Pushdown Automata (PDA's)

Tape :- contains the input string while it is being run.

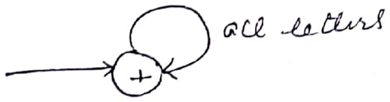
cell i	cell ii	cell iii	cell iv		
a	a	b	Δ	Δ	. . .

⇒ when we reach the first blank cell, we stop. we always presume that once the first blank is encountered, the rest of the tape is also blank. We read from left to right & never go back to a cell that was read before.

states of PDA :

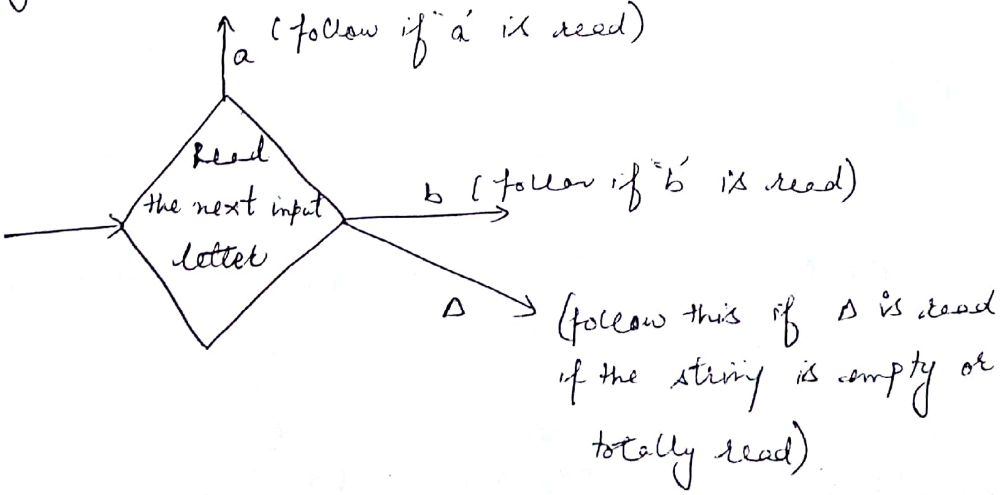
1) start :- same as (−) state.

2). Accept :- It is a dead-end final state. once entered it cannot be left.



3). Reject :- dead end state that is not final.

4) Read :- It reads an input letter & branch to other states depending on what letter has been read.

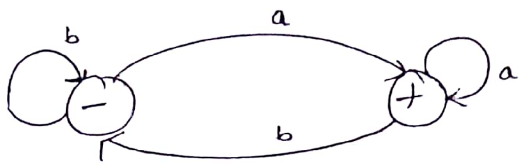


Scanned by CamScanner

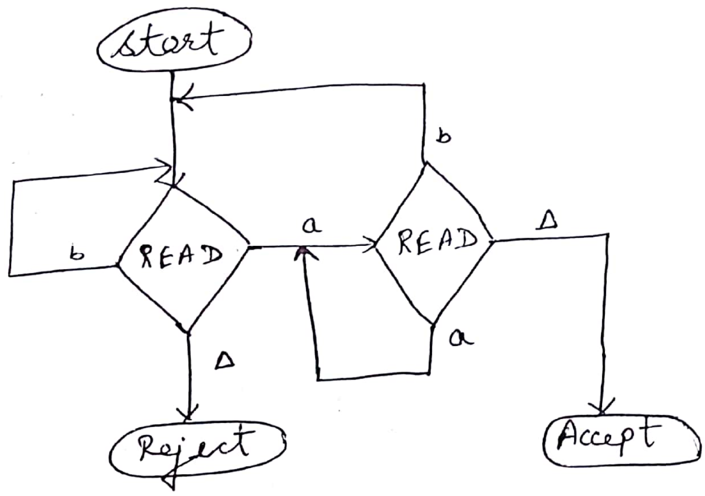
FA's & their Corresponding PDA's

eg-1

FA

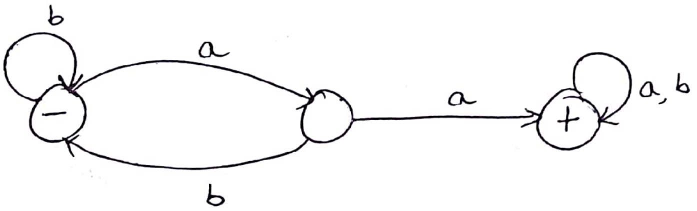


PDA.

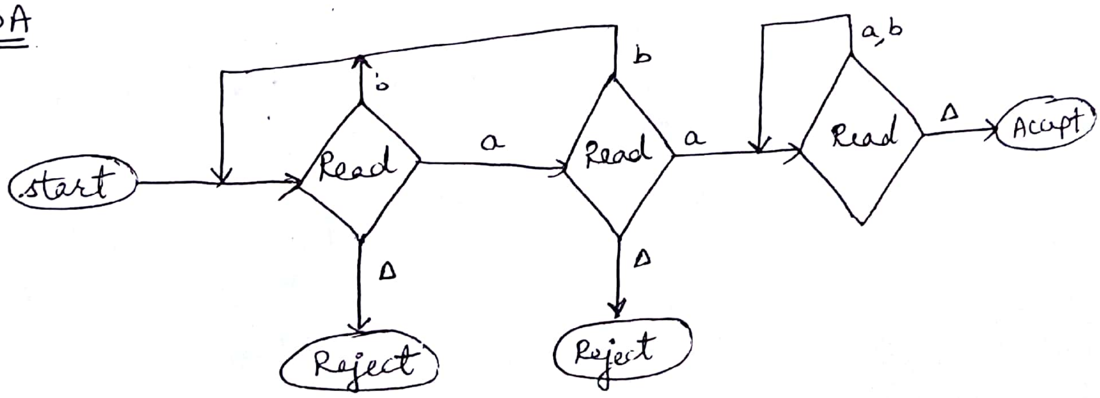


eg-2.

