SNS COLLEGE OF TECHNOLOGY

Coimbatore-36.

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A+’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

COURSE CODE& NAME : *19CSB301 &*AUTOMATA THEORY AND COMPILER DESIGN

III YEAR/ V SEMESTER

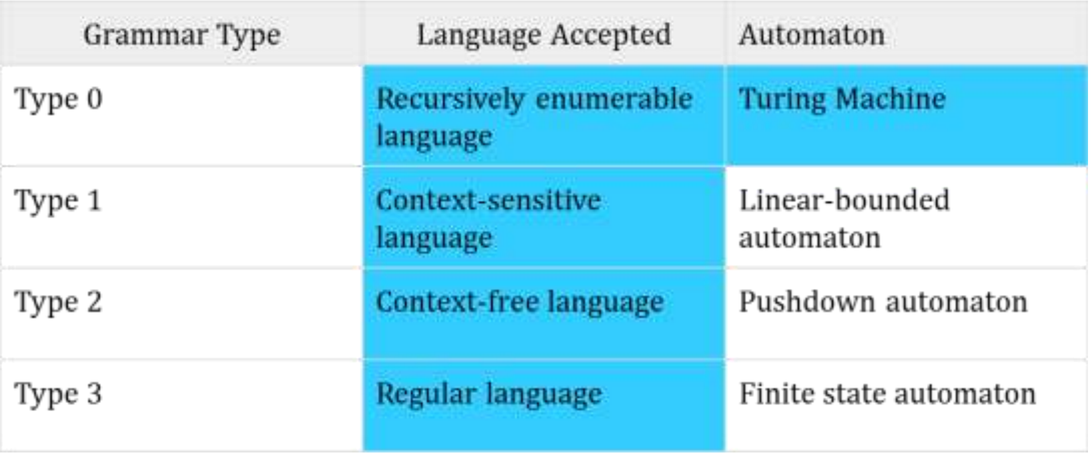
UNIT – I FINITE AUTOMATA AND REGULAR LANGUAGES

Topic: Pushdown Automata

Dr.B.Vinodhini
Assistant Professor
Department of Computer Science and Engineering

PDA – Push Down Automata

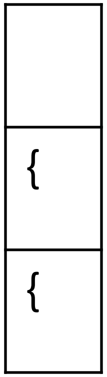
A Push Down Automata(PDA) is a way to implement a context free Grammar in a similar way to design Finite Automata for Regular Grammar



Grammar Type	Language Accepted	Automaton
Type 0	Recursively enumerable language	Turing Machine
Type 1	Context-sensitive language	Linear-bounded automaton
Type 2	Context-free language	Pushdown automaton
Type 3	Regular language	Finite state automaton

