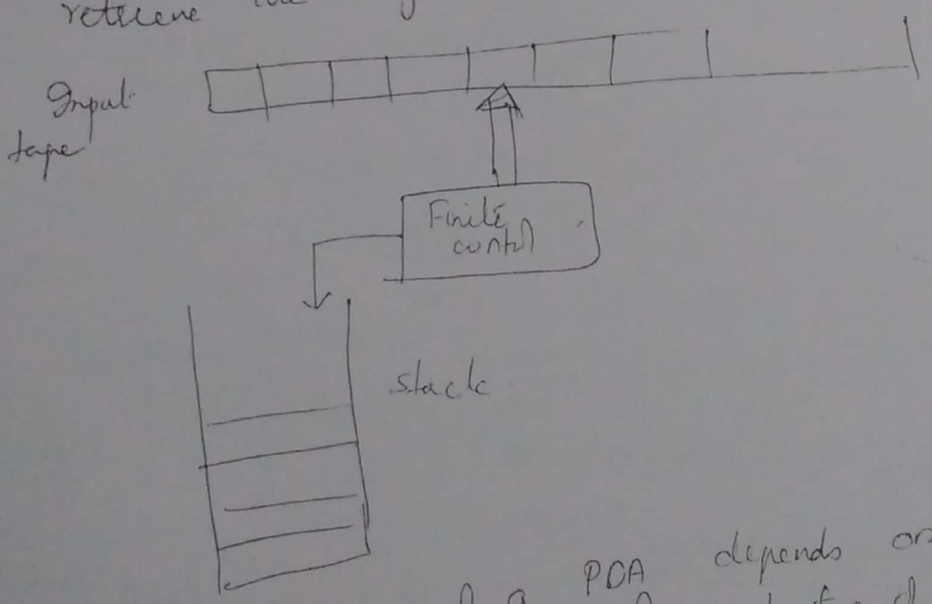
Push Down Automata (PDA)

FA can't be used to recognize all CFLs. A class of automata associated with CFLs is PDA. FA have strictly finite memories, whereas recognition of a CFL requires storing an unbounded amount of information. To scan a string from the language $L = \{a^n b^n\}$ we must not only check whether all a's precede the first b, but also count the no. of a's. Since n is unbounded, counting cannot be done with finite memory. So we use an auxiliary memory in the form of stack. This type of arrangement where a FA has stack leads to the generation of a PDA.

PDA consists of an input tape, a finite control and a stack to store and retrieve the symbol.



Each move of a PDA depends on the current state, input symbol and top of stack symbol.

Each move consists of change of state and replacing top of stack symbol by a string of stack symbols.

Default model of a PDA is <u>non-deterministic</u>

Mathematically a PDA M is a 7-tuple notation $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ where

Γ - capital gamma

Q - finite set of states

Σ - input alphabet

Γ - stack alphabet

q_0 in Q is initial state

z_0 in Γ is the initial stack symbol.

$F \subseteq Q$ is set of final states

and δ is a mapping from $Q \times \Sigma \cup \{\varepsilon\} \times \Gamma$ to a finite subsets of $Q \times \Gamma^*$.

Interpretation of moves

A move $\delta(q, a, z) = \{(p_1, \gamma_1)(p_2, \gamma_2) \cdots (p_m, \gamma_m)\}$ states that: If PDA is in state q with i/p symbol a & z the topmost stack symbol, then PDA can enter into any state p_i and replace z by string γ_i $(1 \le i \le m)$ and advance the i/p head one symbol.

The move $\delta(q, \varepsilon, z) = \{(p_1, \gamma_1)(p_2, \gamma_2) \cdots (p_m, \gamma_m)\}$ states that if PDA is in state q, then independent of the input symbol being scanned with z on stack top, can enter state p_i and replace z by γ_i $(1 \le i \le m)$

Instantaneous Description (ID)

During execution a PDA goes through a sequence of configuration each called ID consists of configuration (i) state (ii) input yet to be scanned and (iii) stack content.

represented by a triplet (q, w, γ) where q is the current state, w is the input yet to be read (leftmost symbol of x is the current input) or γ is the stack content (leftmost symbol of γ is the current top of stack symbol.

