# 04012022-UNIT1

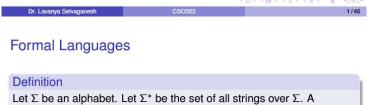
04 January 2022 13:32



Unit1-Slides

CSO322: Theory of Computation
Unit-1: Formal Languages and Finite Automata

Dr. Lavanya Selvaganesh



Let  $\Sigma$  be an alphabet. Let  $\Sigma^*$  be the set of all strings over  $\Sigma$ . A language is a subset of  $\Sigma^*$ .





# Formal Languages

#### Definition

Let  $\Sigma$  be an alphabet. Let  $\Sigma^*$  be the <u>set of all strings over  $\Sigma$ </u>. A language is a subset of  $\Sigma^*$ 

# Example Let $\Sigma = \{a, b\}$ . $\Sigma^* = \{\epsilon, a, b, aa, bb, ab, ba, aaa, ...\}$ . Examples of languages over $\Sigma$ are $L_1 = \{\epsilon, a, aa, aab\}$ .

#### Definition

Let  $\Sigma$  be an alphabet. Let  $\Sigma^*$  be the set of all strings over  $\Sigma$ . A language is a subset of  $\Sigma^*$ .

Let  $\Sigma = \{a, b\}$ .  $\Sigma^* = \{\epsilon, a, b, aa, bb, ab, ba, aaa, ...\}$ .

Examples of languages over  $\Sigma$  are

- $2 L_2 = \{ \underline{x} \in \Sigma^* : |x| \le 8 \}.$

# Formal Languages

#### Definition

Let  $\Sigma$  be an alphabet. Let  $\Sigma^*$  be the set of all strings over  $\Sigma$ . A language is a subset of  $\Sigma^*$ .

#### Example

Let  $\Sigma = \{a, b\}$ .  $\Sigma^* = \{\epsilon, a, b, aa, bb, ab, ba, aaa, ...\}$ .

Examples of languages over  $\Sigma$  are

- **1**  $L_1 = \{\epsilon, a, aa, aab\}.$
- $L_3 = \{X \in \Sigma^* : n_a(X) \ge n_b(X)\}$  On one point on processing the number of a's and b's in x respectively.

## Formal Languages

#### Definition

Let  $\Sigma$  be an alphabet. Let  $\Sigma^*$  be the set of all strings over  $\Sigma$ . A language is a subset of  $\Sigma^*$ .

#### Example

Let  $\Sigma = \{a, b\}$ .  $\Sigma^* = \{\epsilon, a, b, aa, bb, ab, ba, aaa, ...\}$ .

Examples of languages over  $\Sigma$  are

- $\bullet$   $L_1 = \{\epsilon, a, aa, aab\}.$
- ②  $L_2 = \{x \in \Sigma^* : |x| \leq 8\}.$
- **3**  $L_3 = \{x \in \Sigma^* : n_a(x) \ge n_b(x)\}$

where  $n_a$  and  $n_b$  are the number of a's and b's in x respectively.

## Formal Languages

New languages can be constructed using set operations, since languages are sets of strings.

- New languages can be constructed using set operations, since languages are sets of strings.
  - ▶ For any two languages over an alphabet  $\Sigma$ , their union, intersection and difference are also languages over  $\Sigma$ .



## Formal Languages

- New languages can be constructed using set operations, since languages are sets of strings.
  - For any two languages over an alphabet Σ, their union, intersection
  - and difference are also languages over  $\Sigma$ .

    Complement of a language over  $\Sigma$  is defined by  $L' = \Sigma^* L$ .



- New languages can be constructed using set operations, since languages are sets of strings.
  - For any two languages over an alphabet Σ, their union, intersection and difference are also languages over  $\Sigma$ .

  - and difference are also languages over  $\Sigma$ .

    Complement of a language over  $\Sigma$  is defined by  $L' = \Sigma^* L$ .

    Suppose  $L_1 \subseteq \Sigma_1^*$  and  $L_2 \subseteq \Sigma_2^*$ , then  $L_1$  and  $L_2$  are languages over  $(\Sigma_1 \cup \Sigma_2)^*$  i.e.  $L_1, L_2 \subseteq (\Sigma_1 \cup \Sigma_2)^*$

- New languages can be constructed using set operations, since languages are sets of strings.
  - For any two languages over an alphabet Σ, their union, intersection and difference are also languages over Σ.
  - ▶ Complement of a language over  $\Sigma$  is defined by  $L' = \Sigma^* L$ .
  - ▶ Suppose  $L_1 \subseteq \Sigma_1^*$  and  $L_2 \subseteq \Sigma_2^*$ , then  $L_1$  and  $L_2$  are languages over  $(\Sigma_1 \cup \Sigma_2)^*$  i.e.

