Pushdown Automata

Since the DFA's or NFA's can not count & can not store the input for future reference, we ~~are~~ have a new machine called pushdown Automata (PDA).

Def^n of PDA: A pushdown Automata (PDA) is a seven tuple

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

where

Q – is set of finite states

Σ – Set of input alphabets

Γ – Set of stack alphabets.

δ – transition from $Q \times (\Sigma \cup \varepsilon) \times \Gamma$ to finite subset of $Q \times \Gamma^*$

δ is called the transition fn of M

$q_0 \in Q$ is the start state of machine

$Z_0 \in \Gamma$ is the initial symbol on the stack

$F \subseteq Q$ is set of final states

The actions (i.e transitions) performed by the PDA depends on

1. The current state
2. The next input symbol.
3. The symbol on top of the stack.

The actions performed by the m/c consists of

1. changing the states from one state to another
2. Replacing the symbol on the stack.

Note: In general, the transition func accepts three parameters namely a state, an input symbol & stack symbol & returns a new state after changing the top of the stack i.e the transition function has the form:

$$\boxed{\delta(\text{state}, \text{input\_symbol}, \text{stack\_symbol}) = (\text{new\_state}, \text{stack\_symbol})}$$

Instantaneous Description

The current configuration of PDA at any given instant can be described by an instantaneous description (in short we call ID). An ID gives the current state of the PDA, the remaining string to be processed & the entire contents of the stack. Thus, an ID can be defined as shown below.

Defn: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ be a PDA, An ID is defined as 3-tuple or a triple

$$(q, w, \alpha)$$

where

q is the current state

w is the string to be processed

α is the current contents of stack.

Acceptance of a language by PDA

There are two cases wherein a String w is accepted by PDA :

* Get the final state from the start state.
* Get an empty stack from the start state.

→ we say that the language is accepted by a final sta
→ " " " " " " " an empty stack or null stack.

The language L(M) accepted by a final state is defined as

$$L(M) = \{ w \mid (q_0, w, z_0) \vdash^* (p, \varepsilon, \alpha) \quad \text{where } \alpha \in \Gamma^*,\ p \in F,\ w \in \Sigma^*$$

It means that the PDA, currently in state q_0, after scanning the string w enters into a final state p. Once the m/c is in state p, the input symbol should b ε & the contents of the stack are irrelevant. Any thing can be there on the stack.

The language N(M) accepted by an empty stack (Null stack) as

$$N(M) = \{ w \mid (q_0, w, z_0) \vdash^* (p, \varepsilon, \varepsilon) \}$$

where $w \in \Sigma^*$, $q_0, p \in Q$. It means that when the string w is accepted by an empty stack, the final state is irrelevant, the i/p should be completely read & the stac should be empty.

Eg obtain a PDA to accept the language

$L(M) = \{ wCw^R \mid w \in (a+b)^* \}$ where w^R is reverse of w by a final state.

<u>Soln</u>: <u>Step 1</u>: I/p symbols can be a or b

$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$
$$\delta(q_0, b, Z_0) = (q_0, bZ_0)$$

once the first scanned i/p symbol is pushed on to the stack, the stack may contain either a or b.

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

$$\delta(q_0, b, a) = (q_0, ba)$$

$$\delta(q_0, a, b) = (q_0, ab)$$

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

<u>Step 2</u>: I/p symbol is C

$$\delta(q_0, C, Z_0) = (q_1, Z_0)$$

$$\delta(q_0, C, a) = (q_1, a)$$

$$\delta(q_0, C, b) = (q_1, b)$$

30

Step 3: I/p symbols can be a or b

$$\delta(q_1, a, a) = (q_1, \varepsilon)$$

$$\delta(q_1, b, b) = (q_1, \varepsilon)$$

Step 4: Finally, in state q_1, if the string is a palindrom there is no input symbol to be scanned & the stack should be empty i.e the stack should contain Z_0. Now, change the state to q_2 & do not alter the contents of the stack. The transition for this can be of t form:

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

So, the PDA M to accept the language

$$L(M) = \{WCW^R \mid w \in (a+b)^*\}$$

along with transition graph is given by

$$M = (Q, \Sigma, T, \delta, q_0, Z_0, F)$$

where

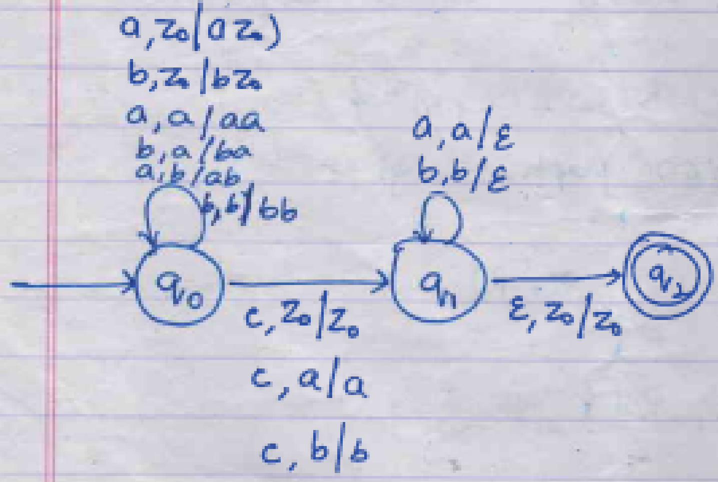
$$Q = \{q_0, q_1, q_2\}$$

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

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

δ: is shown below

$$\delta(q_0, a, z_0) = (q_0, az_0)$$
$$\delta(q_0, b, z_0) = (q_0, bz_0)$$
$$\delta(q_0, a, a) = (q_0, aa)$$
$$\delta(q_0, b, a) = (q_0, ba)$$
$$\delta(q_0, a, b) = (q_0, ab)$$
$$\delta(q_0, b, b) = (q_0, bb)$$
$$\delta(q_0, c, z_0) = (q_1, z_0)$$
$$\delta(q_0, c, a) = (q_1, a)$$
$$\delta(q_0, c, b) = (q_1, b)$$
$$\delta(q_1, a, a) = (q_1, \varepsilon)$$
$$\delta(q_1, b, b) = (q_1, \varepsilon)$$
$$\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$$



$q_0 \in Q$ is the start state of m/c

$z_0 \in \Gamma$ is the initial symbol on the stack

$F = \{q_2\}$ is the final state.

To accept the String: The sequence of moves made by PDA for the string aabCbaa is shown below:

Initial ID

$$(q_0, aabCbaa, Z_0) \vdash (q_0, abCbaa, aZ_0)$$
$$\vdash (q_0, bCbaa, aaZ_0)$$
$$\vdash (q_0, Cbaa, baaZ_0)$$
$$\vdash (q_1, baa, baaZ_0)$$
$$\vdash (q_1, aa, aaZ_0)$$
$$\vdash (q_1, a, aZ_0)$$
$$\vdash (q_1, \varepsilon, Z_0)$$
$$\vdash (q_2, \varepsilon, Z_0)$$

Since q_2 is the final state & i/p string is ε in the final configuration, the string

aabCbaa

is accepted by the PDA.

To reject the String: The sequence of moves made by the PDA for the string aabCbab is shown below:

Initial ID:

$$(q_0, aabcbab, Z_0) \vdash (q_0, abCbab, aZ_0)$$
$$\vdash (q_0, bCbab, aaZ_0)$$
$$\vdash (q_0, Cbab, baaZ_0)$$
$$\vdash (q_1, bab, baaZ_0)$$
$$\vdash (q_1, ab, aaZ_0)$$
$$\vdash (q_1, b, aZ_0) \rightarrow \text{final configuration}$$

Since the transition $\delta(q_1, b, a)$ is not defined, the string

aabcbab

is not a palindrome & the m/c halts & the string is rejected by the PDA.

Note: The same problem can be converted to accept the language by an empty stack. Only the change is, instead of the final transition namely

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

replace it by the transition

$$\delta(q_1, \varepsilon, Z_0) = (q_1, \varepsilon)$$

when a language is accepted by an empty stack, finally the stack should not contain any thing including Z_0. Note that q_1 is not a final state. There is no final state.

Eg Obtain a PDA to accept the language

$L = \{a^n b^n \mid n \geq 1\}$ by a final state.

Soln

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

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

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

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

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

So the PDA to accept the language

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

$M = (Q, \Sigma, \Gamma, \delta_0, q_0, F)$

where

$Q = \{q_0, q_1, q_2\}$

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

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

δ: is shown below

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

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

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

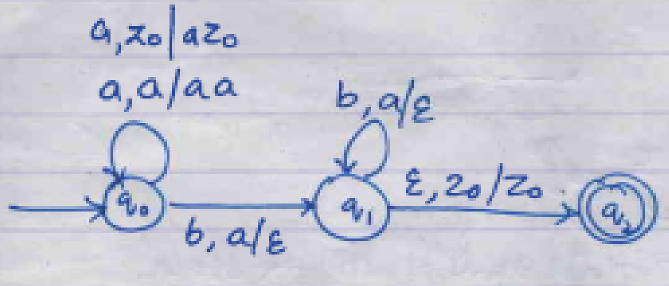
$\delta(q_1, b, a) = (q_1, \varepsilon)$

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

$q_0 \in Q$ is the start state of m/c

$Z_0 \in \Gamma$ is the initial symbol on the stack

$F = \{q_2\}$ is the final state.

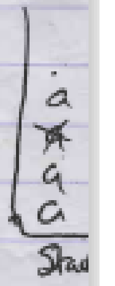


[Note: when to change the state of PDA, if continuous push or pop operation is done then change the state, else remain in the same state].