Each move involves a change from one ID to another. The symbol ⊢ is used to represent a move

A move is defined by the following rule

$(p, ax, A\alpha) \vdash (q, x, \beta\alpha)$ if

$\delta(p, a, A)$ includes (q, β)

$\vdash^*$ represents a sequence of moves

Language acceptance by PDA

PDA can accept a given language in 2 ways
(1) Acceptance by empty stack
(2) Acceptance by final state

Language accepted by empty stack defined as

$N(M) = \{ w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \varepsilon)$ for some p in Q $\}$

Language accepted by final state defined as

$L(M) = \{ w \mid (q_0, w, Z_0) \vdash^* (p, \varepsilon, \gamma)$ for some p in F & γ in Γ $\}$

PDA 2 types
— Deterministic PDA (DPDA)
— Non deterministic PDA (NPDA)

In non deterministic PDA there are finite no of choices of moves in each of the moves will be 2 types situations

In the first type of move depending on state of finite control, input symbol & top symbol on stack a no: of choices are possible. Each choice consists of a next state for finite control and a string of symbols to replace the topmost stack symbol. After selecting a choice, input head is advanced one symbol. The second type of move (ϵ-move) is similar to first except that input symbol not used and input head not advanced after the move. This type of move allows only to manipulate the stack without reading input symbols.

Deterministic PDA (DPDA)

A PDA is said to be deterministic if all the IDs in the design has to give only a single move.

Formally we say a PDA M is deterministic if

(1) for each q in Q and z in Γ, whenever $\delta(q, \epsilon, z)$ is non-empty, then $\delta(q, a, z)$ is empty for all a in Σ.

(2) for no q in Q, z in Γ and a in $\Sigma \cup \{\epsilon\}$ does $\delta(q, a, z)$ contains more than one element.

(1) Design a PDA that accepts $\{wcw^R / w \in (0+1)^*\}$ by empty stack

The central idea is to remember the pattern of w in stack and match the right portion with the pattern already stored. For this purpose we use 2 stack symbols B & G to represent 0 & 1 respectively. In state q_1, PDA pushes B for 0 & G for 1. On scanning c, PDA goes to q_2 state. After scanning the left portion w, stack content from top to bottom is the reverse of w. Also top of stack symbol corresponds to input to be matched.

In state q_2, PDA matches and pops-off B for input 0 and G for input 1. At the end, if input is correct, reverse of left portion (w^R) matches with the input and the initial stack symbol R is exposed. Then PDA goes to accepting state q_2, ~~with~~ ~~top~~ ~~wait~~ making the stack empty.

Top plate	State	Input 0	1	c
Blue	q_1	Add blue plate stay in state q_1	Add green plate stay in state q_1	Go to state q_2
	q_2	Remove top plate stay in q_2	—	—
Green	q_1	Add blue plate stay in state q_1	Add green plate stay in q_1	Go to state q_2
	q_2	—	Remove top plate stay in q_2	—
Red	q_1	Add blue plate stay in state q_1	Add green plate stay in q_1	Go to state q_2
	q_2	Without next input (input is ε)	waiting remove	for the top plate

+) formal description is given as

PDA $M = (\{q_1, q_2\}, \{0, 1, c\}, \{R, B, G\}, \delta, q_1, R, \phi)$

δ defined as

$\delta(q_1, 0, R) = (q_1, BR)$

$\delta(q_1, 0, B) = (q_1, BB)$

$\delta(q_1, 0, G) = (q_1, BG)$

$\delta(q_1, c, R) = (q_2, R)$

$\delta(q_1, c, B) = (q_2, B)$

$\delta(q_1, c, G) = (q_2, G)$

$\delta(q_1, 1, R) = (q_1, GR)$

$\delta(q_1, 1, B) = (q_1, GB)$

$\delta(q_1, 1, G) = (q_1, GG)$

$\delta(q_2, 0, B) = (q_2, \varepsilon)$

$\delta(q_2, 1, G) = (q_2, \varepsilon)$

$\delta(q_2, \varepsilon, R) = (q_2, \varepsilon)$

Transition table for δ

State	Top of stack symbol	0	1	c
q_1	R	(q_1, BR)	(q_1, GR)	(q_2, R)
	B	(q_1, BB)	(q_1, GB)	(q_2, B)
	G	(q_1, BG)	(q_1, GG)	(q_2, G)
q_2	B	(q_2, ε)	—	—
	G	—	(q_2, ε)	—
	R	on ε, PDA removes the Red plate		

