$\delta(q_3, c, z_2) \rightarrow (q_4, \lambda)$
$\delta(q_4, c, z_2) \rightarrow (q_4, \lambda)$
$\delta(q_4, \lambda . z_0) \rightarrow (q_f, z_0)$ accepted by the final state
$\delta(q_4, \lambda, z_0) \rightarrow (q_4, \lambda)$ accepted by the empty stack

Example 7.15 Construct a PDA to accept the following language L=$\{a^{2n}b^n$,where $n > 0\}$.

Solution: The language consists of 2n number of 'a' and n number of 'b' belong to the language set. The PDA can be designed as follows—at the time of traversing the fi rst 'a', two z_1 are added to the stack with a state change from q_0 to q_1 . At the time of traversing the second 'a', one z_1 is popped from the stack with a state change from q_1 to q_0 . By this process after traversing 2n number of 'a', only n number of z_1 exist in the stack. At the time of traversing the first 'b', the stack top is z_1 and the state is q_0. Those z_1 are popped at the time of traversing n number of 'b'. The transitional functions are

$\delta(q_0, a, z_0) \rightarrow (q_1, z_1z_1z_0)$
$\delta(q_1, a, z_1) \rightarrow (q_0, \lambda)$
$\delta(q_0, a, z_1) \rightarrow (q_1, z_1z_1z_0)$
$\delta(q_0, b, z_1) \rightarrow (q_2, \lambda)$
$\delta(q_2, b, z_1) \rightarrow (q_2, \lambda)$
$\delta(q_2, \lambda, z_0) \rightarrow (q_f, z_0)$// Accepted by final state
$\delta(q_2, \lambda, z_0) \rightarrow (q_2, \lambda)$// Accepted by empty stack

Example 7.16 Construct a PDA to accept the language L =$\{a^nb^nc^m$,where n, $m \geq 1\}$ by the empty stack and by the fi nal state.

Solution: The language is in the form $a^nb^nc^m$, where n, $m \geq 1$. In the language set, the number of 'a' and the number of 'b' are the same, but the number of 'c' is different. All the strings in the language set start with n number of 'a's followed by n number of 'b's and ends with m number of 'c's. Here, m and n are both ≥ 1, and the null string does not belong to the language set. The PDA for the language can be designed in the following way. When traversing the 'a's from the input tape, the z_1's are pushed in the stack one by one. At the time of traversing the 'b's, all the z_1's which were pushed into the stack are popped one by one. When the first 'c' will be traversed, at that time the machine is in state q_1 and stack top z_0. At the time of traversal of m number of 'c's, no operation is done on the stack. The transition function for constructing the PDA for the string $a^nb^nc^m$ where m, $n \geq 1$ are

$\delta(q_0, a, z_0) \rightarrow (q_0, z_1z_0)$
$\delta(q_0, a, z_1) \rightarrow (q_0, z_1z_1)$
$\delta(q_0, b, z_1) \rightarrow (q_1, \lambda)$
$\delta(q_1, b, z_1) \rightarrow (q_1, \lambda)$
$\delta(q_1, c, z_0) \rightarrow (q_1, z_0)$
$\delta(q_1, \lambda, z_0) \rightarrow (q_1, \lambda)$ accepted by the empty stack
$\delta(q_1, \lambda, z_0) \rightarrow (q_f, z_0)$ accepted by the fi nal state

Example 7.17 Construct a PDA to accept the language L ={ $a^n b^m c^n$, where n, m $\geq$ 1} by the empty stack and by the fi nal state.

solution: The language is in the form anbmcn, where n, m $\geq$ 1. In the language set, the number of 'a's are equal to the number of 'c's. In the language set, 'a's are separated from 'c's by m number of 'b's. As m, n $\geq$ 1, the null string does not belong to the language set. In the language set, the number of 'a' is equal to the number of 'c' but not with the number of 'b'.

The PDA can be designed in the following way.