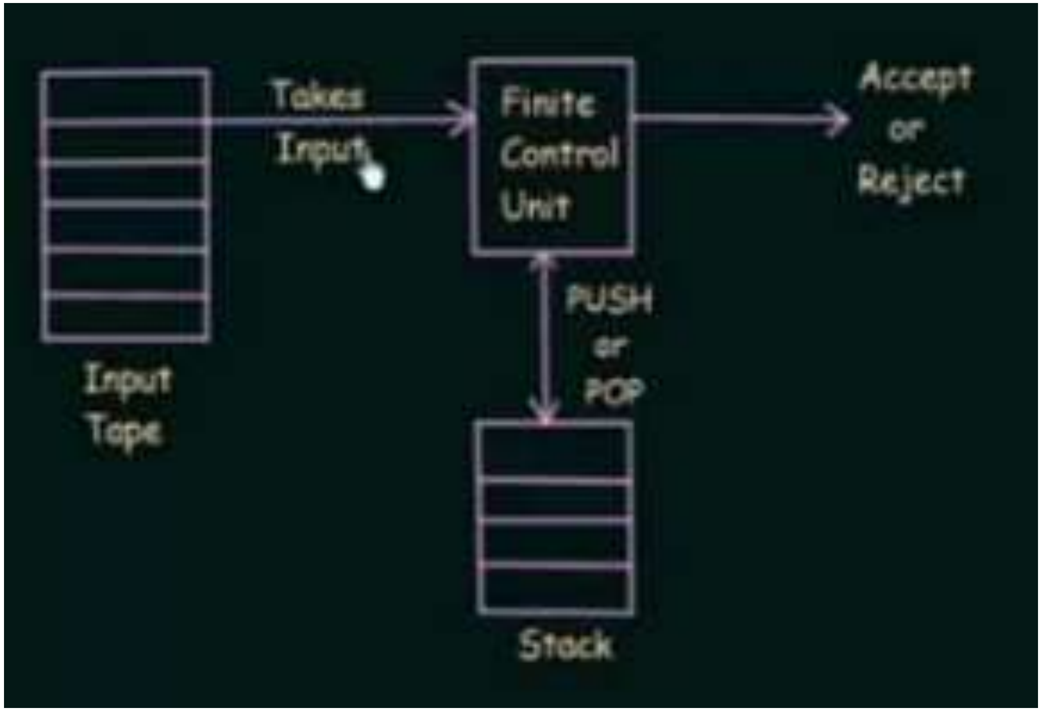
- FSA
  - – not applicable for all domains
  - – Limited Memory
- PDA
  - – It is more Powerful than Finite State Machine
  - – PDA has more memory
  - – **FSA + Stack**
  - – Applications
    - Calculator
    - Java / C Program