eg: Consider a step 01c10

$(q_1, 01c10, R) \vdash (q_1, 1c10, BR)$
$\vdash (q_1, c10, GBR) \vdash (q_2, 0, BR)$
$\vdash (q_2, 10, GBR) \vdash (q_2, \varepsilon, \varepsilon)$
$\vdash (q_2, \varepsilon, R)$

ie acceptance by empty stack

(2) Design a NPDA to recognize $L = \{ww^R \mid w \in (0+1)^*\}$

There are 2 choices of moves. The machine is in q_0 state until middle of string is reached. middle string is read, then M enters to q_1 & tries to match remaining i/p symbols with contents of stack

$M = (\{q_0, q_1\}, \{0,1\}, \{R, B, G\}, \delta, q_0, R, \phi)$

δ defined as

(1) $\delta(q_0, 0, R) = (q_0, BR)$

(2) $\delta(q_0, 1, R) = (q_0, GR)$

(3) $\delta(q_0, 0, B) = \{(q_0, BB), (q_1, \varepsilon)\}$

(4) $\delta(q_0, 0, G) = (q_0 BG)$

(5) $\delta(q_0, 1, B) = (q_0, GB)$

(6) $\delta(q_0, 1, G) = \{(q_0, GG), (q_1, \varepsilon)\}$

(7) $\delta(q_1, 0, B) = (q_1, \varepsilon)$

(8) $\delta(q_1, 1, G) = (q_1, \varepsilon)$

(9) $\delta(q_0, \varepsilon, R) = (q_1, \varepsilon)$

(10) $\delta(q_1, \varepsilon, R) = (q_1, \varepsilon)$

State	TOS symbol	0	1	ε
q_0	R	(q_0, BR)	(q_0, GR)	(q_1, ε)
	B	$\{(q_0, BB), (q_1, \varepsilon)\}$	(q_0, GB)	—
	G	(q_0, BG)	$\{(q_0, GG), (q_1, \varepsilon)\}$	—
q_1	B	(q_1, ε)	—	—
	G	—	(q_1, ε)	—
	R	—	—	(q_1, ε)

eg Consider a string 011110

$(q_0, 011110, R)$ ⊢

$(q_0, 11110, BR)$ ⊢

$(q_0, 1110, GBR)$ → $(q_1, 110, BR)$

$(q_0, 110, GGBR)$ → $(q_1, 10, GBR)$ ⊢

$(q_0, 10, GGGBR)$ → $(q_1, 0, GGBR)$ $(q_1, 0, BR)$ ⊢

$(q_0, 0, GGGGBR)$ (q_1, ε, R) ⊢

$(q_0, \varepsilon, BGGGGBR)$ $(q_1, \varepsilon, \varepsilon)$ ⊢ accept.

(Q) Construct a PDA to recognize all palindromes over {0,1}

Soln is same as ww^R with a slight difference. Any palindromes is of form $w0w^R$ or $w0w^R$ or ww^R. To accept $w0w^R$ or $w1w^R$, whenever the midpoint 0 or 1 is encountered, PDA has to change from q_0 to q_1 without changing the stack. But as midpoint cannot be determined, PDA is given a choice to change over from q_0 to q_1 for each 0 or 1.

Sol: include the foll changes in the above problem

$\delta(q_0, 0, R) = \{(q_0, BR), (q_1, R)\}$

$\delta(q_0, 1, R) = \{(q_0, GR), (q_1, R)\}$

$\delta(q_0, 0, B) = \{(q_0, BB), (q_1, B), (q_1, \varepsilon)\}$

$\delta(q_0, 0, G) = \{(q_0, BG), (q_1, G)\}$

$\delta(q_0, 1, B) = \{(q_0, GB), (q_1, B)\}$

$\delta(q_0, 1, G) = \{(q_0, GG), (q_1, G), (q_1, \varepsilon)\}$

State	Top symbol	0	1	ε
q_0	R	$\{(q_0,BR),(q_1,R)\}$	$\{(q_0,GR),(q_1,R)\}$	(q_1,ε)
	B	$\{(q_0,BB),(q_1,\varepsilon),(q_1,B)\}$	$\{(q_0,GB),(q_1,B)\}$	—
	G	$\{(q_0,BG),(q_1,G)\}$	$\{(q_0,GG),(q_1,G),(q_1,\varepsilon)\}$	—
q_1	B	(q_1,ε)	—	—
	G	—	(q_1,ε)	—
	R	—	—	(q_1,ε)

