4th UNIT PDA

Date ___/___/___

Check given Language is CFL OR NOT :-

1. $L = \{a^n b^n c^m \mid n > m\}$
2. $L = \{a^{m+n} b^n c^n \mid m, n \geq 1\}$ → DCFL, CFL
3. $L = \{a^m b^m c^n d^n \mid m, n \geq 1\}$ → DCFL, CFL
4. $L = \{a^m b^n c^m d^n \mid m, n \geq 1\}$ → ~~DCFL~~, ~~CFL~~ NOT CF
5. $L = \{a^m b^i c^m d^j \mid m, i, j \geq 1\}$ → CFL, DCFL
6. $L = \{a^i b^j c^k \mid k = i + j\}$ → DCFL, CFL
7. $L = \{a^i b^j c^k d^l \mid i = k, j = l\}$ → not DCFL, not CFL
8. $L = \{a^n b^{2n} \mid n \geq 1\}$ → DCFL, CFL
9. $L = \{a^i b^j c^k d^l \mid i, j, k, l \geq 1\}$ DCFL, CFL
10. $L = \{a^n b^n c^n \mid n \geq 1\}$ not DCFL; not CFL

11. $L = a^n b^n c^n d^n \mid n \geq 1$ not DCFL, NOT CFL
12. $L = wcw^R \mid w \in \{a, b\}^+$ DCFL, CFL
13. $L = ww^R \mid w \in \{a, b\}^+$ ~~DCFL~~, CFL only
    b b a
14. $L = WW \mid w \in \{a, b\}^+$ $w = abb$, $ww = abbabb$ NOT CF
15. $L = a^{n^2} \mid n \geq 1 \Rightarrow a, a^4, a^9, a^{16}$ --- not in AP so not DCFL
16. $L = a^{2^n} \mid n \geq 1 \Rightarrow a^2, a^4, a^8,$ --- not in AP so not D, C
17. $L = \cancel{a^n b^{n^2} \mid n \geq 1}$ $a^n b^{n^2} \mid n \geq 1$ not DCFL, not CF
18. $L = a^n b^{2n} \mid n \geq 1$ not DCFL, not CFL
19. $L = a^n b^{2n} c^{3n} \mid n \geq 1$ not DCFL, not CFL
20. $L = n_a(w) = n_b(w) = n_c(w) \mid w \in \{a, b\}^+$ NOT DCFL, CF

Que:- Check give Language CFL or not $L = a^p \mid p$ is prime

Pumping Lemma [For Context Free Language]

* Pumping Lemma [for CFL] is used to prove that a language is NOT context free

Context Free Language:-

In formal Language theory a context free language is a language generated by some context free grammar.

The set of all CFL is identical to the set of Languages accepted by pushdown Automata

context free Grammar is identified by 4 tuples as
$G = \{V, \Sigma, S, P\}$ where
V= set of variables or Non-terminal
Σ= set of Terminal symbol
S = start symbol
P = Production Rule

context free grammar has production Rule of the form.

$$\boxed{\begin{array}{l} A \to a \\ \text{where, } a = \{V \cup \Sigma\}^* \quad \& \quad A \in V \end{array}}$$

* If A is a context free Language, then, A has a Pumping Length "P" such that any string 'S' where $|S| \geq P$ may be divided into 5 pieces $S = uvxyz$ such that the following conditi

conditions must be true:

(1) uv^ixy^iz is in A for every $i \geq 0$

(2) $|vy| > 0$

(3) $|vxy| \leq p$

To Prove that a Language is Not context free using Pumping Lemma [For CFL] follow the steps given below:-

[We Prove using CONTRADICTION]

⇒ Assume that A is context free
⇒ It has to have a Pumping Length (say P)
⇒ All strings longer than P can be pumped $|S| \geq P$
⇒ Now find a string 'S' in such that $|S| \geq P$
⇒ Divide S into 5 parts UVXYZ
⇒ Show that $uv^ixy^iz \notin A$ for some i
⇒ Then consider the ways that S can be divided into uvxyz.
⇒ Show that none of these can satisfy all the 3 pumping condition at the same time
⇒ S cannot be pumped == CONTRADICTION

Que:- Show that $L = \{a^nb^nc^n \mid n \geq 0\}$ is context free.

take a string S such that $S = a^Pb^Pc^P$
Eg: $P=4$

so $S = a^4b^4c^4$

case 1: v & y each contain only one type of

Date ___/___/___

$$L = \underbrace{a}_{u}\underbrace{aa}_{v}\underbrace{abbbbc}_{x}\underbrace{c}_{y}\underbrace{cc}_{z}$$

(2) $|vy| > 0$, $|aac| = 3$ ∴ Satisfy

(3) $|vxy| \le p$ $\boxed{aaabbbbcc \quad |9| \le 4}$ not satisfy

(1) $uv^i xy^i z$ $i = 2$

uv^2xy^2z

$a(aa)^2 abbbbc(c)^2cc$

$a\,aaaa\,abbbb\,c\,cccc$

$a^6 b^4 c^5 \ne L$

Case 2:- Either v or y has more than one kind of symbol.

$$L = \underbrace{aa}_{u}\underbrace{aabbb}_{v}\underbrace{b}_{x}\underbrace{c}_{y}\underbrace{ccc}_{z}$$

(i) $|aabbb| = 5$ $|vy| > 0$.

(ii) $|aabb| \le 4$

(iii) $uv^ixy^iz = aa\,(aabb)^2\,b\,cccc$

$\Rightarrow aa\,aabbaabbb\,cccc \notin L$.

$\Rightarrow a^4 b^2 a^2 b^3 c^4$ not CFG

Que Show that $L = \{ ww \mid w \in \{0,1\}^* \}$ is Not Context free.

Date __/__/__

⇒ Assume that L is Context free

⇒ L must have a pumping length (Say P)

⇒ Now we take a string S such that

$S = 0^P 1^P 0^P 1^P$

⇒ We divides S into parts uvxyz

Eg $P = 5$

$S = 0^5 1^5 0^5 1^5$

CASE 1:- vxy does not straddle a boundary

$\underbrace{00000 1}_{u}\,\underbrace{111}_{vxy}\,\underbrace{1\,00000\,11111}_{z}$

$|vy| > 0$ $|11| > 0$ ✓

$|vxy| \le P$ $|111| \Rightarrow \underset{\text{Length}}{|3|} \le |5|$ which P Length

$uv^i xy^i z$ $i = 2$

$uv^2 xy^2 z$

$000001[1]^2 1[1]^2 100000 11111$

0000011111110000011111

$0^5 1^7 0^5 1^5$

Here 1st half not equal to 2nd half

So

$\boxed{0^5 1^7 0^5 1^5 \notin L}$

CASE 2a :- vxy straddles the first boundary

00000'11111'00000'11111

u vxy z

$|vy| > 0$

$|vxy| \leq p \qquad |5| \leq 5$

$uv^ixy^iz \Rightarrow uv^2xy^2z$

$i = 2$

$000[00]^2 1[11]^2 11 00000 11111$

$0000000 1111111 00000 11111$

$0^7 1^7 0^5 1^5$

So 1st half not equal to 2nd Half.

$\boxed{0^7 1^7 0^5 1^5 \notin L}$

CASE 2b :- vxy straddles the third boundary.

00000'11111'00000'11111

u vxy z

uv^2xy^2z

$00000 11111 0000000 1111111$

$0^5 1^5 0^7 1^7$

So 1st half not equal to 2nd Half.

$\boxed{0^5 1^5 0^7 1^7 \notin L}$

Date ___ / ___ / ______

CASE 3(d):- vxy straddles the midpoint.

$$\underbrace{00000'1 1 1}_{u}\underbrace{1 1 1'0 0}_{vxy}\underbrace{0 0 0'1 1 1 1 1}_{z}$$

uv^2xy^2z

00000 1111111 0000000 11111

$\underbrace{0^5 1^7}\ \underbrace{0^7 0^5}$

1st half not equal to 2nd half

$0^5 1^7 0^7 0^5 \notin L$

In all 4 cases our condition not satisfy so our language is not context free

Date ___ / ___ / ___

PUSHDOWN AUTOMATA

A PDA is a way to implement a context free grammar in a similar way we design finite automata for regular grammar.

