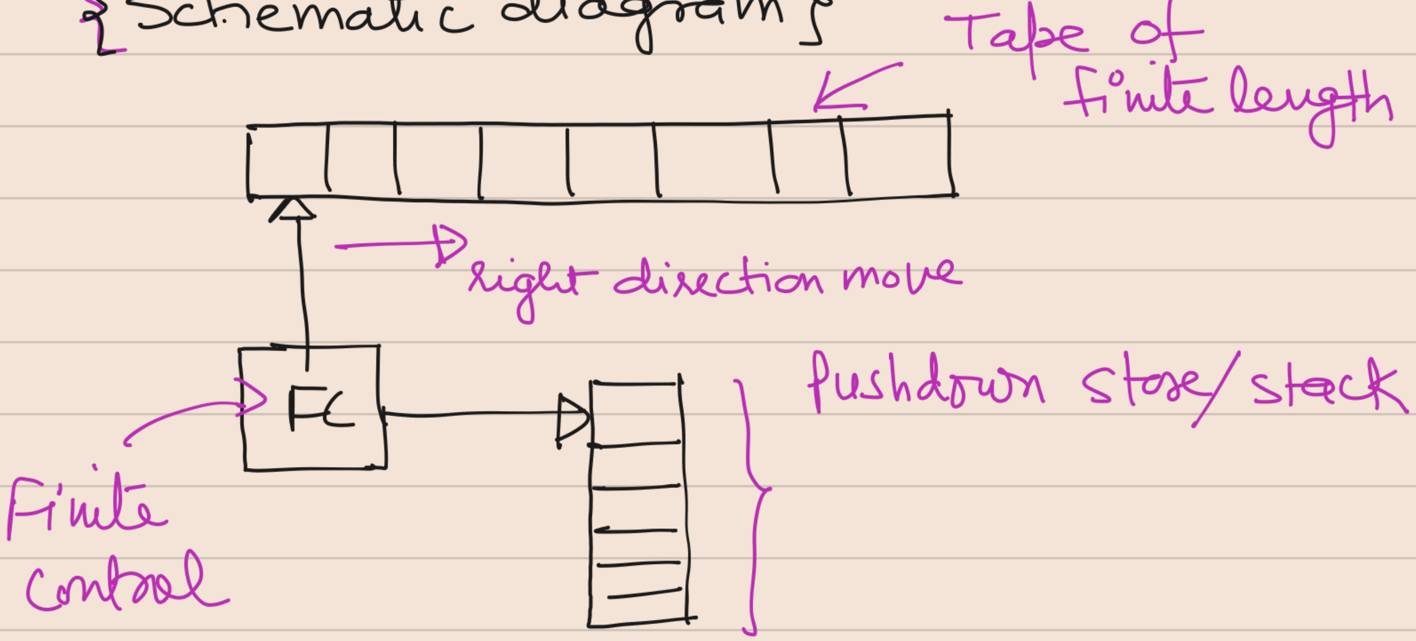
Pushdown Automata:

{Schematic diagram}



{Mathematical description}

$$(Q, \Sigma, \delta, q_0, F, \Gamma, z_0)$$

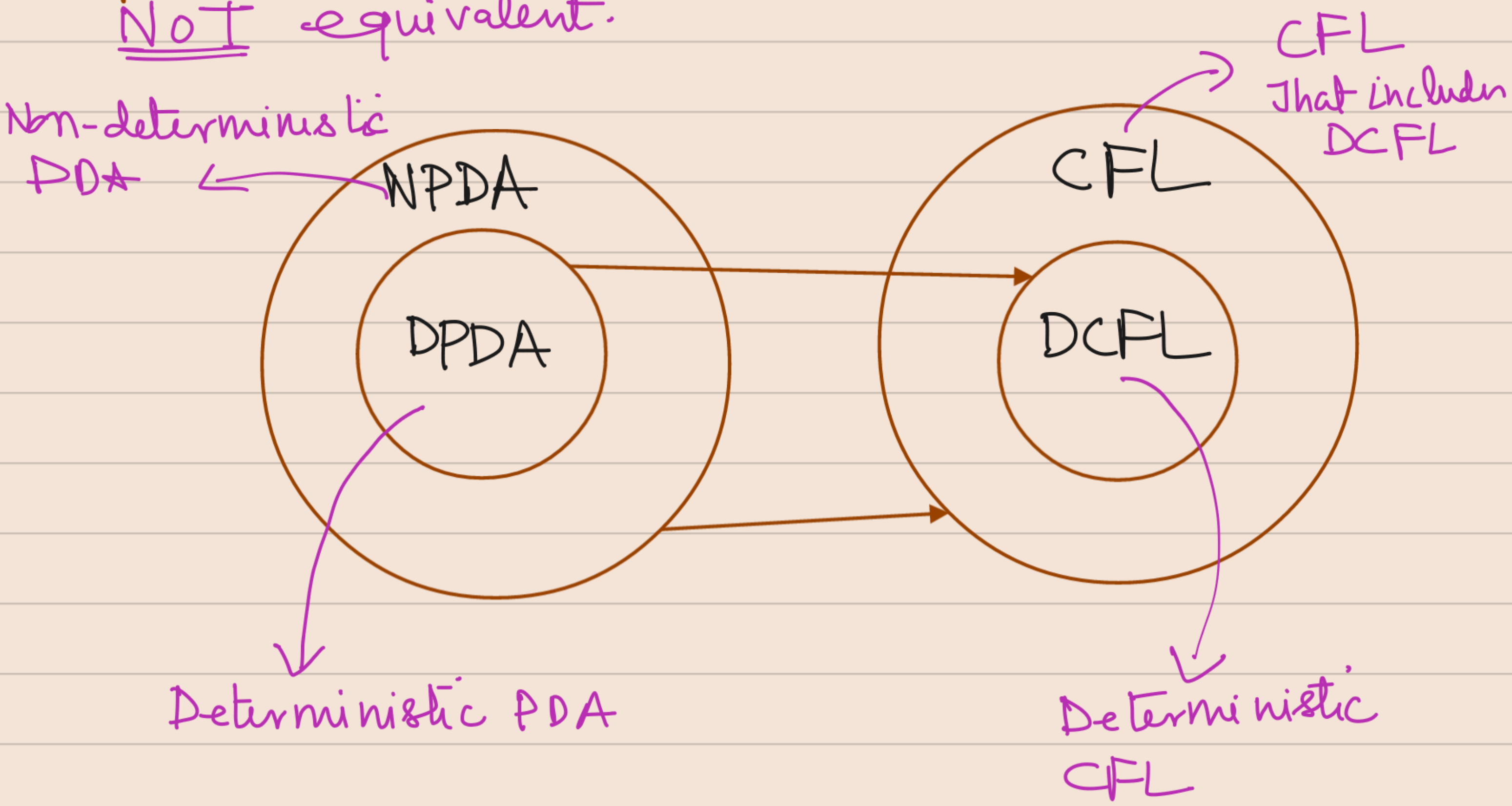
- Q: N.E finite set of states
- Σ: alphabet
- δ: Transition function
- $q_0 \in Q$: initial state
- $F \subseteq Q$: Final states
- Γ: Set of PDS symbols
- $z_0 \in \Gamma$: initial symbol in PDS

$$\delta : Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \rightarrow Q \times \Gamma^*$$

Note:

1. single top of the stack can be replaced by a complete string (of the symbols taken from Γ)
2. can make a move, even without reading anything from the tape.

Note: Capability of NPDA (Non-deterministic pushdown automata) and DPDA (Deterministic pushdown automata) is <u>different</u>. They are <u>NOT</u> equivalent.

