Final

18-03-2025

iv. Intersection:

i- Union
ii- Concatenation
iii- Closure
iv. Intersection

$L_1 = \{aa, aab, baa, \dots\}$

Substring aa appears here

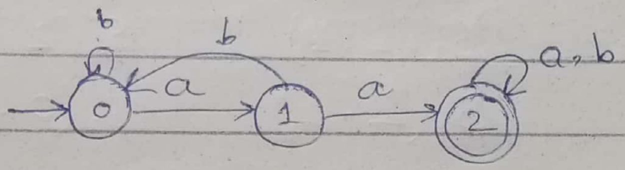
L_2 = Even NO of a's.

$L_2 = \{\lambda, b, bb, \dots, aa, baa, aba, aab, \dots\}$

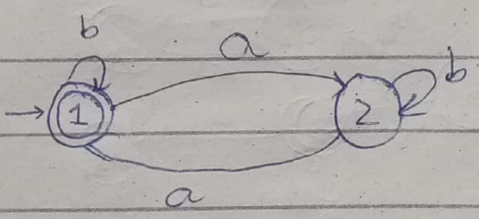
By DeMorgan's law

$$L_1 \cap L_2 = (L_1' \cup L_2')'$$

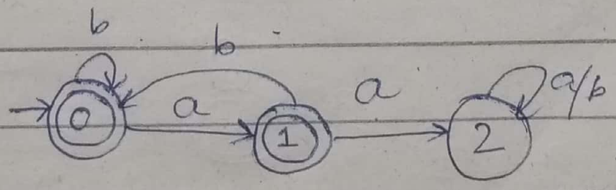
L_1 = DFA1 :



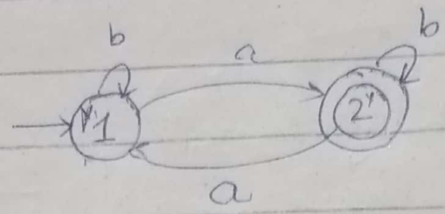
L_2 = DFA2 :



L_1' =

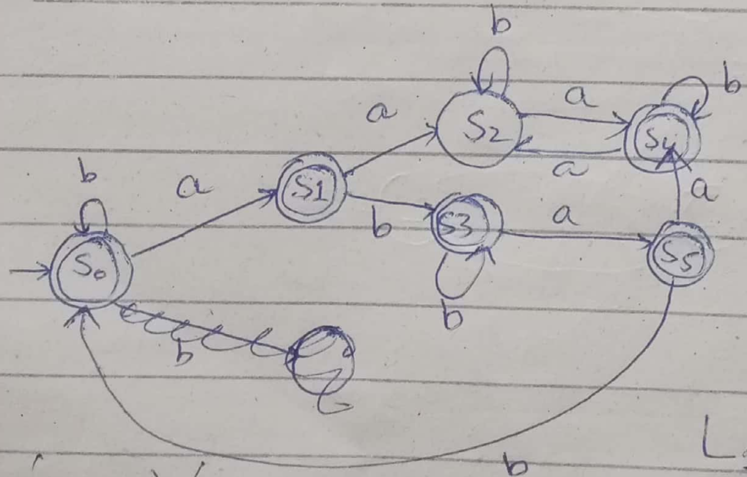


L_2'



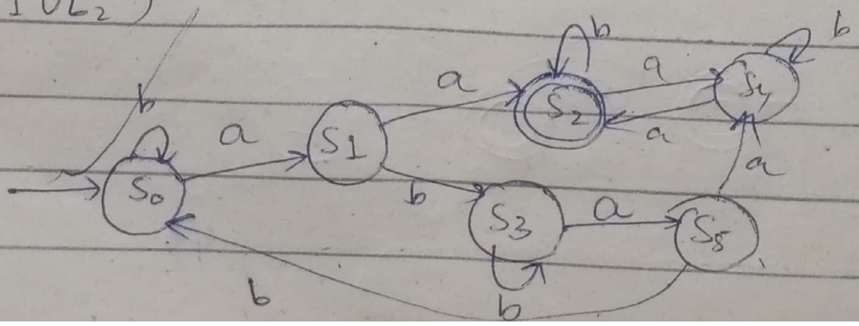
$L_1' \cup L_2'$

	a	b
$+^{-} S_0 (0, 1)$	$S_1(1, 2)$	$S_0(0, 1)$
$+ S_1 (1, 2)$	$S_2(2, 1)$	$S_3(0, 2)$
$S_2 (2, 1)$	$S_4(2, 2)$	$S_2(2, 1)$
$+ S_3 (0, 2)$	$S_5(1, 1)$	$S_3(0, 2)$
$+ S_4 (2, 2)$	$S_2(2, 1)$	$S_4(2, 2)$
$+ S_5 (1, 1)$	$S_4(2, 2)$	$S_0(0, 1)$

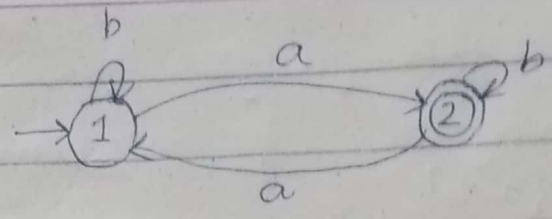


$L_1' \cup L_2'$

$(L_1' \cup L_2')'$

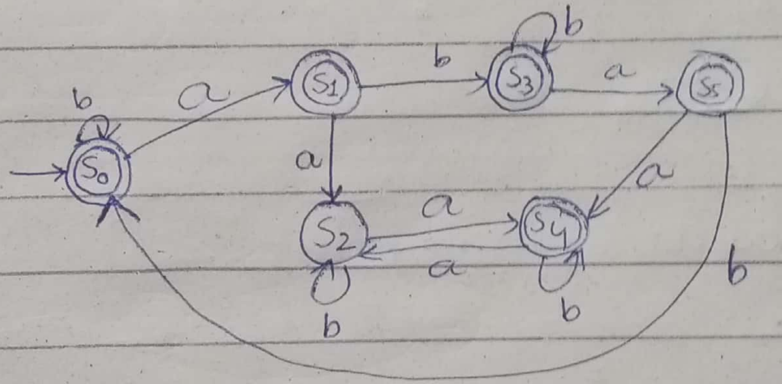


L_2'