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

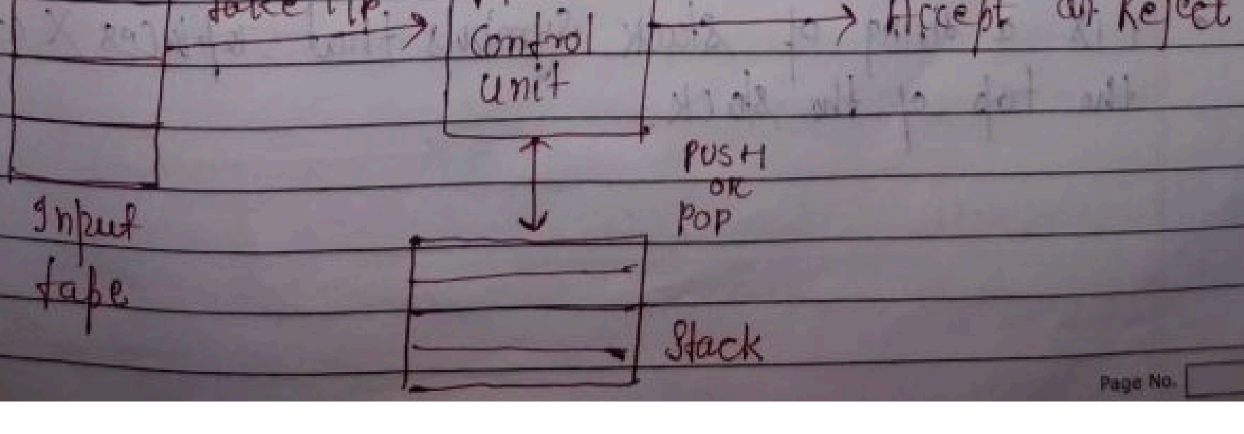
Stack:- It is a data structure, a stack is a way we arrange elements one on top of another.

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 & removed.

Pushdown Automata has 3 components:-

1. Input tape
2. A finite Control
3. A Stack with infinite size.



Date __/__/__

PDA:-

A PDA is formally defiend by 7 tuple as shown.

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

Where,

$Q \Rightarrow$ A finite set of symboles
$\Sigma \Rightarrow$ A finite set of input symbols
$\Gamma \Rightarrow$ A finite Stack Alphabet
$\delta \Rightarrow$ The Transition function
$q_0 \Rightarrow$ The start state
$Z_0 \Rightarrow$ The start stack symbol
$F \Rightarrow$ 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 new state
- Y is a string of stack symbols that replaces X at the top of the stack.

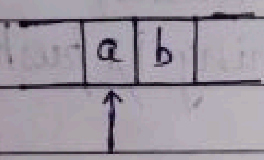
Eg: If $Y = \epsilon$ then the stack is popped.

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

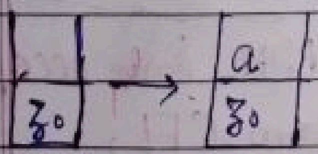
If $Y = YZ$ then X is replaced by Z and Y is pushed onto the stack

* Basic operation on stack:

① PUSH



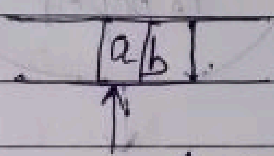
$$q_i \xrightarrow{a, z_0/az_0} q_j$$



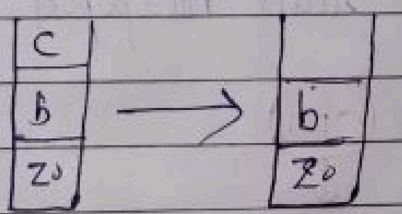
Stack

$$\delta(q_i, a, z_0) \rightarrow (q_j, az_0)$$

② POP

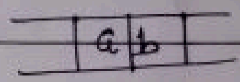


$$q_i \xrightarrow{a, c/\epsilon} q_j$$

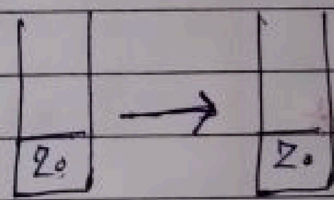


$$\delta(q_i, a, c) \rightarrow (q_j, \epsilon)$$

③ SKIP
(no push & no pop)



$$q_i \xrightarrow{a, z_0/z_0} q_j$$



$$\delta(q_i, a, z_0) \rightarrow (q_j, z_0)$$

PUSHDOWN Automata {Graphical Notation}

Finite state Machine

$$A \xrightarrow{a} B$$

Date ___/___/______

PUSH down Automata

(A) —— a, b → c ——→ (B)

a: Input symbol
Maybe ∈

b: Symbol on Top of the stack, This symbol is poped
'∈' means the Stack is neither read nor poped

c: This Symbol is pushed onto the stack
'∈' means nothing is pushed

* CFG → CFL → PDA

CFL → DPDA: for some
CFL → NPDA: for all
PDA → DPDA, NPDA

DPDA only accept Some CFL, which is known as Deterministic CFL, [DCFL]

NPDA accepts all CFL

DPDA → Determinstic Pushdown Automaton.
NPDA → Nondeterminstic PDA.

$\boxed{P(NPDA) > P(DPDA)}$

(NPDA is more powerful from DPDA)

Date ___/___/___

Transition function of DPDA & NPDA

DPDA:-

$$\boxed{\delta : Q \times \Sigma \times \Gamma \rightarrow Q \times \Gamma^*}$$

NPDA:-

$$\boxed{\delta : Q \times \Sigma \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}}$$

Eg:- Let PDA $A = \{Q, \Sigma, T, \delta, q_0, Z_0, F\}$

$Q = \{q_0, q_1, q_f\}$ $\Sigma = \{a, b\}$ $T = \{a, Z_0\}$ $F = \{q_f\}$

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

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

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

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

$\delta(q_1, \epsilon, Z_0) = \{(q_0, \epsilon)\}$

input string $w = aabb$

Solⁿ: $\delta(q_0, aabb, Z_0) \vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$ (Here 'a' will be poped)

$\vdash (q_1, b, aZ_0)$ (Here 'a' poped)

$\vdash (q_1, \epsilon, Z_0)$

$\vdash (q_0, \epsilon, \epsilon)$

(String accepted)

→ String accepted case:-

Eg:- ~~stack~~

In this example we are on initial state but stack is empty.

Date ___ /___ /______

Eg: Let input string aaabbb

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \Lambda)$
$\delta(q_1, b, a) = (q_1, \Lambda)$
$\delta(q_1, \Lambda, Z_0) = (q_f, Z_0)$

$\rightarrow (q_0, aaabbb, Z_0) \vdash (q_0, aabbb, aZ_0)$
$\vdash (q_0, abbb, aaZ_0)$
$\vdash (q_0, bbb, \cancel{a}aaZ_0)$
$\vdash (q_1, bb, \cancel{a}aZ_0)$
$\vdash (q_1, b, \cancel{a}Z_0)$
$\vdash (q_1, \Lambda, Z_0)$
$\vdash (q_f, \Lambda, Z_0)$

In this example we are on final state but stack is non empty.

PDA Acceptance string:-

- String is accepted by the PDA when we get to the acceptance state and see that the stack is empty.
- PDA can accept by accepting both empty state & empty stack.

Date ___/___/___

⇒ Graphical representation of PDA

Eg:-

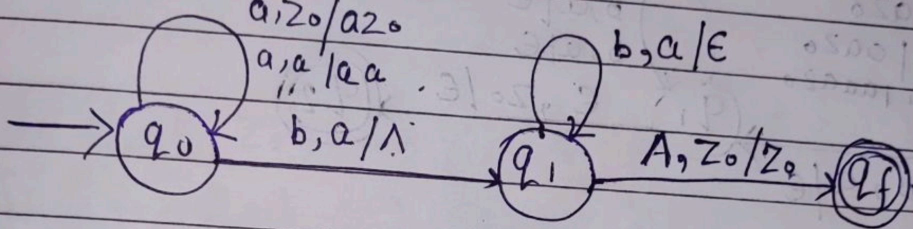
$\delta(q_0, a, Z_0) = (q_0, aZ_0)$

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

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

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

$\delta(q_1, \Lambda, Z_0) = (q_f, Z_0)$



Que Construct the PDA for the language $L = \{a^n b^n \mid n \geq 1\}$

$M = \{Q, \Sigma, \Gamma, \delta, q_0, Z_0, F\}$

Step 1:-

Initially push all a's onto ~~to~~ the Stack

Step 2:-

Whenever "b" occurs change the state & pop "a" from the stack.

