Push Down Automata

PDA

PDA = FA + stack

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

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

This is a CFG, not the Regular language. This langu
cannot be accepted by any finite automata, becaus
it requires the comparision among the equal
number of a and equal no. of b. Finite automa
cannot have the backward information, which
must be required for making the comparison
between the no. of a's and no. of b's. This
limitation can be removed by adding a auxiliary
memory into the form of stack, which works
as follows:

1) All the a's present into the I/p string is pushed into the stack.
2) Corresponding to every b present into the I/p string, pop operation is performed on stack.

3) If after Reading the whole string, stack is empty then I/p string must have equal no. of a's and b's.
Otherwise it doesn't have equal no. of a's and b's.

Finite automata with the stack leads to the generation of new type of machine called PDA. PDA can recognise the strings of CFL.

$$\boxed{PDA = FA + stack}$$

Push Down Automata can be described by 7 tuples.

PDA $A(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

where

Q : finite set of states

Σ : Set of I/p alphabets

Γ : Set of pushdown symbol / Stack alphabet ($\Sigma \subseteq \Gamma$)

δ : Transition Mapping function

q_0 : Initial state of PDA

($Z_0 \in \Gamma$) Z_0 : Bottom / Initial Pushdown (stack) symbol

F : Set of final states

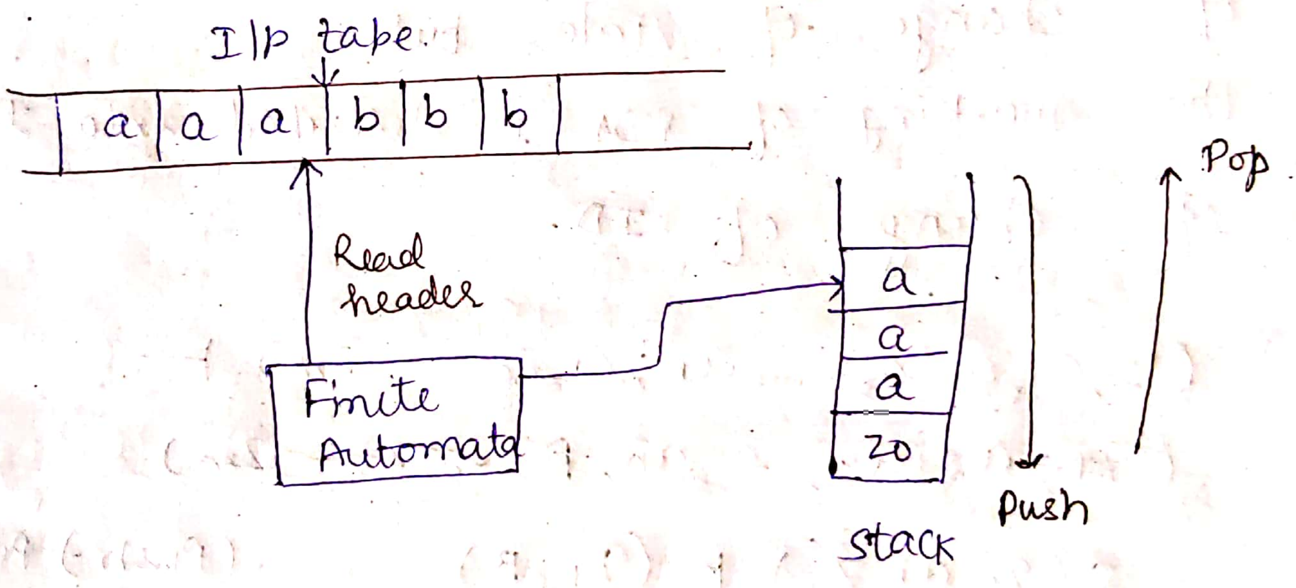
Transition Mapping function

DPDA

$Q \times \{\Sigma \cup \lambda\} \times \Gamma \rightarrow Q \times \Gamma^*$

NDPDA

$Q \times (\Sigma \cup \lambda) \times \Gamma \rightarrow 2^{(Q \times \Gamma^*)}$



Push : Insert into the stack

Pop : Retrieve the element from the stack.

