CS304: Automata and Formal Languages

Lec 2

Refresher on Basic Maths (contd.), Alphabet, Strings, Language, Induction

Rachit Nimavat

July 31, 2025

Outline

Sets

Defn: Unordered collections of distinct elements

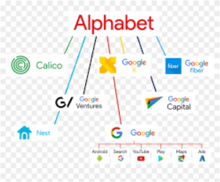
Operations: Union ($\cup$), Intersection ($\cap$), Difference ($\setminus$), Complement (A^c or $\bar{A}$)

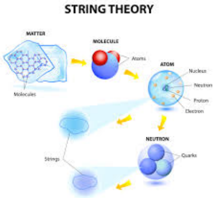
Misc Terms: Emptyset ($\emptyset$ or $\{\}$) Universe (U or $\mathcal{U}$), Superset $A \supset B$ or $A \supseteq B$, Subset $A \subset B$ or $A \subseteq B$

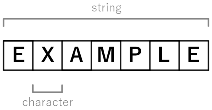
In Automata Theory, our "elements" are **strings**, which, in turn, are sequence of characters of an **alphabet**.

Collections of strings form **languages** – which are fundamentally **sets**

But what are Alphabet and Strings formally?







Alphabet and Strings

Alphabet Σ

A finite, non-empty **set of symbols**

- e.g. $\Sigma = \{0, 1\}$ (binary alphabet)
- e.g. $\Sigma = \{a, b, \dots, z\}$ (lowercase English alphabet)

String w

A finite **sequence of symbols** from Σ

- e.g. For $\Sigma = \{a, b, \dots, z\}$, $w_1 =$ xyz, $w_2 =$ aaba are strings

Length of a string ($|w|$): e.g., $|\text{aaba}| = 4$

Empty String ε with 0 symbols i.e. $|\varepsilon| = 0$

Concatenation uv: Joining strings. e.g., $w_1 w_2 =$ xyzaaba and $w_2 w_1 =$ aabaxyz

Boolean Logic

Proposition: True/False statements

For $\Sigma = \{0, 1\}$, $w_1 = 00010$, and $w_2 = 0111$,

w_1 is longer than w_2 (True)

when viewed as a number, w_1 is bigger than w_2 (False)

Connectives like AND ($\wedge$), OR ($\vee$), NOT ($\neg$) combine propositions

when viewed as numbers (w_1 is prime) $\wedge$ (w_2 is prime) (True)

Truth Table defines precise behavior of statement for all possible truth values of its components

Misc Terms

$\exists$ **"there exists"**

A property holds for **at least one** element in a domain

e.g. $\exists x \in \mathbb{N}, x$ is prime (there exists at least one prime number)

$\forall$ **"for all"**

A property holds for **every** element in a domain

e.g. $\forall x \in \mathbb{N}, x \geq 0$ (for all natural numbers x, x is greater than or equal to 0)

Functions

Defn: For sets A and B, a **function** $f : A \to B$ assigns to each element in a A (**domain**) exactly one element in a B (the **codomain**)

- $\forall x \in A, f(x) \in B$ (True)
- $\{f(x) \mid x \in A\} = B$ (False)
- $\{f(x) \mid x \in A\} \subseteq B$ (True)
- $\forall y \in B,\ \exists x$ with $f(x) = y$ (False)
- $\forall x \in A,\ \exists f(x) \in B$ (True)

$\{f(x) \mid x \in A\}$ is called **range** of f

e.g. $f : \mathbb{N} \to \mathbb{R},\ f(x) = \sqrt{x}$

- Domain: Natural numbers ($\mathbb{N}$)
- Codomain: Real numbers ($\mathbb{R}$)
- Range: $\left\{0, 1, \sqrt{2}, \sqrt{3}, \ldots, \right\} \subset \mathbb{R}$

