CSC-257
Theory Of Computation
(BSc CSIT, TU)

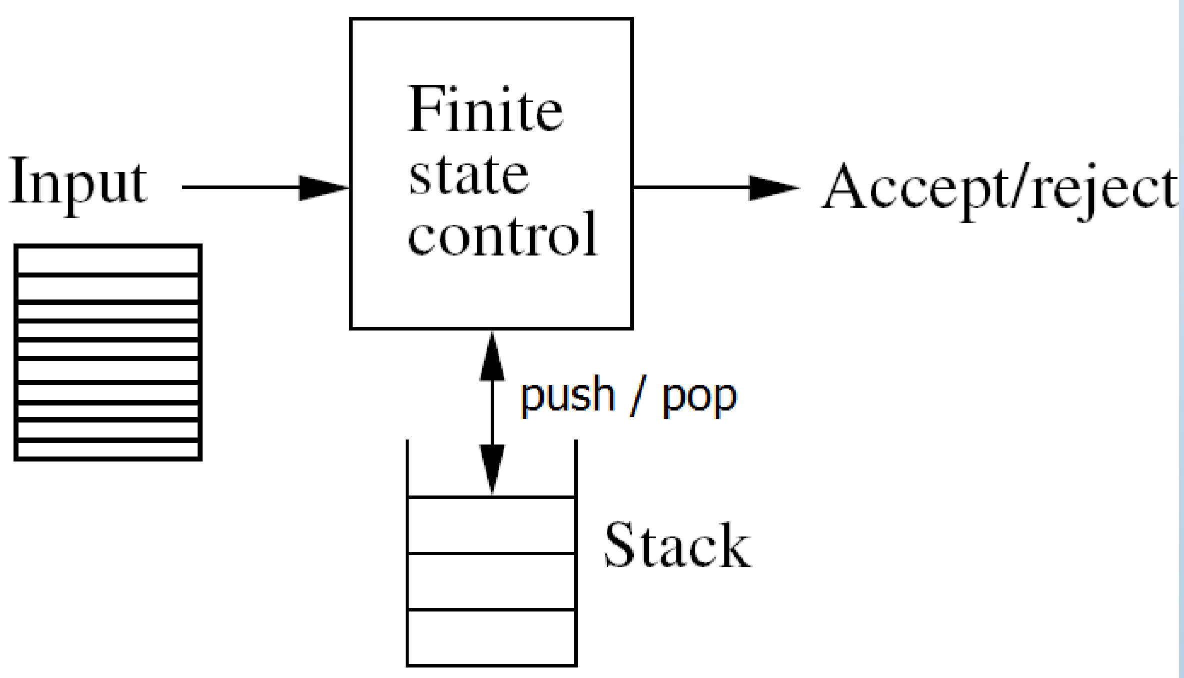
Ganesh Khatri
kh6ganesh@gmail.com

Chomsky Hierarchy of Grammars

Grammar Type	Grammar Accepted	Language Accepted	Automaton
TYPE-0	Unrestricted Grammar	Recursively Enumerable Language	Turing Machine
TYPE-1	Context Sensitive Grammar	Context Sensitive Language	Linear Bounded Automaton
TYPE-2	Context Free Grammar	Context Free Language	Pushdown Automata
TYPE-3	Regular Grammar	Regular Language	Finite State Automaton