Stack allow three operations

→ Push

→ Pop

→ Skip

Instantaneous Description (ID);

In PDA $A(Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ instantaneous description defined by (q, x, α).

where

$q \in Q$

$x \in \Sigma^*$

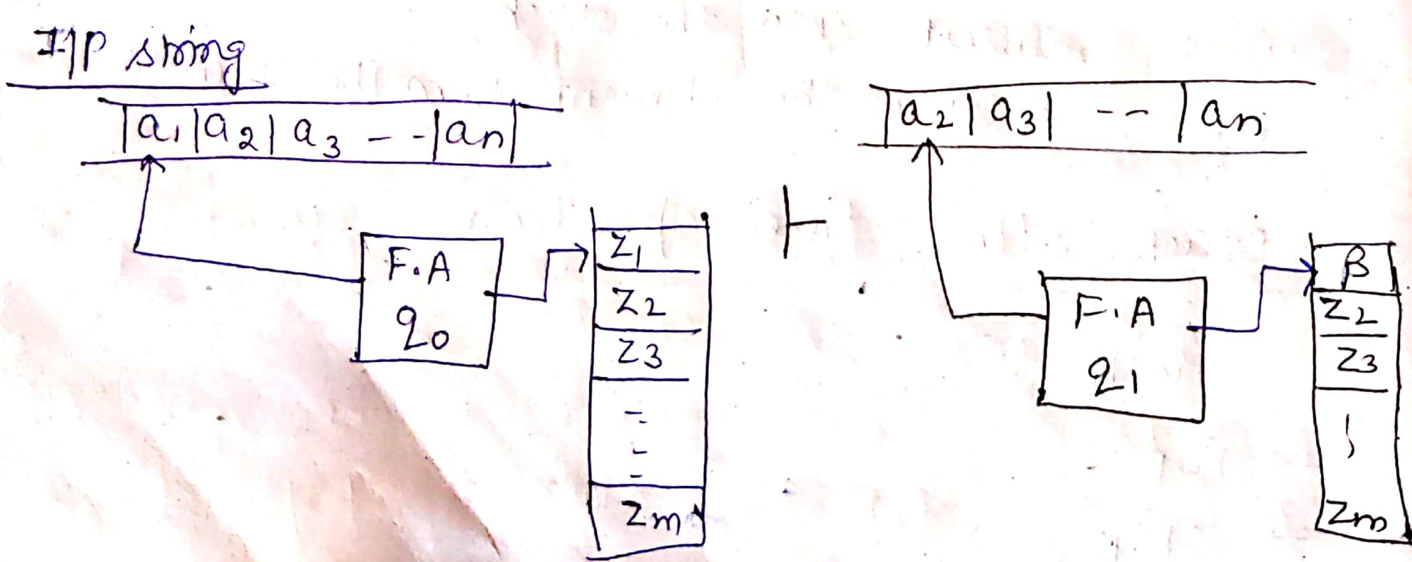
$\alpha \in \Gamma^*$

The working of the FA is described into the form of change of state, but the working of PDA is described into the form of change of ID.

(i) $(q_0, a_1, a_2 a_3 \ldots a_n; Z_1, Z_2 \ldots Z_m) \vdash$

$(q_1, a_2, a_3, \ldots a_n, \beta, Z_2, Z_3 \ldots Z_m)$

$\delta(q_0, a_1, Z_1) \vdash (q_1, \beta)$ (Push & Pop)

