CS116-Automata Theory and Formal Languages

Lecture 0
Course Preliminaries

Computer Science Department

1st Semester 2025-2026

Course Description

- Finite state automata and regular expressions; context free grammars and pushdown automata; Turing machines & recursively enumerable sets

  (3 units Lecture)

  Prerequisites: Discrete and Data Structures

Course Learning Outcomes

1) Construct deterministic and non-deterministic finite automata for given regular languages.
2) Design and test regular expressions equivalent to finite automata.
3) Develop context-free grammars for specific languages and convert them to equivalent pushdown automata.
4) Demonstrate understanding of the computational power and limitations of Turing machines.
5) Apply automata theory concepts to solve computational problems and analyze their complexity.

Course Outline

1. Introduction

- 3 central areas of the theory of Computation
- Historical Perspective
- What is Automata Theory?
- The Chomsky Hierarchy of Languages

2. Finite Automata and Regular Expressions

- Deterministic finite automata(DFA)
- Nondeterministic finite automata(NFA)
- NFA and DFA equivalence
- (Non)Regular languages
- Regular expressions
- Properties of regular languages
- Pumping lemma for regular languages

3. Pushdown Automata and Context-Free Grammars

- (Un)ambiguous context-free grammars (CFGs)
- (Non)context-free languages
- (Non)deterministic pushdown automata (PDA)
- Chomsky normal form (CNF)
- Closure properties of CFLs
- Pumping lemma for CFLs

4. Turing Machines

- Church-Turing thesis
- Language of a Turing machine
- (Non)deterministic Turing machines, single and multi-tape Turing machines
- Universal Turing machine
- Other Turing machine variants
- Decidable and undecidable problems
- Intractability, classes P and NP
- NP-completeness

References

Textbooks

- Sipser, M. (2012). Introduction to the Theory of Computation (3rd ed.)
- Hopcroft, J.E., Motwani, R., & Ullman, J.D. (2006). Introduction to Automata Theory, Languages, and Computation (3rd ed.)
- Linz, P. (2016). An Introduction to Formal Languages and Automata (6th ed.)
- Martin, J.C. (2010). Introduction to Languages and the Theory of Computation (4th ed.)

Other resources:

- JFLAP: Java Formal Languages and Automata Package (http://www.jflap.org/) - simulation tool for automata.
- Online RE simulators and Turing machine simulators (e.g., https://turingmachinesimulator.com/, https://regex101.com/)

Suggested Readings (Articles and Online Resources)

- Sipser, M. (2005). Why Theoretical Computer Science Matters. Communications of the ACM, 48(3), 31–33.
- Crespi-Reghizzi, S. (2020). Teaching Automata and Formal Languages: From the 1970s to the 2020s. ITiCSE Working Group Reports.
- Ginsburg, S. (1966). The Mathematical Theory of Context-Free Languages. McGraw-Hill.
- Turing, A. (1937). On Computable Numbers, with an Application to the Entscheidungsproblem. Proceedings of the London Mathematical Society.
- Gold, E.M. (1967). Language Identification in the Limit. Information and Control, 10(5), 447–474.

Course Requirements & Grading System

• 3 Long Exams	75%
• Quizzes	15%
• Recitation	10%

FB GC and BULMS

- Facebook messenger group chat will be used for instant communication and announcements.
  - Observe the given guidelines on how to behave in the GC
- BU Learning Management System will be utilized for posting and/or submission of learning resources and activities.
  - CS116-Automata Theory and Formal Languages (2025)
  - Self enrollment key: cs116-2025-A (bloc A)

    cs116-2025-B (bloc B)

Mathematical Preliminaries

- Sets
- Sequences and Tuples
- Functions
- Relations
- Graphs
- Proof Techniques

