MODULE 4

PUSH DOWN AUTOMATA (PDA)

EDULINE FOR CSE STUDENTS

Prepared By Mr. EBIN PM, AP, IESCE 1

PUSH DOWN AUTOMATA (PDA)

- Pushdown automata is a way to implement a CFG in the same way we design Finite Automata (FA) for a regular grammar.
- A FA can remember a finite amount of information, but a PDA can remember an infinite amount of information.

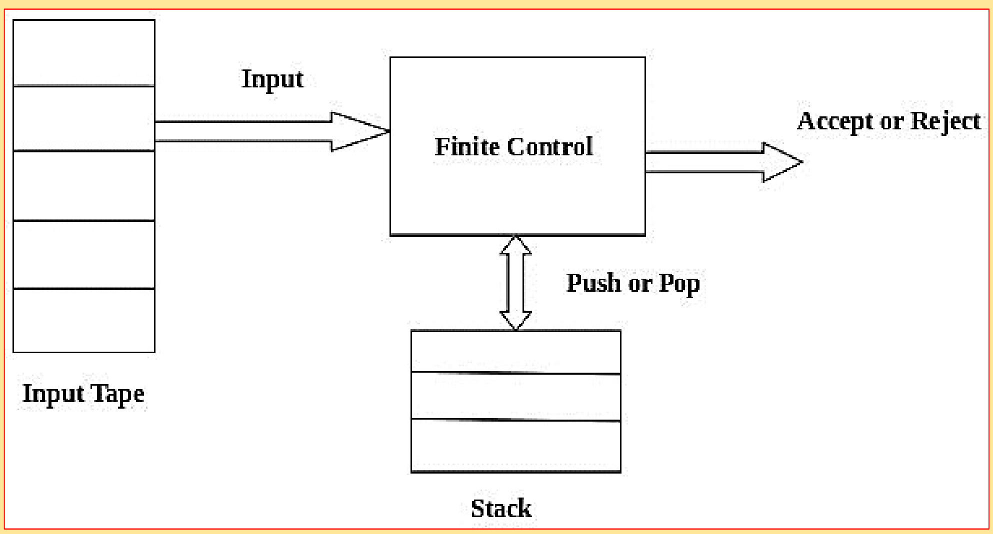
Pushdown automata = "Finite State Machine + stack memory"

- The addition of stack is used to provide a last-in-first-out memory management capability to Pushdown automata.
- Pushdown automata can store an unbounded amount of information on the stack.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 2

- PDA is more powerful than FSM (Finite State Machine)
- A PDA can push an element onto the top of the stack and pop off an element from the top of the stack.
- To read an element into the stack, the top elements must be popped off and are lost.
- A PDA is more powerful than FA.
- Any language which can be acceptable by FA can also be acceptable by PDA.
- PDA also accepts a class of language which even cannot be accepted by FA. Thus PDA is much more superior to FA.

Fig: Push Down Automata (PDA)



❖PDA Components

- **Input tape:** The input tape is divided in many cells or symbols. The input head is read-only and may only move from left to right, one symbol at a time.
- **Finite control:** The finite control has some pointer which points the current symbol which is to be read.
- **Stack:** The stack is a structure in which we can push and remove the items from one end only.

• It has an infinite size.

• In PDA, the stack is used to store the items temporarily.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 5

❖ Formal definition of PDA

The PDA can be defined as a collection of 7 components:

Q : the finite set of states

∑ : the input set

Γ : a stack symbol which can be pushed and popped from the stack

q_0 : the initial state

Z_0 : a start symbol which is in Γ.

F : a set of final states

δ : mapping function which is used for moving from current state to next state.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 6

Transition function δ takes as argument a triple

$\delta(q,a,x)$ where

- **q** is a state in Q
- **a** is either an input symbol in Σ or a can be ε
- **x** is stack symbol , that is a member of Γ

The output of δ is finite set of pairs (p, γ)

p – new state

γ – string of stack symbols that replaces x at the top of the stack

If $\gamma = \varepsilon$ then the stack is popped

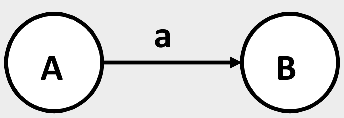
If $\gamma = x$ then the stack is unchanged

If $\gamma = yz$ then x is replaced by z and y is pushed on to the stack

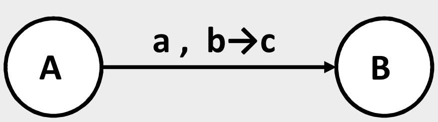
Prepared By Mr.EBIN PM, AP, IESCE EDULINE 7

GRAPHICAL NOTATION OF PDA

Finite Automata (Finite State Machine)



Push Down Automata (PDA)