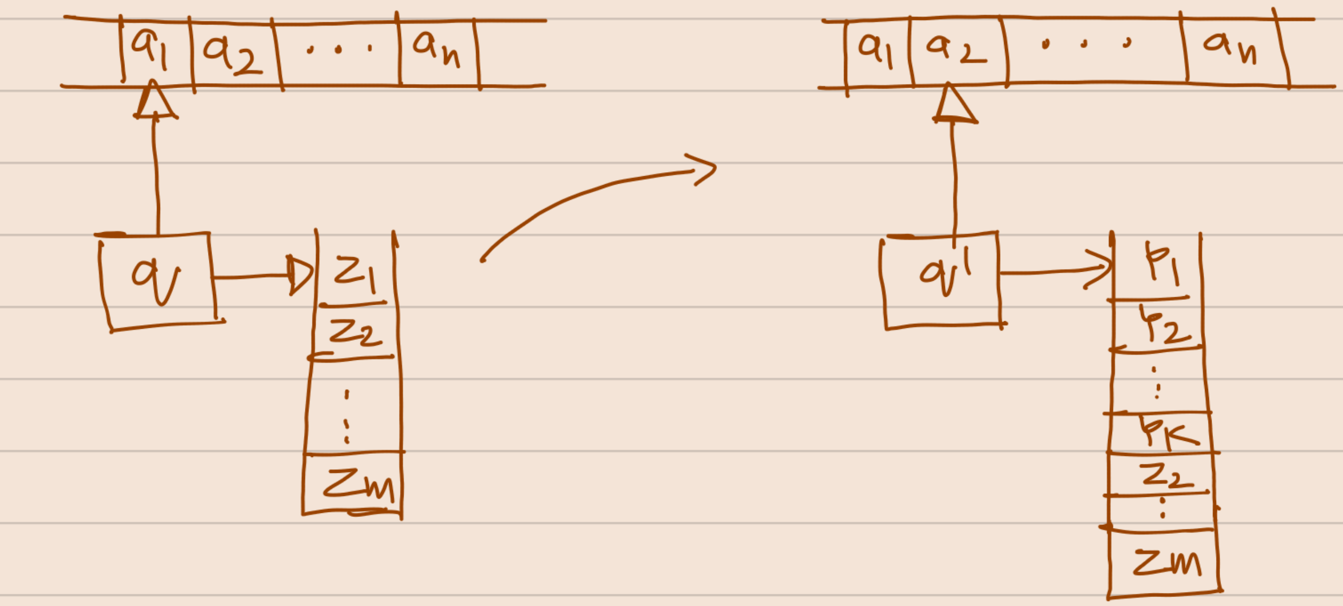


Important terms :

① Instantaneous descriptor (ID) – :

$$(q, x, \alpha)$$

- $q \in Q$: Current state of PDA
- $x \in \Sigma^*$: remaining string to be scanned.
- $\alpha \in \Gamma^*$: current complete content of stack.



ID₁

$$(\underbrace{q}_{\text{current state}}, \underbrace{a_1a_2\cdots a_n}_{\text{Remaining string on tape}}, \underbrace{Z_1Z_2\cdots Z_m}_{\text{Content of stack}})$$

ID₂

$$(q', a_2a_3\cdots a_n, Y_1Y_2\cdots Z_m)$$

This change in ID's (from ID_1 to ID_2) must have occurred due to the transition function.

$$\delta(q, a_1, Z_1) = (q', \underbrace{Y_1Y_2\cdots Y_K}_{})$$

replace Z_1 by $Y_1Y_2\cdots Y_K$ but push $Y_1Y_2\cdots Y_K$ in the reverse order in stack to keep Y_1 on the top.

<u>Note</u>: We study the <u>behaviour of PDA</u> in terms of the <u>changes in the ID's</u>.

(2) Acceptance of a string (w):

Case-1 (Acceptance through Final state)

$$(q_0, w, z_0) \vdash^{*} (q_f, \epsilon, \alpha)$$

Defn: Let $M = (Q, \Sigma, \delta, q_0, F, \Gamma, z_0)$ be a PDA. The set $T(A)$ ~~a~~ accepted by PDA through Final state is defined by

$$T(A) = \{ w \in \Sigma^* \mid (q_0, w, z_0) \vdash^{*} (q_f, \epsilon, \alpha) \text{ for some } q_f \in F \text{ and } \alpha \in \Gamma^* \}$$

Case-2 (Acceptance through null store or empty stack)

$$(q_0, w, z_0) \vdash^{*} (q, \epsilon, \epsilon)$$

Defn: Let $M = (Q, \Sigma, \delta, q_0, F, \Gamma, z_0)$ be a PDA. The set $N(A)$ ~~a~~ accepted by PDA through Null store (or empty stack) is defined by

$$N(A) = \{ w \in \Sigma^* \mid (q_0, w, z_0) \vdash^{*} (q, \epsilon, \epsilon) \text{ for some } q \in Q \}$$

Non-final

An Important theorem:

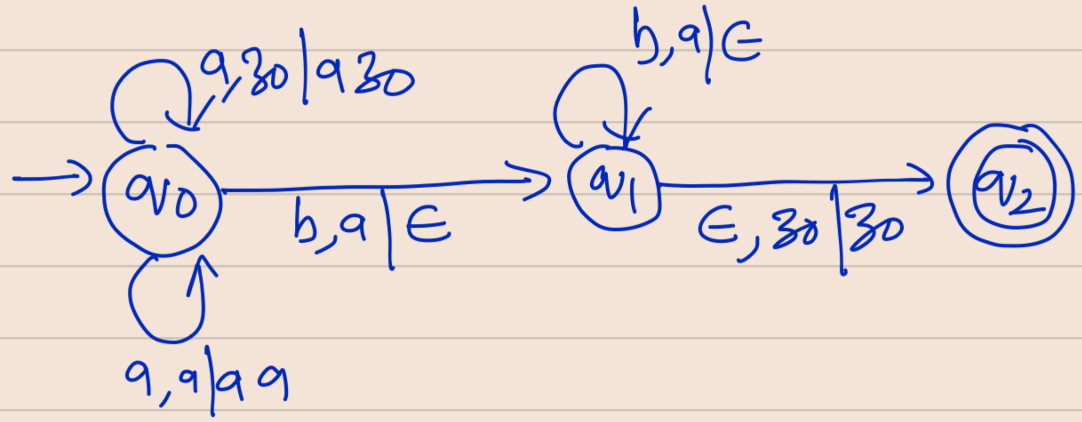
if $M = (Q, \Sigma, \delta, q_0, F, \Gamma, z_0)$ accepts L by final state, we can design another PDA M', which will accept L by the empty store and vice-versa

$$L = T(M) = N(M')$$

Design of Pushdown Automata :

" Develop the logic of the solution in terms of PUSH/POP/SKIP operations "

<u>Ex:</u> $L = \{ a^n b^n \mid n \geq 1 \}$

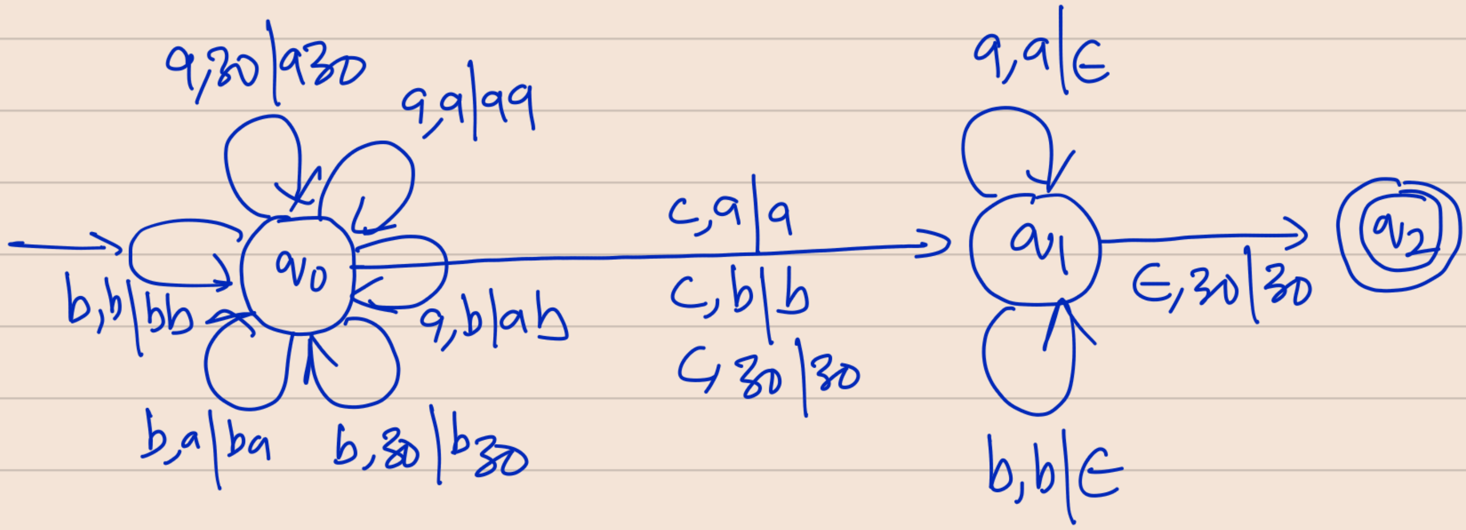


<u>Note:</u> (1) All possible transitions from a state need not to be defined, even for a deterministic pushdown Automata (DPDA).