SETS

A set is a collection of elements

$$A = \{1, \ 2, 3\}$$

$$B = \{train, bus, bicycle, airplane\}$$

We write

$$1 \in A$$

$$ship \notin B$$

Set Representations

$C = \{ a, b, c, d, e, f, g, h, i, j, k \}$

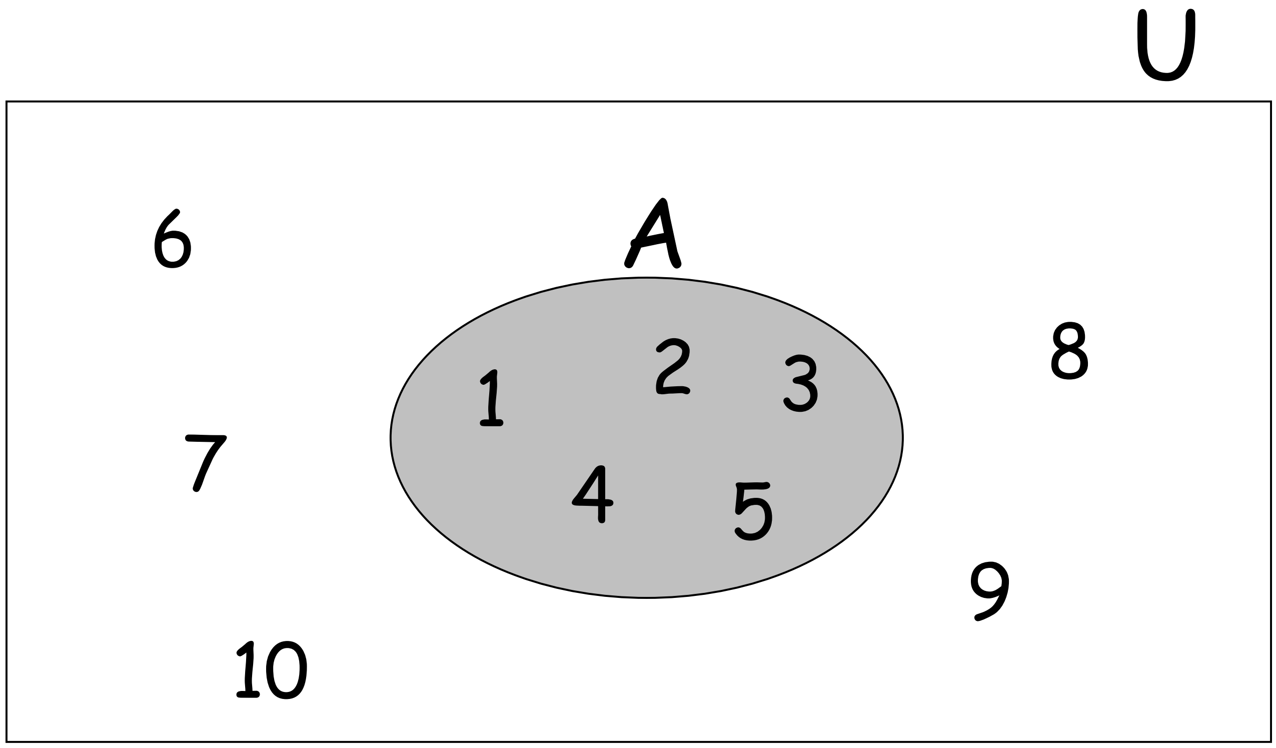
$C = \{ a, b, \ldots, k \}$ $\longrightarrow$ *finite set*