Eg) obtain a PDA to accept the lang $L(M) = \{ w \mid w \in ($
and $n_a(w) = n_b(w)$ by a final state.

Sol? δ :

$$\delta(q_0, a, Z_0) = (q_0, aZ_0)$$
$$\delta(q_0, b, Z_0) = (q_0, bZ_0)$$



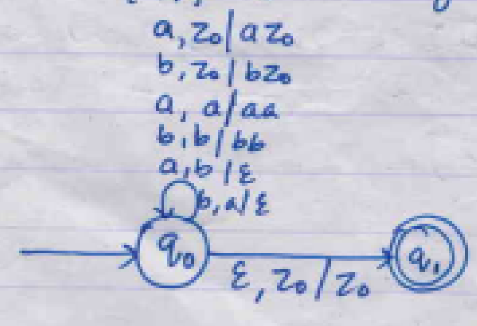
$$\delta(q_0, a, a) = (q_0, aa)$$
$$\delta(q_0, b, b) = (q_0, bb)$$
$$\delta(q_0, a, b) = (q_0, \varepsilon)$$
$$\delta(q_0, b, a) = (q_0, \varepsilon)$$
$$\delta(q_0, \varepsilon, Z_0) = (q_1, Z_0)$$

aaabab

So the PDA to accept the language
$$L = \{ w \mid n_a(w) = n_b(w) \}$$

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

$Q = \{q_0, q_1\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, b, Z_0\}$
δ is shown above
$q_0 \in Q$ is the start state of m/c
$Z_0 \in \Gamma$ is the initial symbol on the stack
$F = \{q_1\}$ is the final state



To accept the lang by an empty stack, the final state is irrelevant where as the next i/p symbol to be scanned be ε & stack should be empty. Even Z_0 should not be there on the stack. So, to obtain the PDA to accept equal number of a's & b's using empty stack, change only the last transaction.

The last transition

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

can be changed as

$$\delta(q_0, \epsilon, Z_0) = (q_1, \epsilon).$$

Eg obtain a PDA to accept a string of balanced parentheses. The parentheses to be considered are (,), [,]

Sol^n

$$\delta(q_0, (, Z_0) = (q_1, (Z_0)$$
$$\delta(q_0, [, Z_0) = (q_1, [Z_0)$$

$$\delta(q_1, (, () = (q_1, (()$$
$$\delta(q_1, (, [) = (q_1, ([)$$
$$\delta(q_1, [, () = (q_1, [()$$
$$\delta(q_1, [, [) = (q_1, [[)$$
$$\delta(q_1,), () = (q_1, \epsilon)$$
$$\delta(q_1,], [) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, Z_0) = (q_0, Z_0)$$

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

where

$Q = \{q_0, q_1\}$

$\Sigma = \{ (,), [,] \}$

$\Gamma = \{ (, [, Z_0 \}$

δ is shown above

$q_0 \in Q$ is the start state of m/c

$Z_0 \in \Gamma$ is the initial symbol on the stack

$F = \{q_0\}$ note that even ε is accepted by PDA & is valid.

Eg) obtain a PDA to accept the language $L = \{ w \mid w \in (a,b)^*$ and $n_a(w) > n_b(w) \}$.

Sol^n The PDA to accept the language

$$L = \{ w \mid n_a(w) > n_b(w) \}$$

is given by

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

where

$Q = \{q_0, q_1\}$

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

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

δ: is shown below

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

$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, \varepsilon)$
$\delta(q_0, b, a) = (q_0, \varepsilon)$
$\delta(q_0, \varepsilon, a) = (q_1, a)$

$q_0 \in Q$ is the start state of m/c
$Z_0 \in \Gamma$ is the initial symbol on the stack
$F = \{q_1\}$ is the final state.

Eg: Obtain a PDA to accept the language

$$L = \{w \mid w \in (a,b)^* \text{ and } n_a(w) < n_b(w)\}$$

Soln: The PDA to accept the language

$$L = \{w \mid n_a(w) < n_b(w)\}$$

is given by

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

where

$Q = \{q_0, q_1\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, b, Z_0\}$

δ is shown below

$\delta(q_0, a, Z_0) = (q_0, aZ_0)$
$\delta(q_0, b, Z_0) = (q_0, bZ_0)$
$\delta(q_0, a, a) = (q_0, aa)$
$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, \varepsilon)$

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

$q_0 \in Q$ is the start state of m/c
$Z_0 \in \Gamma$ is the initial symbol on the stack
$F = \{q_1\}$ is the final state.