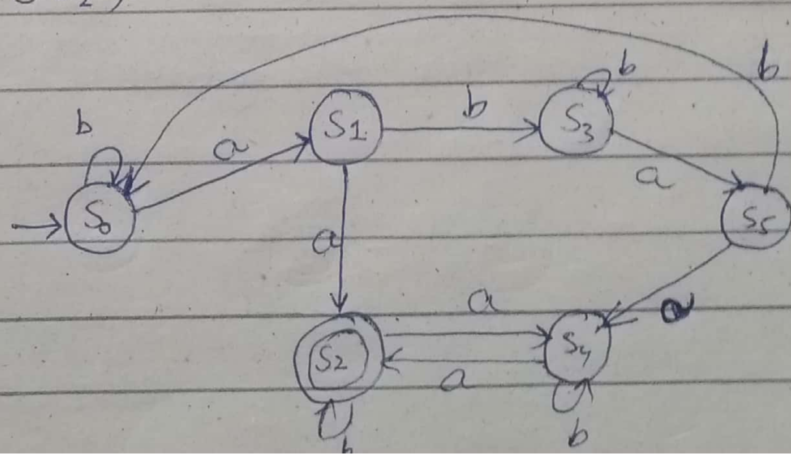


$L_1' \cup L_2'$

	a	b
$\pm S_0(0,1)$	$S_1(1,2)$	$S_0(0,1)$
$+S_1(1,2)$	$S_2(2,1)$	$S_3(0,2)$
$S_2(2,1)$	$S_4(2,2)$	$S_2(2,1)$
$+S_3(0,2)$	$S_5(1,1)$	$S_3(0,2)$
$+S_4(2,2)$	$S_2(2,1)$	$S_4(2,2)$
$+S_5(1,1)$	$S_4(2,2)$	$S_0(0,1)$



$(L_1' \cup L_2')'$



19-03-2024

Non-Regular Languages:

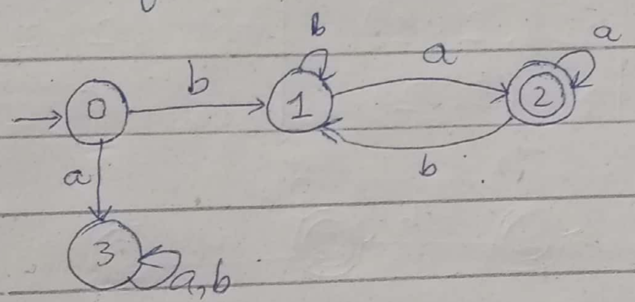
1. Language of palindrome
$\{a, b\}$

$\{a, b, aa, bb, aaa, bbb, aba, bab, \ldots\}$

Properties of R.E

Finite Automata:

1- If a F.A producing infinite number of substrings then there will be a loop in the states and finite states.



$\{ba, bba, baa, bbba, \ldots\}$

2- If a loop repeats then the output of that loop is also the part of our language.

baba

→ Pumping lemma : (Non-regular prove krla

1- Suppose language(L) is a regular language.

2- The Finite automata of language consists of N no. of finite states.

3- Suppose string $w \in L$ where $|w| > N$.

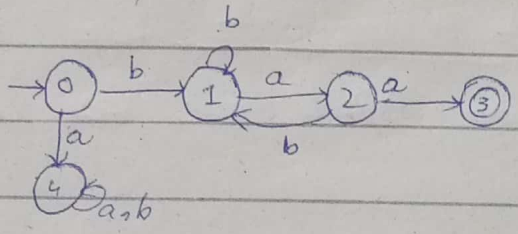
4- Divide string w in a substring $x \cdot y\ z \Rightarrow w = xyz$

i- $|y| \neq 0$

ii- $|x| + |y| \leq N$

5- Repeat y part $w = xy^iz$ $i > 1$

Example:



$|w| > 5$

$|y| \neq 0$

$|x| + |y| = 5 = N$

$w = \underbrace{bbb}_{x}\underbrace{ab}_{y}\underbrace{aba}_{z}$

6- If after repetation $w \in L$ then Language is Regular language. Otherwise non-regular.

→ Forcefullay P.L law ko break krna he.

Palindrome:

{a, b, aa, bb, aba, bab, bbb, ...}

Suppose $N = 5$

→ $W = \underbrace{ba}_{x}\underbrace{ab}_{y}\underbrace{aab}_{z}$

$w = xy^iz$

$W = xy^2z \Rightarrow$ baababaab ☺

→ $W_1 = \underbrace{ba}_{x}\underbrace{a}_{y}\underbrace{baab}_{z}$

$W_1 = xy^iz$

$W_2 = xy^2z = \underbrace{ba}_{x}\underbrace{a}_{y}\underbrace{a}_{y}\underbrace{baab}_{z}$

W_2 is Not belong to L so L is a Non-regular language.

Prove $a^n b^n$ $n \geq 1$ is a non-regular language

{ab, aabb, aaabbb, ...}

$N = 5$

$W_1 = \underbrace{aaa}_{x}\underbrace{a}_{y}\underbrace{bbbb}_{z}$

$W_2 = \underbrace{aaa}_{x}\ \underbrace{a}_{y}\ \underbrace{a}_{y}\ \underbrace{bbbb}_{z}$

$W_2 \notin L$ so L is a non-regular language:

Another way: $N=4$

$w = \underbrace{aa}_{x}\underbrace{a}_{y}\underbrace{bbb}_{z}$

$w = \underbrace{aa}_{x}\ \underbrace{ab}_{y}\ \underbrace{ab}_{y}\ \underbrace{bb}_{z}$

$W \notin L$ So L is a non regular language.

→ Myhill Nerode Theorem: (Prove Regular).

Language L is a regular language over Σ.

1- Identify all classes belong to or not belong to L over Σ.

L = String contain even a's and even b's.

C_1 = String contains even a's and even b's.

C_2 = String contains even a and odd b.

C_3 = odd a's even b's.

C_4 = odd a's odd b's.

2 -Pick string x and y from

same class.

3- Pick another string z from different class.

4- If xz and yz both strings belongs to the language class or non-language classes then it is regular otherwise it is a non-regular.

Example (L)

2- Choose X from C_3 $X = bab$

Choose Y from C_3 $Y = abb$

3- Choose Z from C_4 = $Z = ab$

4- xz = babab $\in C_2$

yz = abbab $\in C_2$

As xz and yz belongs to C_2 (non-language class) so Language is a regular language.

* Should have to belong same classes either language or non-language classes.

⇒ Prove string end with a is a regular language.

L = String ends with a.

1- Classes:

~~C_1 = Ela~~ string ends with a.

C_2 = string not end at a.