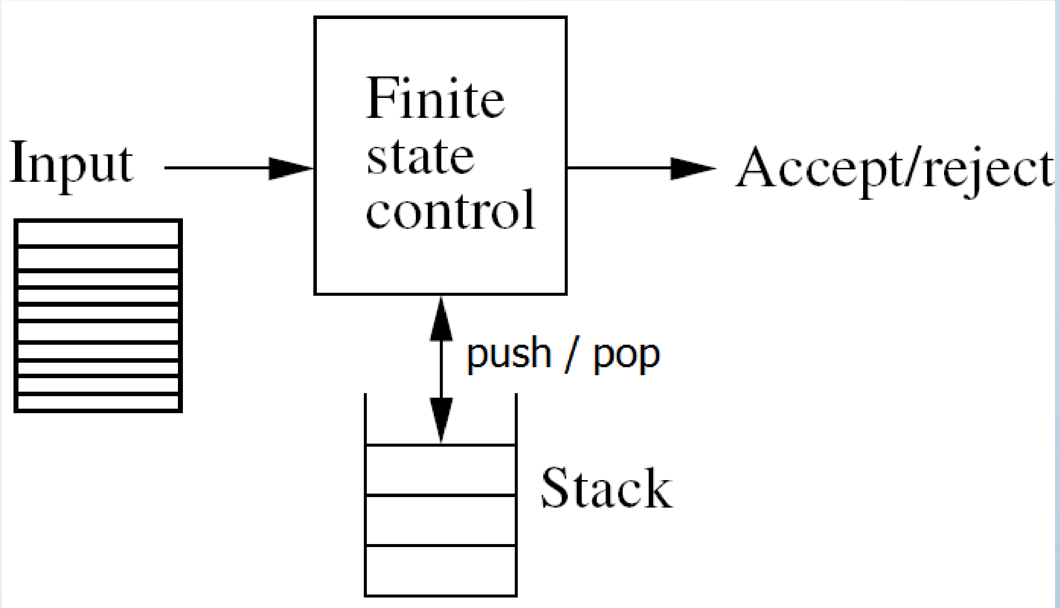
Push Down Automata (PDA)

- The context free languages can also be defined or represented by some automaton which is called Push Down automaton.
- PDA can be thought of as an Є-NFA with the addition of stack.
- The presence of a stack means that, the pushdown automata can remember infinity amount of information.
- PDA can only access the information on its stack in a Last-In-First-Out(LIFO) order.
- We can define PDA informally as device shown in figure



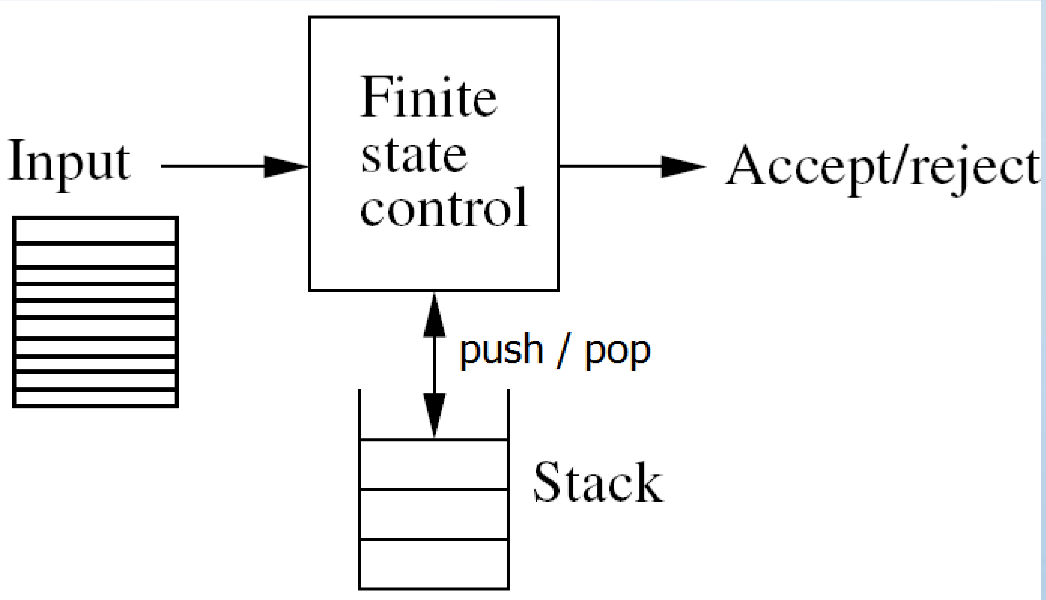
Push Down Automata (PDA)

