UNIT IV: Pushdown Automata – Complete Guide

1. Pushdown Automata (PDA): Definition & Model

A **Pushdown Automaton (PDA)** is a 7-tuple $(Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$, where:

- Q: finite set of states
- Σ: finite input alphabet
- Γ: finite stack alphabet
- δ: transition function $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to Q \times \Gamma^*$
- q_0: start state
- Z_0: start stack symbol
- F: set of accepting states

A PDA is a finite automaton with an extra stack, allowing it to recognize context-free languages (CFLs), unlike finite automata which recognize only regular languages[1][2].

Video Resource: Introduction to Pushdown Automata — Sudhakar Atchala
Watch here

GeeksforGeeks Reference:

- Introduction of Pushdown Automata
- Detailed Study of Pushdown Automata

---

2. Graphical Notation & Instantaneous Description

- **Graphical Notation:** PDAs are often represented as state diagrams, with stack operations labeled on transitions.
- **Instantaneous Description (ID):** The triple (q, w, α) describes the current state (q), unread input (w), and stack contents (α; top at left).

Video Resource: Instantaneous Description of PDA — Sudhakar Atchala
Watch here

GeeksforGeeks Reference:

- Instantaneous Description (ID) GFG

---

3. Language Acceptance of PDA

A language is accepted by a PDA in two ways:

- **By Final State:** Input is completely read and the PDA ends in an accepting state.
- **By Empty Stack:** Input is completely read and the stack is empty.

Both methods are equivalent in expressive power for context-free languages[3].

GeeksforGeeks Reference:

- Acceptance by Final State
- NPDA example

---

4. Design of Pushdown Automata

- **Basic Approach:** Use stack for counting (e.g., matching $a^n b^n$)
- **PDA Examples:**
  - $L = \{a^n b^n\}$: Push 'a's to the stack, pop for each 'b'.

Video Resources:

- Construct PDA for $L = \{a^n b^n\}$ (Sudhakar Atchala)
- Design PDA for $L = \{a^{2n} b^n\}$
- More PDA Designs Playlist (#PDADesign)

GeeksforGeeks Reference:

- Construct Pushdown Automata for given languages
- Problems on Pushdown Automata

---

5. Deterministic vs. Non-Deterministic PDA

- **DPDA (Deterministic):** Each configuration has at most one possible move.
- **NPDA (Non-Deterministic):** Multiple moves possible; more powerful than DPDA for language acceptance.

Key Point:

- DPDA accepts Deterministic Context-Free Languages (DCFL), a subset of NPDA's languages (CFL)[4].

Video Resource: Deterministic Pushdown Automata — Sudhakar Atchala
Watch here

GeeksforGeeks Reference:

- Difference Between NPDA and DPDA

---

6. Equivalence of PDA and Context Free Grammars; Conversion

Any context-free language can be accepted by some PDA, and vice versa.

- **CFG to PDA:** For any CFG, construct a PDA that pushes nonterminals and replaces them per production.
- **PDA to CFG:** For any PDA, develop corresponding grammar rules to generate same language.

Video Resource: Equivalence of CFG and PDA — Sudhakar Atchala
Watch here

GeeksforGeeks Reference:

- Equivalence in Theory of Computation
- CFG Equivalent to PDA Conversion
- CFG to PDA Video

---

7. Two-Stack Pushdown Automata

- **Model:** PDA with two stacks has more computational power.
- **Power:** Equivalent to Turing Machines (can recognize languages not recognized by single-stack PDA).
- **Example:** Language $L = \{a^n b^n c^n\}$ needs at least two stacks.

Video Resources:

- Two Stack PDA (Sudhakar Atchala)
- 2 Stack PDA for $a^n b^n c^n$
- 2 Stack PDA for $a^n b^n c^n d^n$

---

8. Applications of Pushdown Automata

- **Compiler design:** Syntax parsing, expression evaluation, block structure matching.
- **Programming languages:** Ensuring correct nesting of parentheses/braces.
- **XML/HTML document parsing**
- **Natural language processing:** Parsing structured text in linguistics.

Video Resource: PDA in Compiler Design — Sudhakar Atchala
Watch here

GeeksforGeeks Reference:

- CFLs and PDA Applications

---

9. Practice Questions

1. Design a PDA to recognize $L = \{a^n b^n c^n | n \geq 1\}$ using two stacks.
2. Given a CFG, show systematic conversion to PDA.
3. Draw and explain the instantaneous description (ID) of a PDA for a sample input.
4. Compare DPDA and NPDA with examples.
5. Explain a real-world application of PDA in compiler design.

---

10. References

[1] Sudhakar Atchala. (2018–2025). Theory of Computation & Automata Theory Video Series. YouTube.
[2] GeeksforGeeks. Pushdown Automata and CFLs Articles. https://www.geeksforgeeks.org/theory-of-computation/
[3] TutorialsPoint. Converting PDA and CFG. https://www.tutorialspoint.com/automata_theory/cfg_equivalent_to_pda_conversion.htm
[4] GeeksforGeeks. NPDA and DPDA Difference

Essential Sudhakar Atchala PDA Video Playlist:

- PDA Examples and Design Playlist
- Theory of Computation / FLAT Full Playlist

Last Updated: November 24, 2025

Suitable For: B.Tech/CSE TOC/FLAT exam prep, project, and concept review.