Pushdown Automata

Pushdown Automata is a finite automata with extra memory called stack which helps Pushdown automata to recognize Context Free Languages.

Basic Structure of PDA

A pushdown automaton is a way to implement a context-free grammar in a similar way we design DFA for a regular grammar. A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information.

Basically a pushdown automaton is −

"Finite state machine" + "a stack"

A pushdown automaton has three components −

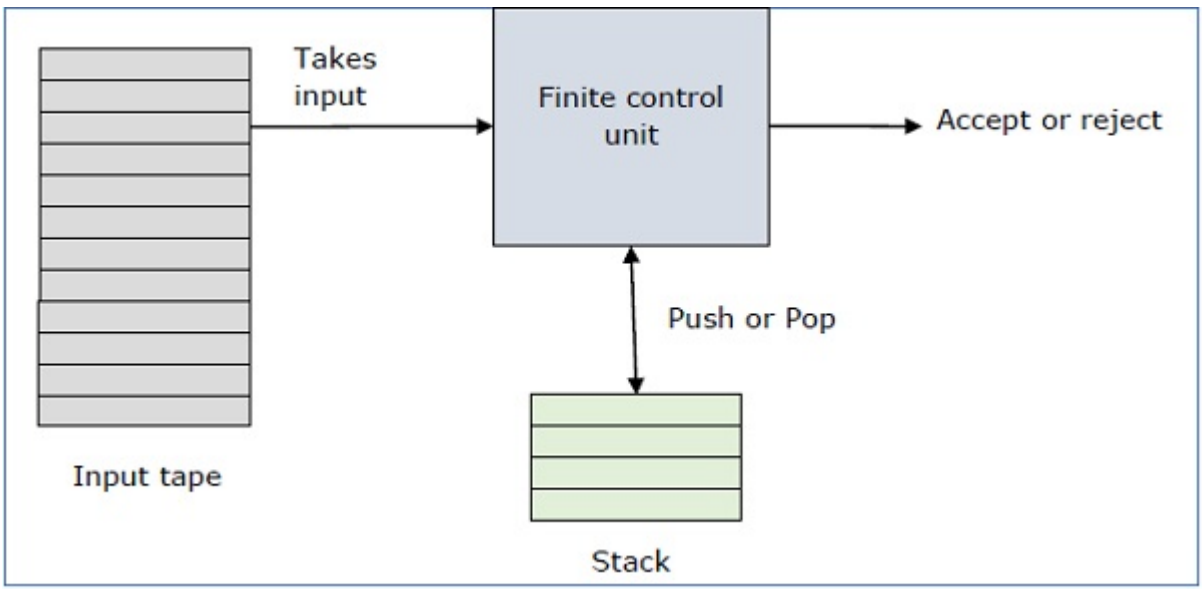
- an input tape,
- a control unit, and
- a stack with infinite size.

The stack head scans the top symbol of the stack.

A stack does two operations −

- **Push** − a new symbol is added at the top.
- **Pop** − the top symbol is read and removed.

A PDA may or may not read an input symbol, but it has to read the top of the stack in every transition.

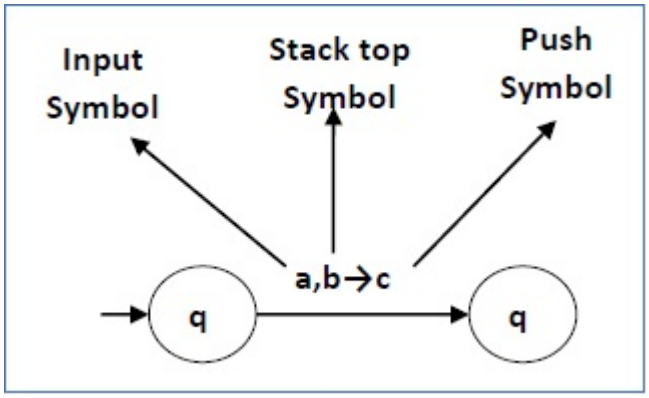


A Pushdown Automata (PDA) can be defined as :

- Q is the set of states
- ∑is the set of input symbols
- Γ is the set of pushdown symbols (which can be pushed and popped from stack)
- q0 is the initial state

- Z is the initial pushdown symbol (which is initially present in stack)
- F is the set of final states
- δ is a transition function which maps Q x {Σ ∪ ∈} x Γ into Q x Γ*. In a given state, PDA will read input symbol and stack symbol (top of the stack) and move to a new state and change the symbol of stack.

The following diagram shows a transition in a PDA from a state q_1 to state q_2, labeled as a,b → c −



This means at state **q_1**, if we encounter an input string '**a'** and top symbol of the stack is '**b'**, then we pop '**b'**, push '**c'** on top of the stack and move to state **q_2**.

Deterministic PDAs

In general terms, a *deterministic* PDA is one in which there is at most one possible transition from any state based on the current input.

Formally, a deterministic PDA is a PDA where:

- for every state P, input symbol 'a' and stack symbol t, there is at most one transition of the form $(P, a, t) \mapsto (Q, u)$ for any state Q and stack symbol u.

Any context-free language that can be converted to a deterministic PDA is called a *deterministic CFL*.

Non-Deterministic PDAs

A nondeterministic pushdown automaton (NPDA) is a 7-tuple
$M = (Q, \Sigma, \Gamma, q_0, \delta, z, F)$, where
Q is a finite set of states
Σ is the input alphabet (a finite set)
Γ is the stack alphabet (a finite set)
$\delta : Q \times (\Sigma \cup \{\lambda\}) \times \Gamma \to$ (finite subsets of $Q \times \Gamma^*$)
is the transition function
$q_0 \in Q$ is the start state
$z \in \Gamma$ is the initial stack symbol
$F \subseteq Q$ is the set of accepting states

Nondeterminism

- Execution: If the machine is in state i and the input letter is b and C is on top of the stack, then pop the stack and enter state j.
- Nondeterminism can occur in two ways, as in the following examples.