*Pharos University in Alexandria*
*Faculty of Computer Science & Artificial Intelligence*
*Course Title: Theory of Computation*
*Code: CS 307*

# Theory of Computation

## Lecturer: Sherine Shawky

Text Books

1. Introduction to formal languages and automata, Peter Linz, 6th edition, 2017.

# Week 1

# Introduction to Theory of Computation

# Agenda

- Hi ☺
- Marks
- Class Rules
- Importance
- Definition
- Mathematical Notation
- Languages
- Grammar
- Automaton

# Importance

- It is the most important course in CS.
- It helps us to understand how people thought of CS as a science in last 50 years.
- It deals with kind of things that can be computed mechanically showing how fast and required space that it needed.
- It provides concepts and principles that help us understand how to design machine.
- What are the mathematical properties of computer hardware and software?
- What are the limitations of computers? Can everything be computed?

# Definition

- **Theory of computation** is the branch that deals with whether and how efficiently problems can be solved on a model of computation, using an algorithm.
- It is divided into three major branches:
  - automata theory
  - computability theory
  - complexity theory
- There are several models in use, but the most commonly examined is the Turing machine.

# Mathematical Notation

- Sets
- Functions
- Graphs and Trees
- Proof Techniques

# Mathematical Notation

- Sets:
  - A set is a collection of elements, without any structure other than membership.
  - $S = \{0,1,2\}$

# Mathematical Notation

- Union
  - Union (U): the combination of the distinct elements from both sets.

- Intersection:
  - Intersection (∩): the elements common to both sets.

- Difference:
  - A difference of two sets is the elements in one set that are NOT in the other.

- Complement :
  - A complement of a set is all the elements that are NOT in the set.
  - Complement symbol is a bar above the set name.

# Mathematical Notation

- Power set:
  - The power set of a set $S$ (written as $2^S$) is the set of all the subsets of the set $S$.

If $S$ is the set $\{a, b, c\}$, then its powerset is

$$2^S = \{\varnothing, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

- Cartesian product:
  - The Cartesian product of two sets $S$ and $T$, is the set of all the ordered pairs created by choosing one element of $S$ and one element of $T$.