F.A.

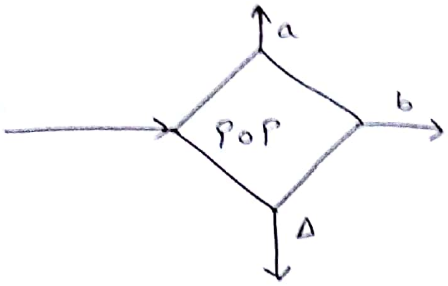


PDA



Scanned by CamScanner

5) Pop :- same as read state becoz each character might appear in stack.



6) Push :- we can leave Push state only by one indicated route, although we can enter a Push state from any direction.

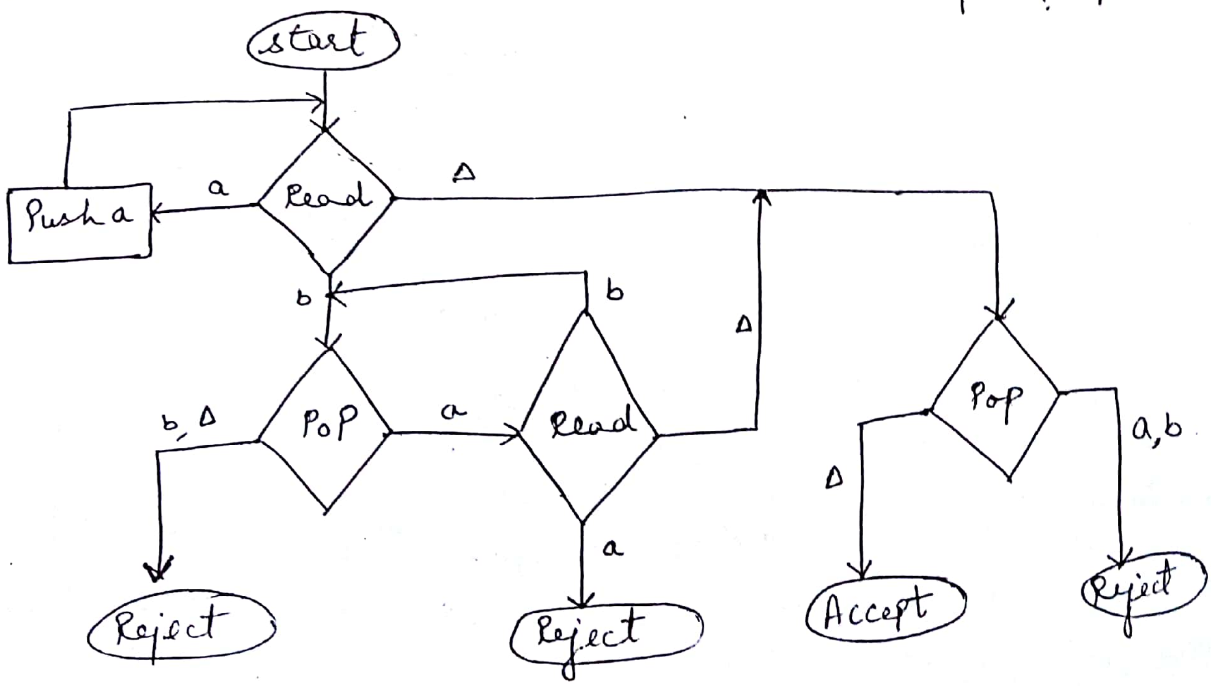
eg :- PDA for the language $a^n b^n$. [deterministic PDA]

initially Tape

a	a	a	b	b	b	Δ	....