When traversing 'a's from the input tape, z_1's are pushed in the stack one by one. When the fi rst 'b' is read by the input head, the machine is in state q_0 and stack top z_1. One state change has occurred. [If after 'b' again 'a' comes as input, the language is in a wrong format. But that 'a' will also be traversed as there will be a function $\delta(q_0, a, z_1) \rightarrow (q_0, z_1 z_1)$ to traverse that 'a'. To avoid this mismatch and to guarantee that the PDA is only for accepting anbncm, where n, m $\geq$ 1, the state change is required.] At the time of traversing 'b', no operation is done on the stack.

When 'c' is going to be traversed, the machine is in state q_1, with the stack top z_1. The stack top is popped and a state change has occurred (to avoid appearance of 'b' after 'c').

The transition function for constructing the PDA for the string $a^n b^n c^m$ where m, n $\geq$ 1 will be

$\delta(q_0, a, z_0) \rightarrow (q_0, z_1 z_0)$
$\delta(q_0, a, z_1) \rightarrow (q_0, z_1 z_1)$
$\delta(q_0, b, z_0) \rightarrow (q_1, z_1)$
$\delta(q_1, b, z_1) \rightarrow (q_1, z_1)$
$\delta(q_1, c, z_1) \rightarrow (q_2, \lambda)$
$\delta(q_2, c, z_1) \rightarrow (q_2, \lambda)$
$\delta(q_2, \lambda, z_0) \rightarrow (q_2, \lambda)$ accepted by the empty stack
$\delta(q_2, \lambda, z_0) \rightarrow (q_f, z_0)$ accepted by the fi nal state

Example 7.18 Construct a PDA to accept the language L = w, where number of 'a' + number of 'b' = number of 'c'.

Solution: According to the given specifi cation, any string that belongs to the language set consists of any combination of 'a', 'b', and 'c', and the number of 'c' is equal to the number of 'a' and the number of 'b'. In the string 'a', 'b' and 'c' can occur in any sequence by fulfi lling the given condition.
The transitional functions are

$\delta(q_0, a, z_0) \rightarrow (q_0, z_1 z_0)$ // 'a' traversed fi rst or when the stack contains only z_0
$\delta(q_0, b, z_0) \rightarrow (q_0, z_1 z_0)$ // 'b' traversed fi rst or when the stack contains only z_0
$\delta(q_0, c, z_0) \rightarrow (q_0, z_2 z_0)$ // 'c' traversed fi rst or when the stack contains only z_0

$\delta(q_0, c, z_2) \rightarrow (q_0, z_2z_2)$ // ‘c’ traversed after ‘c’
$\delta(q_0, a, z_1) \rightarrow (q_0, z_1z_1)$ // ‘a’ traversed after ‘a’ or ‘b’
$\delta(q_0, b, z_1) \rightarrow (q_0, z_1z_1)$ // ‘b’ traversed after ‘a’ or ‘b’
$\delta(q_0, c, z_1) \rightarrow (q_0, \lambda)$ // ‘c’ traversed after ‘a’ or ‘b’
$\delta(q_0, a, z_2) \rightarrow (q_0, \lambda)$ // ‘a’ traversed after ‘c’
$\delta(q_0, b, z_2) \rightarrow (q_0, \lambda)$ // ‘b’ traversed after ‘c’
$\delta(q_0, \lambda, z_0) \rightarrow (q_f, z_0)$ // accepted by the fi nal state
$\delta(q_0, \lambda, z_0) \rightarrow (q_0, \lambda)$ // accepted by the empty stack

7.3 DPDA and NPDA
There are two types of PDA:

1. Deterministic pushdown automata (DPDA)
2. Non-deterministic pushdown automata (NPDA)

7.3.1 Deterministic Pushdown Automata (DPDA)
A PDA is said to be a DPDA if all derivations in the design give only a single move.
If a PDA being in a state with a single input and a single stack symbol gives a single move, then the PDA is called DPDA.
As an example, for L = { a^nb^n where n $\geq$ 1 }, a DPDA can be designed if the transitional functions are as follows:

$\delta(q_0, a, z_0) \rightarrow (q_0, z_1z_0)$
$\delta(q_0, a, z_1) \rightarrow (q_0, z_1z_1)$
$\delta(q_0, b, z_1) \rightarrow (q_1, \lambda)$
$\delta(q_1, b, z_1) \rightarrow (q_1, \lambda)$
$\delta(q_1, \lambda, z_0) \rightarrow (q_1, \lambda)$ accepted by the empty stack
$\delta(q_1, \lambda, z_0) \rightarrow (q_f, z_0)$ accepted by the fi nal state

In the previous PDA, in a single state with a single input and a single stack symbol, there is only one move. So, the PDA is deterministic.
A context-free language is called deterministic context-free language if it is accepted by a DPDA.