Let $S_1 = \{2, 4\}$ and $S_2 = \{2, 3, 5, 6\}$. Then

$$S_1 \times S_2 = \{(2,2), (2,3), (2,5), (2,6), (4,2), (4,3), (4,5), (4,6)\}$$

# Mathematical Notation

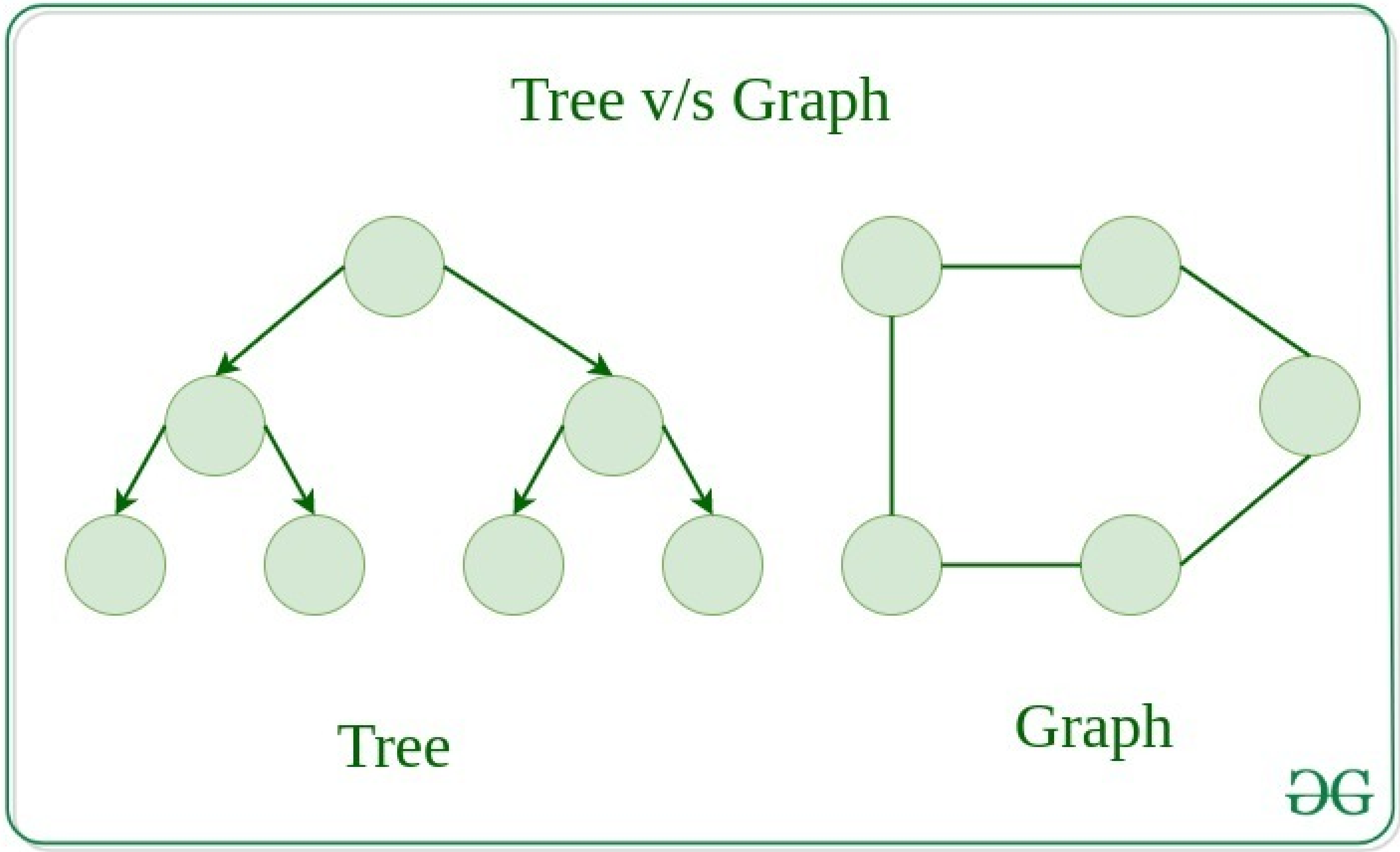
- Functions:
  - A function *f* : is the mapping of elements of *S* to unique elements of *T*.

A **function** is a rule that assigns to elements of one set a unique element of another set. If $f$ denotes a function, then the first set is called the **domain** of $f$, and the second set is its **range.** We write

$$f : S_1 \to S_2$$

to indicate that the domain of $f$ is a subset of $S_1$ and that the range of $f$ is a subset of $S_2$.

# Mathematical Notation




https://www.geeksforgeeks.org/difference-between-graph-and-tree/

# Mathematical Notation

- Proof Techniques:

  - Direct Proof
  - Proof by  contrapositive
  - Proof by contradiction.
  - Proof by induction.

# Languages

- Language: a set of strings based on a certain alphabet.
    - Alphabet (usually denoted by S)
        - Finite
        - Non-Empty
        - Consists of a Set of Symbols
        - Usually show with lowercase letters like a, b, c, …
        - Example: S = {e, u, t, r}, L = { eut, tr, rrr, …}
    - Symbols :
        - Symbols are indivisible objects or entity that cannot be defined.
        - That is, symbols are the atoms of the world of languages.
        - A symbol is any single object such as , *a*, 0, 1, #, begin, or do.