PDA

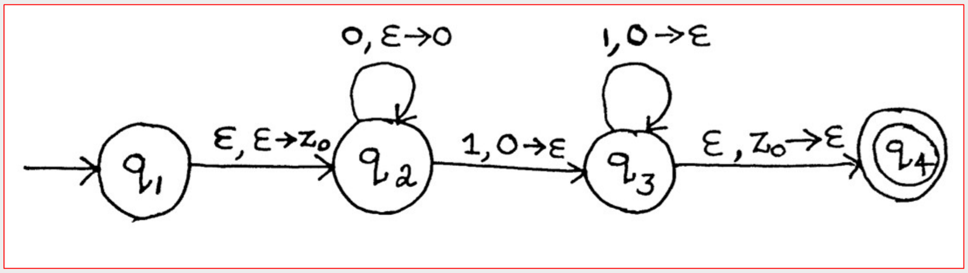
a – Input symbol (This can also be ε)

b – Symbol on top of stack. This symbol is popped (ε means the stack is neither read nor popped)

c – This symbol is pushed on to the stack (ε means nothing is pushed)

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 8

Eg: Construct a PDA that accepts L = $\{0^n 1^n | n \geq 0\}$



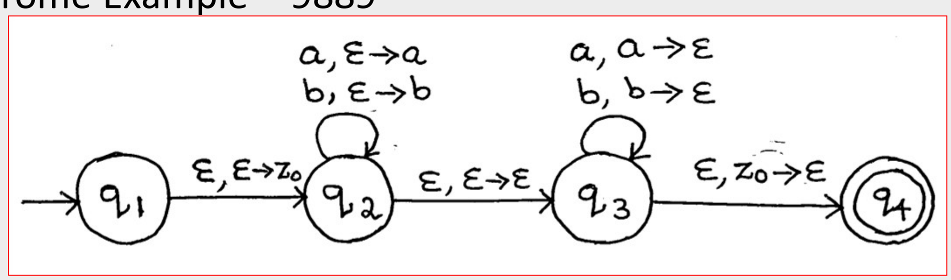
Suppose our input string is 0011

0
0
Z_0

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 9

Eg: Construct a PDA that accepts even palindrome of the form L = $\{ww^R | w = (a+b)^+\}$

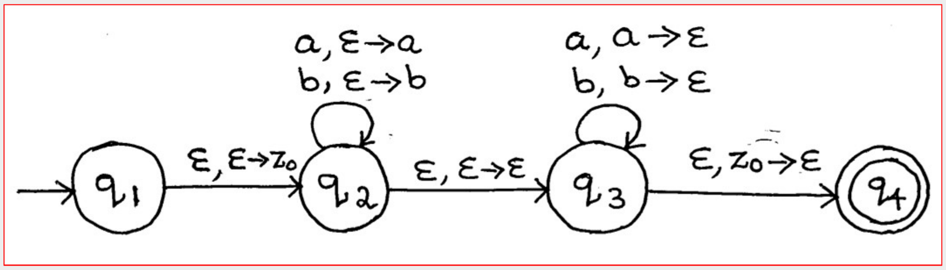
Palindrome Example - 9889



b
a
Z_0

Consider the input strings (1) **abba**
(2) **abab**

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 10

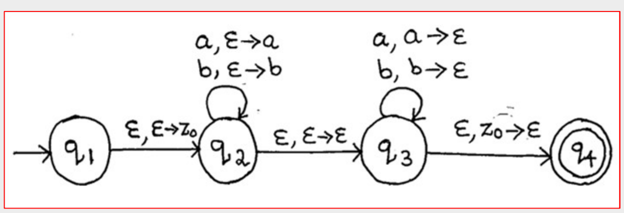


Consider the input strings **abba** .

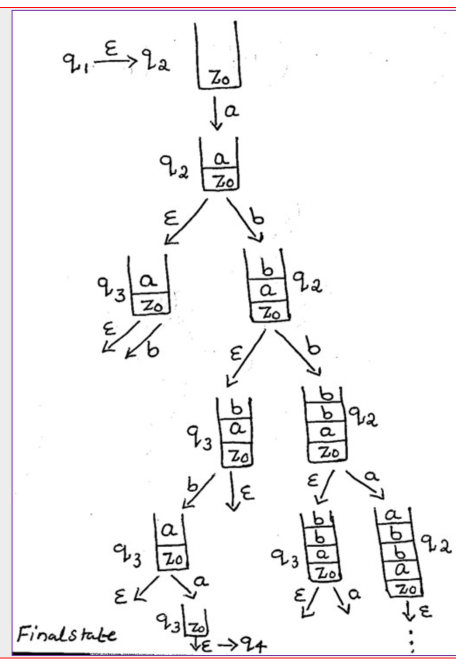
Assume that the ε is present after and before every symbol. Ie.,

ε **a** ε **b** ε **b** ε **a** ε

Prepared By Mr.EBIN PM, AP, IESCE EDULINE



εaεbεbεaε



Prepared By Mr.EBIN PM, AP, IESCE EDULINE

Instantaneous Description (ID) of PDA

- Instantaneous Description (ID) is an informal notation of how a PDA "computes" a input string and make a decision that string is accepted or rejected.