(2) It's suggested to write transition fn also in the answer, apart from the diagram.

$\delta(q_0, a, z_0) = (q_0, a z_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, a) = (q_1, \epsilon)$
$\delta(q_1, b, a) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$

Ex $L = \{ wcw^R \mid w \in \{a,b\}^* \}$



Note: A PDA is deterministic if there is no configuration for which PDA has a choice of more than one move.
In other words, PDA is deterministic, if it satisfies both of the following conditions.

Condition 1:

$\forall q \in Q$, $a \in \Sigma \cup (\epsilon)$, and $X \in \Gamma$
$\delta(q, a, X)$ has at most one element.

Condition 2:

$\forall q \in Q$ and $x \in \Gamma$, if $\delta(q, \epsilon, x) \neq \phi$
then $\delta(q, a, x) = \phi \; \forall a \in \Sigma$

Failing either condition shall make the design "Non-deterministic".

{ Non-deterministic Pushdown automata)

mathematically,

$(Q, \Sigma, \delta, q_0, F, Z, z_0)$

$\delta : Q \times (\Sigma \cup (\epsilon)) \times \Gamma \rightarrow P(Q \times \Gamma^*)$

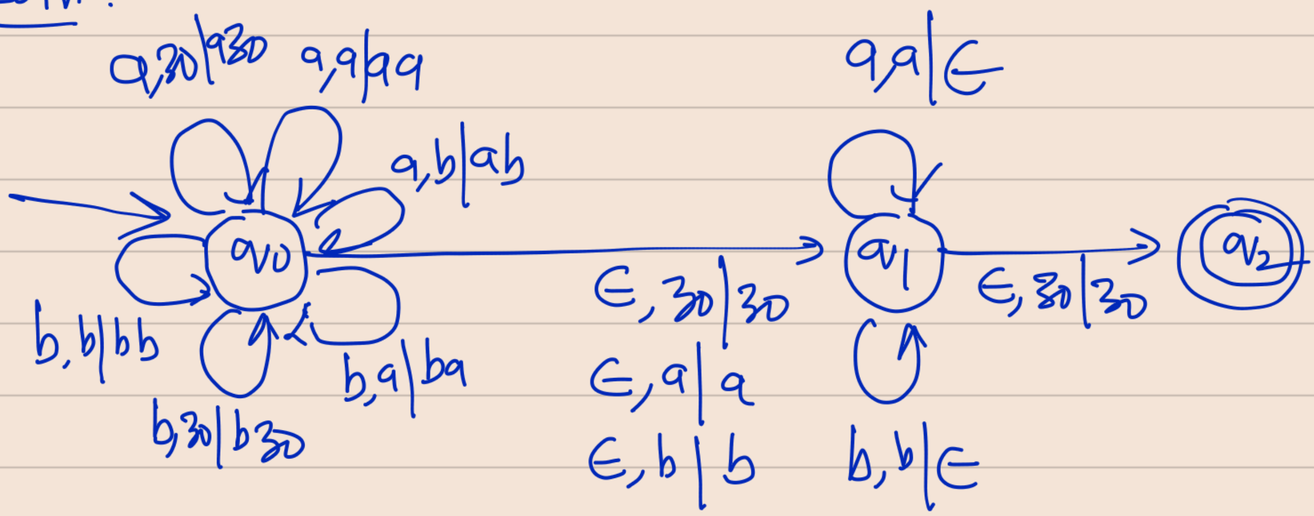
Powerset

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

Ex: PDA (NPDA) for even Palindrome over {a, b}.