Step 3:-

Repeat step 2 until stack is empty.

$L = \{ab, aabb, aaabbb \ldots\}$

a	a	a	b	b	b	ε

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

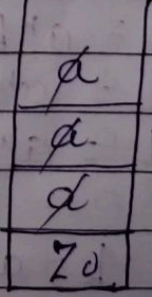
$\delta(q_0, a, a) = (q_0, aaZ_0)$

$\delta(q_0, a, a) = (q_0, aaaZ_0)$

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

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

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



Date __/__/____

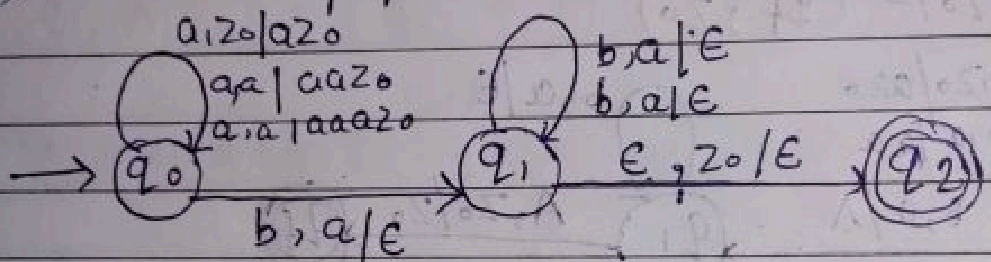
$\delta(q_1, \epsilon, Z_0) = (q_2, \epsilon)$

$\boxed{Z_0}$

↓

Here input string completely ended, so we will change the state.

Z_0 also pop, now stack becom empty.



Now check whether this string accept or not
$w = aabb$

$\delta(q_0, aabbZ_0) \vdash (q_0, abb, aZ_0)$
$\vdash (q_0, bb, aaZ_0)$
$\vdash (q_1, b, aZ_0)$
$\vdash (q_1, \epsilon, Z_0)$
$\vdash (q_2, \epsilon)$

↓ ↳

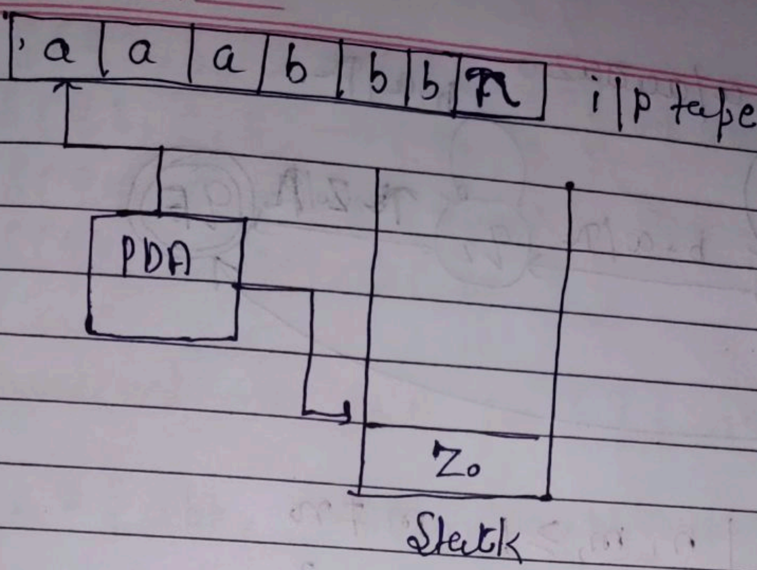
final state & stack is empty.

Design PDA for

(i) $L = a^n b^n \mid n \geq 0$
(ii) $L = a^n b^{2n} \mid n \geq 0$
(iii) $L = a^n b^n c^m \mid n, m, \geq 1,\ m \neq n$
(iv) $L = a^m b^n c^n \mid n, m \geq 1,\ m \neq n$

(i) $L = a^n b^n \mid n \geq 0$
$L = \{\lambda, ab, aabb, aaabbb \ldots\}$

Date ___/___/___



$\delta(q_0, \lambda, Z_0) \rightarrow (q_f, \lambda)$

$\delta(q_0, a, Z_0) \rightarrow (q_1, aZ_0)$

$\delta(q_1, a, a) \rightarrow (q_1, aaZ_0)$

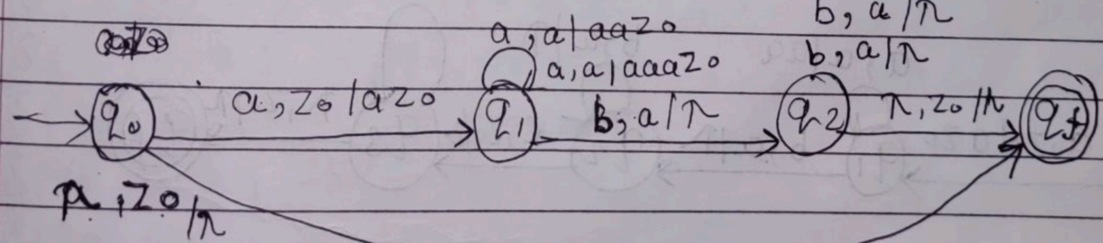
$\delta(q_1, a, a) \rightarrow (q_1, aaaZ_0)$

$\delta(q_1, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, \lambda, Z_0) \rightarrow (q_f, \lambda)$



(ii) $L = a^n b^{2n} \mid n \geq 0 \Rightarrow L = \{\lambda, abb, aabbbb, aaabbbbbb \ldots\}$

$\delta(q_0, \lambda, Z_0) \rightarrow (q_f, \lambda)$

$\delta(q_0, a, Z_0) \rightarrow (q_1, aaZ_0)$

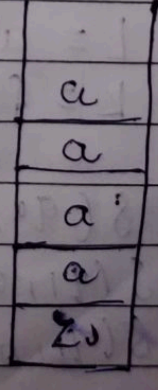
$\delta(q_1, a, a) \rightarrow (q_1, aaaaZ_0)$

$\delta(q_1, b, a) \rightarrow (q_2, \lambda)$

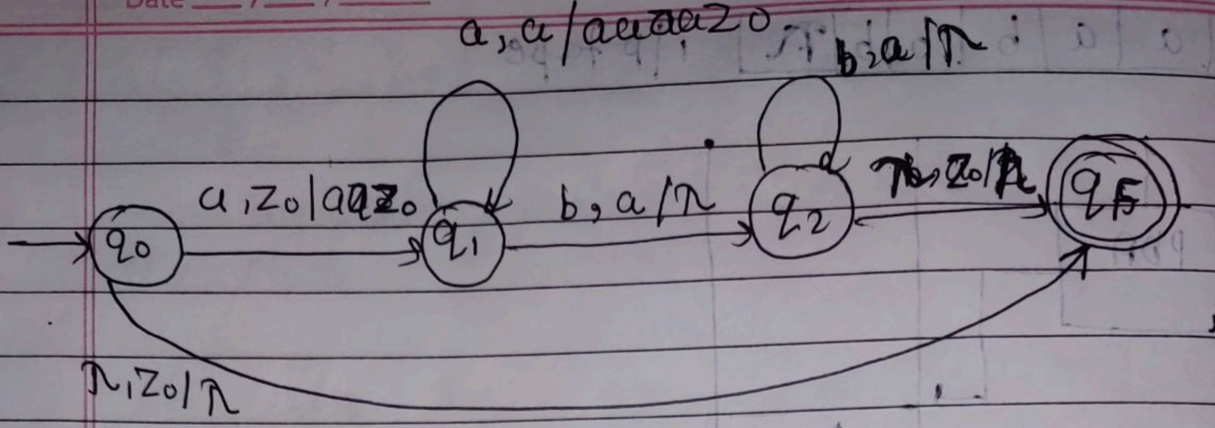
$\delta(q_2, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, \lambda, Z_0) \rightarrow (q_f, \lambda)$