(Eg) obtain a PDA to accept the language

$$L = \{a^n b^{2n} \mid n \geq 1\}.$$

Soln So, the PDA to accept the language

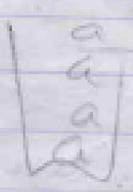
$$L = \{a^n b^{2n} \mid n \geq 1\}$$

is given by

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

where

$Q = \{q_0, q_1, q_2\}$
$\Sigma = \{a, b\}$
$\Gamma = \{a, z_0\}$



δ: is shown below

$$\delta(q_0, a, z_0) = (q_0, aaz_0)$$
$$\delta(q_0, a, a) = (q_0, aaa)$$
$$\delta(q_0, b, a) = (q_1, \varepsilon)$$
$$\delta(q_1, b, a) = (q_1, \varepsilon)$$
$$\delta(q_1, \varepsilon, z_0) = (q_2, \varepsilon)$$

$q_0 \in Q$ is the start state of m/c
$z_0 \in \Gamma$ is the initial symbol on the stack
$F = \{q_2\}$ is the final state.

TO accept the string: The seqⁿ of moves made by the PDA for the string aabbbb is shown be

$(q_0, aabbbb, Z_0) \vdash (q_0, abbbb, aaZ_0)$
$\vdash (q_0, bbbb, aaaaZ_0)$
$\vdash (q_0, bbb, aaaZ_0)$
$\vdash (q_1, bb, aaZ_0)$
$\vdash (q_1, b, aZ_0)$
$\vdash (q_1, \varepsilon, Z_0)$
$\vdash (q_2, \varepsilon, Z_0)$

(Final configuration)

TO reject the string: The seqⁿ of moves made by t PDA for the string aabbb is shown below:

Initial ID

$(q_0, aabbb, Z_0) \vdash (q_0, abbb, aaZ_0)$
$\vdash (q_0, bbb, aaaaZ_0)$
$\vdash (q_0, bb, aaaZ_0)$
$\vdash (q_0, b, aaZ_0)$
$\vdash (q_0, \varepsilon aZ_0)$
(final configuration)

Eg. obtain a PDA to accept the language $L = \{ww^R \mid w \in (a+b)^*\}$ by a final state.

Solⁿ.

$$L = \{ww^R \mid w \in (a+b)^*\}$$

1 $\delta(q_0, a, z_0) = (q_0, az_0)$

2 $\delta(q_0, b, z_0) = (q_0, bz_0)$

3 $\delta(q_0, a, a) = (q_0, aa)$

4 $\delta(q_0, b, a) = (q_0, ba)$

5 $\delta(q_0, a, b) = (q_0, ab)$

6 $\delta(q_0, b, b) = (q_0, bb)$

7 $\delta(q_0, a, a) = (q_1, \epsilon)$

8 $\delta(q_0, b, b) = (q_1, \epsilon)$

9 $\delta(q_1, \epsilon, z_0) = (q_2, z_0)$

Note: that the transⁿ numbered 3 & 7, 6 & 8 can obtained & the transition can be written as

$\ldots{}^R \mid w \in (a,b)^*\}$

$\ldots T, \delta, q_0, z_0, F)$

$\ldots, q_1, q_2\}$

$q_0 \in \ldots, b\}$

$z_0 \in \ldots a, b, z_0\}$

$F = \{\ldots$

δ : is shown below

$$\delta(q_0, \varepsilon, z_0) = (q_1, z_0)$$
$$\delta(q_0, a, z_0) = (q_0, az_0)$$
$$\delta(q_0, b, z_0) = (q_0, bz_0)$$
$$\delta(q_0, a, a) = \{(q_0, aa), (q_1, \varepsilon)\}$$
$$\delta(q_0, b, a) = \{q_0, ba)$$
$$\delta(q_0, a, b) = (q_0, ab)$$
$$\delta(q_0, b, b) = \{(q_0, bb), (q_1, \varepsilon)\}$$
$$\delta(q_1, a, a) = (q_1, \varepsilon)$$
$$\delta(q_1, b, b) = (q_1, \varepsilon)$$
$$\delta(q_1, \varepsilon, z_0) = (q_2, z_0)$$

TO accept the string : the seq^n of moves made by the PDA for the string aabbaa is shown below:

Initial ID.

$(q_0, aabbaa, z_0) \rightarrow (q_1, aabbaa, z_0) \rightarrow (q_2, aabaa,$

$\downarrow$

$(q_0, abbaa, az_0)$

$\downarrow$

$(q_0, bbaa, aaz_0)$

$\downarrow$

$(q_0, baa, baaz_0) \rightarrow (q_0, aa, bbaaz_0) \rightarrow (q_0, a, abbaaz$

$\rightarrow (q_0, \varepsilon, a$

$(q_0, aa, bbaaz_0) \downarrow (q_1, \varepsilon, bbaaz_0)$ Reject

$\downarrow$

(q_1, aa, aaz_0)

$\downarrow$

(q_1, a, az_0)

$\downarrow$

$(q_1, \varepsilon, z_0) \rightarrow (q_2, \varepsilon, z_0)$ (accept)

Deterministic & Non-deterministic PDA

Def^n: Let $M = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$ be a PDA. The PDA is deterministic if.