 $L_1, L_2 \subseteq (\Sigma_1 \cup \Sigma_2)^*$ 

► However, there is a possibility for complement of L₁ to be either

 $L_1 = \Sigma_1^* - L_1 \text{ or } L_1' = (\Sigma_1 \cup \Sigma_2)^* - L_1.$ 

Depending on the context, it will be clear which alphabet is referred to

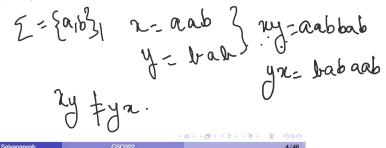
4 D +	(5)	121	12:	Mark	200

### Formal Languages

Concatenation operation on strings will allow us to construct new languages.



- Concatenation operation on strings will allow us to construct new languages.
  - If x and y are elements of Σ\*, the concatenation of x and y is the string xy formed by writing the symbols of x followed by symbols of y.



- Concatenation operation on strings will allow us to construct new languages.
  - If x and y are elements of Σ\*, the concatenation of x and y is the string xy formed by writing the symbols of x followed by symbols of y.

#### Example

If x = abb and y = ba, then xy = abbba and yx = baabb.

		+ 🖂 +	10	121	121	100	200
Dr. Lavanya Selvaganesh	CSO322						4/46

#### Formal Languages

- Concatenation operation on strings will allow us to construct new languages.
  - If x and y are elements of Σ\*, the concatenation of x and y is the string xy formed by writing the symbols of x followed by symbols of y.

#### Example

If x = abb and y = ba, then xy = abbba and yx = baabb.

For any string x,  $x\epsilon = \epsilon x = x$ .

E e E L = { E} empty string. L= \$ = empty string. L= \$ - empty languige

# Dr. Lavanya Selvaganesh CSO322 4/4

#### Formal Languages

- Concatenation operation on strings will allow us to construct new languages.
  - If x and y are elements of Σ\*, the concatenation of x and y is the string xy formed by writing the symbols of x followed by symbols of y.

#### Example

If x = abb and y = ba, then xy = abbba and yx = baabb.

- ▶ For any string x,  $x\epsilon = \epsilon x = x$ .
- For strings x, y, z; (xy)z = x(yz) i.e. concatenation is associative.

by 12 or sulys)



- Substring of a string
  - x is a substring of another string y, if there are strings w and z, either or both may be null, so that y = wxz.

#### Substring of a string

x is a substring of another string y, if there are strings w and z, either or both may be null, so that y = wxz.

#### Example

The string "car" is a substring of "descartes", "vicar", "cartridge" and "car", but not a substring of "charity".



#### Formal Languages

#### Substring of a string

x is a substring of another string y, if there are strings w and z, either or both may be null, so that y = wxz.

#### Example

The string "car" is a substring of "descartes", "cartridge" and "car", but not a substring of "charity".

► A **prefix** of a string is an initial substring. Example - Prefixes of <u>abaa</u> – ε, a, ab, aba, abaa.



#### Formal Languages

#### Substring of a string

x is a substring of another string y, if there are strings w and z, either or both may be null, so that y = wxz.

#### Example

The string "car" is a substring of "descartes", "vicar", "cartridge" and "car", but not a substring of "charity".

- A **prefix** of a string is an initial substring. Example Prefixes of  $abaa \epsilon$ , a, ab, aba, aba.
- A suffix of a string is a final substring.
   Example Suffixes of abaa ε, a, aa, baa, abaa.

Concatenation of languages is also possible.

► If 
$$L_1, L_2 \subseteq \Sigma^*, L_1L_2 = \{xy | x \in L_1, y \in L_2\}$$

Dr. Lavanya Selvaganesh	CSO322				6/46
		10 1 15 1	451451	100	200



$$\qquad \qquad \textbf{If } L_1, L_2 \subseteq \Sigma^*, L_1L_2 = \{xy | x \in L_1, y \in L_2\}$$