# Languages

- **Concatenation**
  - Example: $w = a_1a_2\ldots a_n$ $u = b_1b_2\ldots b_m$
  - $w\ u = a_1a_2\ldots a_n\ b_1b_2\ldots b_m$
  - $\boldsymbol{\lambda}\ w = w\ \boldsymbol{\lambda} = w$
  - $|uv| = |u| + |v|$ “Length”

# Languages

- **Star Closure (Universe)**
  - Set of all strings over S (alphabet)

Let $\Sigma = \{a, b\}$. Then

$$\Sigma^* = \{\lambda, a, b, aa, ab, ba, bb, aaa, aab, \ldots\}.$$

The set

$$\{a, aa, aab\}$$

is a language on $\Sigma$. Because it has a finite number of sentences, we call it a finite language. The set

$$L = \{a^n b^n : n \geq 0\}$$

# Grammars

- **Grammars:**

A grammar $G$ is defined as a quadruple

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

- <u>**Definition:**</u>
  - **V** = Set of Variables (non-empty)
  - **T** = Set of Terminal Symbols
  - **S** = Start Symbol
  - **P** = Set of Productions

# Grammars

- Example 1:

  Consider the grammar: $G = (\{S\}, \{a, b\}, S, P)$,

  with $P$ given by

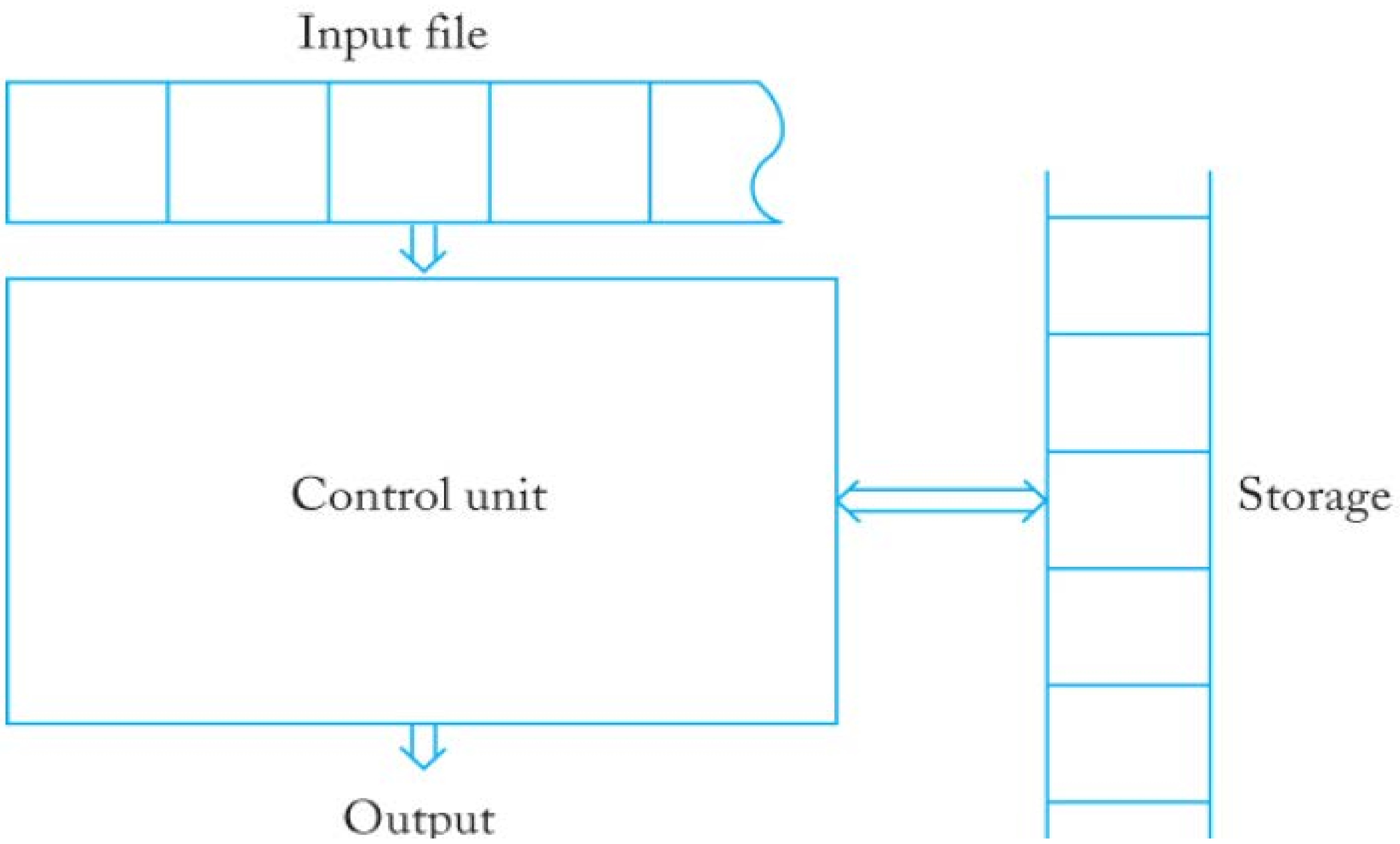
  $$S \rightarrow aSb,$$
  $$S \rightarrow \lambda.$$