1) $\delta(q, a, z)$ has only one element.
2) If $\delta(q, \epsilon, z)$ is not empty then $\delta(q, a, z)$ should be empty.

Both the conditions should be satisfied for the PDA to be deterministic. If one of the cond^ns fails, the PDA is non-deterministic.

Eg Is the PDA to accept the language $L(M) = \{wcw^R \mid w \in (a+b)^*\}$ is deterministic?

Sol^n ~~$\delta(q_0, a, z_0)$~~

All the transitions are solved in eg ① so by referring those transitions the PDA satisfies of both the cond^ns

i.e ① $\delta(q, a, z)$ has only one element: so that in the transitions, for each $q \in Q$, $a \in \Sigma$ & $Z \in \Gamma$, there is only one component defined & the first cond^n is satisfied.

② The 2^nd cond^n states that if $\delta(q, \epsilon, z)$ is not empty, then $\delta(q, a, z)$ should be empty, i.e, there is an ϵ-transition (in this case it is $\delta(q_1,$ then there should not be any trans^n from the state q_1, when the top of the stack is z_0, w true.

Since, the PDA satisfies both the cond^ns, the PD deterministic.

(Eg) Is the PDA corresponding to the lang $L=\{a^nb^n \mid n\geq 1\}$ by a final state is deterministic

Sol^n

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

Transitions are

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

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

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

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

$\delta(q_1, \epsilon, z_0) = (q_2, \epsilon)$.

(1) The first cond^n satisfied $\delta(q, a, z)$ have only one component

(2) To satisfy 2nd cond^n, consider the transition

$\delta(q_1, \epsilon, z_0) = (q_2, \epsilon)$

Since the trans^n is defined, the tran^n $\delta(q_1, a, z_0)$ where $a \in \Sigma$ should not be defined which is true. Since both the cond^ns are satisfied, the given PDA is deterministic.

(Eg) Is the PDA to accept the language $L(M)=\{w \mid w\in(a+b)^*$ and $n_a(w)=n_b(w)\}$ is deterministic?

Sol^n

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

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

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

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

$$\delta(q_0, a, b) = (q_0, \varepsilon)$$
$$\delta(q_0, b, a) = (q_0, \varepsilon)$$
$$\delta(q_0, \varepsilon, z_0) = (q_1, z_0).$$

First condⁿ satisfied

To satisfy second cond².

$$\delta(q_0, \varepsilon, z_0) = (q_1, z_0)$$

Since this transition is defined the transⁿ $\delta(q_0, a, z_0)$ where $a \in \Sigma$ should not be defin
But there are two transitions.

$$\delta(q_0, a, z_0) = (q_0, az_0)$$
$$\delta(q_0, b, z_0) = (q_0, bz_0)$$

So the given PDA is non-deterministic PD

Eg Is the PDA to accept the lang consisting of balanced parentheses is deterministic.

Ans yes.

Eg Is the PDA to accept the language $L = \{w \mid w \in (a,b)^* \text{ and } n_a(w) > n_b(w)\}$ is deterministic

Solⁿ
$$\delta(q_0, a, z_0) = (q_0, az_0)$$
$$\delta(q_0, b, z_0) = (q_0, bz_0)$$
$$\delta(q_0, a, a) = (q_0, aa)$$

$\delta(q_0, b, b) = (q_0, bb)$
$\delta(q_0, a, b) = (q_0, \varepsilon)$
$\delta(q_0, b, a) = (q_0, \varepsilon)$
$\delta(q_0, \varepsilon, a) = (q_1, a)$

First condition satisfied i.e $\delta(q, a, z)$ should have only one component for each $q \in Q, a \in \Sigma$ $z \in \Gamma$

to satisfy 2nd condition, consider the trans

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

Since this trans is defn the transition $\delta(q_0, f, \ldots$ where $f \in \Sigma$ should not be defined. But there are two trans.

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

So the given PDA is non-deterministic

Eg Is the PDA to accept the lang $L = \{a^n b^n \mid n \geq 1\}$ is deterministic?

Sol: yes

Eg) Is the PDA to accept the language $L = \{ww^R \mid w \in (a+b)^*\}$ is deterministic?

Sol)

$$\delta(q_0, a, z_0) = (q_0, az_0)$$
$$\delta(q_0, b, z_0) = (q_0, bz_0)$$
$$\delta(q_0, a, a) = (q_0, aa), (q_1, \epsilon)$$
$$\delta(q_0, b, a) = (q_0, ba)$$
$$\delta(q_0, a, b) = (q_0, ab)$$
$$\delta(q_0, b, b) = (q_0, bb), (q_1, \epsilon)$$
$$\delta(q_1, a, a) = (q_1, \epsilon)$$
$$\delta(q_1, b, b) = (q_1, \epsilon)$$
$$\delta(q_1, \epsilon, z_0) = (q_2, z_0)$$

The first cond^n to be deterministic is $\delta(q, a, z)$ should have only one component. But there are two transitions each having two components

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

So the first cond^n fails. Given PDA is non deterministic PDA.