2- Chooses from xyz:

x from C1 = abaa

y from C1 = baba

z from C_2 = bb

3- Concatenation:

$xz \Rightarrow$ abaabb $\in C_2$

$yz \Rightarrow$ bababb $\in C_2$.

As xz and yz belong to C_2 (Same classes) then L is a regular language.

→ Palindrome:

C1 : Palindrome

C_2 : Non-palindrome

x,y from C1

z from C_2:

$x = aba$

$y = aa$

$z = ba$

$xz = ababa \in C_1$

$yz = aaba \in C_2$

As xz and yz does not belong to same classes

So L is a non-regular language.

25-03-24

Effectively Solvable Problem

→ any problem for which there exist any algorithm and that algorithm has finite steps or problem could solve in finite steps.

Decision procedure:

→

→ Prove 2 languages having same strings

$L_1 = \{1, 2, \ldots 10\}$

$L_2 = \{1, 2, \ldots 10\}$

$U = \{1, 2, \ldots 10\}$

$$(L_1 \cap L_2') \cup (L_1' \cap L_2)$$

* Result should be empty in case of same languages.

* Result can be in form of F.A or R.E.

* For Regular Expression:

$$(a+b)^*(abb+\lambda)(a^+ + aa)$$

1 → Remove all Kleen stars

$$(a+b)(abb+\lambda)(a^+ + aa)$$

2 → Remove all Kleen plus operators

$$(a+b)(abb+\lambda)(a+aa)$$

3 → Remove all right sides of union operators. $(a\underline{+b})(abb\underline{+\lambda})(a\underline{+aa})$

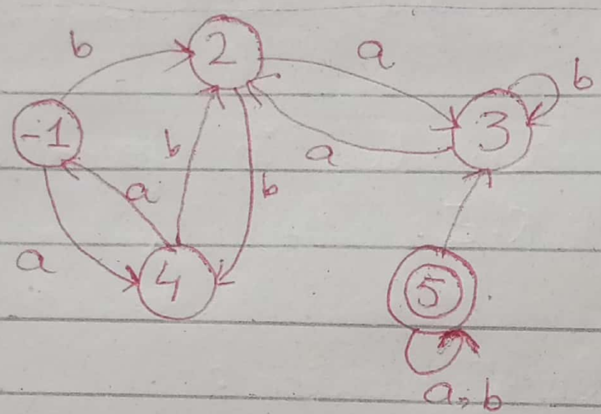
$$(a)(abb)(a)$$

$$aabba$$

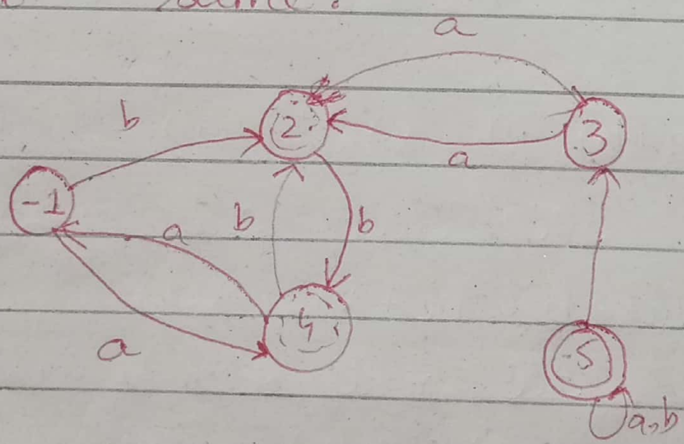
4 → If the resultant is a string

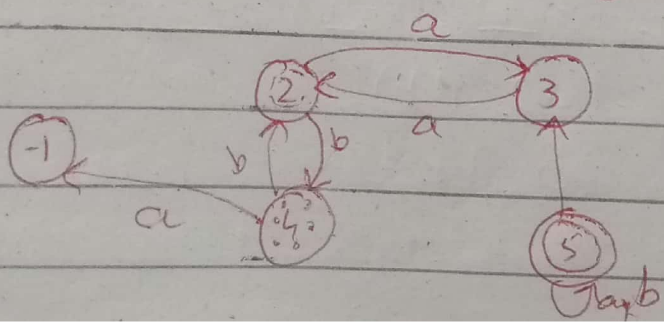
then the languages are not same if the result is empty (ϕ) then the languages are same.

* For Finite automata:



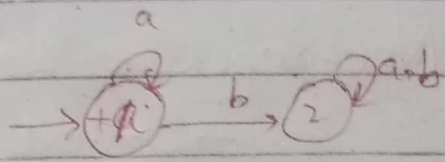
→ If we can reach at final state then the languages are not same.



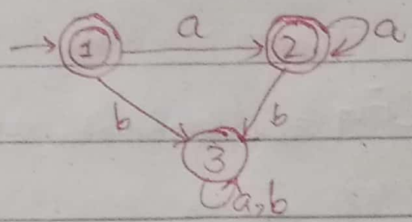


Example:

$L_1 = a^*$

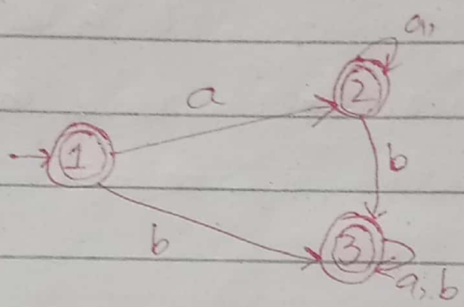


$L_2 = \lambda + aa^*$

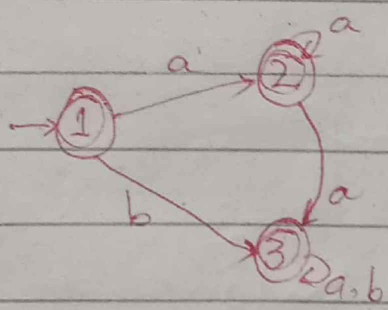


$(FA_1' \cap FA_2)'$

verify krna he



$(FA_1 \cap FA2')'$



26-03-24

TOA

Non-Regular Language:

→ Context Free Grammer (CFG)

CFG = {terminal, non-terminal, Production}

Terminal ⇒ Characters info (like Σ)

Non-terminal ⇒ variables: can be replace with a value.

Production ⇒ rules to replace variables with a value.

$S \to a$ — Capital letters ⇒ Non-terminal

$S \to b$ — Small letters ⇒ Terminal

$S \to A$

$A \to aA$

$A \to \lambda$

Non-terminal = {S, A}

Terminal = {a, b}

Production:

→ Left side pr non-terminal

→ Right side pr (terminal, non-terminal or combo of both).