stack

Δ
⋮



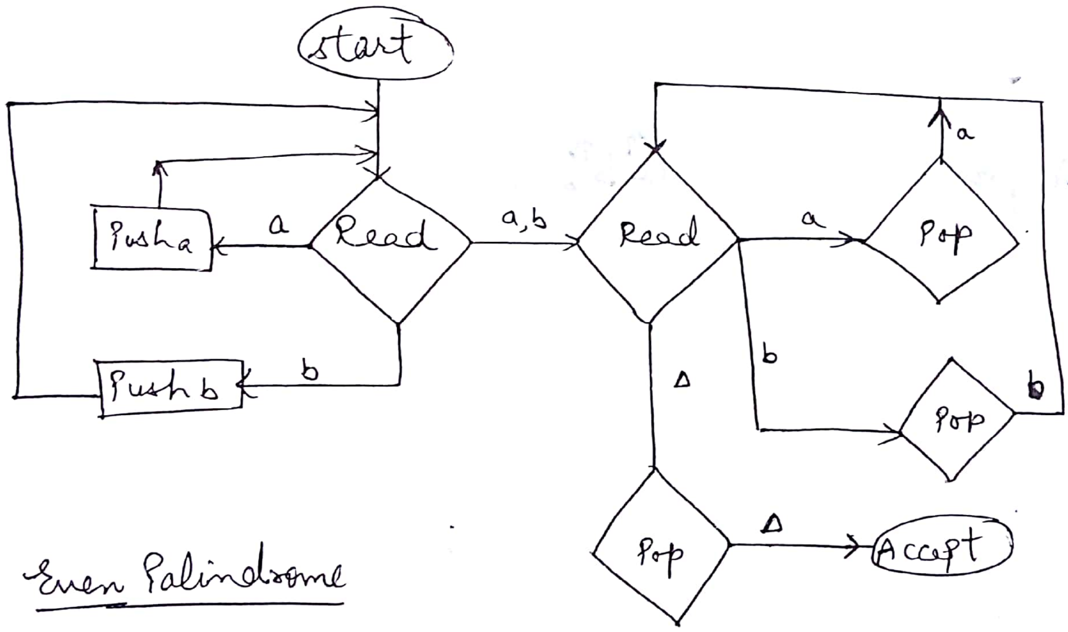
Deterministic PDA :- is one for which every input string has a unique path through the machine.

Non-deterministic PDA :- is one for which at certain times we may have to choose among possible paths through the machine.

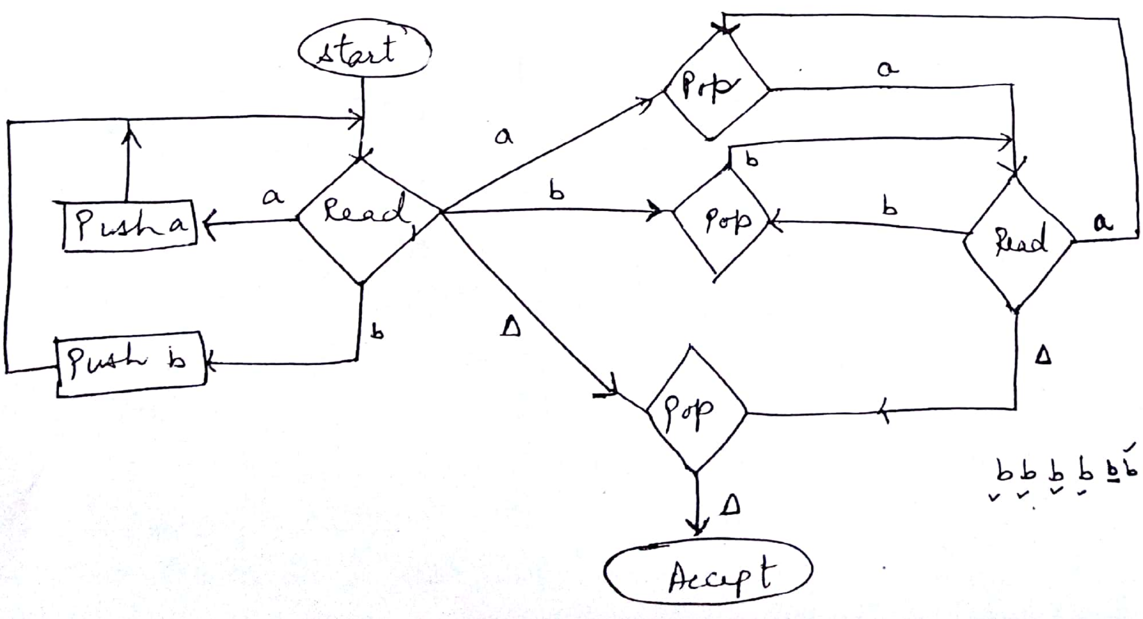
eg:- PDA (Non-deterministic).

odd Palindrome.

ab$\underline{a}$ba.



Even Palindrome



bbbbbbbbb

Definition of PDA.

A Pushdown Automata is a collection of eight things:

1) An alphabet Σ of input letters.

2) An input tape (infinite in one direction). Initially the string of input letters is placed on the TAPE starting in cell i. The rest of the tape is blank.

3) An alphabet Γ of stack letters.

4) A pushdown stack (infinite in one direction). Initially the stack is empty (contains all blanks).

5) One start-state that has only out-edges, no in-edges.

start

6) Halt states of two kinds: some Accept & some Reject. They have in-edges & no out-edges.

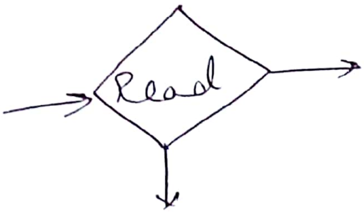
$\rightarrow$ Accept $\quad\quad$ $\rightarrow$ Reject

7) Finitely many non-branching PUSH states that introduce characters onto the top of the stack. They are of the form

$\boxed{\text{Push } x}$ where x is any letter in Γ.