CFG to PDA

The steps to be followed to convert a grammar to its equivalent PDA are shown below

1. Convert the grammar into GNF
2. Let q_0 be the start state & Z_0 is the initial symbol on the stack. The transition for this can be

$$\delta(q_0, \epsilon, Z_0) = (q_1, SZ_0)$$

3. For each prodn of the form

$$A \to a\alpha$$

introduction the transn

$$\delta(q_1, a, A) = (q_1, \alpha)$$

4. Finally in state q_1, without consuming any input, change the state to q_f, which is an accepting state.
The transn for this can be of the form

$$\delta(q_1, \epsilon, Z_0) = (q_f, Z_0).$$

Eg For the below grammar

$$S \to aABC$$
$$A \to aB | a$$
$$B \to bA | b$$
$$C \to a$$

Obtain the corresponding PDA.

Soln Let q_0 be the start state & Z_0 the initial symbol on the stack

Step 1: Push the start symbol S on to the stack & change the state to q_1. The transn for this can be of the form

$$\delta(q_0, \varepsilon, Z_0) = (q_1, SZ_0)$$

Step 2: For each prodn $A \to a\alpha$ introduce the transn

$$\delta(q_1, a, A) = (q_1, \alpha)$$

This can be done as shown below

Prodn	Transition
$S \to aABC$	$\delta(q_1, a, S) = (q_1, ABC)$
$A \to aB$	$\delta(q_1, a, A) = (q_1, B)$
$A \to a$	$\delta(q_1, a, A) = (q_1, \varepsilon)$
$B \to bA$	$\delta(q_1, b, B) = (q_1, A)$
$B \to b$	$\delta(q_1, b, B) = (q_1, \varepsilon)$
$C \to a$	$\delta(q_1, a, C) = (q_1, \varepsilon)$

Step 3: Finally in state q_1, with consuming an input change the state to q_f which is accep state

i.e)

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

So the PDA M is given by

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

where

$Q = \{q_0, q_1, q_2\}$

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

$\Gamma = \{S, A, B, C, Z_0)$

δ is shown below.

$\delta(q_0, \varepsilon, Z_0) = (q_1, SZ_0)$

$\delta(q_1, a, S) = (q_1, ABC)$

$\delta(q_1, a, A) = (q_1, B)$

$\delta(q_1, a, A) = (q_1, \varepsilon)$

$\delta(q_1, b, B) = (q_1, A)$

$\delta(q_1, b, B) = (q_1, \varepsilon)$

$\delta(q_1, a, C) = (q_1, \varepsilon)$

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

$q_0 \in Q$ is the SS of m/c

$Z_0 \in \Gamma$ is the initial symbol on the stack.

$F = \{q_f\}$ is the final state.

The der^n from the grammar is shown below

$S \Rightarrow aABC \Rightarrow aaBBC \Rightarrow aabBC \Rightarrow aabbC$

$\Rightarrow aabba.$

The String aabba is derived from the Start Symbol S. The Same String should be accepted by PDA also. The moves made by the PDA are shown below:

Initial ID:

$$
\begin{aligned}
(q_0, aabba, Z_0) &\vdash (q_1, aabba, SZ_0) \\
&\vdash (q_1, abba, ABCZ_0) \\
&\vdash (q_1, bba, BBCZ_0) \\
&\vdash (q_1, ba, BCZ_0) \\
&\vdash (q_1, a, CZ_0) \\
&\vdash (q_1, \varepsilon, Z_0) \\
&\quad (q_f, \varepsilon, Z_0)
\end{aligned}
$$

Since q_f is the final State the String aabba, a by the PDA.

(eg) for the grammar

$$
\begin{aligned}
S &\rightarrow aABB \mid aAA \\
A &\rightarrow aBB \mid a \\
B &\rightarrow bBB \mid A \\
C &\rightarrow a
\end{aligned}
$$

obtain the Corresponding PDA.

Soln: To obtain PDA from CFG the grammar should be in GNF All the prodn except the prodn

$$B \rightarrow A$$

are in GNF.

By Substituting for A in above prodn we get B-prodn also in GNF as shown below

$$B \rightarrow bBB \mid aBB \mid a.$$

So new grammar is GNF can be taken.