eg $(q_0, 00100, R) \rightarrow (q_1, 0100, R)$

$\downarrow$

$(q_0, 0100, BR) \rightarrow (q_1, 100, BR)$

$\searrow (q_1, 100, R)$

$\downarrow$

$(q_0, 100, BBR) \searrow (q_1, 00, BBR)$

$\downarrow$

$(q_0, 00, GBBR) \searrow (q_1, 0, BR)$

$\downarrow$

$(q_0, 0, BGBBR) \quad (q_1, 0, GBBR) \quad (q_1, \varepsilon, R)$

$\downarrow$

$(q_0, \varepsilon, BBGBBR) \quad (q_1, 0, BGBBR) \quad (q_1, \varepsilon, \varepsilon)$

$(q_1, \varepsilon, GBBR)$

$\downarrow$

Accept

(4) Construct a PDA for language $L = \{ w \in \{a,b\}^+ \mid n_a(w) = n_b(w)$ such that $L = L(M) \}$.

~~Whenever~~ a is read first push A to stack then when b is read pop A ~~out~~ ~~of stack~~.

~~Otherwise~~ when b is read first push B to stack then when a is read pop B.

$M = (\{q_0\}, \{a,b\}, \{A, B, z_0\}, \delta, q_0, z_0, \phi)$.

where δ defined as :

$\delta(q_0, \varepsilon, z_0) = (q_0, \varepsilon)$
$\delta(q_0, a, z_0) = (q_0, Az_0)$
$\delta(q_0, b, z_0) = (q_0, Bz_0)$
$\delta(q_0, a, A) = (q_0, AA)$
$\delta(q_0, b, B) = (q_0, BB)$
$\delta(q_0, a, B) = (q_0, \varepsilon)$
$\delta(q_0, b, A) = (q_0, \varepsilon)$

eg
$(q_0, abab, z_0)$
$\vdash (q_0, bab, Az_0)$
$\vdash (q_0, ab, z_0)$
$\vdash (q_0, b, Az_0)$
$\vdash (q_0, \varepsilon, z_0)$
$\vdash (q_0, \varepsilon, \varepsilon)$

(5) Design a PDA to recognize the language $L = \{a^n b^n \mid n \geq 1\}$:

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

where δ defined as :

$\delta(q_0, a, z_0) = (q_0, Az_0)$
$\delta(q_0, a, A) = (q_0, AA)$
$\delta(q_0, b, A) = (q_1, \varepsilon)$
$\delta(q_1, b, A) = (q_1, \varepsilon)$
$\delta(q_1, \varepsilon, z_0) = (q_1, \varepsilon)$

Here when a is read push A to stack.
when b is read after A ~~push~~ pop A.
Finally when all the strings are read stack contains Z_0 pop Z_0 and make the stack empty

eg $(q_0, aabb, Z_0) \vdash (q_0, abb, AZ_0)$
$\vdash (q_0, bb, AAZ_0) \vdash (q_1, b, AZ_0)$
$\vdash (q_1, \varepsilon, Z_0) \vdash (q_1, \varepsilon, \varepsilon)$

(b) Construct a PDA to recognize $L = \{a^n b^m c^{m+n} / n, m \geq 0\}$
Here stack symbol A used to count no of a's & b's
In start state q_0, PDA pushes A for each a. In state q_1 PDA scans b's & pushes one A for each b. On reading c, PDA goes to q_2 state while popping off one A
In q_2, it reads c & pops off one A for each c. At end if all A's are exhausted, PDA goes to final state q_2 making stack empty

$M = (\{q_0, q_1, q_2, \cancel{q_3}\}, \{a, b, c\}, \{A, Z_0\}, \delta, q_0, Z_0, \cancel{F}\ \phi\}$

