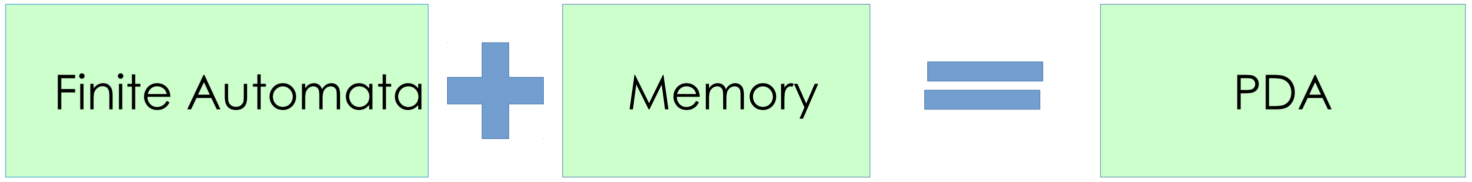
Push Down Automata (PDA)

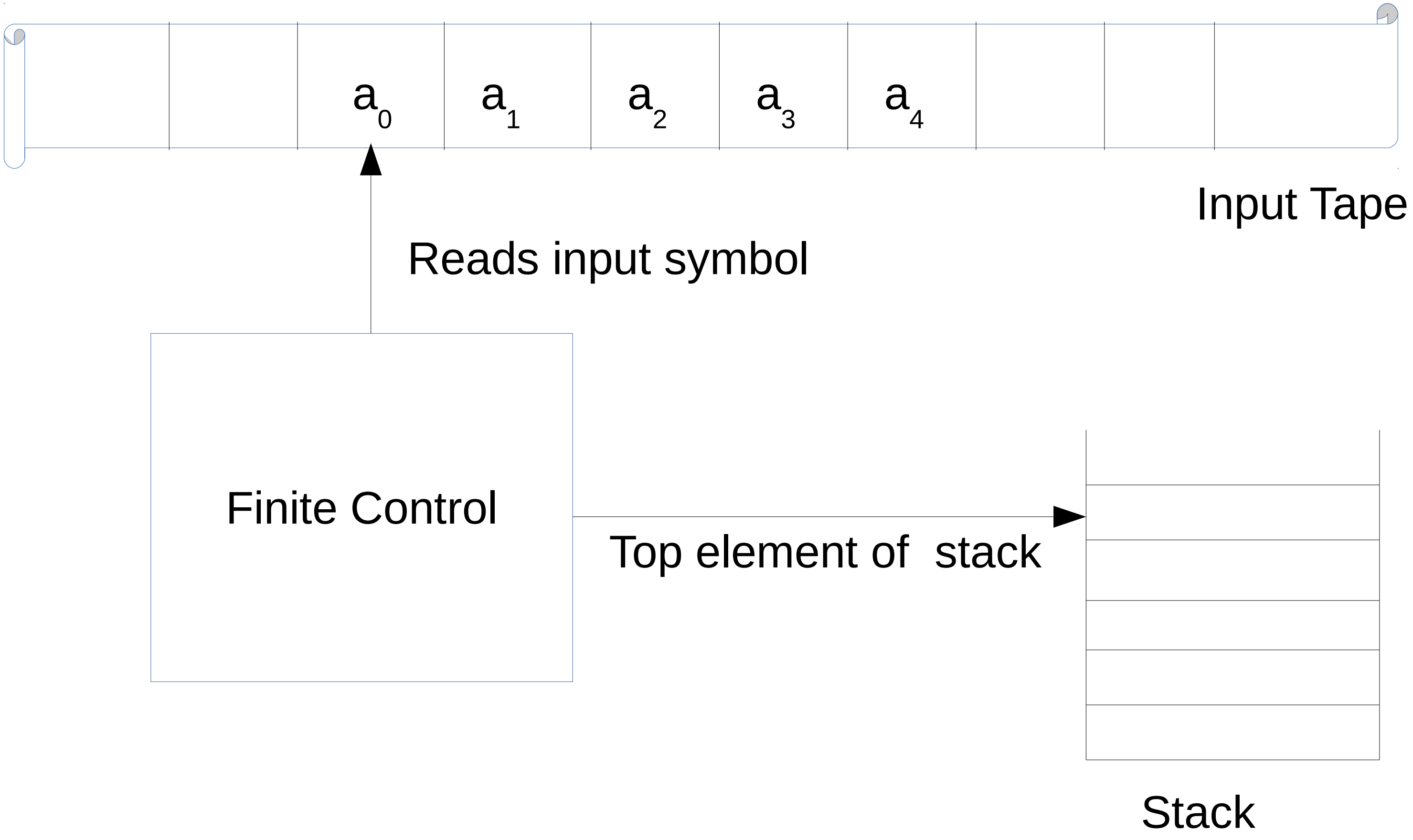
Sagar Giri

- When we add a memory stack to a finite automata, we get a new machine called Push Down Automata.



Finite Automata + Memory = PDA

- Pushdown Automata is no deterministic in nature. It may have choices of moves in each situation.
- PDA contains a input tape, finite control and a stack



- The string of input symbol which the PDA is going to read is written on the input tape
- Two types of move:
  - Read input symbol, read top element of stack change state, manipulates stack and points next element of input tape
  - Empty move: doesn't read input symbol from from tape, read top element from stack, manipulates stack, change state

Operations performed on stack

1. Top element is removed i.e. pop operation
2. A string of stack symbols is pushed in stack
3. Top element is replaced by string of stack symbols

Definition

A PDA is formally defined as a 7 tuple

P= (Q,∑,т, δ,q_0 ,Z_0 ,F)

Q is a finite set of states

∑ is a set of input symbols

т is a set of stack symbols

δ is a transition function

q_0 is a start state

Z_0 is a Initial stack top symbol

F is a Final/accepting states

- PDA are more capable than finite state machines but less capable than Turing machines.
- Deterministic pushdown automata can recognize all deterministic context free languages while nondeterministic ones can recognize all context free languages.

- Pushdown automata differ from finite state machines in two ways:
  - 1. They can use the top of the stack to decide which transition to take.
  - 2. They can manipulate the stack as part of performing a transition.

Finite automata can't recognize the language

$L = \{a^n b^n \mid n>0\}$

but **PDA** can, due to presence of memory

Thank You !