UNIT III: Context Free Grammar – Complete Guide

1. Context Free Grammar (CFG): Definition & Fundamentals

A **Context-Free Grammar (CFG)** is a 4-tuple (V, Σ, P, S), where:

- V: finite set of non-terminals
- Σ: finite set of terminals ($V \cap \Sigma = \emptyset$)
- P: finite set of productions ($A \rightarrow \alpha$ with $A \in V$ and $\alpha \in (V \cup \Sigma)^*$)
- S: start symbol ($S \in V$)

CFGs describe the syntax of programming languages, arithmetic expressions, and are fundamental for compilers and parsing algorithms[1][2].

Video Resource: Definition of CFG with an Example — Sudhakar Atchala
Watch here

GeeksforGeeks Reference: What is Context-Free Grammar?

---

2. Leftmost and Rightmost Derivations, Parse Trees

- **Leftmost Derivation:** At each step, replace the leftmost non-terminal.
- **Rightmost Derivation:** Always replace the rightmost non-terminal.
- **Parse Tree (Derivation Tree):** Hierarchical tree showing how the start symbol derives the string.

Example for CFG:

$S \rightarrow aSb \mid \varepsilon$

String: "aabb"
Leftmost: $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb \Rightarrow aabb$
Rightmost: $S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb \Rightarrow aabb$

Video Resource: Leftmost/Rightmost Derivations & Parse Trees — Sudhakar Atchala
Watch here

GeeksforGeeks Reference: Parse Trees and Derivations

---

3. Ambiguous Grammars

A **grammar is ambiguous** if at least one string can be generated in more than one way (i.e., multiple leftmost derivations or parse trees).

Example: $S \to S + S \mid S * S \mid a$

Video Resource: Ambiguous Grammar — Sudhakar Atchala
Watch here

GeeksforGeeks Reference: Ambiguity in Context-Free Grammar

---

4. Simplification of CFG

- **Elimination of Useless Symbols:** Remove non-generating or unreachable symbols.
  - Removal of Useless Symbols (Video) – Sudhakar Atchala
  - GFG: Simplifying Context Free Grammars
- **Elimination of ϵ-Productions:** Remove rules of the form $A \to \epsilon$ except possibly for start symbol.
  - Video: Removal of ϵ-Productions
  - GFG: Simplifying CFG
- **Elimination of Unit Productions:** Rules where a non-terminal produces another single non-terminal ($A \to B$).
  - Video: Removal of Unit Productions
  - GFG: Simplifying CFG
- **Comprehensive Example:**
  Minimization of CFG (Sudhakar Atchala)

---

5. Normal Forms: Chomsky Normal Form (CNF) and Greibach Normal Form (GNF)

Chomsky Normal Form (CNF)

- All productions are of the form $A \to BC$ or $A \to a$ (or possibly $S \to \epsilon$).
- Used for efficient parsing (CYK algorithm).

Steps for conversion:

1. Remove useless, unit, and ϵ-productions.
2. Replace terminals on RHS with new non-terminals if mixed.
3. Ensure RHS is either two non-terminals or a single terminal.

- CNF Conversion Video (Sudhakar Atchala)
- GFG: CFG to CNF

Greibach Normal Form (GNF)

- Every production: $A \to a\alpha$ (a is a terminal, α is a string of non-terminals).
- GNF Conversion Video (Sudhakar Atchala)
- GFG: CFG to GNF

---

6. Pumping Lemma for Context-Free Languages

The **Pumping Lemma** is used to prove certain languages are NOT context-free.
Statement: For any context-free language L, there exists $p > 0$ such that every $w \in L$ with $|w| \geq p$ can be written as $w = uvxyz$, satisfying:

- $|vxy| \leq p$
- $vx \neq \epsilon$
- For all $i \geq 0, uv^ixy^iz \in L$

Application:

- Used to show languages like $\{a^nb^nc^n\}$ are not context-free.

Video Resource: Pumping Lemma for CFL — Sudhakar Atchala
Watch here

GeeksforGeeks Reference: Pumping Lemma for CFL

---

7. Closure Properties of Context-Free Languages

Context-Free Languages (CFLs) are:

- Closed under: **Union, Concatenation, Kleene Star**
- Not closed under: **Intersection, Complement**

Table:

Operation	Regular	Context-Free	Context-Sensitive
Union	Yes	Yes	Yes
Concatenation	Yes	Yes	Yes
Kleene Star	Yes	Yes	Yes
Intersection	Yes	No	Yes
Complement	Yes	No	Yes

Video Resource: Closure Properties of CFL — Sudhakar Atchala
Watch here

GeeksforGeeks Reference:

- CFL Closure Properties (Summary Table)

- Closure Properties Detailed

8. Applications of Context-Free Grammars

- Syntax analysis in compilers (parse tree, syntax checker)
- Programming language design (formal syntax specification)
- XML/HTML parsing (document structure validation)
- Artificial intelligence (natural language modeling)
- Expression evaluation (arithmetic, logical, mathematical expressions)

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

GeeksforGeeks Reference: Applications of CFG

9. Practice Questions

1. Construct a CFG for $L = \{a^n b^n \mid n \geq 0\}$.
2. Show with an example an ambiguous grammar and explain why it is ambiguous.
3. Convert $\{S \to aSb|bSa|\epsilon\}$ to CNF.
4. Use the Pumping Lemma to show $L = \{a^n b^n c^n|n \geq 1\}$ is not context-free.
5. Describe closure properties of CFLs with one language example for each closed and non-closed operation.

10. References

[1] Sudhakar Atchala. (2018–2025). Theory of Computation and Automata Theory Video Series. YouTube.
[2] GeeksforGeeks. Various TOC & CFL Articles. https://www.geeksforgeeks.org/theory-of-computation/

Essential Sudhakar Atchala Video 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.