OR $L = \{ x x^R \mid x \in \{a,b\}^* \}$

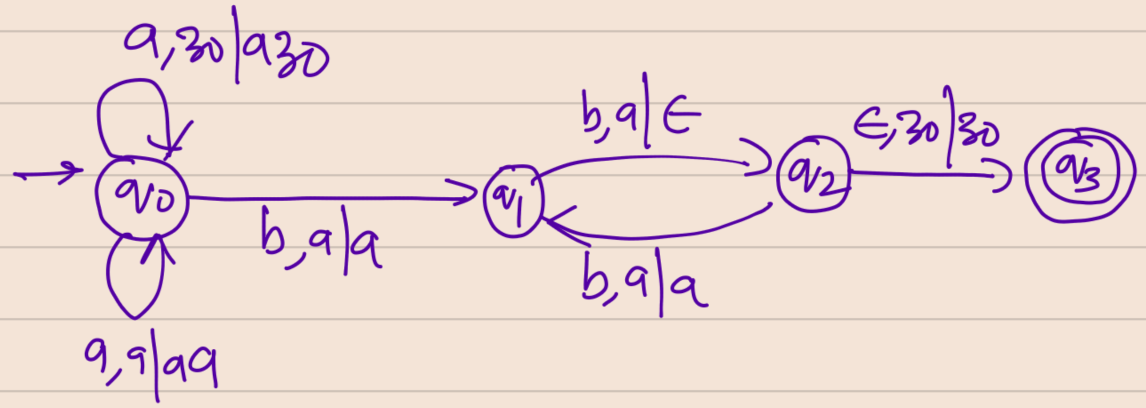
Soln:



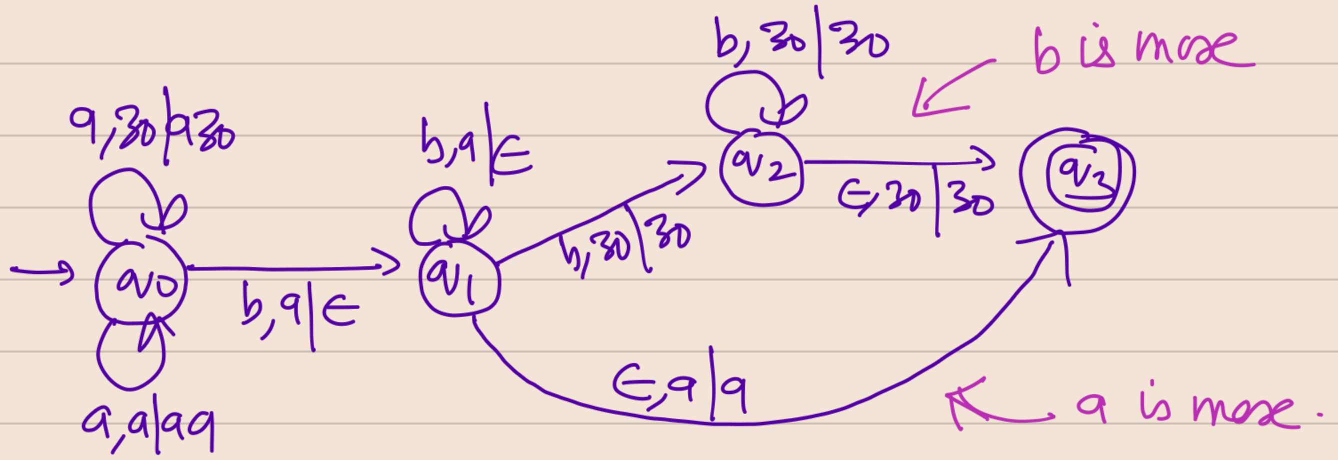
Exercise: Design PDA to accept palindrome (even and odd, both) over $\Sigma = \{a, b\}$.

Solved Examples :

