Parul® University

NAAC A++

Chomsky hierarchy of languages

Chapter - 1: Introduction

Prof. Riddhi Atulkumar Mehta
Assistant Professor
Department of Computer Science and Engineering

Parul® University

NAAC A++

Content

INDEX

What is the Chomsky Hierarchy?

Definition:

The Chomsky Hierarchy, introduced by Noam Chomsky in 1956, classifies formal languages into four types based on generative grammars.

Purpose:

To understand the power and limitations of different types of grammars and machines.

The Four Types of Grammars

Type	Grammar Name	Recognized By
Type 0	Unrestricted Grammar	Turing Machine
Type 1	Context-Sensitive Grammar	Linear Bounded Automaton (LBA)
Type 2	Context-Free Grammar	Pushdown Automaton (PDA)
Type 3	Regular Grammar	Finite Automaton (DFA/NFA)

Type 0 – Unrestricted Grammars

- Most powerful type
- **Production Rule Form**: $\alpha \to \beta$
  - No restriction, except $\alpha \neq \varepsilon$
- **Generated Language**: Recursively Enumerable Language (REL)
- Machine Model: Turing Machine
- **Example**:

  $S \to aSb \mid SS \mid \varepsilon$

Type 1 – Context-Sensitive Grammars

- Production Rule Form: $\alpha A\beta \rightarrow \alpha\gamma\beta$
  - γ is non-empty and $|\gamma| \geq |A|$

- Context on both sides of A determines its replacement

Generated Language: Context-Sensitive Language (CSL)

Machine Model: Linear Bounded Automaton (LBA)

Example:

$a^n b^n c^n$ ($n \geq 1$)

Type 2 – Context-Free Grammars (CFGs)

Production Rule Form: A → γ

- A is a single non-terminal
- $\gamma \in (V \cup \Sigma)^*$

Generated Language: Context-Free Language (CFL)

Machine Model: Pushdown Automaton (PDA)

Common Use: Programming languages, compilers

Example:

$S \rightarrow aSb \mid \varepsilon$

→ Generates: $\{a^n b^n \mid n \geq 0\}$

Type 3 – Regular Grammars

Production Rule Form:

- A → aB or A → a
- Only one non-terminal on the left, and one terminal + optional non-terminal on the right

Generated Language: Regular Language

Machine Model: Finite Automaton (DFA/NFA)

Use Case: Text searching, lexical analysis

Example:

S → aS | bS | ε

Summary Table

Type	Grammar	Production Rules	Language Class	Machine
0	Unrestricted	$\alpha \to \beta$	Recursively Enumerable	Turing Machine
1	Context-Sensitive	$\alpha A\beta \to \alpha\gamma\beta$,	γ	$\geq$
2	Context-Free	$A \to \gamma$	Context-Free	Pushdown Automaton
3	Regular	$A \to aB$ / a	Regular	Finite Automaton

Parul®
University

NAAC GRADE A++

https://paruluniversity.ac.in/