δ defined as

$\delta(q_0, \varepsilon, Z_0) = (q_2, \varepsilon)$
$\delta(q_0, a, Z_0) = (q_0, AZ_0)$
$\delta(q_0, a, A) = (q_0, AA)$
$\delta(q_0, b, Z_0) = (q_1, AZ_0)$
$\delta(q_0, b, A) = (q_1, AA)$
$\delta(q_1, b, A) = (q_1, AA)$
$\delta(q_1, c, A) = (q_2, \varepsilon)$
$\delta(q_2, c, A) = (q_2, \varepsilon)$
$\delta(q_2, \varepsilon, Z_0) = (q_2, \varepsilon)$ $\delta(q_1, b, Z_0) = (q_1, AZ_0)$
~~$\delta(q_0, \varepsilon, Z_0)$~~ = ~~$(q_2, Z_0)$~~
$\delta(q_0, c, A) = (q_2, \varepsilon)$

eg: $(q_0, abbccc, z_0) \vdash (q_0, bbccc, Az_0) \vdash (q_1, ccc, AAAz_0)$

$\vdash (q_1, bccc, AAz_0)$

$\vdash (q_2, cc, AAz_0) \vdash (q_2, c, Az_0)$

$\vdash (q_2, \varepsilon, z_0) \vdash (q_3, \varepsilon, \varepsilon)$

Accept.

(7) Construct a PDA to recognize $L = \{a^i b^j c^k \mid j = i+k, i \geq 0, k \geq 0\}$.

Here 2 stack symbols A & B used.
Symbol A used to count no: of a's.
B is used to count excess no: of b's.
In state q_0, PDA pushes A for each a.
In state q_1 as long as A is in stack, PDA pops off one A for each b. When A's are exhausted, PDA starts pushing one B for each b.
On encountering c, PDA goes to q_2.
In q_2 it pops off one B for each c.
At end, when B's are exhausted and z_0 is exposed, it ~~goes to state~~ makes stack empty and accepts L.

$M = (\{q_0, q_1, q_2\}, \{a, b, c\}, \{A, B, z_0\}, \delta, q_0, z_0, \phi)$

where δ defined as

$\delta(q_0, a, z_0) = (q_0, Az_0)$

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

$\delta(q_0, b, z_0) = (q_1, Bz_0)$

$\delta(q_0, b, A) = (q_1, \varepsilon)$

$\delta(q_1, b, A) = (q_1, \varepsilon)$

$\delta(q_1, b, z_0) = (q_1, Bz_0)$

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

$\delta(q_1, c, B) = (q_2, \varepsilon)$

$\delta(q_2, c, B) = (q_2, \varepsilon)$

$\delta(q_2, \varepsilon, z_0) = (q_2, \varepsilon)$ $\quad \delta(q_1, \varepsilon, z_0) = (q_2, \varepsilon)$.

$\delta(q_0, \varepsilon, z_0) = (q_2, \varepsilon)$

eg $(q_0, aabbbc, z_0) \vdash (q_0, abbbc, Az_0)$

$\vdash (q_0, bbbc, AAz_0) \vdash (q_1, bbc, Az_0)$

$\vdash (q_1, bc, z_0) \vdash (q_1, c, Bz_0)$

$\vdash (q_2, \varepsilon, z_0) \vdash (q_2, \varepsilon, \varepsilon)$

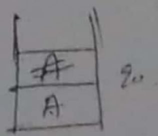
$\downarrow$ accept

(8) Construct a PDA to recognize $\{a^n b^{2n} / n \geq 0\}$.

Here we keep count of twice the no: of a's to be matched by b's. Here in start state q_0, push A's for each a. On reading i/p b, PDA ~~pops off~~ first goes to q_1 and goes to q_0 again. next b pops off one A and goes to q_1. At the end, PDA if all A's are exhausted and z_0 is exposed goes to accepting state making stack empty

$M = (\{q_0, q_1\}, \{a, b\}, (A, z_0), \delta, q_0, z_0, \emptyset\})$

where δ defined as



$\delta(q_0, a, z_0) = (q_0, Az_0)$

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

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

$\delta(q_1, b, A) = (q_0, \varepsilon)$

$\delta(q_1, \varepsilon, z_0) = (q_0, \varepsilon)$

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

Scanned by CamScanner

eg $(q_0, abb, z_0) \vdash (q_0, bb, AZ_0)$
$\vdash (q_1, b, AZ_0) \vdash (q_0, \varepsilon, z_0) \vdash (q_0, \varepsilon, \varepsilon)$

(9) Construct a PDA to recognize $L = \{a^{2n}b^n / n \geq 0\}$.

