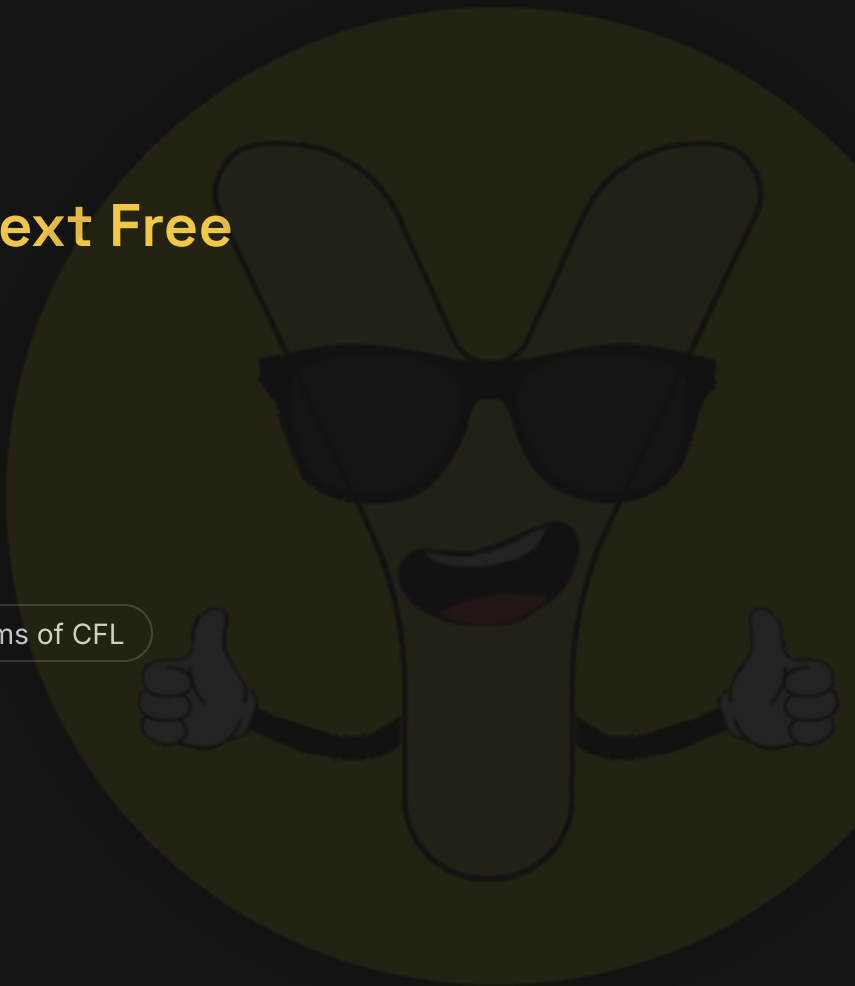
AKTU · 2ND YEAR · SEMESTER 4 · BCS402

Theory of Automata and Formal Languages

Unit 4 — Push Down Automata and Properties of Context Free Languages

- Nondeterministic Pushdown Automata (NPDA)- Definition, Moves, A Language Accepted by NPDA
- Deterministic Pushdown Automata(DPDA) and Deterministic Context free Languages(DCFL)
- Pushdown Automata for Context Free Languages
- Context Free grammars for Pushdown Automata
- Two stack Pushdown Automata
- Pumping Lemma for CFL
- Closure properties of CFL
- Decision Problems of CFL



DECODED BY

Chirag Jain

Founder, Ycotes · Gold Medalist in Artificial Intelligence, IIT Madras · taught 10,000+ students

theycotes.com

Introduction to Pushdown Automata (PDA)

A PDA is a finite automaton equipped with an external stack memory, allowing it to recognize Context-Free Languages (CFLs).

- PDA = Finite Automaton + Stack
- The stack provides infinite memory, but access is restricted to LIFO (Last-In-First-Out).
- It can recognize languages that require counting or matching symbols, such as $L = \{a^n b^n | n \geq 0\}$.
- The stack allows the machine to 'remember' symbols seen earlier.

Nondeterministic PDA (NPDA)

NPDA DEFINITION

An NPDA is a 7-tuple $M = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F$.