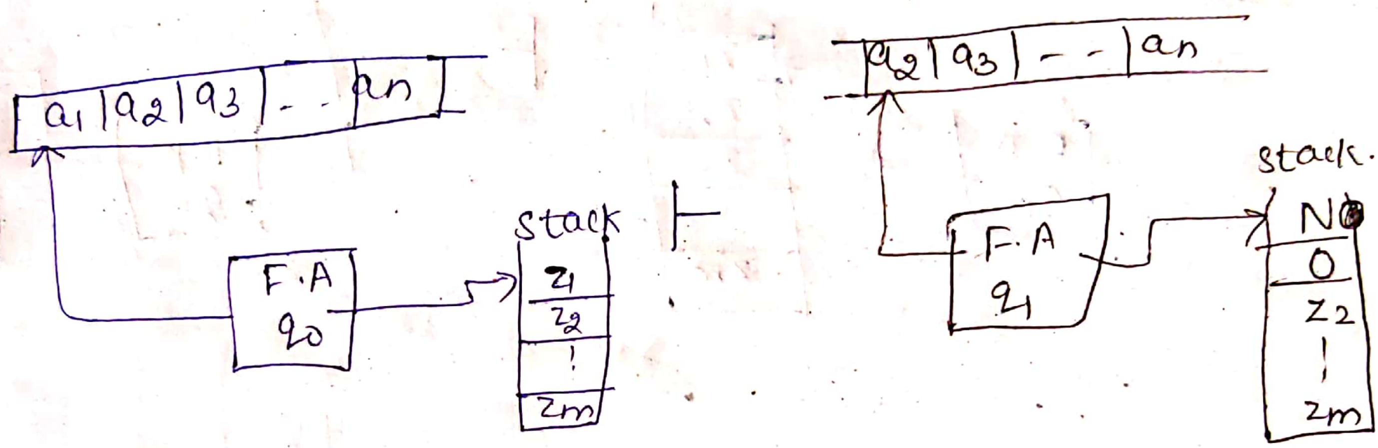


(Push pop)

(ii) $(q_0, a_1, a_2 -- a_n, Z_1, Z_2 -- Z_m) \vdash$

$q_1, a_2, a_3 -- a_n, NO\,Z_2 -- Z_m)$

if $(q_0, a_1, Z_1) \vdash (q_1, NO)$

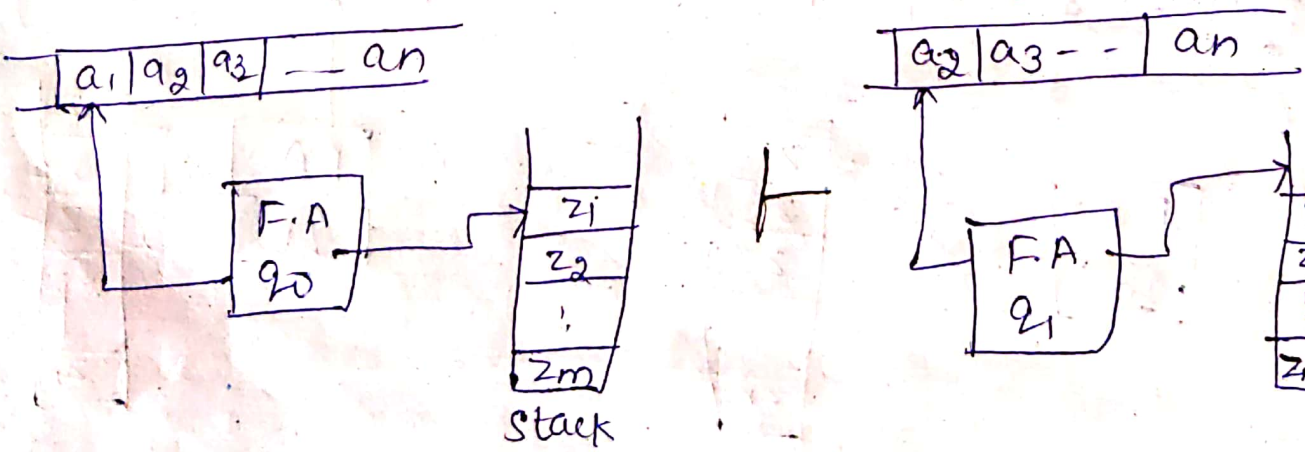


(Push)

(iii) $(q_0, a_1, a_2, a_3 -- a_n, Z_1, Z_2 -- Z_m) \vdash$

$(q_1, a_2, a_3 -- a_n, \beta, Z_1, Z_2 -- Z_m)$

$\delta(q_0, a_1, Z_1) \vdash (q_1, \beta Z_1)$.