farisinga
we insert two aa's into stack



Date __/__/__



(iii) $L = a^n b^n c^m \mid n, m \geq 1, m \neq n$

$L = \{ abcc, aabbccc \text{ ---- } \}$

$\delta(q_0, a, Z_0) \rightarrow (q_1, aZ_0)$

$\delta(q_1, a, a) \rightarrow (q_1, aa)$

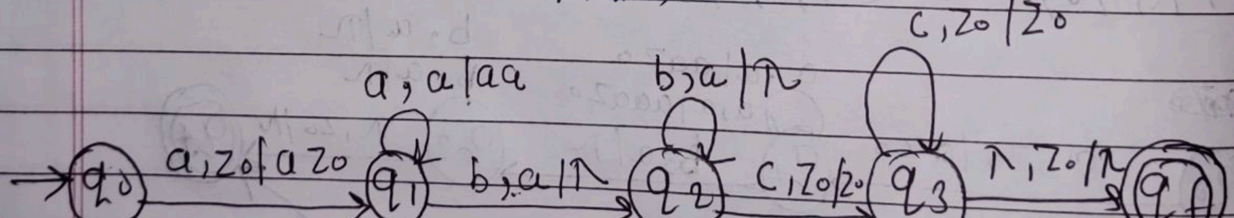
$\delta(q_1, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, c, Z_0) \rightarrow (q_3, Z_0)$

$\delta(q_3, c, Z_0) \rightarrow (q_3, Z_0)$

$\delta(q_3, \lambda, Z_0) \rightarrow (q_f, \lambda)$



(iv) $L = a^m b^n c^n \mid n, m \geq 1 \quad m \neq n$

$L = \{ abbcc, aabbbccc, aaabbbbcccc \text{ --- } \}$

$\delta(q_0, a, Z_0) \rightarrow (q_1 \; Z_0)$

$\delta(q_1, a, Z_0) \rightarrow (q_1, Z_0)$

$\delta(q_1, b, Z_0) \rightarrow (q_2, bZ_0)$

Date ___/___/___

$\delta(q2, b, b) \rightarrow (q2, bb)$

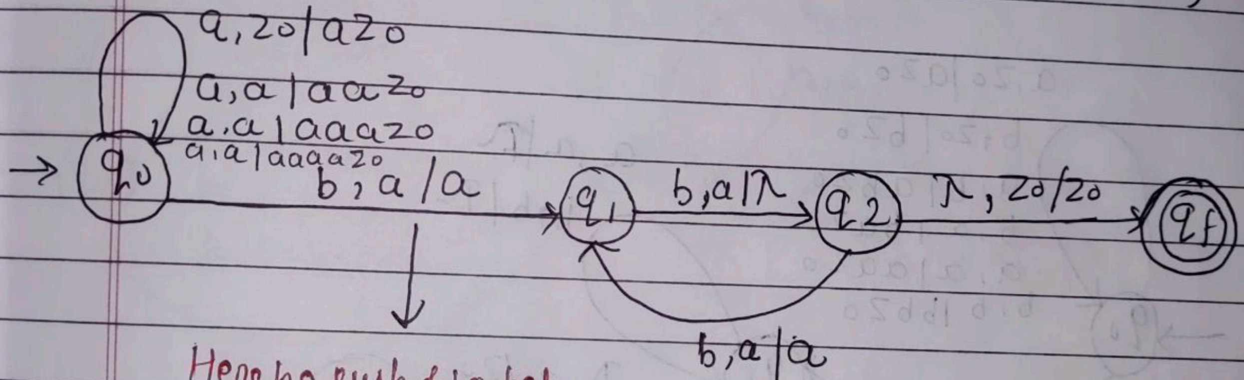
$\delta(q3, c, b) \rightarrow (q4, \lambda)$

$\delta(q4, c, b) \rightarrow (q4, \lambda)$

$\delta(q4, \lambda, z_0) \rightarrow (qf, \lambda)$

Que construct a PDA for $L = a^n b^{2n} \mid n \geq 1$

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



Here no push & no pop operation will perform

$\delta(q_0, a, z_0) \rightarrow (q_0, az_0)$

$\delta(q_0, a, a) \rightarrow (q_0, aaz_0)$

$\delta(q_0, a, a) \rightarrow (q_0, aaaz_0)$

$\delta(q_0, a, a) \rightarrow (q_0, aaaaz_0)$

$\delta(q_0, b, a) \rightarrow (q_1, aaaaz_0)$

$\delta(q_1, b, a) \rightarrow (q_2, \lambda)$

$\delta(q_2, b, a) \rightarrow (q_1, aaaz_0)$

$\delta(q_2, \lambda, z_0) \rightarrow (qf, z_0)$ accepted

Que construct PDA for the given $L = (wcw^r \mid w \in (a, b)^*$

$w = ab$	$w^r = ba$	$wcw^r = abcba$
$w = baba$	$w^r = abab$	$wcw^r = babacabab$
$w = a$	$w^r = a$	$wcw^r = aca$
$w = \epsilon$	$w^r = \epsilon$	$wcw^r = \epsilon c \epsilon = c$