Stack

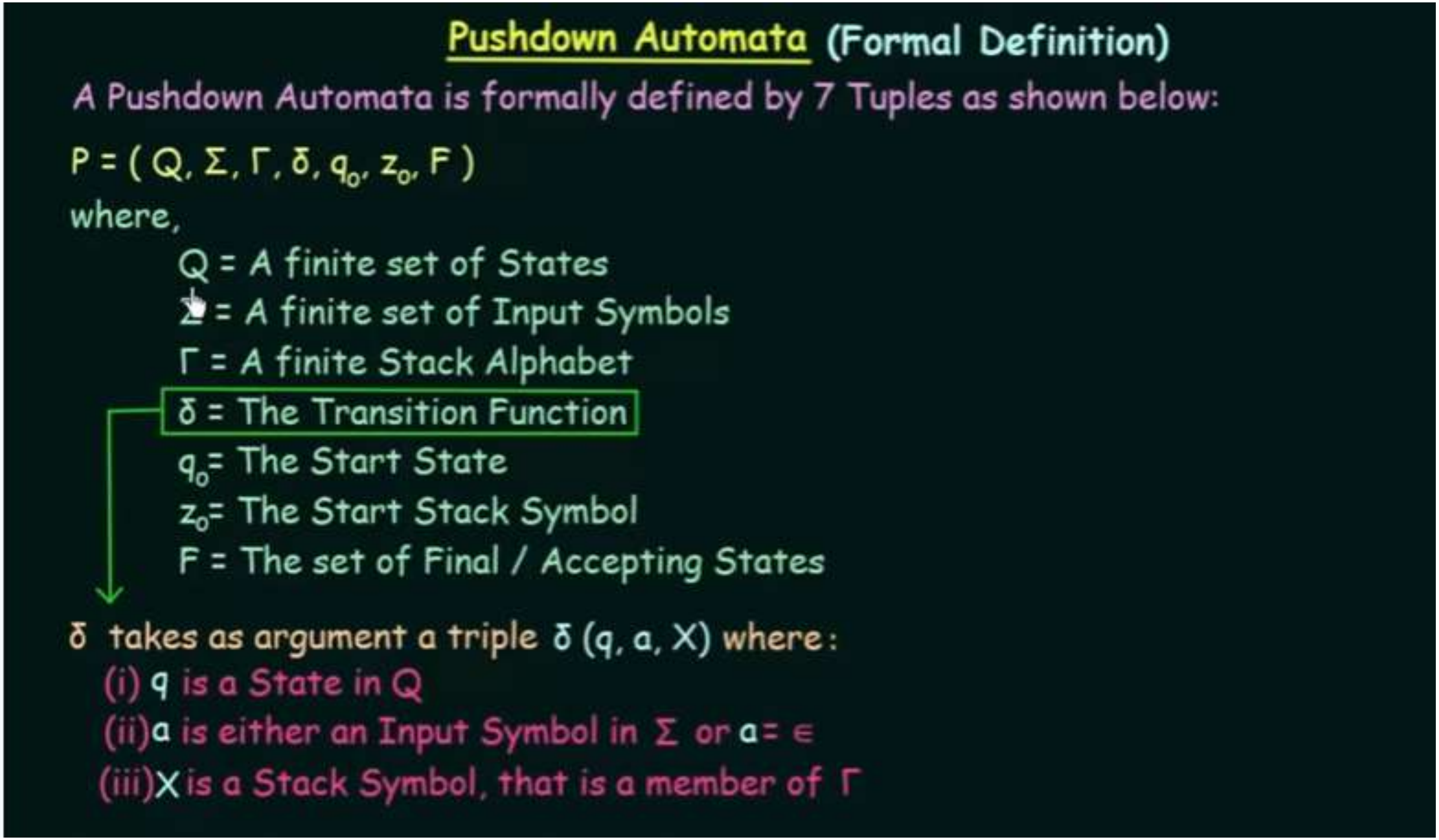


Components of PDA

- Input Tape
- Finite Control Unit
- Stack



Formal Definition of PDA



Pushdown Automata (Formal Definition)

A Pushdown Automata is formally defined by 7 Tuples as shown below:

$P = (Q, \Sigma, \Gamma, \delta, q_0, z_0, F)$

where,

Q = A finite set of States

Σ = A finite set of Input Symbols

Γ = A finite Stack Alphabet

δ = The Transition Function

q_0 = The Start State

z_0 = The Start Stack Symbol

F = The set of Final / Accepting States

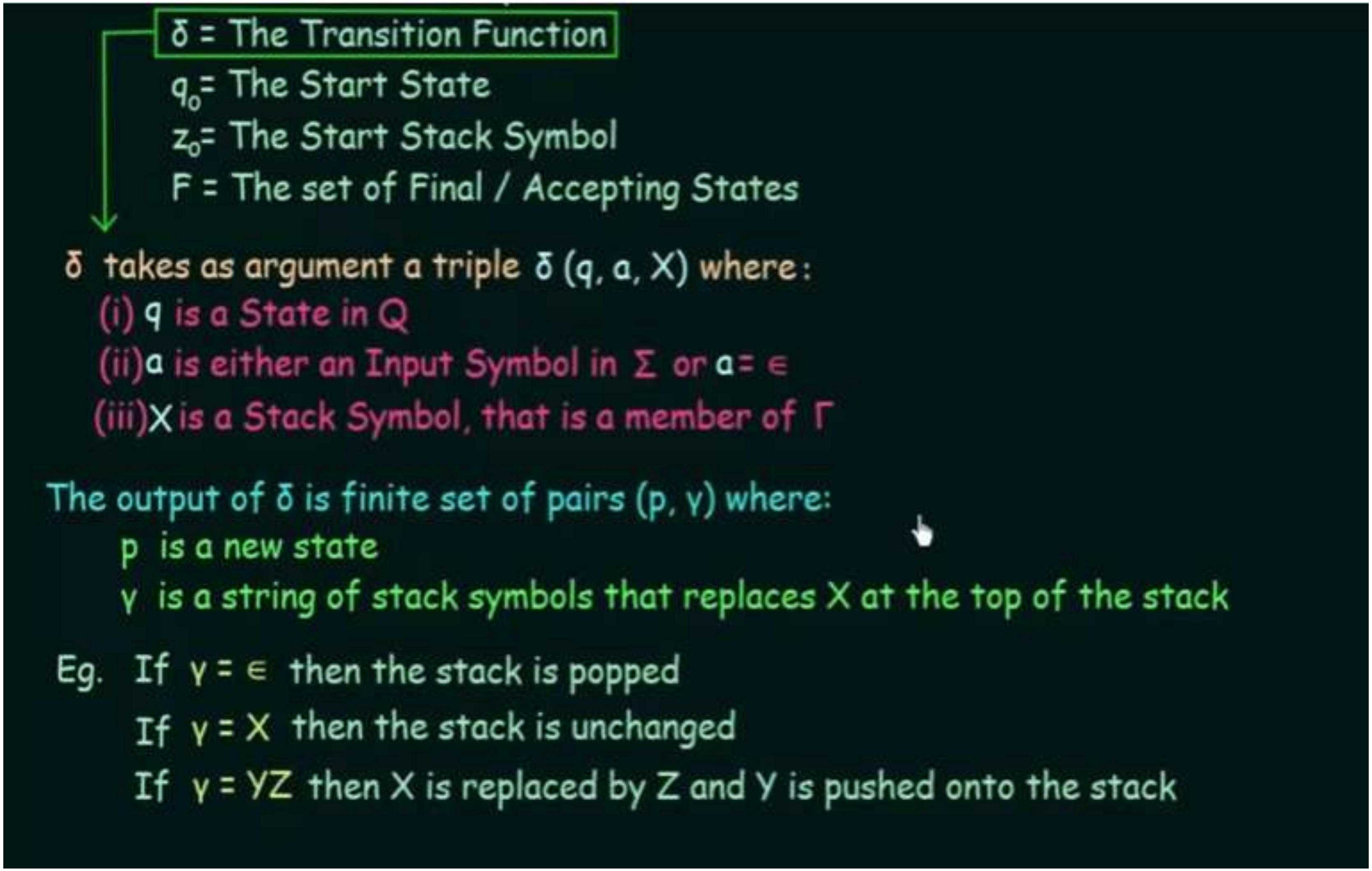
δ takes as argument a triple $\delta(q, a, X)$ where:

(i) q is a State in Q

(ii) a is either an Input Symbol in Σ or $a = \epsilon$

(iii) X is a Stack Symbol, that is a member of Γ

Formal Definition of PDA



δ = The Transition Function

q_0= The Start State

z_0= The Start Stack Symbol

F = The set of Final / Accepting States

δ takes as argument a triple $\delta(q, a, X)$ where:

(i) q is a State in Q

(ii)a is either an Input Symbol in Σ or a= ϵ

(iii)X is a Stack Symbol, that is a member of Γ

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

p is a new state

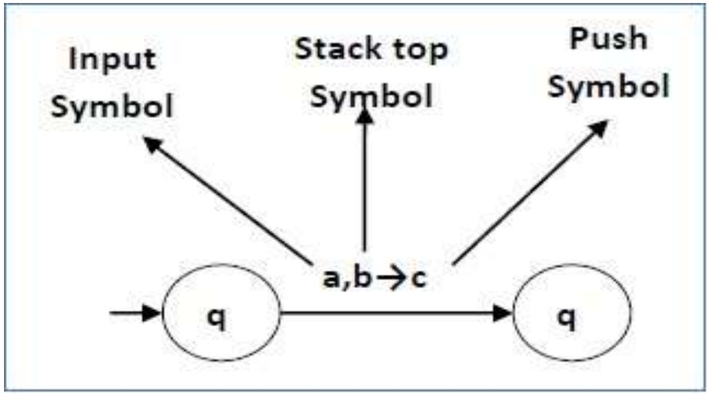
γ is a string of stack symbols that replaces X at the top of the stack

Eg. If $\gamma = \epsilon$ 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 onto the stack

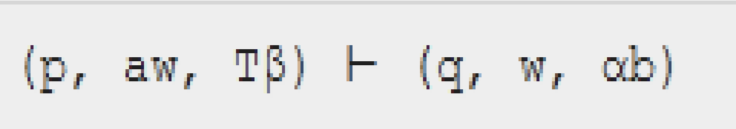
- The following diagram shows a transition in a PDA from a state q_1 to state q_2, labeled as a,b → c −



- This means at state $\mathbf{q_1}$, 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 $\mathbf{q_2}$.