→ Top wala starting non-terminal hota. (often represented by S).
* can be a complete word.
Start.

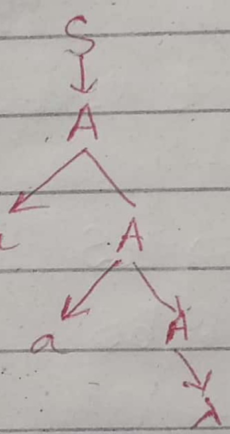
S
a

S
b

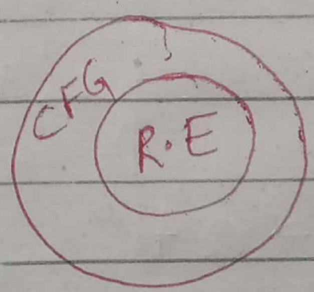
→ aa

S
↓
A
↓
aA
↓
a(aA)
↓
aa(A)
↓
aa(λ)
↓
aa

Derivation:
Process of generation a string using context free grammar.



$S \Rightarrow A$
$\Rightarrow aA$
$\Rightarrow aaA$
$\Rightarrow aa\lambda$
$\Rightarrow aa$



→ CFG ki Computational power zyada he.

i- Produce C.F.G for

$\{\lambda, a, aa, aaa, \ldots\}$

$= a^*$

$S \to aS$ $\qquad$ $S \to aS \mid \lambda$

$S \to \lambda$

ii- Produce C.F.G for

$\{a, aa, aaa, \ldots\}$

$= a^+$

$S \to a$ $\qquad$ $S \to a \mid aS$

$S \to aS$

iii- Produce C.F.G for

$(a+b)^*$

$= \{\lambda, a, b, aa, bb, ab, ba, \ldots\}$

$S \to aS$

$S \to bS$ $\qquad$ $S \to Sa \mid Sb \mid \lambda$

$S \to \lambda$

iv- S $\qquad$ $a(a+b)^* b$

$S \to aA$	$S \to aXb$
$A \to aA \mid bA$	$X \to aX \mid bX \mid \lambda$
$A \to b$	

v- Must appears aa.

$$(a+b)^* aa (a+b)^*$$

$S \to XaaX$

$X \to aX \mid bX \mid \lambda$

Non-Regular Expression

vi- Equal no of a's and b's in which a comes before b.

$a^n b^n \qquad n > 0$

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

~~$S \to aA \mid aA$~~
~~$A \to Sb$~~

$S \to aSb \mid ab$

vii- Even length palindrome:

$\{\lambda, aa, bb, abba, baab, \ldots\}$

$S \to \lambda \mid \cancel{aa \mid bb} \mid aSa \mid bSb \mid$

odd length palindrome

S → a|b|asa|bSb

All palindromes

S → a|b|aSa|bSb|λ

All length palindrome except λ.

S → a|b|aa|bb|asa|bsb|

Compiler Construction

DS → DT V;
DT → int | float | char
V → id, V | id

int a;	float a, b ;	char a, b, c;
DS → DT V;	DS → DT V;	DS → DT V;
→ int id;	→ float id, V;	→ char id, V;
	→ float id, id;	→ char id, id, V;
		→ char id, id, id;

01-04-24

Even number of a's and b's.

$(aa + bb + ((ab + ba)(aa + bb)^*(ab + ba)))^*$

$S \rightarrow \cancel{aa \mid bb} \mid (ab + ba) S (ab + ba)$

$S \rightarrow aaS \mid bbS \mid \underbrace{(ab + ba)}_{X} S \underbrace{(ab + ba)}_{X} \mid \lambda$

$X \rightarrow abS \mid baS$

XSX

= abSaaSbaS

= abaaSaa λ ba λ

= abaaλaaba

$S \rightarrow aaS \mid bbS \mid \lambda \mid XSX \mid SS$

$X \rightarrow ab \mid ba \mid S$

XSX

X S X

ab XSX ba

ab aaS ba

aaλ

XSX

abXSX

λ

abba

ababaababa

XSX

aSa Sb

aSSb

aaSSb

aa

Sb

Sbb

Sabb

Sb

aSb

aSb aaSb aaSb

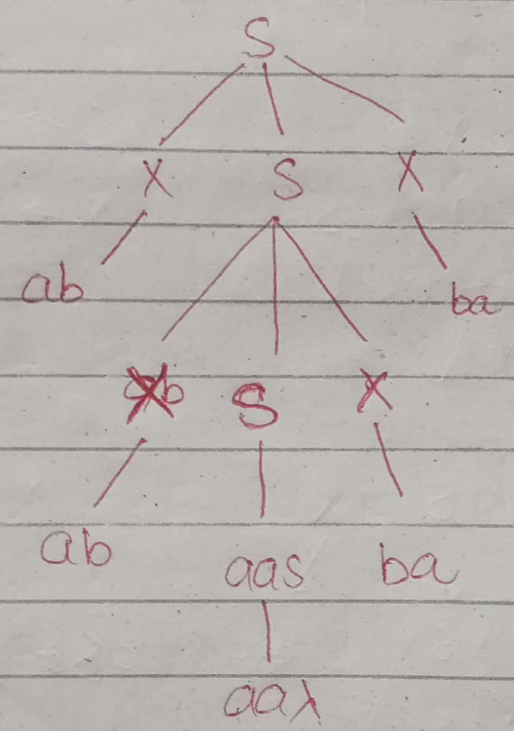
Even no of a's and b's

$(aa+bb+(ab+ba)(aa+bb)^*(ab+ba))^*$

$S = aaS \mid bbS \mid \lambda \mid XSX \mid SS$

$X = ab \mid ba$

ababaababa



ababaababa

$\boxed{\text{abbaaa}}$

$a^* b^*$

$S \to \lambda \mid aS \mid Sb \mid$	$S \to aS \mid \lambda \mid X$ $X \to bX \mid \lambda$

Ambigous Grammer:

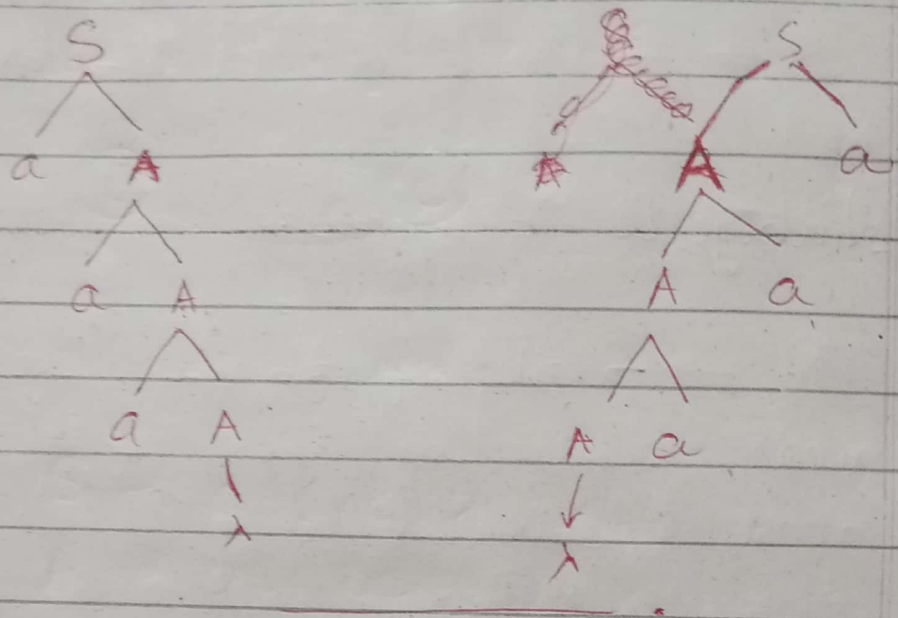
→ aik sy zayada left most derivation ya right most derivation

Derivation
i- Left ⇒ leftmost non-terminal ko solve krna
ii - Right ⇒ rightmost non-

Left	Right	
$S \to AB$	$S \to AB$	$S \to AB$
$S \to aAB$	$S \to AbB$	$A \to Aa \mid \lambda$
$S \to aaAB$	$S \to Ab\lambda$	$B \to Bb \mid \lambda$
$S \to aa\lambda B$	$S \to Ab$	
$S \to aaB$	$S \to aAb$	
$S \to aaBb$	$S \to aaAb$	
$S \to aa\lambda b$	$S \to aa\lambda b$	
$S \to aab$	$S \to aab$	

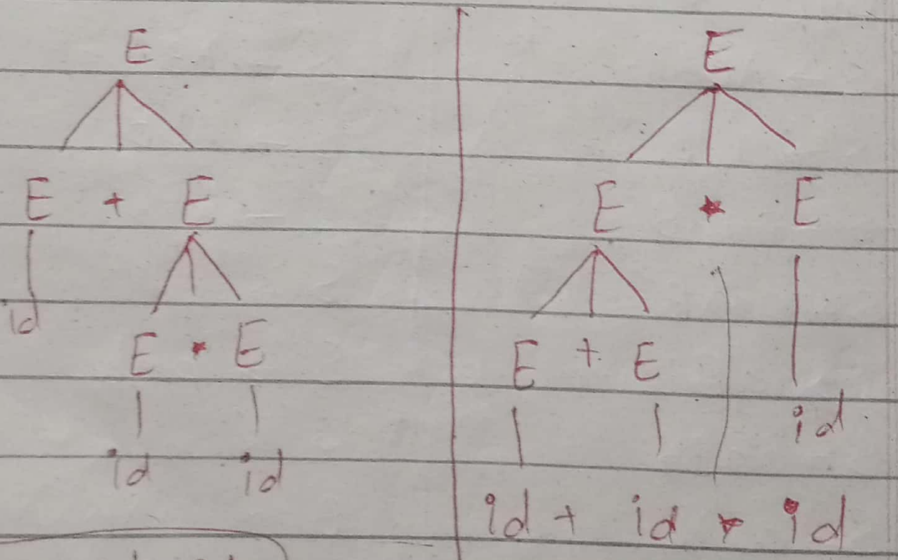
$S \rightarrow aA \mid Aa \mid \lambda$

aaa



id + id * id ✓

$E \rightarrow E+E \mid E*E \mid id$ ✓



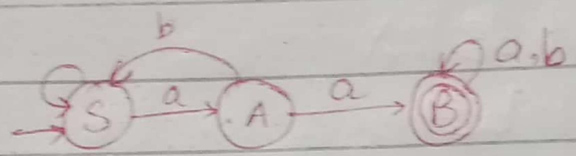
$\boxed{id+id*id}$ $\boxed{id+id*id}$

$E = E + E$
$E = id + E * E$
$E = id + id * E$
$E = id + id * id$

$E = E * E$
$E = E + E * E$
$E = id + E * E$
$E = id + id * E$
$E = id + id * id$

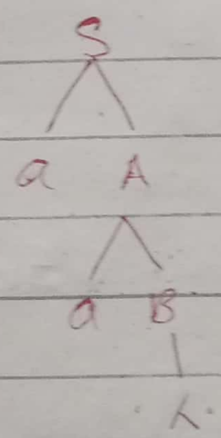
Regular Grammer

$(a+b)^* aa (a+b)^*$



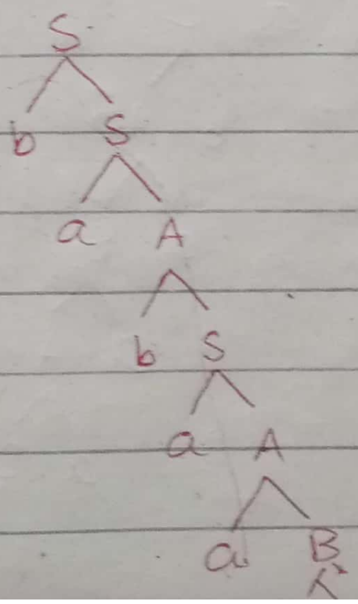
$S \rightarrow aA \mid bS$
$A \rightarrow aB \mid bS$
$B \rightarrow aB \mid bB \mid \lambda$ (final k sath λ).

aa

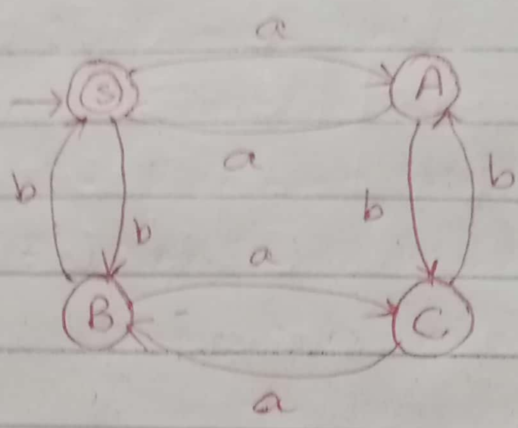


aa

babaa



Even no. of a's and b's.