A ID is a triple **(q, w, α),** where:

q is the current state.

w is the remaining input (unconsumed input)

α is the stack contents, top at the left.

Turnstile notation

⊢ sign is called a "turnstile notation" and represents one move.

⊢* sign represents a sequence of moves.

Eg: (p, b, T) ⊢ (q, w, α)

This implies that while taking a transition from state p to state q, the input symbol 'b' is consumed, and the top of the stack 'T' is replaced by a new string 'α'

δ is a transition function which maps **$Q \times \{\Sigma \cup \varepsilon\} \times \Gamma \rightarrow Q \times \Gamma^*$**

- 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.

Example : Define the pushdown automata for language $\{a^n b^n \mid n > 0\}$

Let Q = { q0, q1 }

Σ = { a, b }

Γ = { A, Z_0 } . Consider the input string aaabbb.

Row	State	Input	δ(transition function used)	Stack(Leftmost symbol represents top of stack)	State after move
1	q0	aaabbb		Z	q0
2	q0	aaabbb	δ(q0,a,Z)={(q0,AZ)}	AZ	q0
3	q0	aaabbb	δ(q0,a,A)={(q0,AA)}	AAZ	q0
4	q0	aaabbb	δ(q0,a,A)={(q0,AA)}	AAAZ	q0
5	q0	aaabbb	δ(q0,b,A)={(q1,∈)}	AAZ	q1
6	q1	aaabbb	δ(q1,b,A)= {(q1,∈)}	AZ	q1
7	q1	aaabbb	δ(q1,b,A)= {(q1,∈)}	Z	q1
8	q1	∈	δ(q1, ∈,Z)= {(q1,∈)}	∈	q1

- Initially, the state of automata is q0 and symbol on stack is Z and the input is aaabbb as shown in row 1.
- On reading 'a' (shown in bold in row 2), the state will remain q0 and it will push symbol A on stack.
- On next 'a' (shown in row 3), it will push another symbol A on stack. After reading 3 a's, the stack will be AAAZ with A on the top.
- After reading 'b' (as shown in row 5), it will pop A and move to state q1 and stack will be AAZ.
- When all b's are read, the state will be q1 and stack will be Z. In row 8, on input symbol '∈' and Z on stack, it will pop Z and stack will be empty. This type of acceptance is known as acceptance by empty stack.

❖**Important points**

- The above pushdown automaton is deterministic in nature because there is only one move from a state on an input symbol and stack symbol.
- The non-deterministic pushdown automata can have more than one move from a state on an input symbol and stack symbol.
- It is not always possible to convert non-deterministic pushdown automata to deterministic pushdown automata.
- The push down automata can either be implemented using acceptance by empty stack or acceptance by final state and one can be converted to another.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 17

CLOSURE PROPERTIES OF CFL

1. Union : If L1 and If L2 are two context free languages, their union L1 ∪ L2 will also be context free. For example,

- L1 = { $a^nb^nc^m$ | m >= 0 and n >= 0 } and
- L2 = { $a^nb^mc^m$ | n >= 0 and m >= 0 }
- **L3 = L1 ∪ L2** = {$a^nb^nc^m \cup a^nb^mc^m$ | n >= 0, m >= 0 } is also context free.

- L1 says number of a's should be equal to number of b's and L2 says number of b's should be equal to number of c's. Their union says either of two conditions to be true. So it is also context free language.

Prepared By Mr.EBIN PM, AP, IESCE EDULINE 18

2. Concatenation : If L1 and If L2 are two context free languages, their concatenation L1.L2 will also be context free. For example,

$L1 = \{ a^n b^n \mid n >= 0 \}$ and $L2 = \{ c^m d^m \mid m >= 0 \}$

$L3 = L1.L2 = \{a^n b^n c^m d^m \mid m >= 0 \text{ and } n >= 0\}$ is also context free.

- L1 says number of a's should be equal to number of b's and L2 says number of c's should be equal to number of d's.
- Their concatenation says first number of a's should be equal to number of b's, then number of c's should be equal to number of d's.
- So, we can create a PDA which will first push for a's, pop for b's, push for c's then pop for d's. So it can be accepted by pushdown automata, hence context free.

3. Kleene Closure : If L1 is context free, its Kleene closure **L1*** will also be context free. For example,

$L1 = \{ a^n b^n \mid n >= 0 \}$

$L1^* = \{ a^n b^n \mid n >= 0 \}^*$ is also context free.

- So CFL are closed under Kleen Closure.

4. Intersection and complementation :

- If L1 and If L2 are two context free languages, their intersection

  $L1 \cap L2$ need not be context free.
- Similarly, complementation of context free language L1 which is $\sum^* - L1$, need not be context free.
- So CFL are not closed under Intersection and Complementation.