7.3.2 Non-deterministic Pushdown Automata (NPDA)
A pushdown automata is called non-deterministic if one of the derivations generates more than one move. If a PDA being in a state with a single input and a single stack symbol gives more than one move for any of its transitional functions, then the PDA is called NPDA.
A context-free language is called non-deterministic context-free language if it is accepted by a NPDA.
The following are some examples of NPDA.

Example 7.19 Design a non-deterministic PDA for accepting the string $\{WW^R$ where W $\in(a, b)^+$ and W^R is the reverse of W$\}$ by the empty stack and by the fi nal state.

Solution: W is a string which consists of 'a' and 'b'. W^R is the reverse string of W. Let W = abaa, then W^R will be aaba. WW^R will be abaaaaba. Traversing 'a' in the input tape, z_1 will be pushed into the stack, and traversing 'b' in the input tape z_2 will be pushed into the stack. When W^R will start, for the traversal of the fi rst symbol, the state of the PDA will be changed and the stack top will be popped. From the next input symbols belonging to W^R , the stack top will be popped.

Here, there is a problem. To the PDA, it is not assigned where W is ended and W^R is started. From the beginning state, the PDA will traverse the total string assuming it as W. Sometimes (in some books)attempt has been done to differentiate W and W^R by traversing a λ symbol and changing the state in between W and W^R. But this is also wrong, as all the input symbols are placed on the input tape from left to right in a continuous fashion. There is no gap in between the two input symbols as they are placed one by one in each square from left to right on the input tape.

From the previous discussion, it is clear that a DPDA cannot be designed for WW^R.

But, this is possible for an NPDA. Our problem is to fi nd the middle of the string where W ends and W^R starts. There is a chance to be the middle of the string where two 'a's or two 'b's appear, because the last alphabet of W is the first alphabet of W^R. Make two transitional functions when the stack top is z_1 and 'a' is traversed as input and the stack top is z_2 and 'b' is traversed as input. (All the other δ functions are the same as WCW^R). The PDA for accepting the string $WW^R/W{\in}(a,b)^+$, where W^R is the reverse of W, is

$$\begin{gathered}
\text{Q}=\{q_0, q_1, q_2, q_f\}\\
\Sigma=\{a, b\}\\
\Gamma=\{z_0, z_1, z_2\}\\
q_0=\{q_0\}\\
z_0=\{z_0\}\\
\text{F}=\{q_f\}
\end{gathered}$$

The transitional function will be as follows:

$$\begin{gathered}
\delta(q_0, a, z_0) \to (q_1, z_1 z_0)\\
\delta(q_0, b, z_0) \to (q_1, z_2, z_0)\\
\delta(q_1, a, z_1) \to (q_1, z_1 z_1), (q_2, \lambda)\\
\delta(q_1, b, z_1) \to (q_1, z_2 z_1)\\
\delta(q_1, a, z_2) \to (q_1, z_1 z_2)\\
\delta(q_1, b, z_2) \to (q_1, z_2 z_2), (q_2, \lambda)\\
\delta(q_2, a, z_1) \to (q_2, \lambda)\\
\delta(q_2, b, z_2) \to (q_2, \lambda)\\
\delta(q_2, \lambda, z_0) \to (q_2, \lambda)\\
\delta(q_2, \lambda, z_0) \to (q_f, z_0)
\end{gathered}$$

Consider a string abbbba. The string is in the form WW^R, where W = abb.

$(q_0, abbbba, z_0)$

$$\begin{array}{c}\downarrow \\ (q_1, bbbba, z_1z_0) \\ \downarrow \\ (q_1, bbba, z_2z_1z_0)\end{array}$$