$S \to aABB \mid aAA$

$A \to aBB \mid a$

So $B \to bBB \mid aBB \mid a$

$C \to a$

$\delta(q_0, \epsilon, z_0) = (q_1, S z_0)$

prodn	Transition
$S \to aABB$	$\delta(q_1, a, S) = (q_1, ABB)$
$S \to aAA$	$\delta(q_1, a, S) = (q_1, AA)$
$A \to aBB$	$\delta(q_1, a, A) = (q_1, BB)$
$A \to a$	$\delta(q_1, a, A) = (q_1, \epsilon)$
$B \to bBB$	$\delta(q_1, b, B) = (q_1, BB)$
$B \to aBB$	$\delta(q_1, a, B) = (q_1, BB)$
$B \to a$	$\delta(q_1, a, B) = (q_1, \epsilon)$
$C \to a$	$\delta(q_1, a, C) = (q_1, \epsilon)$

$\delta(q_1, \epsilon, z_0) = (q_f, z_0)$.

$\delta(q_0, a, z_0) = (q_0, az_0)$ $\delta(q_0, a, a) = \delta(q_0, aa)$ 9480088606
$\delta(q_0, b, z_0) = (q_0, bz_0)$

a, b — ab
b, a — bb
b, b

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

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

$M = (Q, \Sigma, \Gamma, \delta, q_0, z, F)$

$(q_0, aaabbaaa, z_0)$
$(q_0, aabbaaa, az_0)$ — $(q_1, abbaaa$
$(q_0, abbaaa, aaz_0)$

27, 37, 6

2, 7, 12, 15, 16, 20, 2
22, 31, 35, 39, 49, 53, 54, 62
~~22, 23, 2, 53~~ 64, 65, 124, 45

PDA to CFG

We can convert a given PDA to CFG. The general procedure for conversion is

1) The input symbols of PDA will be the terminals of CFG
2) If the PDA moves from state to q_i to state q_j on consuming the input $a \in Z$ when Z is the top of the stack, then the non-terminals of CFG are the triplets of the form $(q_i Z q_j)$
3) If q_0 is the start state & q_f is the final state then $(q_0 Z q_f)$ is the start symbol of CFG.
4) The production of CFG can be obtained from the transitions of PDA as shown below:

a) For each transition of the form

$$\delta(q_i, a, Z) = (q_j, AB)$$

introduce the productions of the form

$$(q_i Z q_k) \rightarrow a(q_j A q_l)(q_l B q_k)$$

where q_k & q_l will take all possible values from Q

b) For each transition of the form

$$\delta(q_i, a, Z) = (q_j, \varepsilon)$$

introduce the production

$$(q_i Z q_j) \rightarrow a$$

NOT contxt f

$z \Rightarrow uvwxy$

$|vwx| \le$

$vx \ne \varepsilon$

$n = 2$

aa

(Eg) Obtain a CFG that generates the langua[ge] accepted by PDA $M = (\{q_0, q_1\}, \{a, b\}, \{A, Z\}, \delta, \ldots$ with the transitions

$$\delta(q_0, a, Z) = (q_0, AZ)$$
$$\delta(q_0, b, A) = (q_0, AA)$$
$$\delta(q_0, a, A) = (q_1, \varepsilon)$$

Sol: Now, the transition

$$\delta(q_0, a, A) = (q_1, \varepsilon)$$

can be converted into production as shown

For δ of the form $\delta(q_i, a, Z) = (q_j, \varepsilon)$	Resulting produ[ction] $(q_i Z q_j) \to a$
$\delta(q_0, a, A) = (q_1, \varepsilon)$	$(q_0 A q_1) \to a$

Now the transitions

$$\delta(q_0, a, Z) = (q_0, AZ)$$
$$\delta(q_0, b, A) = (q_0, AA)$$

can be converted into prod^ns as shown below

For δ of the form $\delta(q_i, a, Z) = (q_j, AB)$	Resulting prod^ns $(q_i Z q_k) \to a(q_j A q_l)(q\ldots$
$\delta(q_0, a, Z) = (q_0, AZ)$	$(q_0 Z q_0) \to a(q_0 A q_0)(q_0 Z q_0) \mid a(q_0\ldots$ $(q_0 Z q_1) \to a(q_0 A q_0)(q_0 Z q_1) \mid a(q_0\ldots$
$\delta(q_0, b, A) = (q_0, AA)$ ~~$\delta(q_0, b, A) = (q_0, AA)$~~	$(q_0 A q_0) \to b(q_0 A q_0)(q_0 A q_0) \mid b(q_0\ldots$ $(q_0 A q_1) \to b(q_0 A q_0)(q_0 A q_1) \mid b(q_0\ldots$

Start symbol of the grammar will be $q_0 Z q_1$

(Eg) Obtain a CFG for the PDA shown below:

$$\delta(q_0, a, Z) = (q_0, AZ)$$
$$\delta(q_0, a, A) = (q_0, A)$$
$$\delta(q_0, b, A) = (q_1, \varepsilon)$$
$$\delta(q_1, \varepsilon, Z) = (q_2, \varepsilon)$$

Solⁿ To obtain a CFG from the PDA, all the transitions should be of the form

$$\delta(q_i, a, Z) = (q_j, AB)$$
$$\delta(q_i, a, Z) = (q_j, \varepsilon)$$

Except 2nd transⁿ, all transⁿˢ are in the required form, so let we take 2nd transⁿ

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

& convert it into the required form.