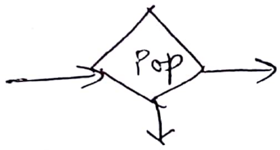
8) Finitely many branching states of two kinds:

(i). states that read the next unused letter from the tape,



which may have out-edges labeled within letters from Σ & the blank character Δ, with no restrictions on duplication of labels & no compulsion that there be a label for each letter of Σ, or Δ.

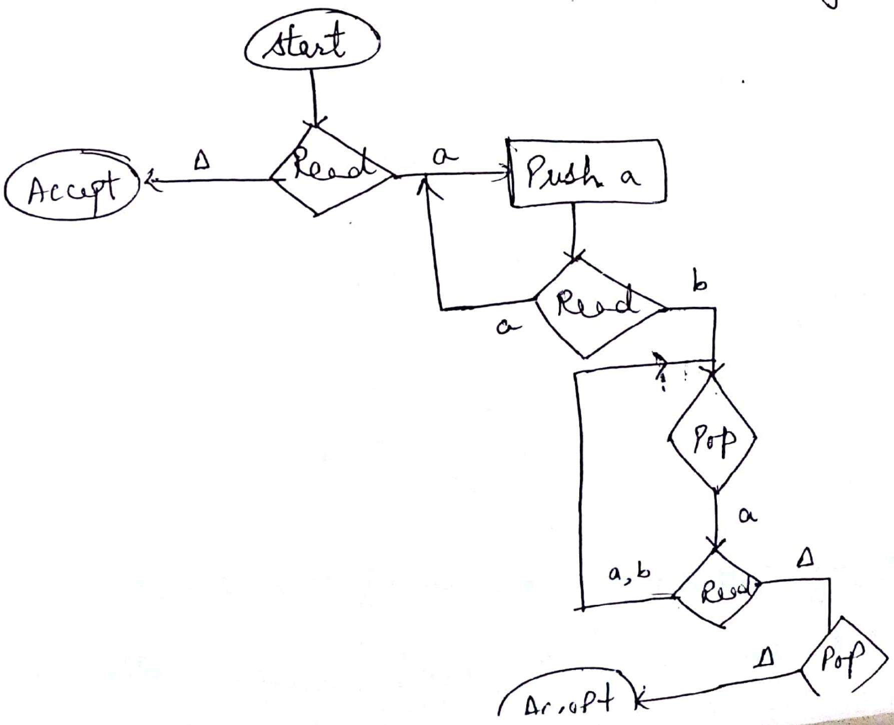
ii) states that read the top character of the stack.



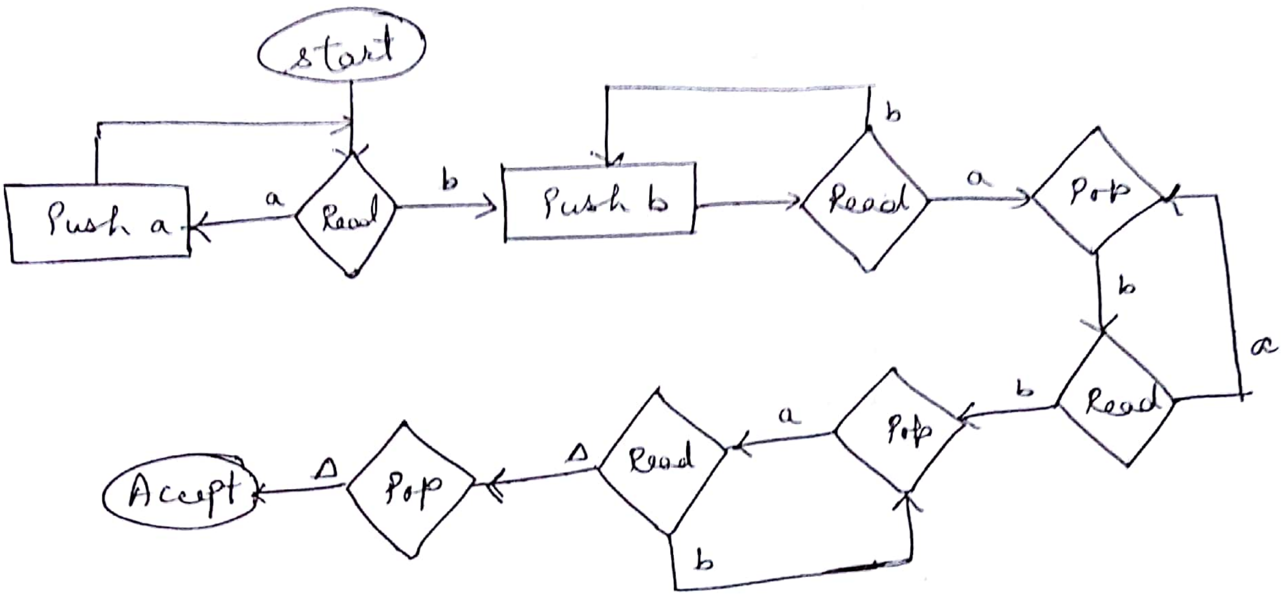
which may have the out-edges labeled with the letters of Γ and the blank character Δ, again with no restrictions.