Homework: Look up Relations on Wikipedia

Language

Alphabet Σ is a finite set

String w over Σ is a finite sequence of elements of Σ

Σ^* is the set of all strings over Σ

e.g. $\Sigma = \{0, 1\} \implies \Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \dots, \}$

Language over Σ is a set of strings over Σ

Any language over Σ is a **subset** of Σ^*

e.g. $\Sigma = \{0, 1\}$, then $A = \{w \mid w \text{ is a prime when viewed as a number}\}$ is a language

e.g. $\Sigma = \{a, b, \dots, z\}$, then $B = \{w \mid w \text{ is an English word}\}$ is a language

Languages = Problems

Language over Σ is a set of strings over Σ

Language is a **subset** of Σ^*

Problem: Given a string $w \in \Sigma^*$ and a language $L \subseteq \Sigma^*$, is $w \in L$?

Languages $\equiv$ Functions that take a string as input, and output a single bit denoting whether the string is in language or not

Theorem 2.1

Every language L over Σ uniquely corresponds to a function $f : \Sigma^* \mapsto \{0, 1\}$.

Proof Idea.

Given L, define f s.t

$$f(x) = \begin{cases} 1 & \text{if } x \in L \\ 0 & \text{otherwise} \end{cases}$$

Given f, define

$L := \{x \in \Sigma^* \mid f(x) = 1\}$

$\square$

HW: Write formal proof

Mathematical Proof[1]

A **good proof** should be:

- easy to understand
- correct!
- has 3 levels of detail
  1. the “hint”
     e.g. Proof by contradiction, Proof by induction...
  2. short para with main ideas
  3. the full proof (and nothing but the proof!)

The slides will only have hint/main ideas, but in your homeworks/quizzes/exams, you are required to provide all the details

[1] Ryan Williams’ slides!

Induction: The Power of Stepping Stones

Like proving you can climb an infinitely tall ladder

Often need to prove properties about structures that are defined **recursively**
Induction allows us to prove an infinite number of cases with a finite number of steps

Core Idea:

Imagine an infinitely tall ladder

Base Case: If you can climb the first step (prove the statement for the smallest value)

Inductive Step: And if you show that if a step can be climbed, the next one can also be climbed (prove that if the statement holds for k, it holds for $k+1$)

Conclusion: Then an infinitely tall ladder can be climbed (the statement holds for all values)

Induction - Example

Flipped Reversals

$\Sigma = \{0, 1\}$ and a string w over Σ,

- flipped(w) or $\overline{w}$ is obtained by flipping all bits of w
- reverse(w) or w^R is obtained by "reading" w from right-to-left

Observation 1

For any binary string w, $\overline{(w^R)} = (\overline{w})^R$.

Proving: For any binary string w, $\overline{(w^R)} = (\overline{w})^R$

Base Case

Proof by Induction on $|w|$.
Base case: $w = \varepsilon$ (the empty string)

On Board!

- Let $w = \varepsilon$
- LHS: $\overline{(\varepsilon^R)} = \overline{\varepsilon} = \varepsilon$ (since $\varepsilon^R = \varepsilon = \overline{\varepsilon}$)
- RHS: $(\overline{\varepsilon})^R = \varepsilon^R = \varepsilon$
- Since LHS $=$ RHS, the property holds for $|w| = 0$

Proving: For any binary string w, $\overline{(w^R)} = (\overline{w})^R$ ✔

Induction

Proof by Induction on $|w|$.
Base case: $w = \varepsilon$ ✔

Inductive Hypothesis Assume that the property holds for some arbitrary string x with length $k \geq 0$. i.e. assume $\overline{(x^R)} = (\overline{x})^R$
Inductive Step Let w be an arbitrary string of length $k + 1$ ✔ On Board!
Conclusion:

- Since the base case holds, and the inductive step has shown that if the property holds for strings of length k, it also holds for strings of length $k + 1$,
- By Induction, $\overline{(w^R)} = (\overline{w})^R$ for all binary strings w.

$\square$