- Q: Finite set of states.
- Σ: Input alphabet.
- Γ: Stack alphabet.
- δ: Transition function $Q \times (\Sigma \cup \{\epsilon\}) \times \Gamma \to$ finite subsets of $Q \times \Gamma^*$.
- q_0: Initial state; Z_0: Initial stack symbol; F: Final states.

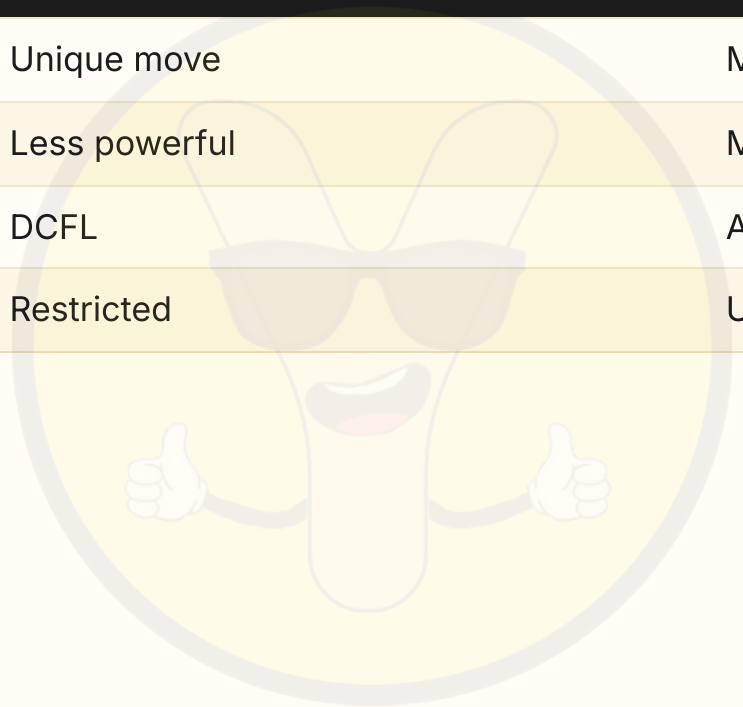
NPDA Moves and Acceptance

The transition function defines how the PDA changes state and modifies the stack.

- Transition: $\delta(q, a, X) = \{(p_1, \gamma_1), (p_2, \gamma_2), ...\}$.
- If $a = \epsilon$, it is an epsilon-move (spontaneous transition).
- Acceptance by Final State: The machine enters a state $q \in F$ after consuming the input.
- Acceptance by Empty Stack: The machine empties its stack after consuming the input.
- Both modes of acceptance are equivalent in power for NPDA.

Deterministic vs Nondeterministic PDA

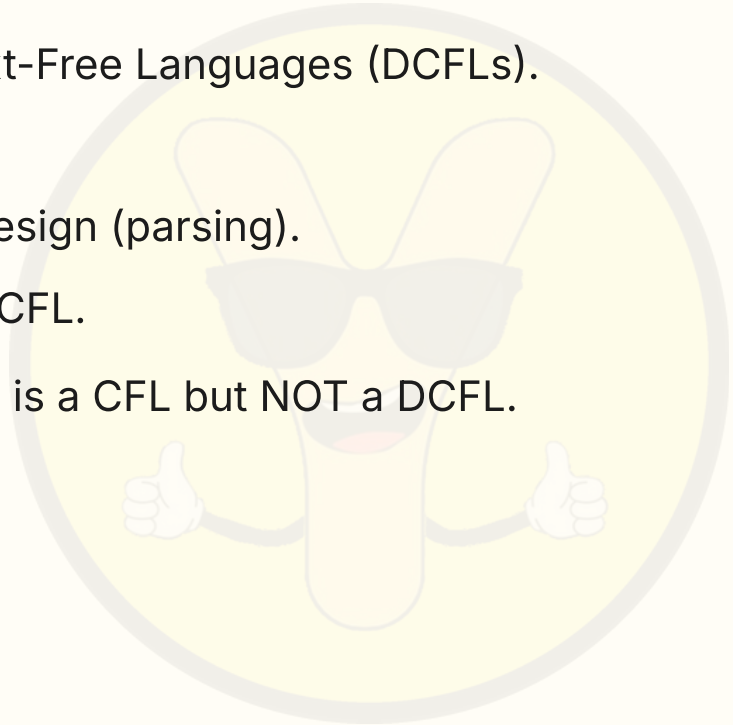
Feature	DPDA	NPDA
Transition	Unique move	Multiple choices
Power	Less powerful	More powerful
CFL Type	DCFL	All CFLs
Epsilon moves	Restricted	Unrestricted



Deterministic PDA (DPDA) and DCFL

A DPDA is a PDA where for every state, input, and stack symbol, there is at most one transition.