Terminologies Related to PDA

- The "turnstile" notation is used for connecting pairs of ID's that represent one or many moves of a PDA. The process of transition is denoted by the turnstile symbol "⊢".
- Consider a PDA (Q, ∑, S, δ, q_0, I, F). A transition can be mathematically represented by the following turnstile notation –



(p, aw, Tβ) ⊢ (q, w, αb)

- (p, aw, Tβ) ⊢ (q, w, αb) This implies that while taking a transition from state **p** to state **q**, the input symbol **‘a’** is consumed, and the top of the stack **‘T’** is replaced by a new string **‘α’**.
- **Note** – If we want zero or more moves of a PDA, we have to use the symbol (⊢*) for it.

Language of PDA- Final State Acceptability

In final state acceptability, a PDA accepts a string when, after reading the entire string, the PDA is in a final state. From the starting state, we can make moves that end up in a final state with any stack values. The stack values are irrelevant as long as we end up in a final state.

For a PDA (Q, ∑, S, δ, q_0, I, F), the language accepted by the set of final states F is −

$$\mathbf{L(PDA) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, x), q \in F\}}$$

for any input stack string **x**.

Language of PDA- Empty Stack Acceptability

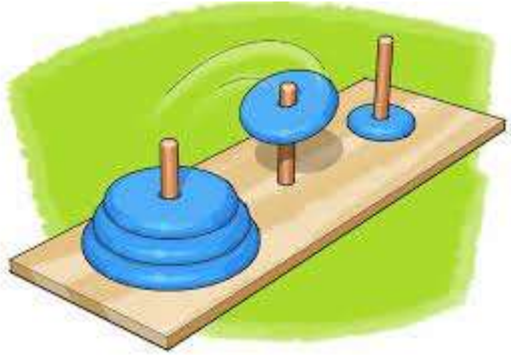
Here a PDA accepts a string when, after reading the entire string, the PDA has emptied its stack.

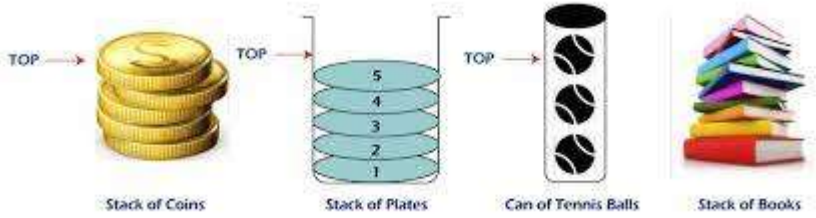
For a PDA (Q, ∑, S, δ, q_0, I, F), the language accepted by the empty stack is −

$$L(PDA) = \{w \mid (q_0, w, I) \vdash^* (q, \varepsilon, \varepsilon), q \in Q\}$$

PDA Applications

- Syntax Analysis phase in Compiler
- Towers of Hanoi
- Smart phone – calculator
- Stack Applications





References

- John E. Hopcroft and Rajeev Motwani and Jeffrey D. Ullman, “Introduction to Automata Theory, Languages and Computation”, Second Edition, Pearson Education, New Delhi, (2007) (UNIT-I)
- Linz P. An introduction to formal languages and automata. Sixth edition, Jones and Bartlett Publishers; 2016.(UNIT-I)
- Ramaiah k. Dasaradh “Introduction to Automata and Compiler Design “ First Edition ,Prentice Hall India Learning Private Limited(2011)(UNIT-I to V)