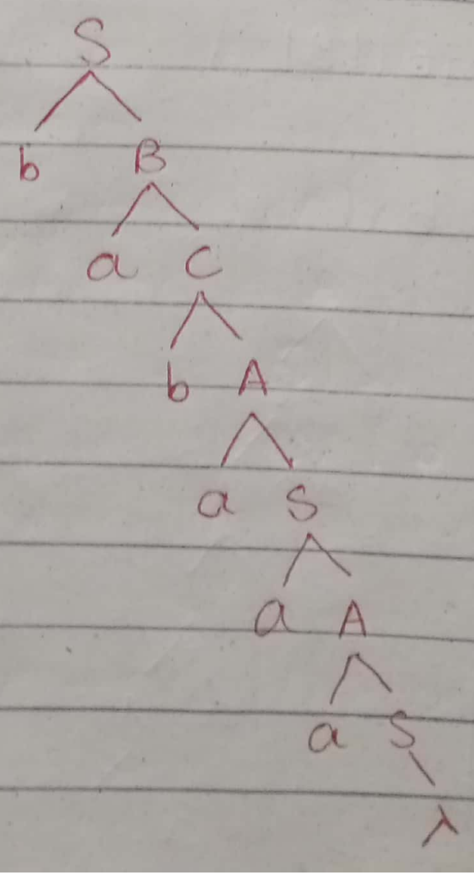
$S \rightarrow aA \mid bB \mid \lambda$

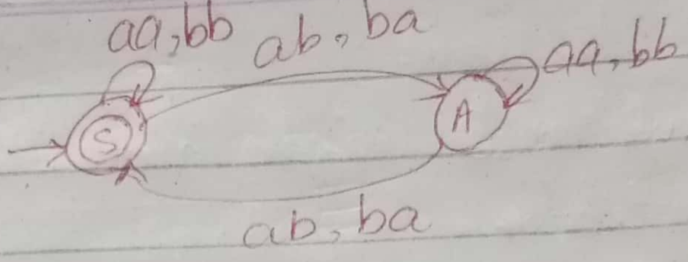
$A \rightarrow aS \mid bC$

$B \rightarrow aC \mid bS$

$C \rightarrow aB \mid bA$

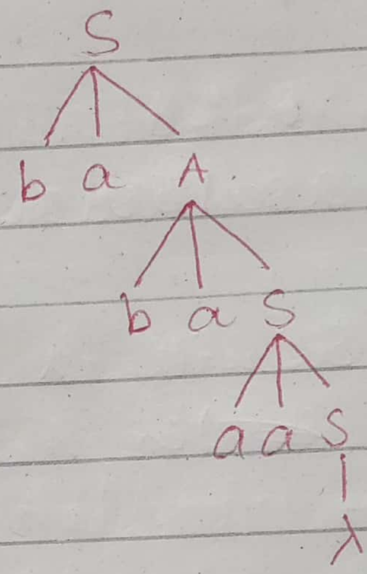
babaaa





S → aaS | bbS | abA | baA | λ
A → aaA | bbA | abS | baS

babaaa



Regular Grammer:

R.G → Semiword | word

R.G ki production ya tu word hoti ya semiword.

Semiword ⇒ (terminal)* . non-terminal

word ⇒ terminal , λ

CHOMSKY's Normal Form (CNF)

1 - Remove null and nullable productions.

2 - Remove Unit production

3 - ensure Non terminal → two non-terminal | 1 terminal

$S \rightarrow ab \mid bA \mid \lambda$

$A \rightarrow asa \mid bS \mid ab$

after removing: Null.

$S \rightarrow ab \mid bA$

$A \rightarrow asa \mid bs \mid ab \mid aa \mid b$

replace S with λ in A and write it ~~(with)~~ in A.

Nullable :

$S \rightarrow ab \mid bA \mid \lambda$

$A \rightarrow aSa \mid Abs \mid ab \mid S$ → nullable

derived way

$S \rightarrow ab \mid bA \mid b$

$A \rightarrow asa \mid Abs \mid ab\cancel{s} \mid aa \mid bs \mid Ab \mid b$

$A \rightarrow asa \mid Abs \mid ab \mid S \mid aa \mid bs \mid Ab \mid b \mid \boxed{\cancel{b}}$

Remove null and nullable

S → Xa | YY | aX | ZYX
X → Za | bZ | ZZ | Yb
Y → Ya | XY | λ
Z → aX | ~~X~~YY

S → Xa | YY | aX | ZYX | Y | ZX | YX | X | a | ZY | Z
X → Za | bZ | ZZ | Yb | b | a | ~~b~~ | Z
Y → Ya | XY | a | X | Y
Z → aX | YY~~Y~~ | YY | Y | a

Step by step
Remove λ and replace Y from

S → Xa | YY | aX | ZYX | Y | ZX
X → Za | bZ | ZZ | Yb | b
Y → Ya | XY | a | X
Z → aX | YYY | YY | Y

Unit production:
Non-terminal consists of Non-terminal.

$S \rightarrow ab \mid A \mid bA$ unit production

$A \rightarrow b \mid aS$

Replace A in S with all values of A.

$S \rightarrow ab \mid b \mid aS \mid bA$

$A \rightarrow b \mid aS$

For cyclic unit production:

$S \rightarrow ab \mid A \mid bA$

$A \rightarrow b \mid aS \mid S$

$S \rightarrow ab \mid b \mid aS \mid S \mid bA$

$A \rightarrow b \mid aS \mid ab \mid S \mid bA$

S ki values replace kring sy phir wohi aaen gi
So ignore it.

$S \rightarrow ab \mid b \mid aS \mid bA$

$A \rightarrow b \mid aS \mid ab \mid bA$

Remove Unit production

$S \to A \mid bb$

$A \to B \mid b$

$B \to S \mid a$

$S \to B \mid b \mid bb \to S \mid a \mid b \mid bb \to a \mid b \mid bb$

$A \to S \mid a \mid b \Rightarrow S \mid a \mid b \mid bb \to a \mid b \mid bb$

$B \to S \mid a \mid b \mid bb \to a \mid b \mid bb$

Remove unit production

$S \to ab \mid bb$

$A \to B \mid b$

$B \to ba \mid a$

3

3. Ensure

Non-terminal → Two Non-terminal
Non-terminal → Single terminal

S → a|b|bb
A → a|b|bb.
B → a|b|bb

S → a|b|YY
A → a|b|YX
B → a|b|YY
Y → b

There should be only two non-terminal or single terminal

S → a|b|YYZ → isko be 2-non-terminal banao.
~~A~~ → a|b|YY

S → a|b|WZ
A → a|b|YY
W → YY

Convert into CNF:

$S \to aA \mid bB \mid \lambda$

$A \to aa \mid SA$

$B \to aS$

Remove Null and nullable production

$S \to aA \mid bB$

$A \to aa \mid SA \mid A$

$B \to aS \mid a$

Remove Unit production

$S \to aA \mid bB$

$A \to aa \mid SA$

$B \to aS \mid a$

Ensure 2 non-terminal and 1 terminal.