Here we have to keep count of half the no: of a's to be matched by b's. While scanning a's, we use stack symbol A and push one A for each aa. For this, initial phase divided to 2 states q_0 & q_1. In q_0, PDA pushes one A for each a and goes to q_1. In q_1, after reading the next a, it switches back to q_0 with no change in stack. In q_0, if it scans the i/p b, PDA goes to q_2. In q_2, it pops off one A for each b. At the end when all A's are exhausted and z_0 is exposed make the stack empty & PDA reaches the acceptance state

$M = (\{q_0, q_1, q_2\}, \{a, b\}, \{A, z_0\}, \delta, q_0, z_0, \phi)$ $aa \to b$

where δ defined as

$\delta(q_0, a, z_0) = (q_1, Az_0)$
$\delta(q_0, a, A) = (q_1, AA)$
$\delta(q_1, a, A) = (q_0, A)$
$\delta(q_0, b, A) = (q_2, \varepsilon)$
$\delta(q_2, b, A) = (q_2, \varepsilon)$
$\delta(q_2, \varepsilon, z_0) = (q_2, \varepsilon)$
$\delta(q_0, \varepsilon, z_0) = (q_2, \varepsilon)$

eg: $(q_0, aaaabb, z_0) \vdash (q_1, aaabb, Az_0)$
$\vdash (q_0, aabb, Az_0) \vdash (q_1, abb, AAz_0)$
$\vdash (q_0, bb, AAz_0) \vdash (q_2, b, Az_0)$
$\vdash (q_2, \varepsilon, z_0) \vdash (q_2, \varepsilon, \varepsilon)$
accept

(10) Construct a PDA to recognize $L = \{a^i b^j c^k \mid i \neq j\}$.

Idea: In start state q_0, PDA pushes one A for each a. In state q_1 when it reads b's it pops off A for each b. The process ends when
(1) when A's are exhausted
(2) b's are exhausted while A's are still in stack
Case 1 indicates $i < j$. Then PDA goes to q_2 to skip b's and then goes to q_3 when c is encountered
Case 2 $i > j$. Then PDA goes to state q_3 and halts
In q_3, PDA simply skips all c's and accepts the input at the end of i/p

$m = (\{q_0, q_1, q_2, q_3\}, \{a, b, c\}, \{A, z_0\}, \delta, q_0, z_0, \phi)$

where δ defined as

$\delta(q_0, a, z_0) = (q_0, Az_0)$
$\delta(q_0, a, A) = (q_0, AA)$
$\delta(q_0, b, A) = (q_1, \varepsilon)$
$\delta(q_0, b, z_0) = (q_2, z_0)$
$\delta(q_1, b, A) = (q_1, \varepsilon)$
$\delta(q_1, b, z_0) = (q_2, z_0)$
$\delta(q_1, c, A) = (q_3, A)$

$\delta(q_2, b, z_0) = (q_2, z_0)$

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

$\delta(q_3, c, A) = (q_3, A)$ $\quad \delta(q_3, \varepsilon, A) = (q_3, \varepsilon)$

~~$\delta(q_3, c, z_0) = (q_3, z_0)$~~

$\delta(q_3, \varepsilon, z_0) = (q_3, \varepsilon)$.

eg. $(q_0, abbc, z_0) \vdash (q_0, bbc, Az_0)$
$\vdash (q_1, bc, z_0) \vdash (q_2, c, z_0)$
$\vdash (q_3, \varepsilon, z_0) \vdash (q_3, \varepsilon, \varepsilon)$

$(q_0, aabc, z_0) \vdash (q_0, abc, Az_0)$
$\vdash (q_0, bc, AAz_0) \vdash (q_1, c, Az_0) \vdash (q_3, \varepsilon, z_0)$
$\vdash (q_3, \varepsilon, \varepsilon)$

Extra Questions.

P. PDA to recognize $L = \{a^i b^j \mid i < j \text{ and } i, j \geq 1\}$

Q PDA to recognize $L = \{a^i b^j \mid i > j \text{ and } i, j \geq 1\}$

Q PDA to " $L = \{a^n b^m a^n \mid m, n \geq 1\}$