



# Theory of Machines and Languages

Fatemeh Deldar

1403-1404

# References

1. **P. Linz and S. H. Rodger. An introduction to formal languages and automata.** 7th Edition, Jones and Bartlett Publishers, 2022
2. **M. Sipser. Introduction to the Theory of Computation.** 3rd Edition, Cengage Learning, 2012

# Grading

- ❑ **Midterm Exam**

- 6 pts

- ❑ **Final Exam**

- 9 pts

- ❑ **Quiz**

- 1 pts

- ❑ **Homework**

- 4 pts

- ❑ **Class Activity**

- 1 pts

# Syllabus

## Introduction to the Theory of Computation

## Finite Automata

- Finite Automata
- Regular Languages and Regular Grammars
- Properties of Regular Languages

## Context-Free Languages

- Context-Free Languages
- Simplification of Context-Free Grammars and Normal Forms
- Pushdown Automata
- Properties of Context-Free Languages

## Turing Machines and Computability

- Turing Machines
- A Hierarchy of Formal Languages and Automata
- Limits of Algorithmic Computation
- An Overview of Computational Complexity

# Introduction to the Theory of Computation

# Three Basic Concepts

## 1. Languages

### ➤ Alphabet

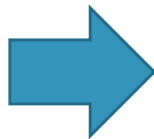
- A finite and nonempty set of symbols ( $\Sigma$ )

### ➤ Strings

- Finite sequences of symbols from the alphabet
  - **Example:** If the alphabet  $\Sigma = \{a, b\}$ , then *abab* and *aaabbbba* are strings on  $\Sigma$
- The concatenation of two strings  $w$  and  $v$  is the string obtained by appending the symbols of  $v$  to the right end of  $w$

$$w = a_1 a_2 \cdots a_n$$

$$v = b_1 b_2 \cdots b_m$$



$$wv = a_1 a_2 \cdots a_n b_1 b_2 \cdots b_m$$

- The reverse of a string is obtained by writing the symbols in reverse order

$$w^R = a_n \cdots a_2 a_1$$

# Three Basic Concepts

## 1. Languages

### ➤ Strings

- The length of a string  $w$ , denoted by  $|w|$ , is the number of symbols in the string
- Empty string is a string with no symbols at all ( $\lambda$ )
- Any string of consecutive symbols in  $w$  is said to be a substring of  $w$
- If  $w = vu$ , then the substrings  $v$  and  $u$  are said to be a prefix and a suffix of  $w$ , respectively
- **Example:** if  $w = abbab$ , then  $\{\lambda, a, ab, abb, abba, abbab\}$  is the set of all prefixes of  $w$
- $|uv| = |u| + |v|$
- If  $w$  is a string, then  $w^n$  stands for the string obtained by repeating  $w$   $n$  times
  - $w^0 = \lambda$
- If  $\Sigma$  is an alphabet, then we use  $\Sigma^*$  to denote the set of strings obtained by concatenating zero or more symbols from  $\Sigma$

$$\Sigma^+ = \Sigma^* - \{\lambda\}$$

# Three Basic Concepts

## 1. Languages

- A language is defined very generally as a subset of  $\Sigma^*$
- A string in a language  $L$  will be called a sentence of  $L$
- Any set of strings on an alphabet  $\Sigma$  can be considered a language
- **Example:**
  - Let  $\Sigma = \{a, b\}$ , then  $\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \dots\}$
  - The set  $\{a, aa, aab\}$  is a finite language on  $\Sigma$
  - The set  $L = \{a^n b^n : n \geq 0\}$  is also an infinite language on  $\Sigma$



# Three Basic Concepts

## 2. Grammars

- To study languages mathematically, we need a mechanism to describe them
- A grammar for the English language tells us whether a particular sentence is well formed or not

$$\begin{aligned}\langle sentence \rangle &\rightarrow \langle noun\_phrase \rangle \langle predicate \rangle \\ \langle noun\_phrase \rangle &\rightarrow \langle article \rangle \langle noun \rangle \\ \langle predicate \rangle &\rightarrow \langle verb \rangle\end{aligned}$$

- The grammar tells us that the sentences “a boy runs” and “the dog walks” are properly formed

# Three Basic Concepts

## 2. Grammars

A grammar  $G$  is defined as a quadruple

$$G = (V, T, S, P),$$

where  $V$  is a finite set of objects called **variables**,  
 $T$  is a finite set of objects called **terminal symbols**,  
 $S \in V$  is a special symbol called the **start** variable,  
 $P$  is a finite set of **productions**.

It will be assumed without further mention that the sets  $V$  and  $T$  are non-empty and disjoint.

# Three Basic Concepts

## 2. Grammars

- The production rules are the heart of a grammar
  - They specify how the grammar transforms one string into another
  - Through this they define a language associated with the grammar
- If  $x \rightarrow y$ , where  $x$  is an element of  $(V \cup T)^+$  and  $y$  is in  $(V \cup T)^*$

$$w = uxv \quad \longrightarrow \quad z = uyv$$

- We say that  $w$  derives  $z$  or that  $z$  is derived from  $w$  ( $w \Rightarrow z$ )
- If  $w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow w_n$  we say that  $w_1$  derives  $w_n$  ( $w \stackrel{*}{\Rightarrow} z$ )
  - The  $*$  indicates that an unspecified number of steps (including zero)

# Three Basic Concepts

## 2. Grammars

Let  $G = (V, T, S, P)$  be a grammar. Then the set

$$L(G) = \left\{ w \in T^* : S \xRightarrow{*} w \right\}$$

is the language generated by  $G$ .

# Three Basic Concepts

## 2. Grammars

Consider the grammar

$$G = (\{S\}, \{a, b\}, S, P),$$

with  $P$  given by

$$S \rightarrow aSb,$$

$$S \rightarrow \lambda.$$

Then

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb,$$

so we can write

$$S \xRightarrow{*} aabb.$$

$$\Rightarrow L(G) = \{a^n b^n : n \geq 0\}$$

- The string ***aabb*** is a ***sentence*** in the language generated by  $G$ , while ***aaSbb*** is a ***sentential form***