  *It generates the language:*

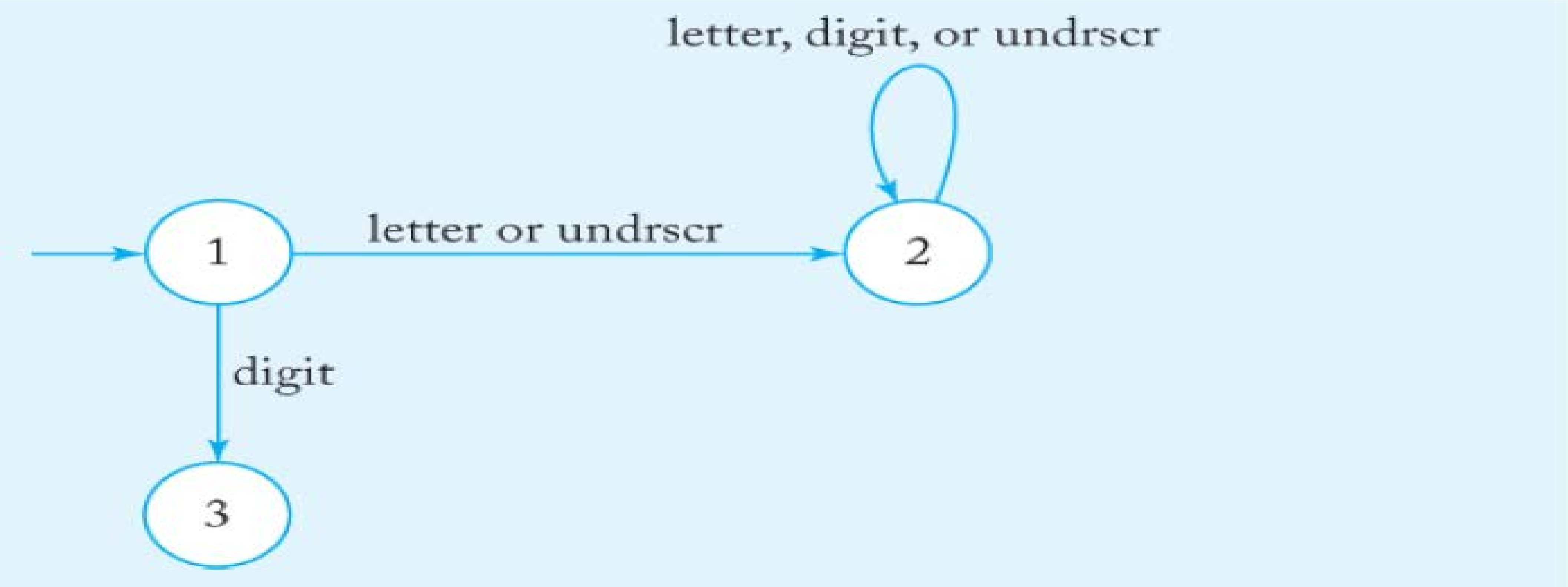
  $L(G) = \{\lambda$, ab, aabb, aaabbb, …}

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

# Automaton



# Automaton




Thank you