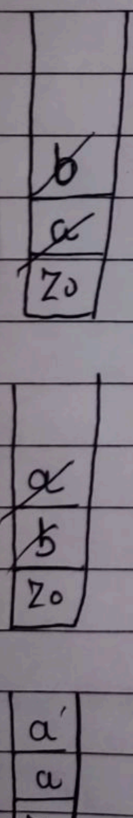
Here C will be decision making

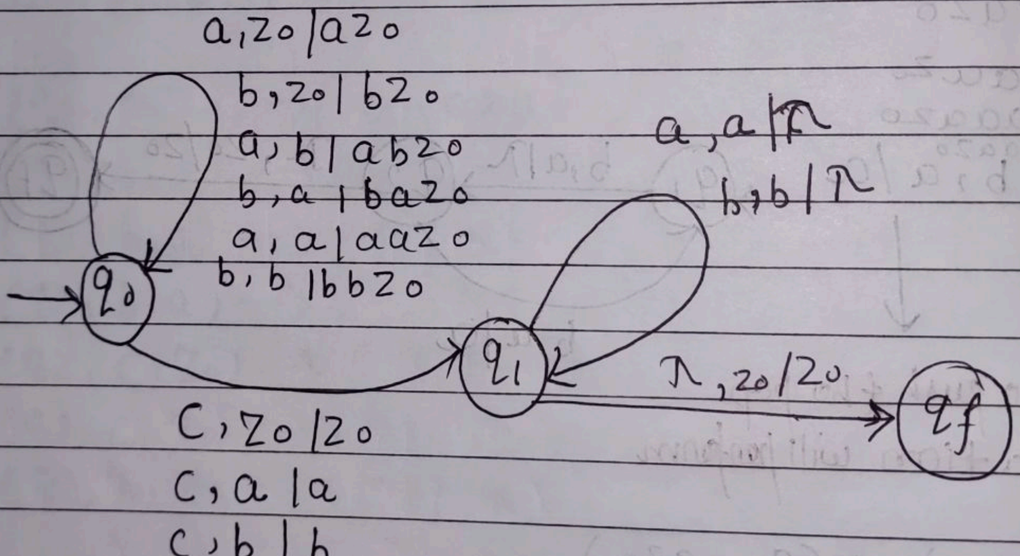
Date ___/___/_____

Language will be

L = {C, aca, bCb, abcba, babaCabab ___}

b a b a C a b a b





Date __/__/__

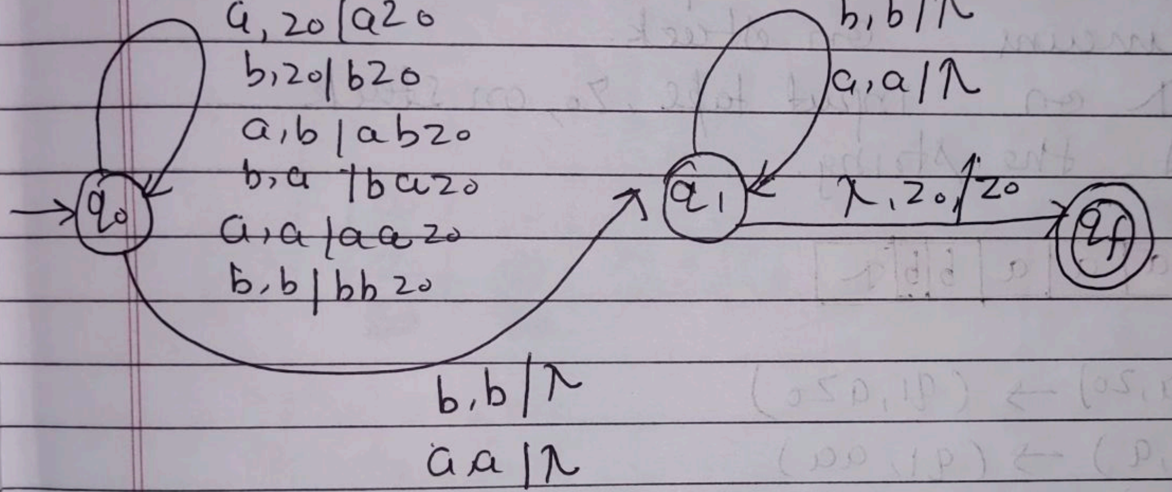
Ques construct PDA for $L = \{ww^r \mid w \in (a,b)^+\}$

$w = a \quad w^r = a \quad ww^r = aa$

$w = ab \quad w^r = ba \quad ww^r = abba$

$w = abab \quad w^r = baba \quad ww^r = ababbaba$

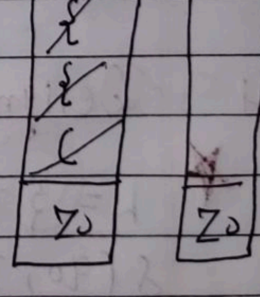
$L = \{aa, abba, ababbaba - - -\}$



Ques:- Design PDA for Balanced parenthese

({{ }})

- 1st push (into stack
- Now push all { into stack
- When } comes, pop each {
- when) comes, pop each (



$\delta(q_0, (, Z_0) \rightarrow (q_1, (Z_0)$

$\delta(q_1, \{, () \rightarrow (q_2, \{()$

$\delta(q_2, \{, \{) \rightarrow (q_2, \{\{()$

$\delta(q_2, \}, \{) \rightarrow (q_3, \lambda)$

$\delta(q_2, \}, \{) \rightarrow (q_3, \lambda)$

$\delta(q_3,), () \rightarrow (q_4, \lambda)$

$\delta(q_3, \lambda, Z_0) \rightarrow (q_f, \lambda)$

Date __/__/__

Qu Design PDA for $\{L = a^m b^n \quad m > n \geq 1.\}$

$L = \{aab, aaab, aaabb, \ldots\}$

⇒ Push all a's into stack
⇒ When b's comes, POP a's from stack
⇒ If empty string λ comes, and a's on stack, POP a's from stack till Z_0 remains on stack.
⇒ If λ on input tape, Z_0 on stack accept the string.

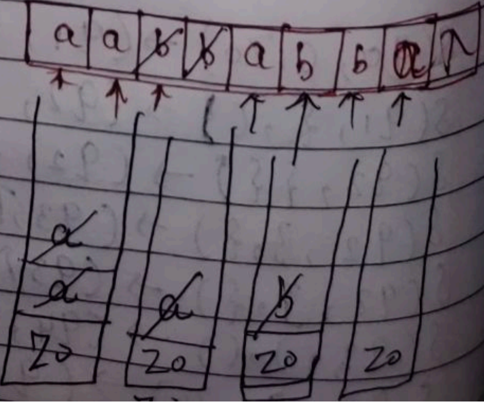
a	a	a	a	b	b	λ

$\delta(q_0, a, Z_0) \rightarrow (q_1, aZ_0)$
$\delta(q_1, a, a) \rightarrow (q_1, aa)$
$\delta(q_1, b, a) \rightarrow (q_2, \lambda)$
$\delta(q_2, b, a) \rightarrow (q_2, \lambda)$
$\delta(q_2, \lambda, a) \rightarrow (q_2, \lambda)$
$\delta(q_2, \lambda, Z_0) \rightarrow (q_f, \lambda)$

Qu Design PDA $L = \{n_a(w) = n_b(w) \quad w \in \{a, b\}^*$

★ $L = \{\lambda, a, b, ab, ba, aabb, bbaa \ldots\}$

$\delta(q_0, \lambda, Z_0) \rightarrow (q_f, \lambda)$
$\delta(q_0, a, Z_0) \rightarrow (q_0, aZ_0)$
$\delta(q_0, b, Z_0) \rightarrow (q_0, bZ_0)$
$\delta(q_0, a, a) \rightarrow (q_0, aa)$
$\delta(q_0, b, b) \rightarrow (q_0, bb)$
$\delta(q_0, a, b) \rightarrow (q_0, \lambda)$
$\delta(q_0, b, a) \rightarrow (q_0, \lambda)$



Date ___ / ___ / ___

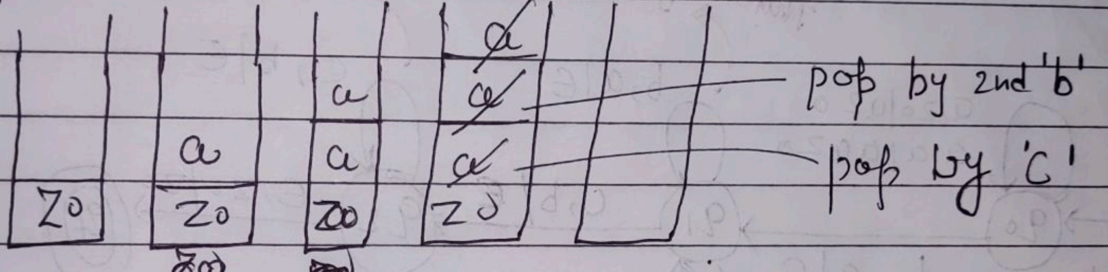
Qu Construct PDA $L = a^{m+n} b^m c^n \mid m, n \geq 1$

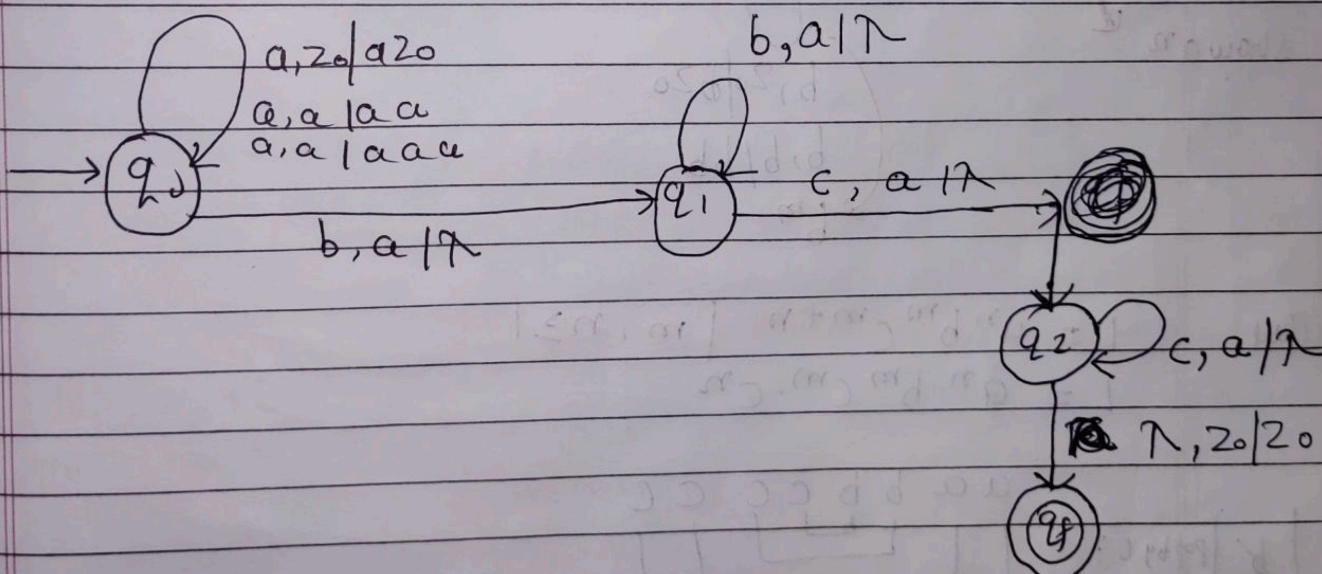
$\{m = 1, n = 1\} \{m = 2, n = 1\}$

$\{ L = aabc,\ aaabbc \ldots\ldots \}$

* In this all 'a's will be PUSH into stack & 'a' will be pop when insert 'b', & 'a' also pop when insert 'c'