$S \to YA \mid ZB$

$A \to YY \mid SA$

$B \to YS \mid a$

$Y \to a$

$Z \to b$

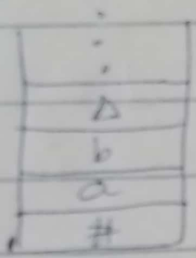
15-04-24

PushDown Automata (PDA)

1: Input Tape
2: States ⇒ {Accept, Reject, Start, Read}
3: Stack ⇒ {Push, Pop}
4: Σ ⇒ Input alphabets...

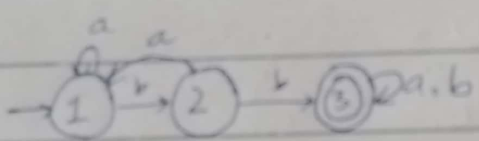
Input tape

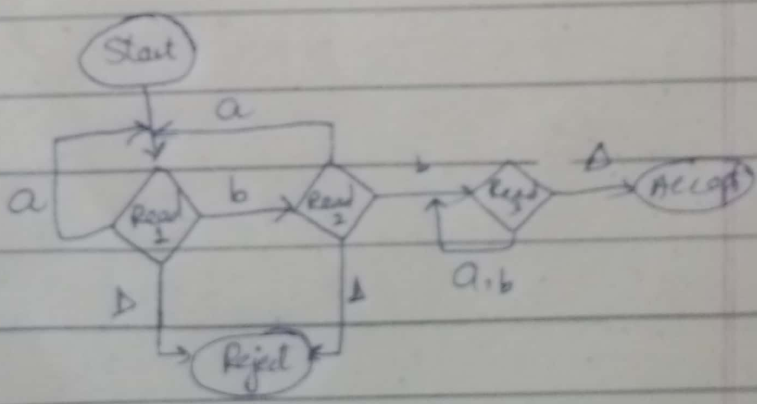
Empty	Δ	Δ	Δ	Δ...
aba	a	b	a	Δ...



Stack

$(a+b)^* bb (a+b)^*$



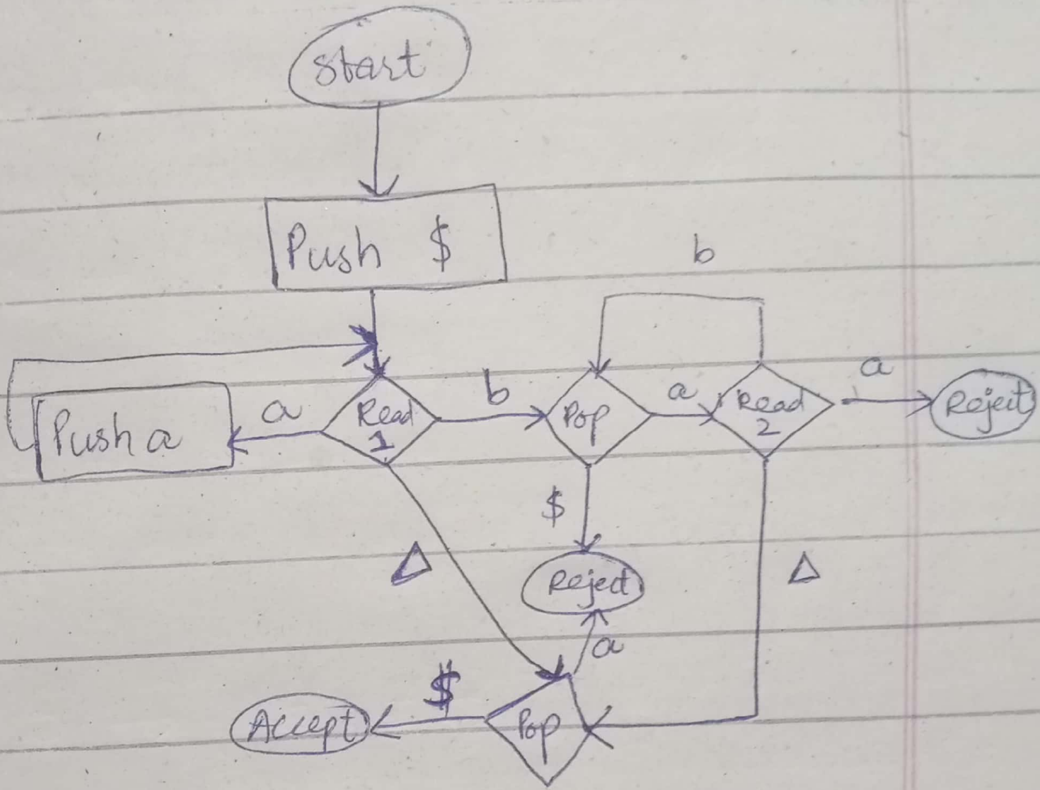


aaaabbbb

Stack is not used for regular languages.