Examples of PDA's

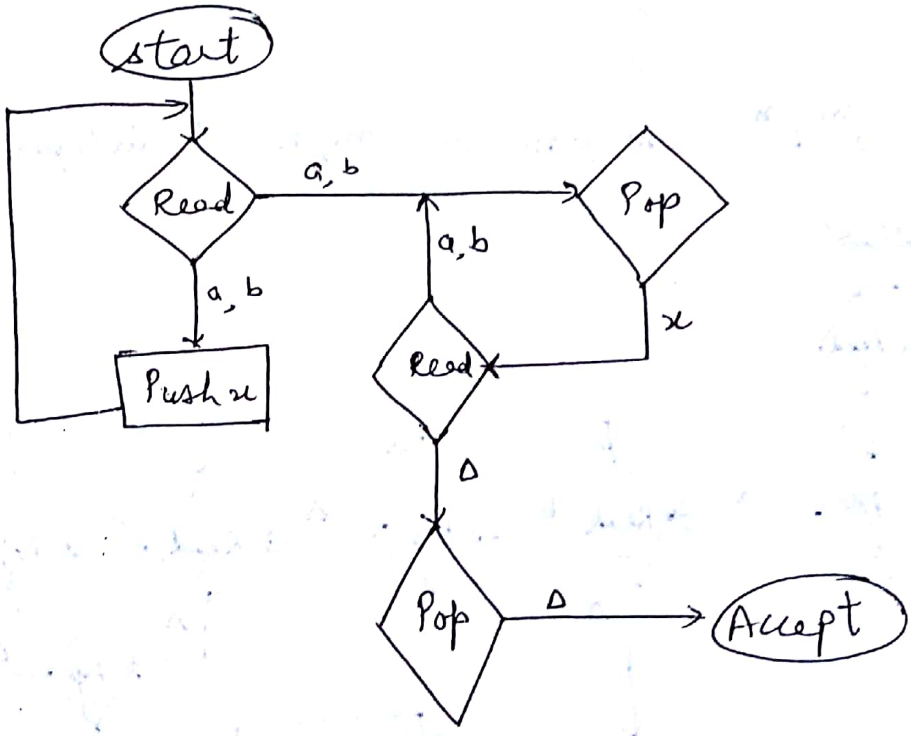
1) $L = \{a^n S$, where S starts with b & length$(S) = n\}$.



2) $L = a^n b^m a^m b^n$, where m, n are independent +ive integers.

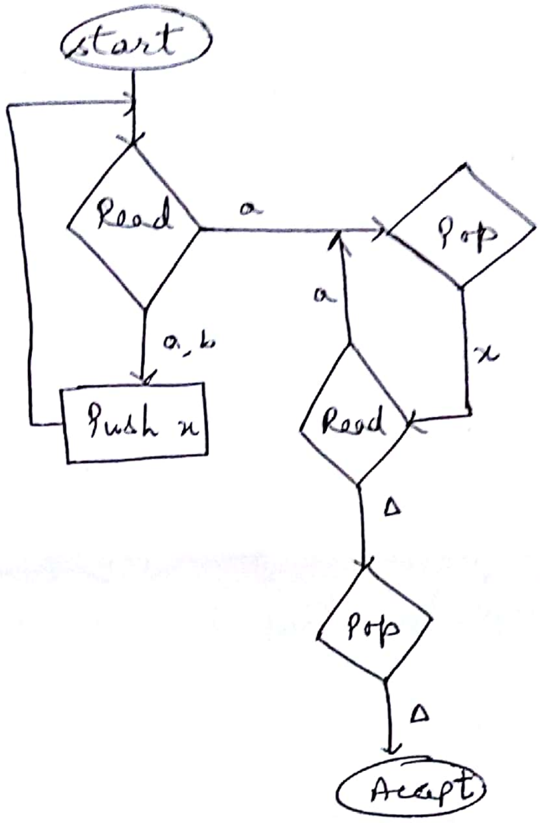


3) PDA that accepts the language of all words with an even no. of letters.