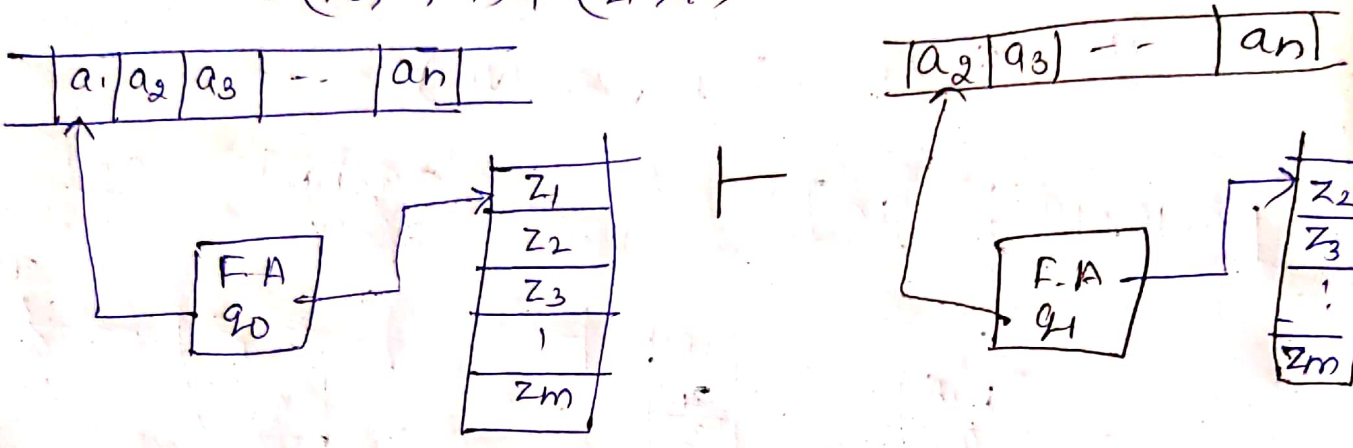


Pop

(iv) $(q_0, a_1, a_2, a_3 - a_n, z_1, z_2, z_3 \ldots z_m) \vdash$

$(q_1, a_2, a_3 \ldots z_2, z_3, - z_m)$

$\delta(q_0, a_1, z_1) \vdash (q_1, \lambda)$

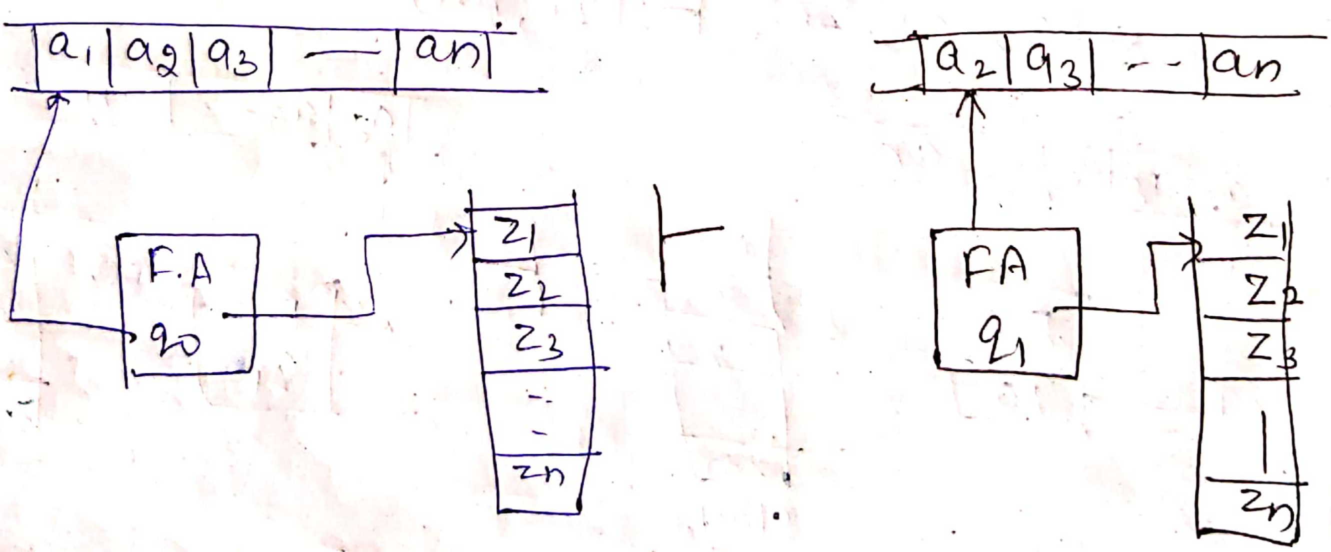


Skip

(v) $(q_0, a_1, a_2, a_3 \ldots a_n, z_1, z_2 \ldots z_m) \vdash$

$(q_1, a_2, a_3 \ldots a_n, z_1, z_2 \ldots z_m)$

$\delta(q_0, a_1, z_1) \vdash (q_1, z_1)$



Acceptance by PDA

Two methods

Acceptance by final state

Acceptance by NULL Store | empty stack

① **Acceptance by final state:**