- DPDA accepts Deterministic Context-Free Languages (DCFLs).
- DCFLs are a proper subset of CFLs.
- DCFLs are important for compiler design (parsing).
- Example: $L = \{a^n b^n | n \geq 0\}$ is a DCFL.
- Example: $L = \{ww^R | w \in \{a, b\}^*\}$ is a CFL but NOT a DCFL.



PDA Architecture

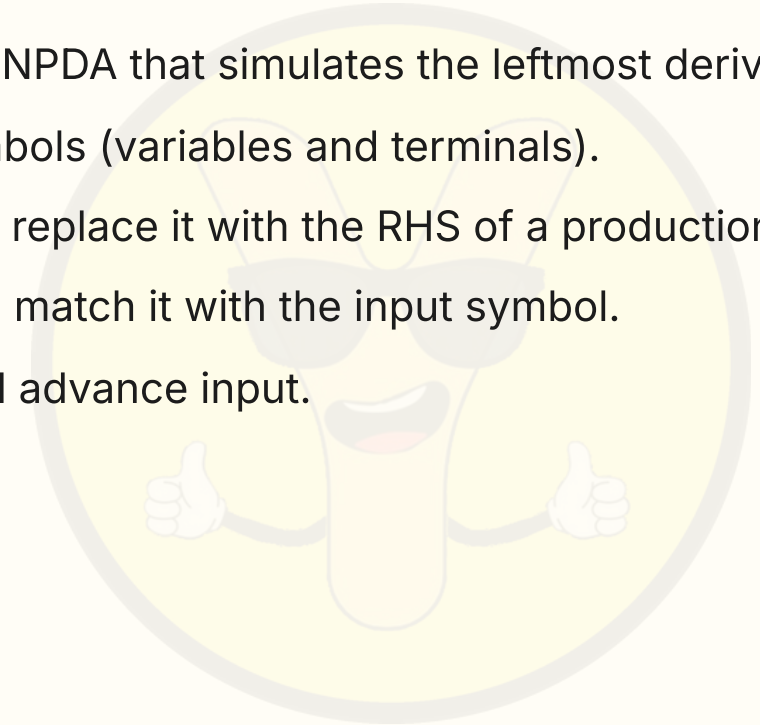


The Finite Control reads the input tape and performs Push/Pop operations on the Stack.

PDA for Context-Free Languages

Every CFL can be recognized by an NPDA.

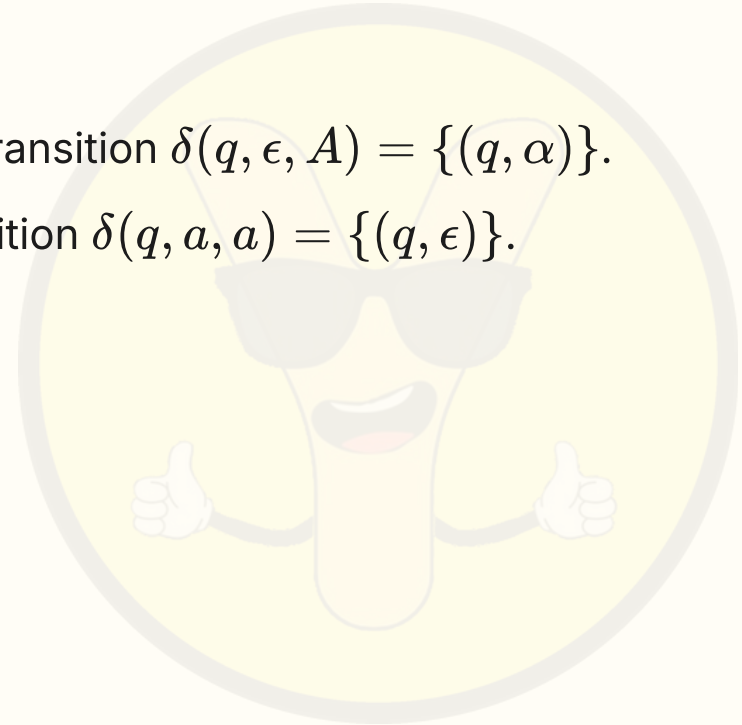
- Given a CFG, we can construct an NPDA that simulates the leftmost derivation.
- The stack stores the grammar symbols (variables and terminals).
- If the top of the stack is a variable, replace it with the RHS of a production.
- If the top of the stack is a terminal, match it with the input symbol.
- If they match, pop the terminal and advance input.