- PDA is an abstract machine determined by following three things :
  - Input Tape
  - Finite State Control
  - A Stack with infinite size
- Each moves of the machine is determined by three things :
  - The current state
  - Next input symbol
  - Symbol on the top of stack
- The moves consist of :
  - Changing state | staying on same state
  - Replacing the stack top by string of zero or more symbols



Push Down Automata (PDA)

- **Popping** the top symbol of the stack means replacing it by Є
- **Pushing** Y on the stack means replacing stack's top, say X, by YX
- The single move of machine contains only one stack operation either push or pop.
- A DFA can remember a finite amount of information, but a PDA can remember an infinite amount of information



Push Down Automata (PDA) : Forma Definition

- A PDA is defined by seven tuples (Q, Σ, Γ, δ, q_0, z_0, F) where,

  Q - Finite set of states

  Σ - Finite set of input symbols | alphabets

  Γ - Finite set of stack alphabet

  q_0- Start state of PDA; q_0 ε Q

  z_0 - Initial stack symbol; z_0 ε Γ

  F - Set of final states; F ε Q

  δ - Transition function that maps Q × (Σ U {Є}) × Γ → Q × Γ*

Push Down Automata (PDA) : Forma Definition

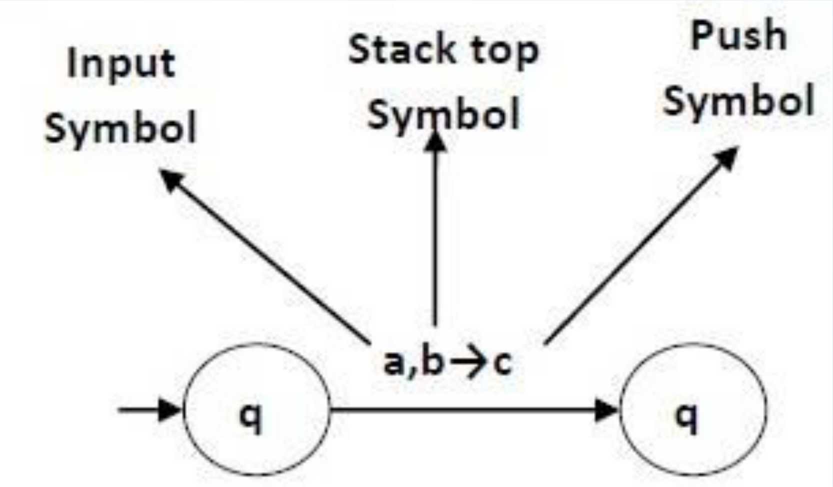
- As for finite automata, δ governs the behavior of PDA. Formally, δ takes as argument a triple δ(q, a, X) where
  - ‘q’ is a state
  - ‘a is either an input symbol in Σ or Є. (note Є does not belong to Σ)
  - X is a stack symbol
- The output of δ is a finite set of pairs (p, γ), where p is the new state and γ is the string on the stack after transition.
- i.e the moves of PDA can be interpreted as :

  $\delta(q, a, X) = \{(p_1, \gamma_1)\ (p_2, \gamma_2)\ \ldots\ldots\ldots\ (p_m, \gamma_m)\}$ here q, p_i ε Q, a ε $\Sigma U Є$ & X ε Γ, γ_i ε Γ^*

Push Down Automata (PDA) : Graphical Notation

- We can use transition diagram to represent a PDA, where :
  - Any state is represented by a node in graph (diagram)
  - Any arc labeled with 'start' indicates the start state and doubly circled states are accepting / final states
  - The arc corresponds to transition of PDA as :

    arc labeled as a, x | ɑ means transition δ(q,a,x) = (p,ɑ) for arc from state p to q



Push Down Automata (PDA) : Graphical Notation

- **Example :** A PDA accepting a string over {a, b} such that number of a’s and b’s are equal. i.e. L = {w | w ε {a, b}* and a’s and b’s are equal}.
- The PDA that accepts the above language can be constructed using the idea that the PDA should push the input symbol if the top of the stack symbol is same as it otherwise Pop the stack.
- For this, let us construct a PDA as :
- $P = \{Q, \Sigma, \Gamma, \delta, q_0, z_0, F\}$ be the PDA recognizing the given language. where, let us suppose