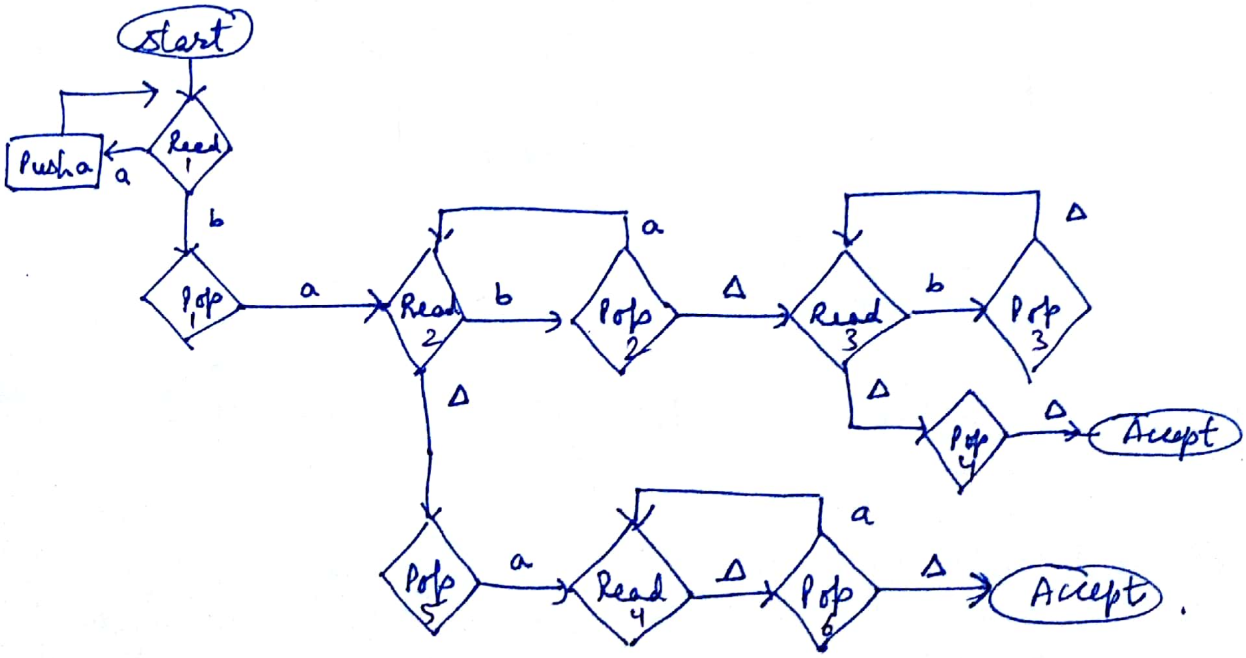


Scanned by CamScanner

4) $L = \{ S\, a^{length(S)}$ that is any string S followed by as many a's as S has letters. $\}$

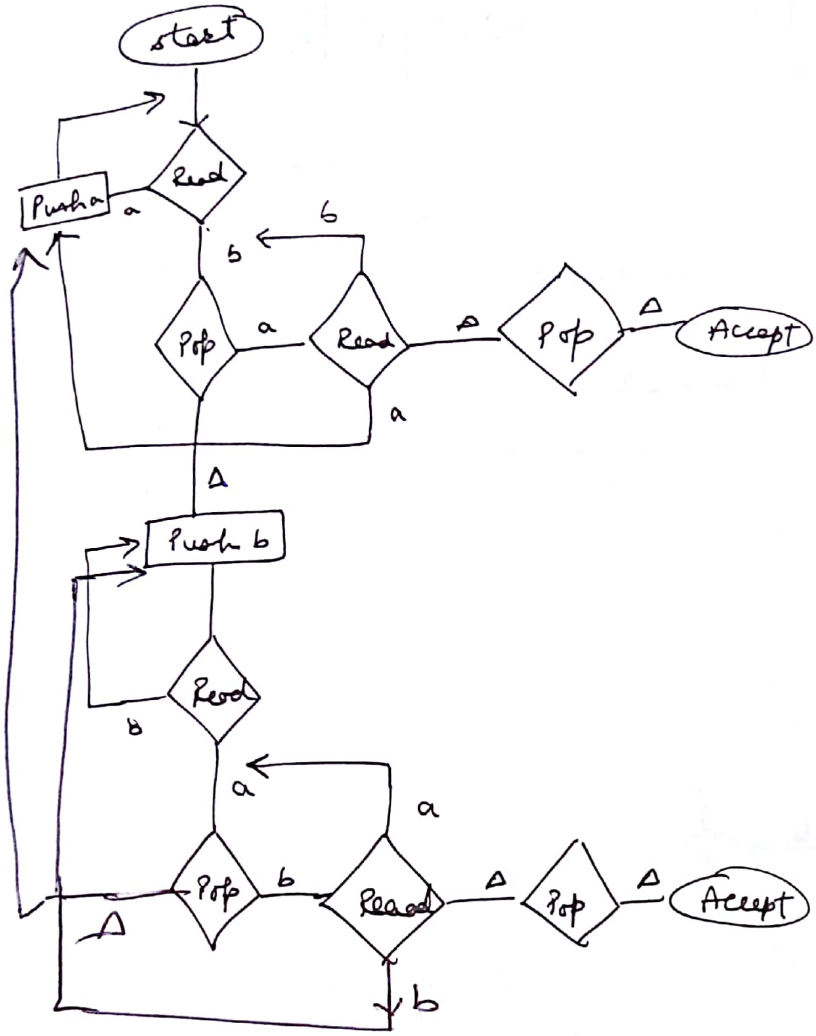


5) $L = \{ a^m b^n \quad m \neq n$ & m, n & independent + integers

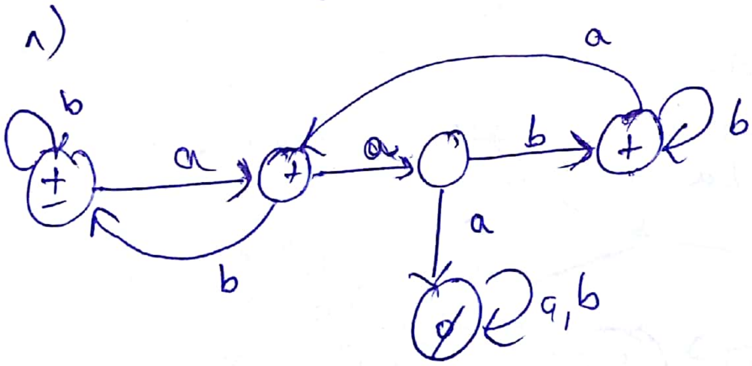


Scanned by CamScanner

PDA for EQUAL.