In a PDA $A (Q, \Sigma, \Gamma, S, q_0, Z_0, F)$ acceptance by final state is represented by $T(A)$ which is described as follows:

$$T(A) = \{ w \mid w \in \Sigma^* , (q_0, w, Z_0) \vdash^* (q_f, \lambda, Z_0) \}$$

where $q_f \in F$.

② **Acceptance by Empty stack**

In a PDA $A (Q, \Sigma, \Gamma, S, q_0, Z_0, F)$ acceptance by empty stack is represented by $N(A)$, which is described as follows:

$$N(A) = \{ w \mid w \in \Sigma^*, (q_0, w, Z_0) \vdash^* (q, \lambda, \lambda)$$

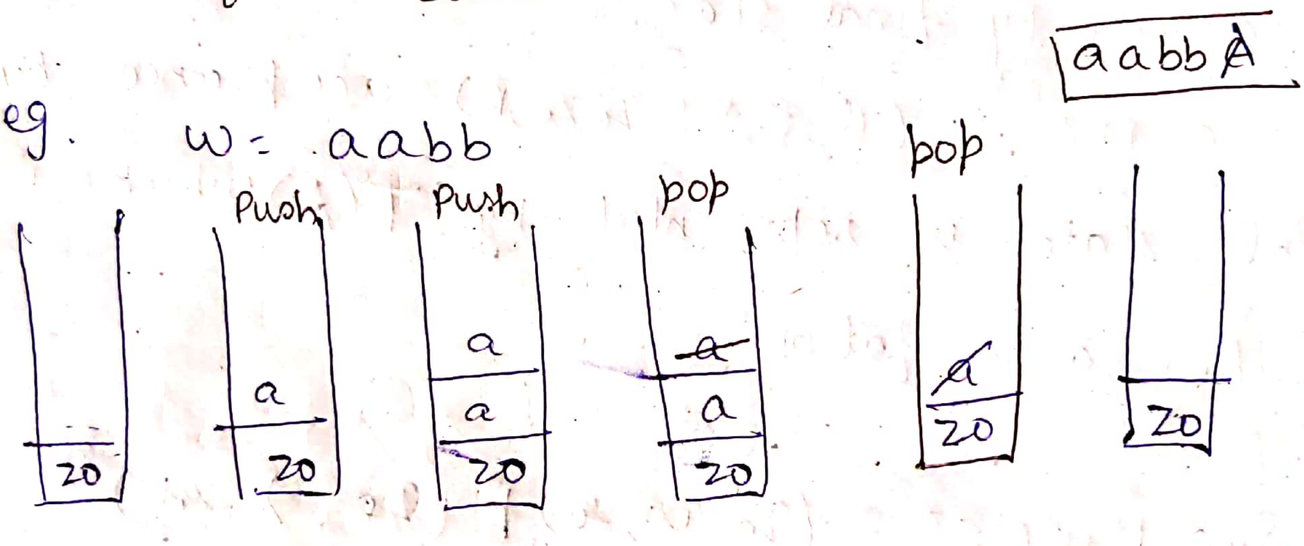
PDA Accepted by final state or accepted by empty stack, both are equivalent in power.