It is clear from the transⁿ that when input symbol a is encountered the top of the stack the PDA remains in state q_0 & contents of stack are not altered. This can be interpreted as delete A from the stack & insert A onto the s

So, once A is deleted from the stack we enter into new state q_3. But, in state q_3 without consuming any input we add A onto stack. The corresponding transitions are:

$$\delta(q_0, a, A) = (q_3, \varepsilon)$$
$$\delta(q_3, \varepsilon, Z) = (q_0, AZ)$$

So, the PDA can be written using the f
transitions

$\delta(q_0, a, Z) = (q_0, AZ)$
$\delta(q_0, a, A) = (q_3, \epsilon)$
$\delta(q_3, \epsilon, Z) = (q_0, AZ)$

$\delta(q_0, b, A) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, Z) = (q_2, \epsilon)$

Now, the transitions

$\delta(q_0, a, A) = (q_3, \epsilon)$
$\delta(q_0, b, A) = (q_1, \epsilon)$
$\delta(q_1, \epsilon, Z) = (q_2, \epsilon)$

can be converted into productions as shown

for δ of the form $\delta(q_i, a, Z) = (q_j, \epsilon)$	Resulting prod^{ns} $(q_i Z q_j) \to a$
$\delta(q_0, a, A) = (q_3, \epsilon)$	$(q_0 A q_3) \to a$
$\delta(q_0, b, A) = (q_1, \epsilon)$	$(q_0 A q_1) \to b$
$\delta(q_1, \epsilon, Z) = (q_2, \epsilon)$	$(q_1 Z q_2) \to \epsilon$

Now, the transitions

$\delta(q_0, a, Z) = (q_0, AZ)$
$\delta(q_3, \epsilon, Z) = (q_0, AZ)$

For δ of the form	Resulting prod^n
$\delta(q_i, a, Z) = (q_j, AB)$	$(q_i Z q_k) \rightarrow a(q_j A q_l)(q_l B q_k)$ $k = 0,1,2,3$ $\quad l = 0,1,2,3$
$\delta(q_0, a, Z) = (q_0, AZ)$	$(q_0 Z q_0) \rightarrow a(q_0 A q_0)(q_0 Z q_0) \mid a(q_0 A q_1)(q_1 Z$ $a(q_0 A q_2)(q_2 Z q_0) \mid a(q_0 A q_3)(q_3 Z q$ $(q_0 Z q_1) \rightarrow a(q_0 A q_0)(q_0 Z q_1) \mid a(q_0 A q_1)(q_1 Z q_1)$ $a(q_0 A q_2)(q_2 Z q_1) \mid a(q_0 A q_3)(q_3 Z q_1$ $(q_0 Z q_2) \rightarrow a(q_0 A q_0)(q_0 Z q_2) \mid a(q_0 A q_1)(q_1 Z q_2$ $a(q_0 A q_2)(q_2 Z q_2) \mid a(q_0 A q_3)(q_3 Z q_2$ $(q_0 Z q_3) \rightarrow a(q_0 A q_0)(q_0 Z q_3) \mid a(q_0 A q_1)(q_1 Z q_3)$ $a(q_0 A q_2)(q_2 Z q_3) \mid a(q_0 A q_3)(q_3 Z q_3$
$\delta(q_3, \varepsilon, Z) = (q_0, AZ)$	$(q_3 Z q_0) \rightarrow (q_0 A q_0)(q_0 Z q_0) \mid (q_0 A q_1)(q_1 Z q_0) \mid$ $(q_0 A q_2)(q_2 Z q_0) \mid (q_0 A q_3)(q_3 Z q_0)$ $(q_3 Z q_1) \rightarrow (q_0 A q_0)(q_0 Z q_1) \mid (q_0 A q_1)(q_1 Z q_1)$ $(q_0 A q_2)(q_2 Z q_1) \mid (q_0 A q_3)(q_3 Z q_1$ $(q_3 Z q_2) \rightarrow (q_0 A q_0)(q_0 Z q_2) \mid (q_0 A q_1)(q_1 Z q_2$ $(q_0 A q_2)(q_2 Z q_2) \mid (q_0 A q_3)(q_3 Z q_2)$ $(q_3 Z q_3) \rightarrow (q_0 A q_0)(q_0 Z q_3) \mid (q_0 A q_1)(q_1 Z q_3) \mid$ $(q_0 A q_2)(q_2 Z q_3) \mid (q_0 A q_3)(q_3 Z q_3)$

The start symbol of the grammar will be $q_0 Z q_2$.