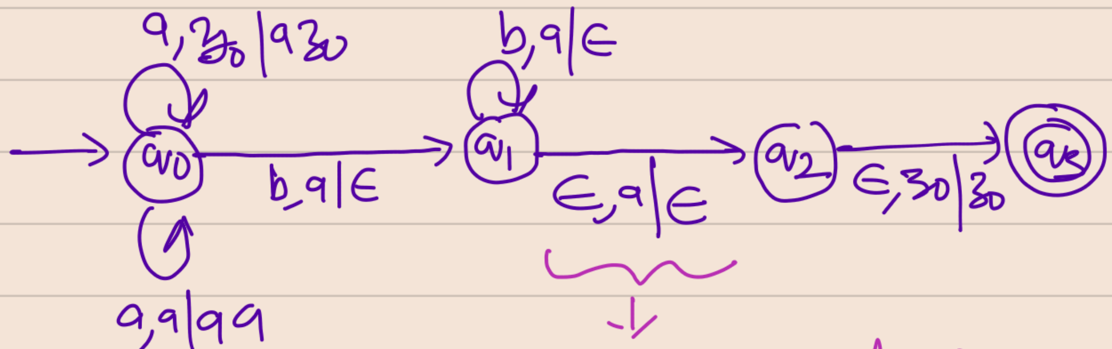
① $L = \{a^n b^{2n} \mid n \geq 1\}$



② $L = \{a^i b^j \mid i, j \geq 1,\ i \neq j\}$



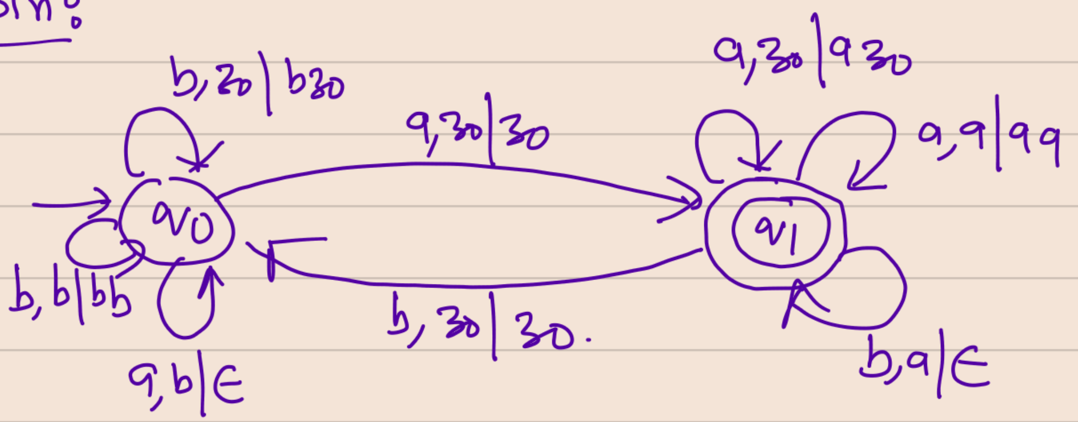
③ $L = \{a^i b^j \mid i, j \geq 1,\ i = j + 1\}$



To ensure only One a is more

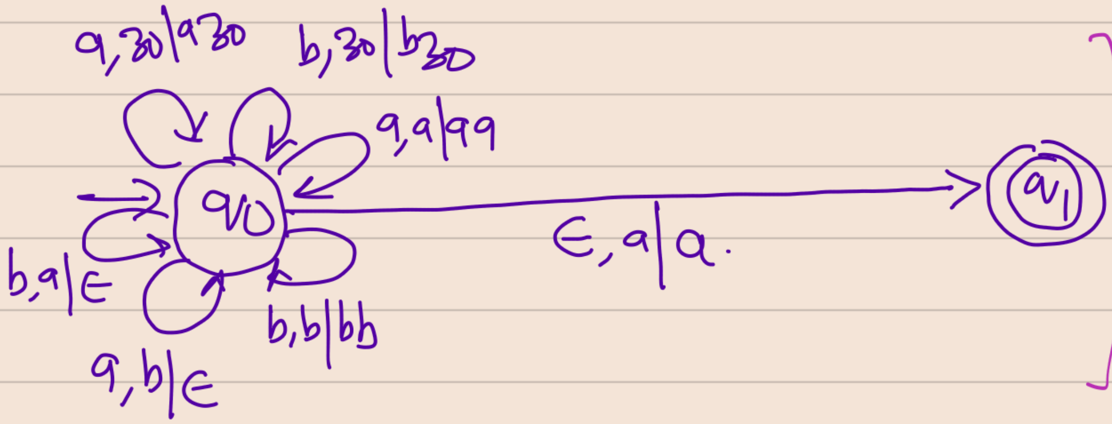
④ $L = \{ x \in \{a,b\}^* \mid \underbrace{n_a(x)}_{\text{No. of a's in } x} > \underbrace{n_b(x)}_{\text{No. of b's in } x} \}$

<u>Soln:</u>



Deterministic PDA (DPDA)

<u>OR</u>



Non-deterministic PDA (NPDA)

[Ref: "Introduction to languages and theory of computation" by John. C. Martin (TMH)]