For Non-Regular Languages

$a^n b^n$

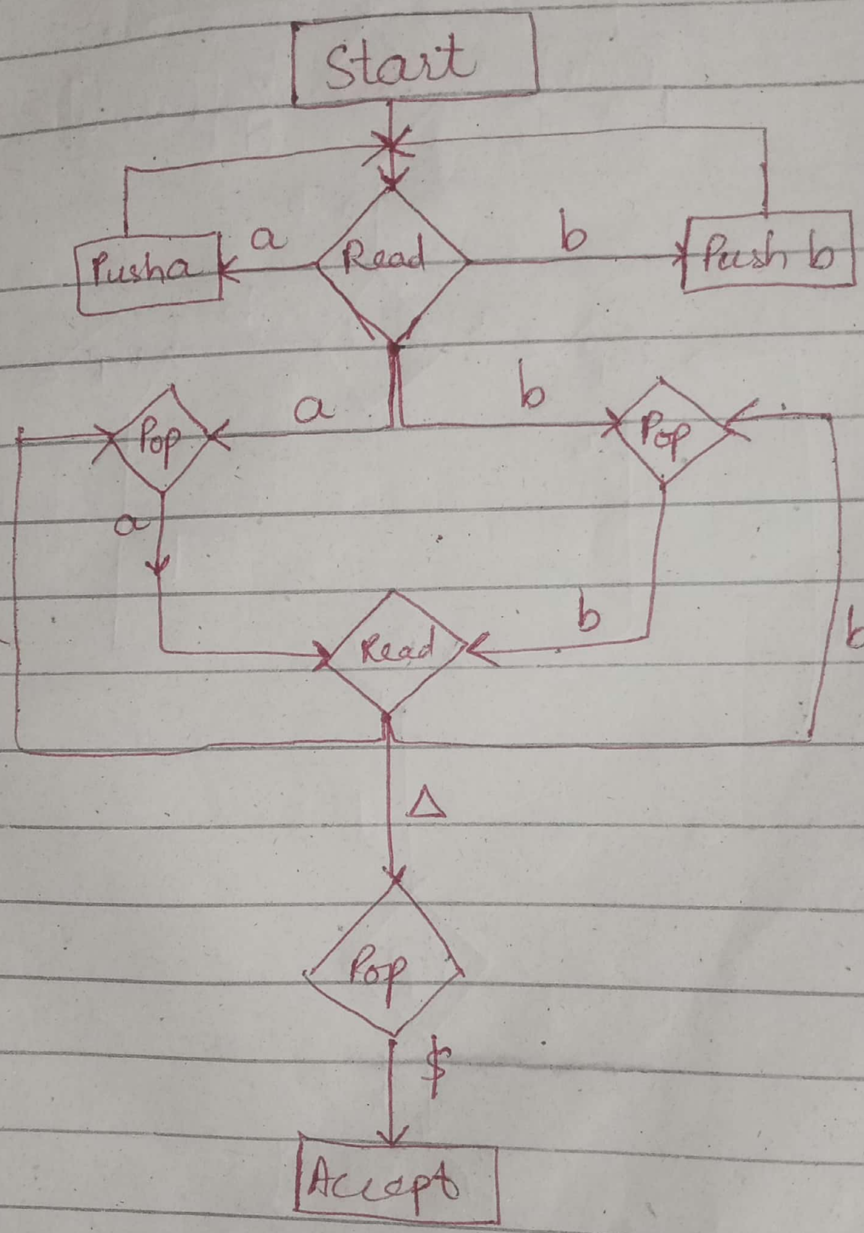


$a^n b^{2n}$

Non-Deterministic PDA

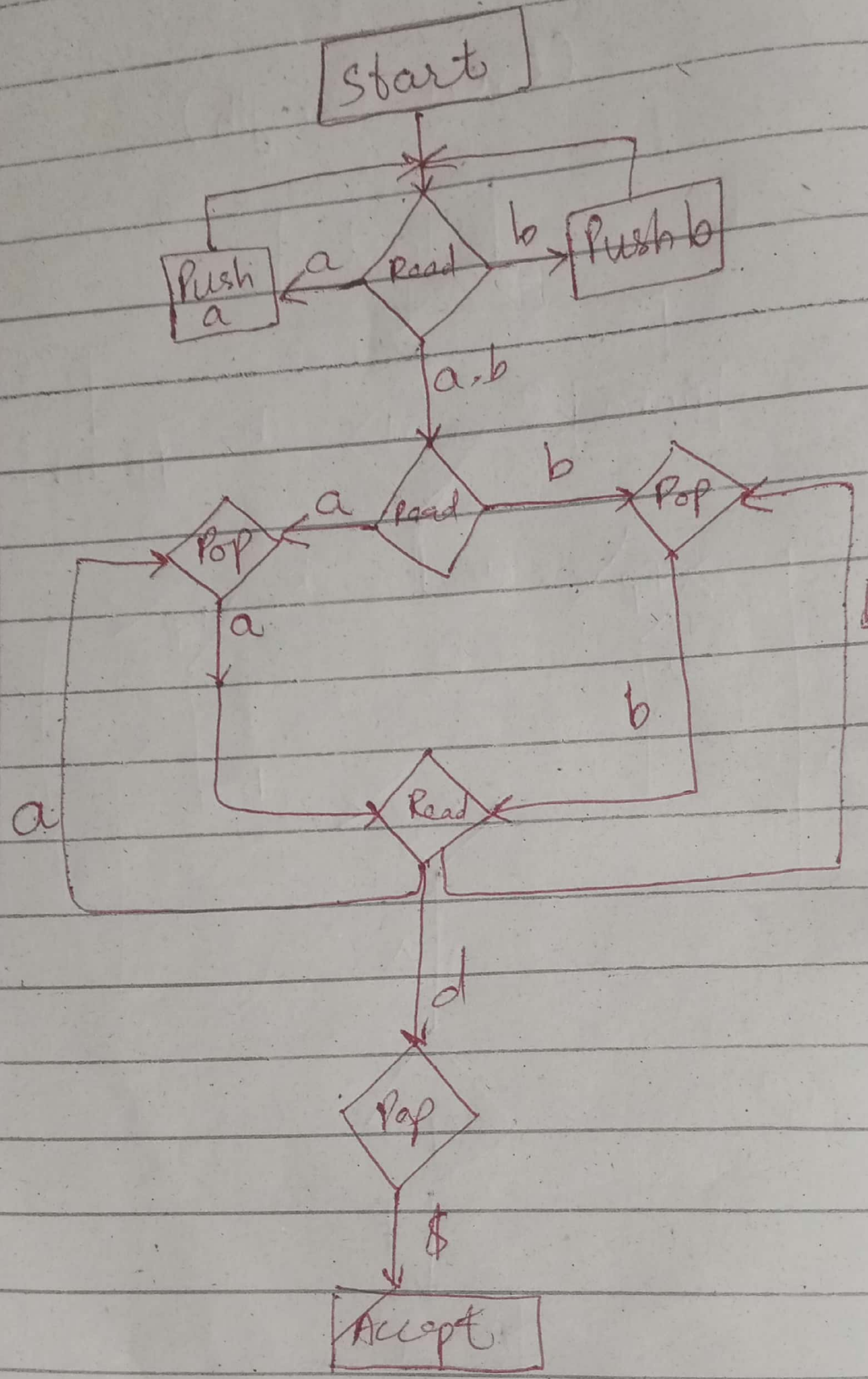
PDA for palindrome:

(Even length)



bbaabb

Odd length palindrome



bbabb