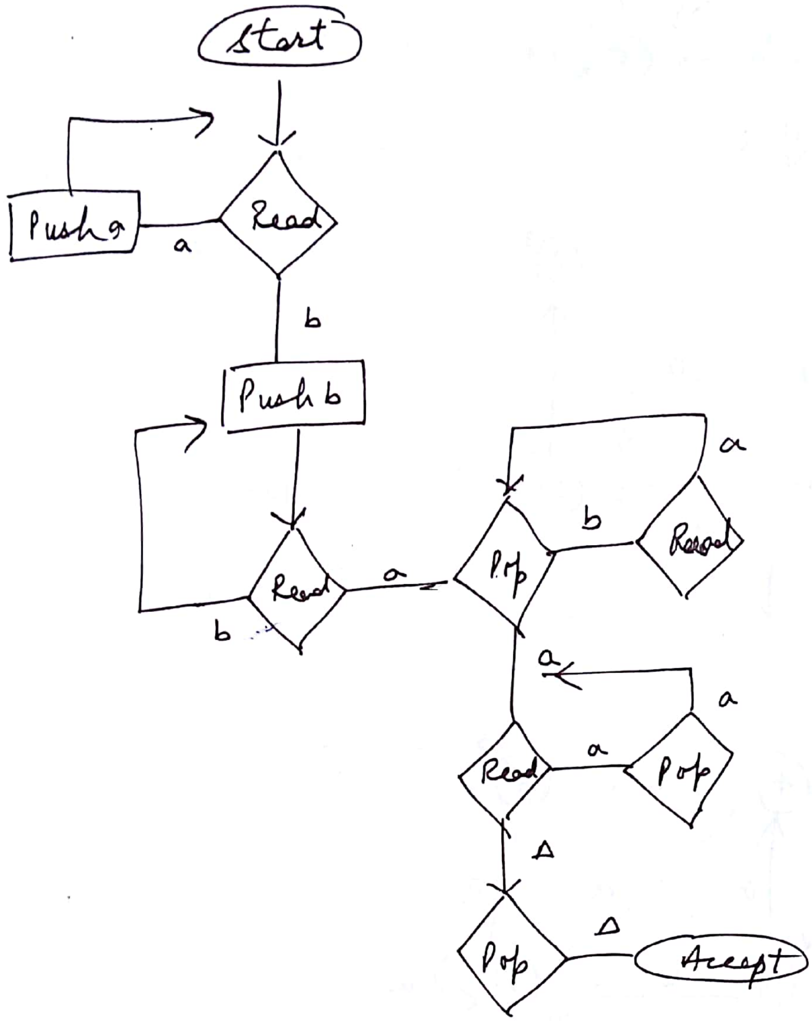


FA in which every 'a' is foll. immediately by 'b'.
(incl ∧)



PDA for

$a^m b^n a^{m+n}$, $n \geq 1$, $m \geq 1$



$m, n \geq 1$

$a^m b^n$, $m \neq n$

$S \to aAb \mid aBb$

$A \to aAb \mid aA \mid a$

$B \to aBb \mid bB \mid b$

CFG for Non-Palindrome

~~S → A|B|A~~

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

$C \to aC \mid bC \mid \Lambda$

~~B → bBb | bCa | ba | ab | aBa~~

$a^m b^n$ acbba $m \neq n$, $m, n \geq \phi$

$S \to aSb \mid B \mid C$

$B \to bB \mid b$

$C \to aC \mid a$

$m > n$

$S \to aBb$

$B \to aBb \mid aB \mid a$

$a^m b^n$, $m > n$

~~S → aSb | B~~

~~B → aB | a~~ $m > n$

$S \to aBb$

$B \to aBb \mid aB \mid a \mid \Lambda$

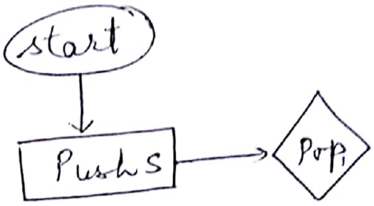
CFG = PDA.

Building a PDA for every CFG.

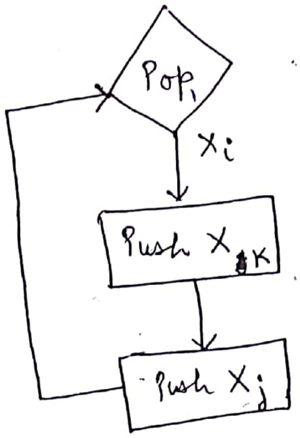
Rules:

1). Convert the CFG in CNF.