CFG to PDA Conversion Algorithm

Standard method to build an NPDA from a grammar.

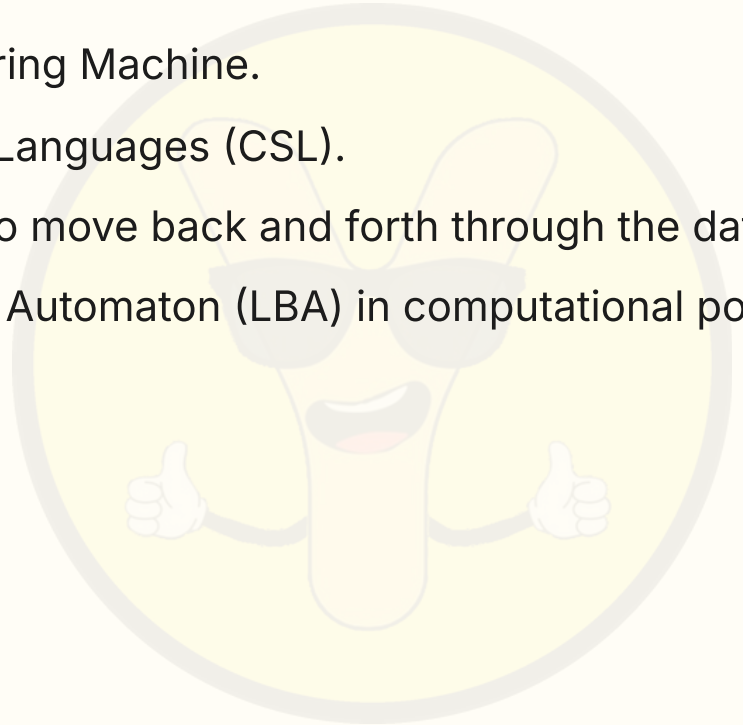
- Create a single state PDA.
- For each production $A \to \alpha$, add transition $\delta(q, \epsilon, A) = \{(q, \alpha)\}$.
- For each terminal $a \in \Sigma$, add transition $\delta(q, a, a) = \{(q, \epsilon)\}$.
- Initialize stack with start symbol S.
- The PDA accepts by empty stack.



Two-Stack Pushdown Automata

A PDA with two stacks is significantly more powerful than a standard PDA.

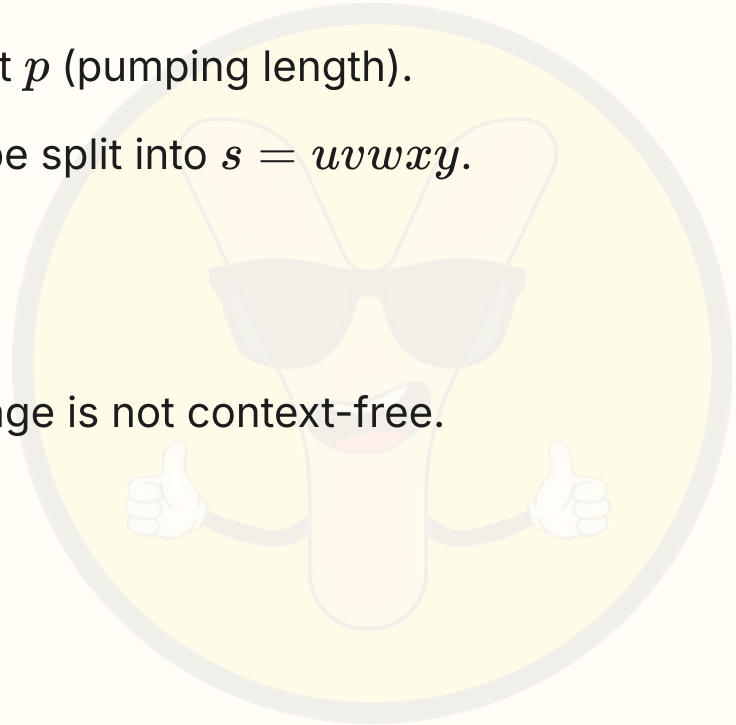
- A two-stack PDA can simulate a Turing Machine.
- It can recognize Context-Sensitive Languages (CSL).
- The two stacks allow the machine to move back and forth through the data.
- It is equivalent to a Linear Bounded Automaton (LBA) in computational power.