From language to PDA

① Construct the PDA for the language
$a^n b^n \mid n \geq 1$

The string should be accepted by
1). Null store (empty stack)
2) final state

eg. $w = aabb$



I/P	(h)	a	a	b	b	(ε) λ
state	q_0	q_0	q_0	q_1	q_1	q_f

Acceptance by Null store

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

$Q = \{q_0\}$ $\qquad F = \phi$

$\Sigma = \{a, b\}$

$\Gamma = \{a, z_0\}$

$q_0 = \{q_0\}$

$z_0 = \{z_0\}$

Scanned with CamScanner

δ is defined by

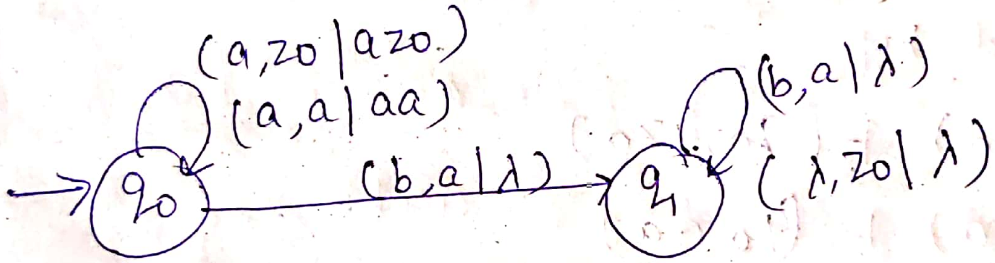
$\delta(q_0, a, z_0) \vdash (q_0, az_0)$ (Push)

$\delta(q_0, a, a) \vdash (q_0, aa)$ (Push)

$\delta(q_0, b, a) \vdash (q_1, \lambda)$ (Pop)

$\delta(q_1, b, a) \vdash (q_1, \lambda)$

$\delta(q_1, \lambda, z_0) \vdash (q_1, \lambda)$



State-transition diagram of PDA

eg. Let us consider the string aaabbb.

Steps: $(q_0, aaabbb, z_0) \vdash (q_0, aabbb, az_0)$
$\vdash (q_0, abbb, aaz_0)$
$\vdash (q_0, bbb, aaaz_0)$
$\vdash (q_1, bb, aaz_0)$
$\vdash (q_1, b, az_0)$
$\vdash (q_1, \lambda, z_0)$
$\vdash (q_1, \lambda, \lambda)$

Acceptance by final state :

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

$Q = \{q_0, q_1, q_f\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, z_0\}$

$q_0 = \{q_0\}$

$z_0 = \{z_0\}$

$F = q_f$

δ is defined by

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

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

$\delta(q_0, b, a) \vdash (q_1, a)$

$\delta(q_1, b, a) \vdash (q_1, \lambda)$

$\delta(q_1, \lambda, z_0) \vdash (q_f, z_0)$

eg. Consider string aaabbb

$\delta(q_0, aaabbb, z_0) \vdash (q_0, aabbb, az_0)$

$\vdash (q_0, abbb, aaz_0)$

$\vdash (q_0, bbb, aaaz_0)$

$\vdash (q_1, bb, aaz_0)$

$\vdash (q_1, b, az_0)$

$\vdash (q_1, \lambda, z_0)$

$\vdash \cancel{(q_f, \lambda, z_0)}\ (q_f, z_0)$

(2) Construct the PDA for accepting set of all strings of a and b, with equal no. of a and equal no. of b. or

$n_a(w) = n_b(w)$ it means

a and b comes in any order.

The string should be accepted by

i) Null store

(ii) final state

I/p	Top of stack	
a	Zo	Push
b	Zo	
a	a	Push
b	b	
a	b	Pop.
b	a	
λ	Zo	Pop.

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

$\delta(q_0, b, Z_0) \vdash (q_0, bZ_0)$

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

$\delta(q_0, b, b) \vdash (q_0, bb)$

$\delta(q_0, a, b) \vdash (q_0, \lambda)$

$\delta(q_0, b, a) \vdash (q_0, \lambda)$

$\delta(q_0, \lambda, Z_0) \vdash (q_0, \lambda)$

Acceptance by null store

PDA A $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = \{q_0\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, Z_0\}$

$q_0 = \{q_0\}$

$Z_0 = \{Z_0\}$

$F = \phi$.