Eg: a a a b b c





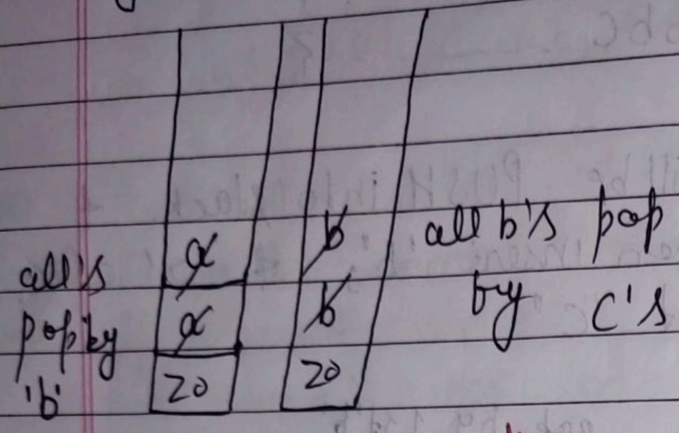
Qu $L = a^n b^{m+n} c^m \ [m, n \geq 1]$

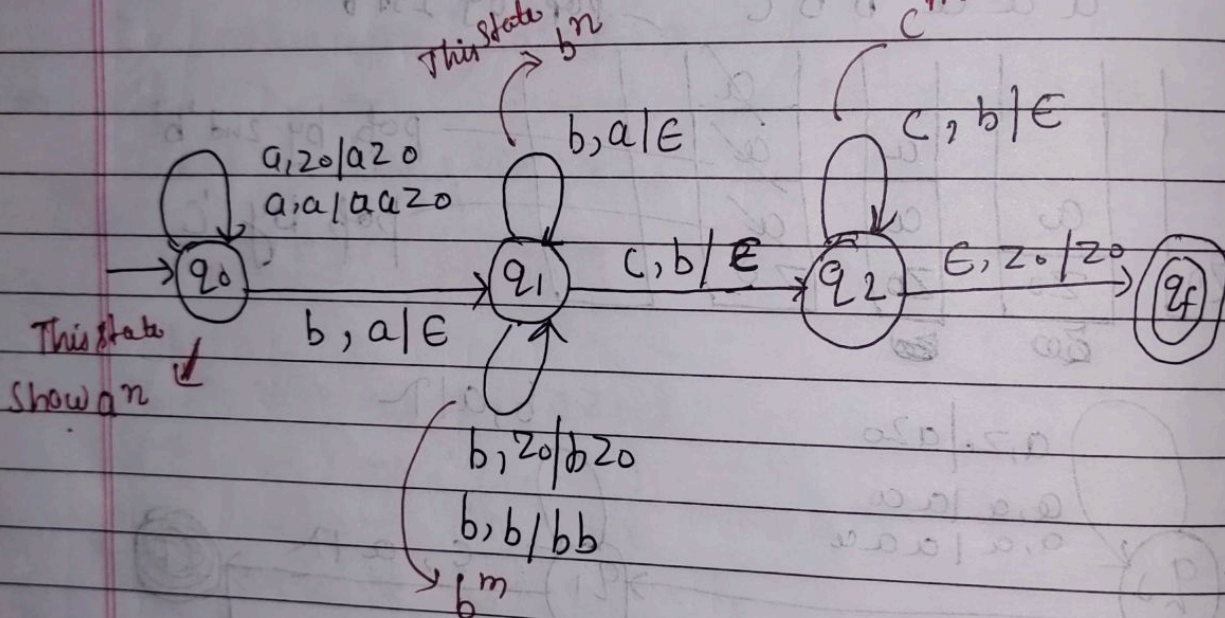
$= a^n b^n b^m c^m$

- all no a's (a^n) PUSH into stack & Pop by all no of b's (b^n)
- then all no of b's (b^m) pop by all no of c's (c^m)

Date ___/___/___

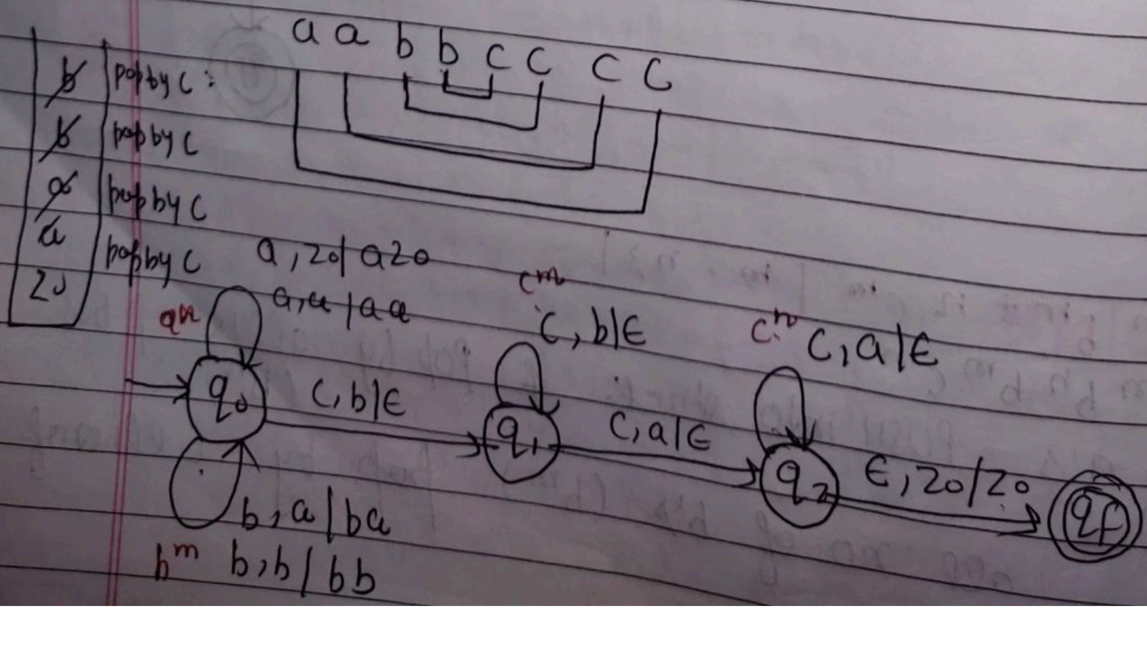
Eg $aa\cancel{b}\cancel{b}\cancel{b}\cancel{b}\cancel{c}\cancel{c}$