Pumping Lemma for CFLs

A tool to prove that a language is NOT context-free.

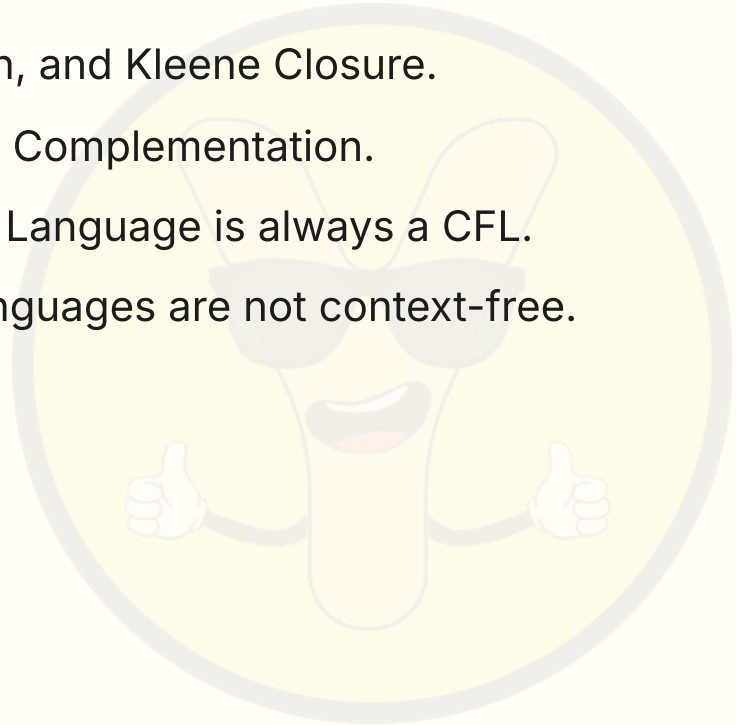
- If L is a CFL, there exists a constant p (pumping length).
- Any string $s \in L$ with $|s| \geq p$ can be split into $s = uvwxy$.
- Conditions: $|vwx| \leq p$, $|vx| \geq 1$.
- For all $i \geq 0$, $uv^iwx^iy \in L$.
- If this fails for any string, the language is not context-free.



Closure Properties of CFLs

CFLs are closed under certain operations but not others.

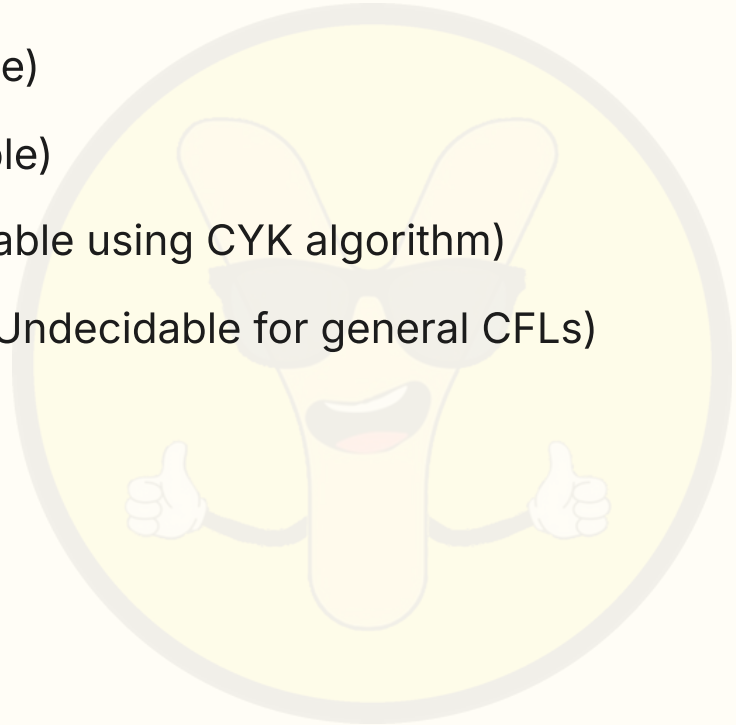
- Closed under: Union, Concatenation, and Kleene Closure.
- NOT closed under: Intersection and Complementation.
- Intersection of a CFL and a Regular Language is always a CFL.
- These properties help in proving languages are not context-free.