  Q = { q0, q1, q2 }

  Σ = {a, b}

  Γ = {a, b, z0}

  z0 = z0

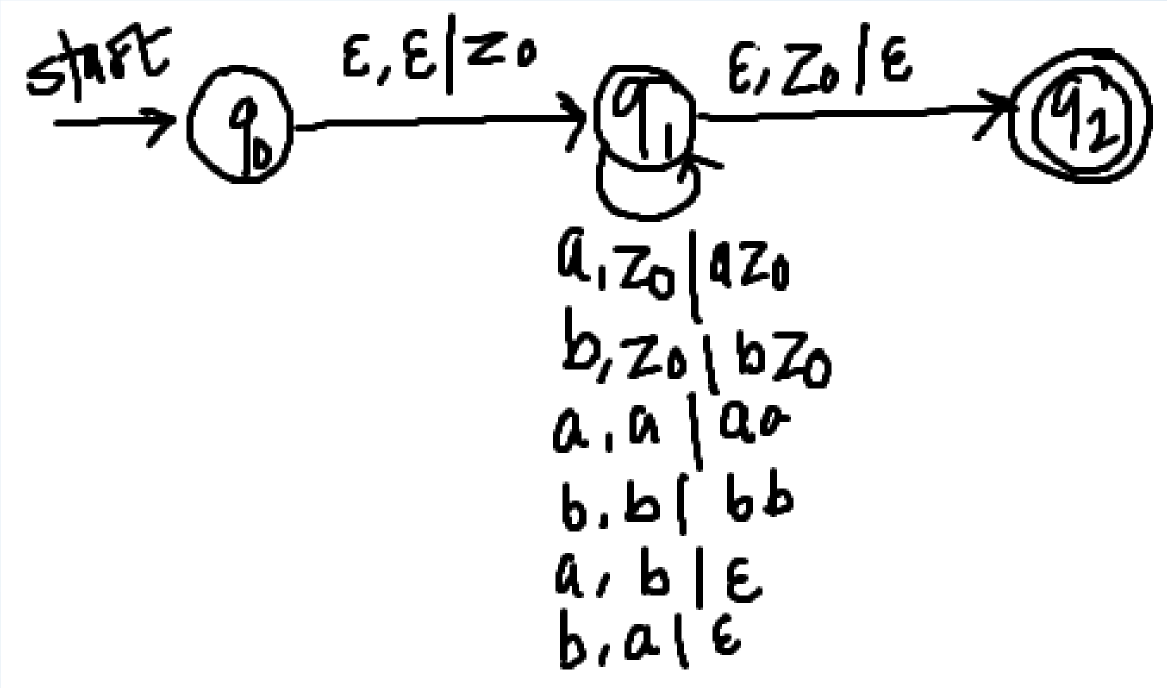
  q0 = q0

  F = { q2 }

Push Down Automata (PDA) : Graphical Notation

- Now δ is defined as :
  - δ(q0, Є, Є) = (q1, z0) //initialize stack with Є to indicate the bottom of stack
  - δ(q1, a, z0) = (q1, az0)
  - δ(q1, b, z0) = (q1, bz0)
  - δ(q1, a, a) = (q1, aa)
  - δ(q1, b, b) = (q1, bb)
  - δ(q1, a, b) = (q1, Є)
  - δ(q1, b, a) = (q1, Є)
  - δ(q1, Є, z0) = (q2, Є)