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

$L = a^n b^m c^m \cdot c^n$



$\omega \in \{a,b\}^*$

Date ___ / ___ / _____

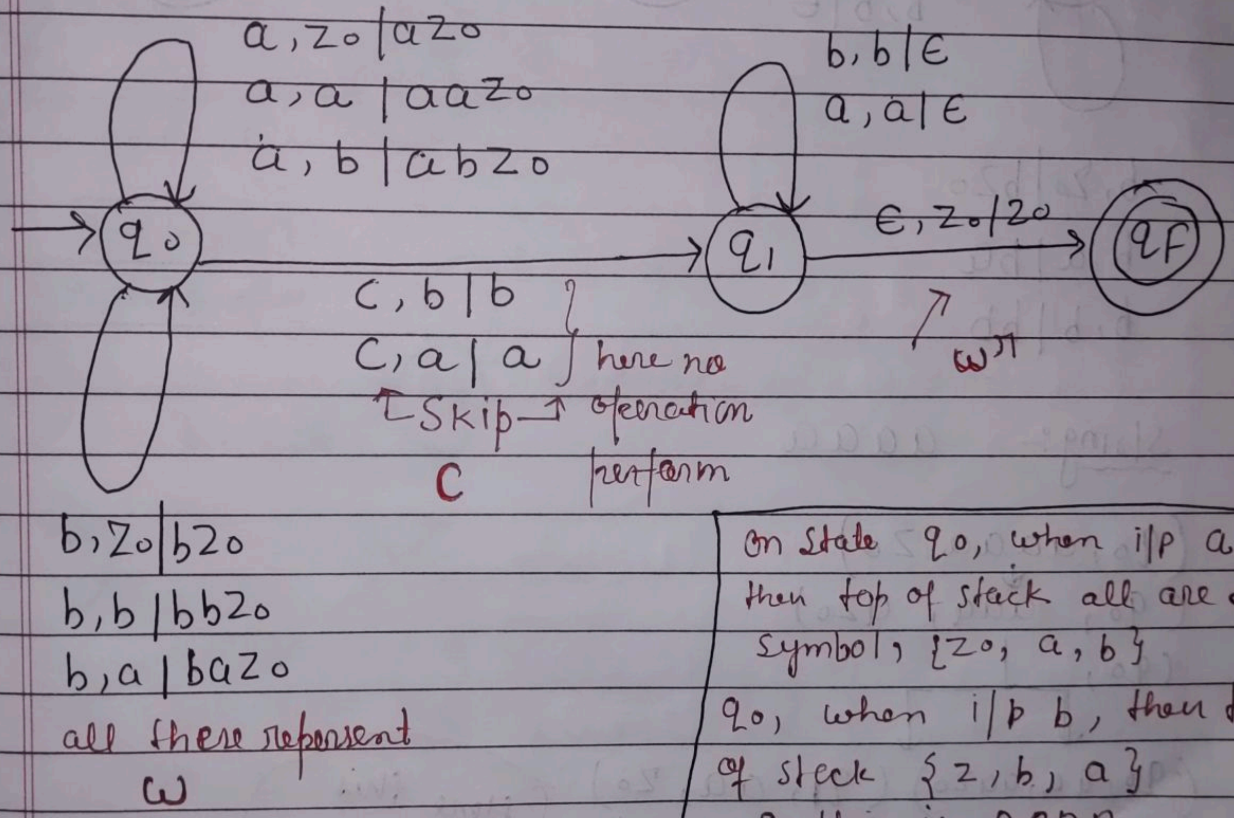
Que Design PDA $L = \{\omega c \omega^{r} \mid \omega \in \{a,b\}^+\}$

a b b c b b a

ω ω^{r}

b a a c a a b

ω ω^{r}



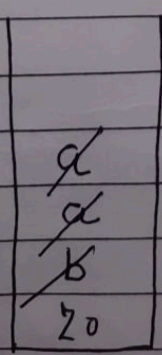
On state q_0, when i/p a then top of stack all are different symbol, $\{Z_0, a, b\}$

q_0, when i/p b, then top of stack $\{Z, b, a\}$

So this is DPDA

Check string : b a a c a a b

for c skip operation



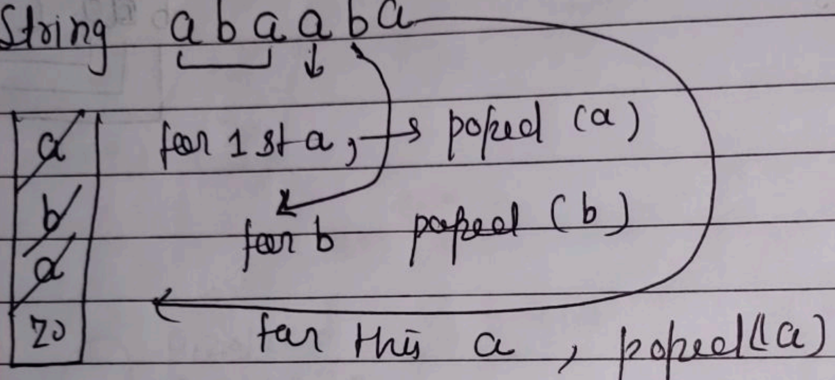
Date ___/___/___

Que Design PDA for the given Language $L = ww^r \mid w \in (a+b)^*$ and Check whether it is DPDA & NPDA.

$w = aba$ $w^r = aba$ $ww^r = abaaba$

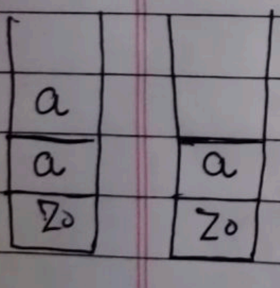
$w = aa$ $w^r = aa$ $ww^r = aaaa$

$\Rightarrow$ String abaaba



$$\overset{w}{\overbrace{a\,a}}\;\overset{w^r}{\overbrace{a\,a}}$$

$\Rightarrow$ String a a a a

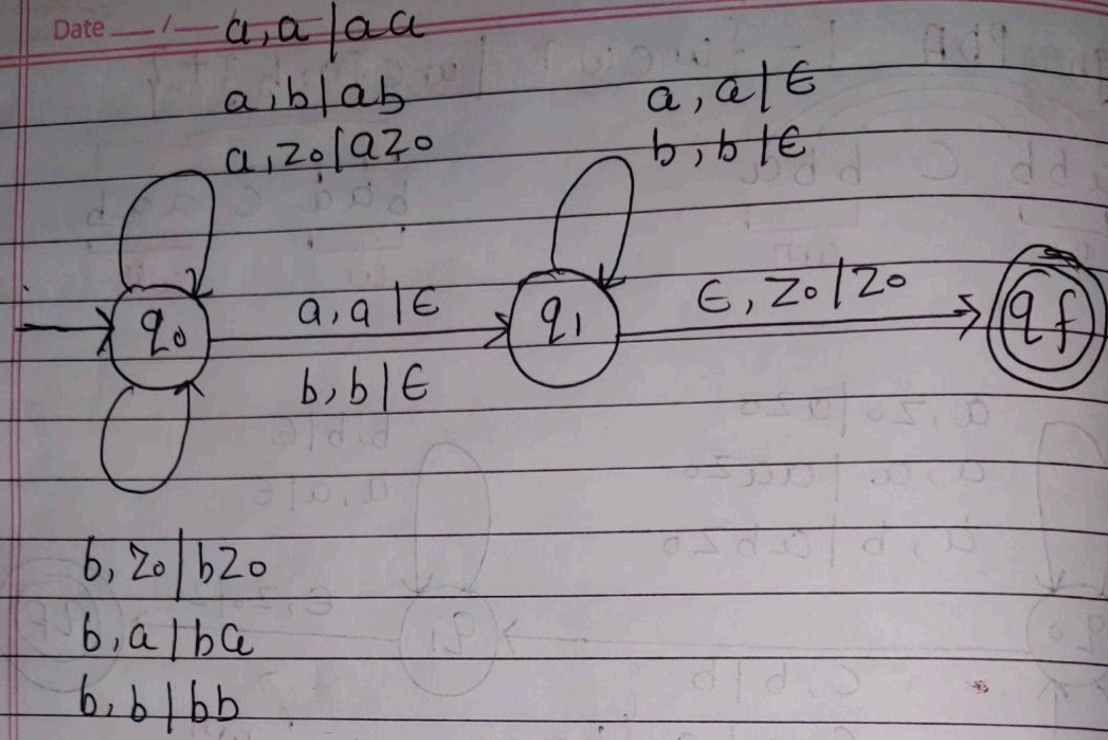


for this 'a' we have two choice
'a' will be PUSH,
'a' will be POP

* means when top of stack 'a' & input 'a' then we will pop.
* on other side i/p 'a' & top of stack a then we will PUSH 'a'