Decision Problems of CFLs

Algorithmic problems to determine properties of a given CFL.

- Emptiness: Is $L(G) = \emptyset$? (Decidable)
- Finiteness: Is $L(G)$ finite? (Decidable)
- Membership: Is $w \in L(G)$? (Decidable using CYK algorithm)
- Equivalence: Is $L(G_1) = L(G_2)$? (Undecidable for general CFLs)



Membership Problem (CYK Algorithm)

Check if string 'baab' is in a grammar.

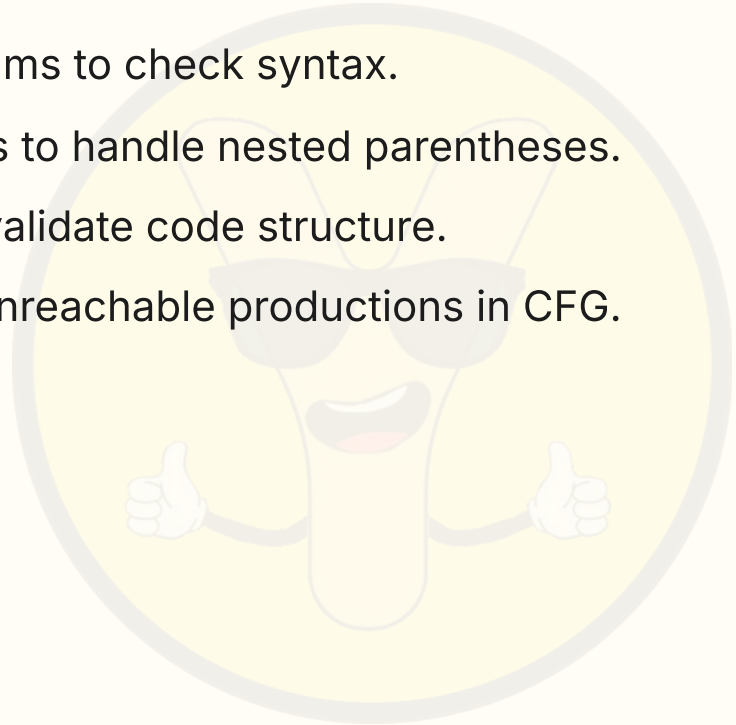
1. Convert grammar to Chomsky Normal Form (CNF).
2. Construct a triangular table of size $n \times n$.
3. Fill the table bottom-up based on productions.
4. If the top cell contains the start symbol, the string is accepted.

If the start symbol is in the final cell, the string belongs to the language.

Programming Properties of CFLs

Practical applications of CFL properties in software.

- Parsing: Using stack-based algorithms to check syntax.
- Expression Evaluation: Using stacks to handle nested parentheses.
- Compiler Front-end: Using PDA to validate code structure.
- Dead code elimination: Removing unreachable productions in CFG.



Exam corner

Q1 Define NPDA and explain its transition function with an example.

Q2 Differentiate between DPDA and NPDA with respect to power and language classes.

Q3 Construct a PDA for the language $L = \{a^n b^n | n \geq 1\}$.

Q4 State and prove the Pumping Lemma for Context-Free Languages.

Q5 Explain why CFLs are not closed under intersection using an example.

Q6 Describe the algorithm to convert a CFG into an equivalent PDA.

Q7 Discuss the decidability of the Emptiness and Membership problems for CFLs.

Unit 4 done. You are exam-ready.

Revise the Exam corner once more, then open Unit 5 on theycotes.com. Every AKTU subject, unit by unit, free. Notes by Chirag Jain, Ycotes.