δ is defined by

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

$\delta(q_0, b, z_0) \vdash (q_0, bz_0)$

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

$\delta(q_0, b, b) \vdash (q_0, b, b)$

$\delta(q_0, b, a) \vdash (q_0, \lambda)$

$\delta(q_0, a, b) \vdash (q_0, \lambda)$

$\delta(q_0, \lambda, z_0) \vdash (q_0, \lambda)$

Consider the string baaabb

$(q_0, baaabb, z_0) \vdash (q_0, aaabb, bz_0)$

$\vdash (q_0, aabb, z_0)$

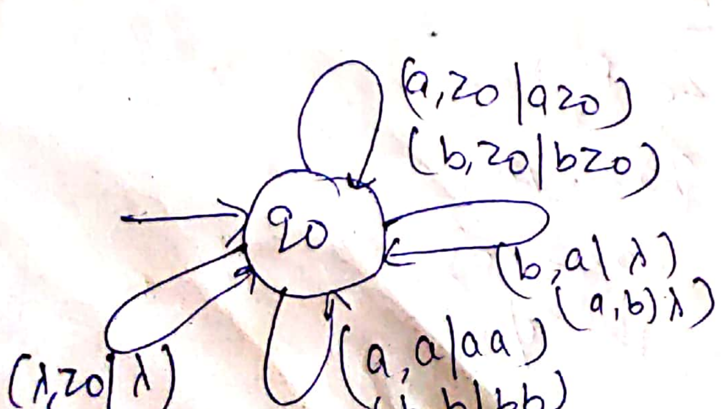
$\vdash (q_0, abb, az_0)$

$\vdash (q_0, bb, aaz_0)$

$\vdash (q_0, b, az_0)$

$\vdash (q_0, \lambda, z_0)$

$\vdash (q_0, \lambda, \lambda)$



state transition diagram

Acceptance by final state

PDA A $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$

$Q = (q_0, q_f)$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, Z_0\}$

$q_0 = \{q_0\}$

$Z_0 = \{Z_0\}$

$F = q_f$

δ is defined by

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

$\delta(q_0, b, Z_0) \vdash (q_0, bZ_0)$

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

$\delta(q_0, b, b) \vdash (q_0, bb)$

$\delta(q_0, b, a) \vdash (q_0, \lambda)$

$\delta(q_0, a, b) \vdash (q_0, \lambda)$

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

Consider string baaabb

$\delta(q_0, baaabb, Z_0) \vdash$

$(q_0, aaabb, bZ_0)$

$\vdash (q_0, aabb, Z_0)$

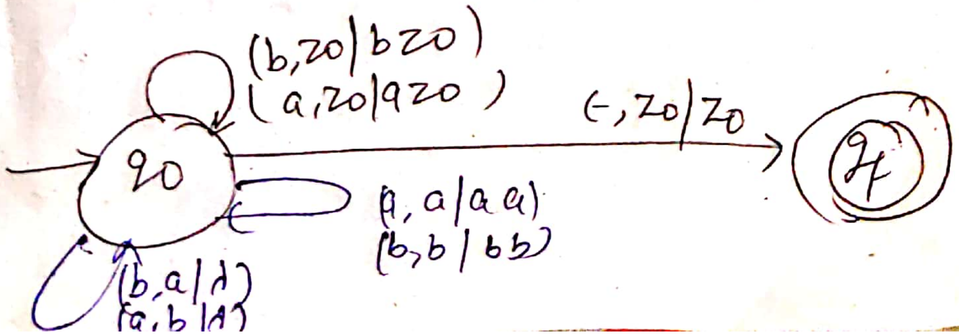
$\vdash (q_0, abb, aZ_0)$

$\vdash (q_0, bb, aaZ_0)$

$\vdash (q_0, b, aZ_0)$

$\vdash (q_0, \lambda, Z_0)$

$\vdash (q_f, Z_0)$



State transition diagram