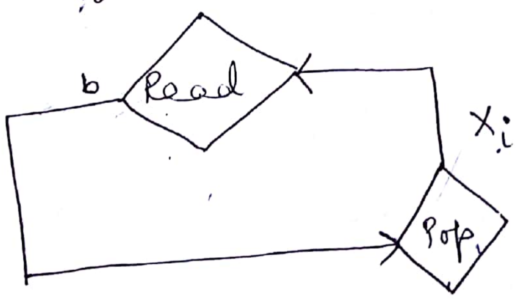
2). Then Begin with



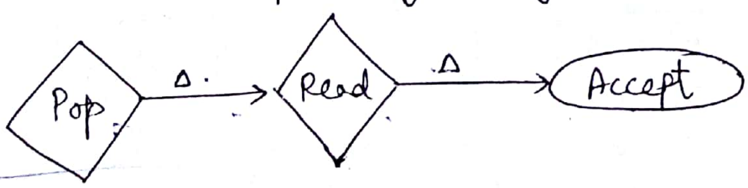
3). For each production of the form $X_i = X_j X_k$, include the circuit from the PoP back to itself:



4). For all productions of the form $X_i \rightarrow b$, we include the circuit

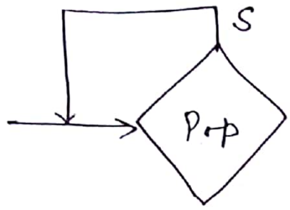


5). When the stack is finally empty, we follow the path;



Scanned by CamScanner

(6). To include $\wedge$, we need to add another circuit to the PDA, a simple loop at the Pop.



Example :- CFG (in CNF & plus one $\wedge$-production).

$S \to AR_1$ $\quad$ $S \to a$

$R_1 \to SA$ $\quad$ $S \to b$

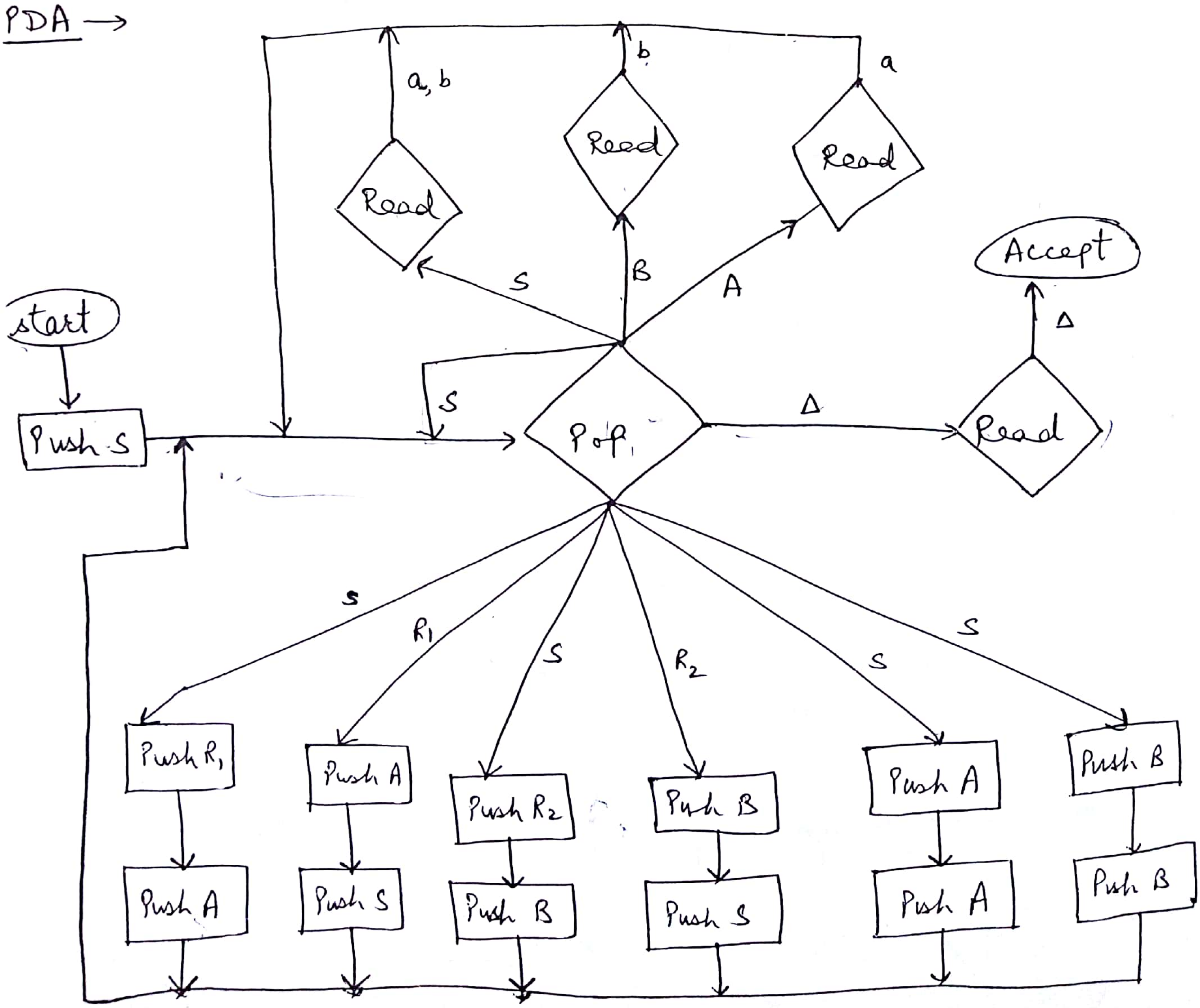
$S \to BR_2$ $\quad$ $A \to a$

$R_2 \to SB$ $\quad$ $B \to b$

$S \to AA$ $\quad$ $S \to \wedge$

$S \to BB$

PDA $\to$



Scanned by CamScanner