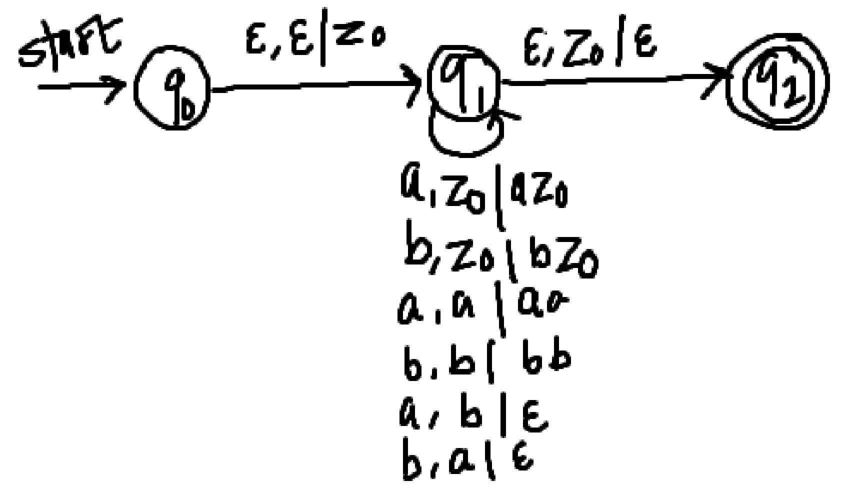
  // last line indicates the acceptance of string



Push Down Automata (PDA) : Graphical Notation

- Let us trace for w= aabbbaab

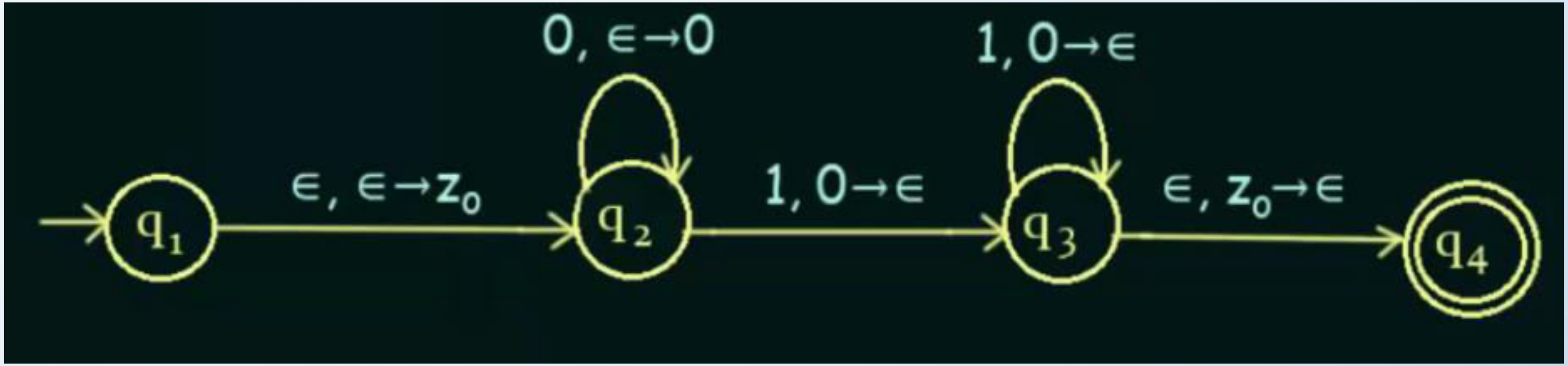
S.N	State	unread string	Stack	Transition Used
1	q_0	aabbbaab	ε	--
2	q_1	aabbbaab	z_0	1
3	q_1	abbbaab	az_0	2
4	q_1	bbbaab	aaz_0	4
5	q_1	bbaab	az_0	7
6	q_1	baab	z_0	7
7	q_1	aab	bz_0	3
8	q_1	ab	z_0	6
9	q_1	b	az_0	2
10	q_1	ε	z_0	7
11	q_2	ε	ε	8



Finally we have reached state q2. Hence accepted

Push Down Automata (PDA) : Graphical Notation

- **Exercise 1 :** Construct a PDA that accepts L = { 0^n1^n | n >=0 }
  - https://www.youtube.com/watch?v=eY7fwj5jvC4



Push Down Automata (PDA) : Graphical Notation

- **Exercise 2 :** Construct a PDA that accepts even palindromes of the form $L = \{ ww^R \mid w = (a+b)^+ \}$ [accepting strings : abba, aba, baab etc]
  - https://www.youtube.com/watch?v=TEQcJybMMFU