At a time same input symbol 'a' we will perform two transition on 2 State. (operation PUSH & POP)
Which is the case of NPDA

Date __ / __



String:- aaaa

(q_0, aaaa, Z_0)

(q_0, aaa, aZ_0)

(q_0,

(q_0, aa, aaZ_0) (PUSH) (q_1, aa, Z_0) (POP) → ~~Here~~ this transition is not define so stop

(q_0, a, $aaaZ_0$) (q_1, a, aZ_0)

(q_0, ε, $aaaaZ_0$) → stop

(q_1, ε, aaZ_0) → stop

(q_1, ε, Z_0) ⇒ qf

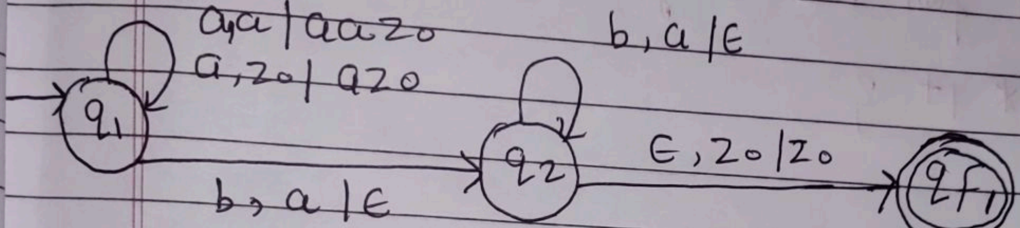
find final State.

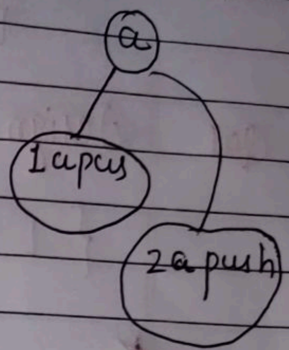
Date ___ /___ /_____

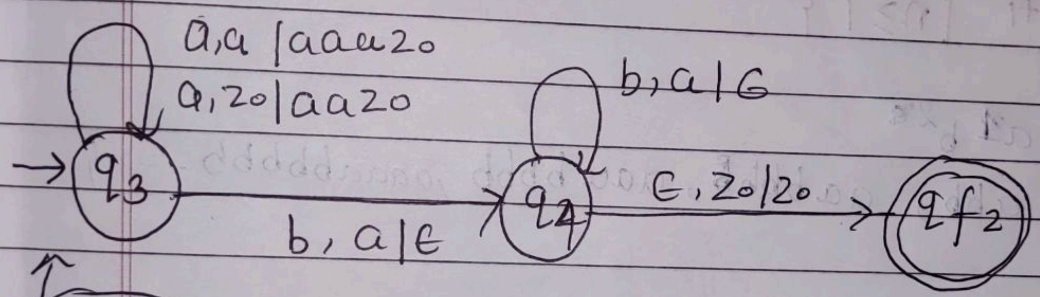
Que Design PDA for $\{ L = a^n b^n$ UNION $a^n b^{2n} \mid n \geq 1. \}$

$L = [\{a^n b^n\} \cup \{a^n b^{2n}\} \; n \geq 1]$

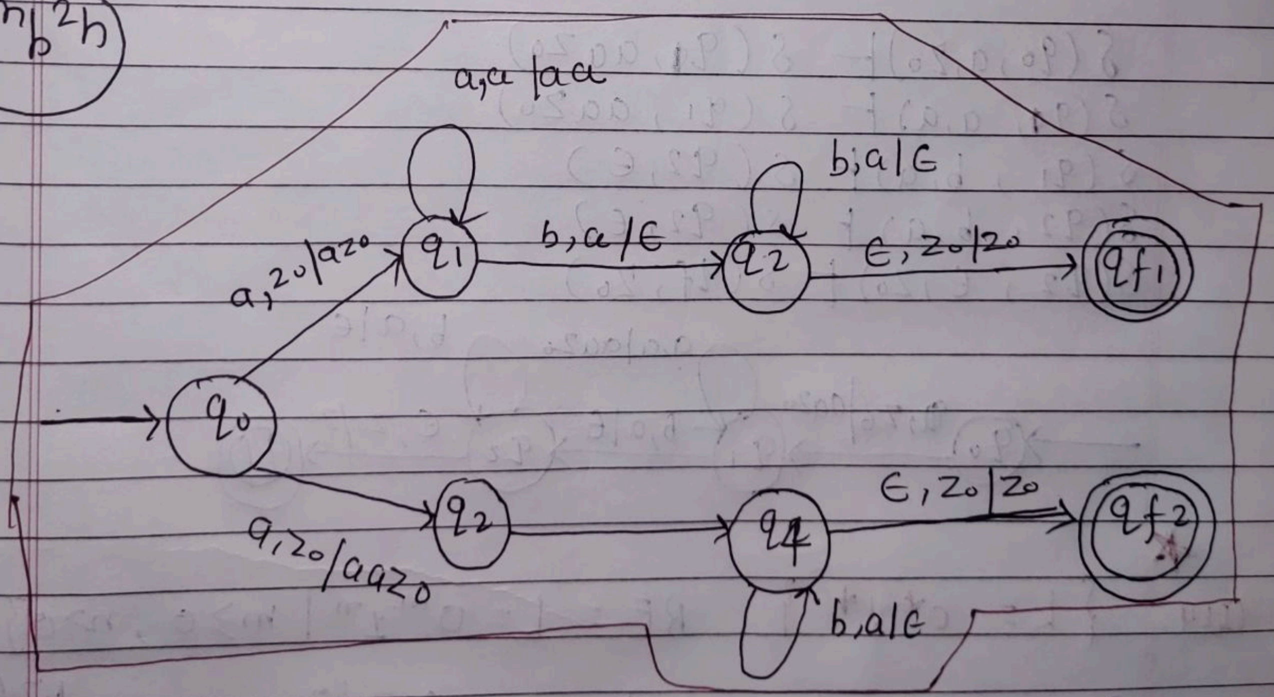
$a^n b^n$







$a^n b^{2n}$



This is also NDPDA

Date __/__/__

Que Design a PDA for the Language
$\{L = a^n b^{n+1} \mid n \geq 1\}$

Que Design PDA for the regular Expression

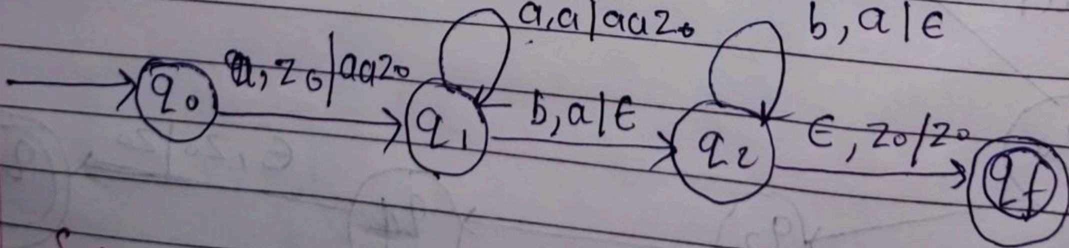
$\{L = 0^* 1^+\}$

★

Que: $\{L = a^n b^{n+1} \mid n \geq 1\}$

$L = a^1 b^2$
$\{ L = abb, aabbb, aaabbbb, aaaabbbbb \ldots \}$

$\delta(q_0, a, Z_0) \vdash \delta(q_1, aaZ_0)$
$\delta(q_1, a, a) \vdash \delta(q_1, aaZ_0)$
$\delta(q_1, b, a) \vdash \delta(q_2, \epsilon)$
$\delta(q_2, b, a) \vdash \delta(q_2, \epsilon)$
$\delta(q_2, \epsilon, Z_0) \vdash \delta(q_f, Z_0)$



★

Que $\{L = 0^* 1^+\}$ $\quad RE = L = 0^m 1^n \mid m \geq 0, n > 0\}$

$\delta(q_0, 0, Z_0) \vdash (q_0, Z_0)$
$\delta(q_0, 1, Z_0) \vdash (q_f, Z_0)$
$\delta(q_1, 1, Z_0) \vdash (q_f, Z_0)$