Parul® University
Vadodara, Gujarat
NAAC GRADE A++

Information and Communication Technology

Introduction

Study Guide

Prof. Riddhi Atulkumar Mehta
CSE, PIT
Parul University

1.1 Introduction to Alphabet, Languages, and Grammars

Alphabet (Σ)

- An **alphabet** is a finite set of symbols.
- Symbols are atomic units used to construct strings.
- Denoted by **Σ** (capital Greek sigma).
- **Examples:**
  - Σ = {0, 1} → Binary alphabet
  - Σ = {a, b, c, ..., z} → Lowercase English letters
  - Σ = {a, b} → Simple two-symbol alphabet

Strings

- A **string** is a finite sequence of symbols from an alphabet.
- **Length** of a string w is denoted by **|w|**.
- **Empty string:** ε (epsilon), with |ε| = 0
- **Example:**
  - If Σ = {a, b}, then:
    - "ab", "aa", "bba" are strings.
    - "c" is not a valid string over Σ.

Language (L)

- A **language** is a set of strings formed from an alphabet.
- Formally: **L ⊆ Σ*** (Kleene star denotes all possible strings over Σ including ε)
- **Examples:**
  - L = {w ∈ {0,1}* | w contains an even number of 0s}
  - L = {$a^n b^n$ | n ≥ 1} → Strings like "ab", "aabb", "aaabbb"

Grammar

- A **formal grammar** is a system that describes how strings in a language can be generated.
- Defined as a 4-tuple:

  G = (V, Σ, P, S)

  where:
  - **V**: A finite set of variables (non-terminal symbols)
  - **Σ**: A finite set of terminal symbols (alphabet), disjoint from V
  - **P**: A finite set of **production rules**
  - **S**: A special start symbol (S ∈ V)

1.2 Productions and Derivation

Productions

- Also called **rewrite rules**.
- Used to replace a variable with another variable or string.
- Format: **A → α**, where:
  - A ∈ V (non-terminal)
  - α ∈ (V ∪ Σ)* (string of terminals/non-terminals)
- **Example:**
  - A → aB
  - B → b

Derivation

- A **derivation** is a sequence of rule applications starting from the start symbol.
- Shows how strings in the language are generated.
- Written as:
  - $\mathbf{S \Rightarrow \alpha_1 \Rightarrow \alpha_2 \Rightarrow \ldots \Rightarrow w}$, where w is a string in Σ*

📌 **Leftmost Derivation**

- In each step, replace the **leftmost** non-terminal first.

📌 **Rightmost Derivation**

- In each step, replace the **rightmost** non-terminal first.

🎯 **Sentential Form**

- A string derived from the start symbol, possibly containing both terminals and non-terminals.

✅ **Example Derivation**

Given grammar:

- S → aSb | ε

Derive "aabb":

- S ⇒ aSb ⇒ aaSbb ⇒ aabb

1.3 Chomsky Hierarchy of Languages

Noam Chomsky categorized formal languages into a hierarchy based on the complexity of their grammars and the computational machines that recognize them.

Type-0: Unrestricted Grammar

- **Grammar rules:** $\alpha \rightarrow \beta$ (where $\alpha \neq \varepsilon$, α and β are strings with at least one non-terminal in α)
- **Language class:** Recursively enumerable languages
- **Recognized by:** Turing Machine
- **Most general class** – no restriction on production rules

Type-1: Context-Sensitive Grammar (CSG)

- **Grammar rules:** $\alpha A\beta \rightarrow \alpha\gamma\beta$ (where $|\gamma| \geq 1$)
- **A** can be replaced by γ only when it appears in context α and β.
- **Language class:** Context-sensitive languages
- **Recognized by:** Linear Bounded Automaton (LBA)
- **Example language:** $L = \{a^n b^n c^n \mid n \geq 1\}$

Type-1: Context-Sensitive Grammar (CSG)

- **Grammar rules:** $\alpha A\beta \rightarrow \alpha\gamma\beta$ (where $|\gamma| \geq 1$)
- **A** can be replaced by γ only when it appears in context α and β.
- **Language class:** Context-sensitive languages
- **Recognized by:** Linear Bounded Automaton (LBA)
- **Example language:** $L = \{a^n b^n c^n \mid n \geq 1\}$

Type-3: Regular Grammar

- **Grammar rules:** $A \rightarrow aB$ or $A \rightarrow a$ (Right-linear or Left-linear)
- **Language class:** Regular languages
- **Recognized by:** Finite Automaton (DFA/NFA)
- **Simplest class**, used in **lexical analysis**
- **Example:** $L = \{a^*b^*\}$

Chomsky defined **four classes of grammars**, each with increasing expressive power:

Type	Grammar Class	Language Type	Automaton	Production Form
0	Unrestricted	Recursively Enumerable	Turing Machine	$\alpha \to \beta$ ($\alpha \neq \varepsilon$)
1	Context-Sensitive	Context-Sensitive	Linear Bounded Automaton	$\alpha A\beta \to \alpha\gamma\beta$ (
2	Context-Free	Context-Free	Pushdown Automaton	$A \to \gamma$ ($A \in V, \gamma \in (V \cup \Sigma)^*$)
3	Regular	Regular	Finite Automaton	$A \to aB$ or $A \to a$ (Right-linear)

References:

1. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2006). Introduction to Automata Theory, Languages, and Computation (3rd ed.). Pearson Education.
2. Chomsky, N. (1959). On Certain Formal Properties of Grammars. Information and Control, 2(2), 137–167. https://doi.org/10.1016/S0019-9958(59)90362-6
3. GeeksforGeeks. (n.d.). *Chomsky Hierarchy in Theory of Computation*. Retrieved May 10, 2025, from https://www.geeksforgeeks.org/chomsky-hierarchy/
4. TutorialsPoint. (n.d.). *Formal Grammar and Language*. Retrieved May 10, 2025, from https://www.tutorialspoint.com/automata_theory/formal_grammar.htm
5. Wikipedia contributors. (n.d.). *Chomsky hierarchy*. Wikipedia. Retrieved May 10, 2025, from https://en.wikipedia.org/wiki/Chomsky_hierarchy

Parul® University
Vadodara, Gujarat
NAAC GRADE A++

https://paruluniversity.ac.in/