$S = \{ 2, 4, 6, \ldots \}$ $\longrightarrow$ *infinite set*

$S = \{ j : j > 0, \text{ and } j = 2k \text{ for some } k>0 \}$

$S = \{ j : j \text{ is nonnegative and even} \}$

$A = \{ 1, 2, 3, 4, 5 \}$



Universal Set: all possible elements

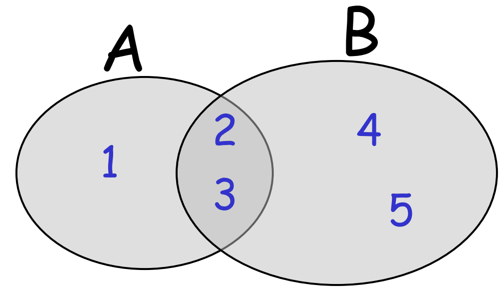
$U = \{ 1 , \ldots , 10 \}$

Set Operations

$A = \{1, 2, 3\}$ $B = \{2, 3, 4, 5\}$

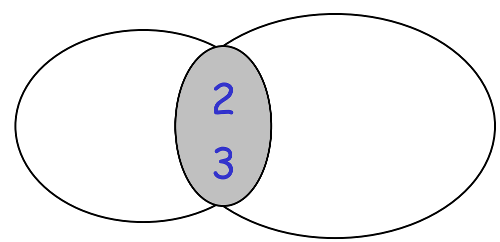
- Union

  $A \cup B = \{1, 2, 3, 4, 5\}$



- Intersection

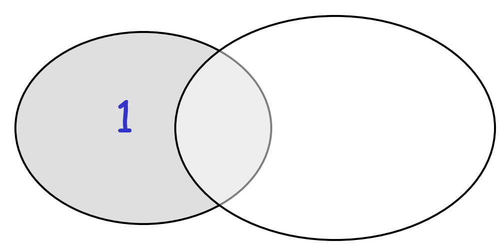
  $A \cap B = \{2, 3\}$



- Difference

  $A - B = \{1\}$

  $B - A = \{4, 5\}$

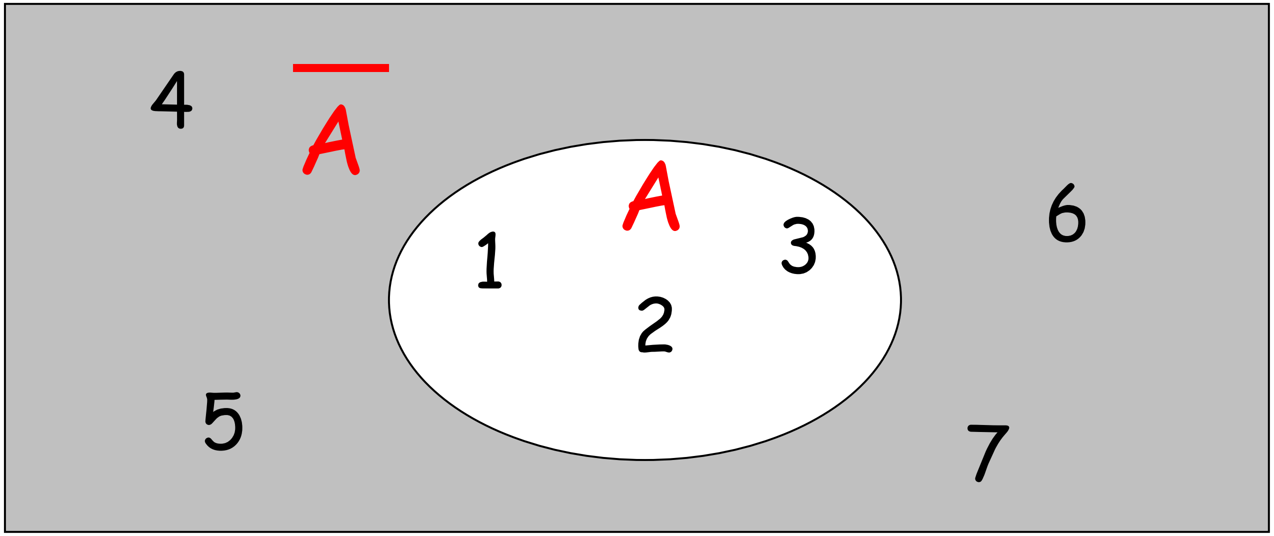


Venn diagrams

- Complement

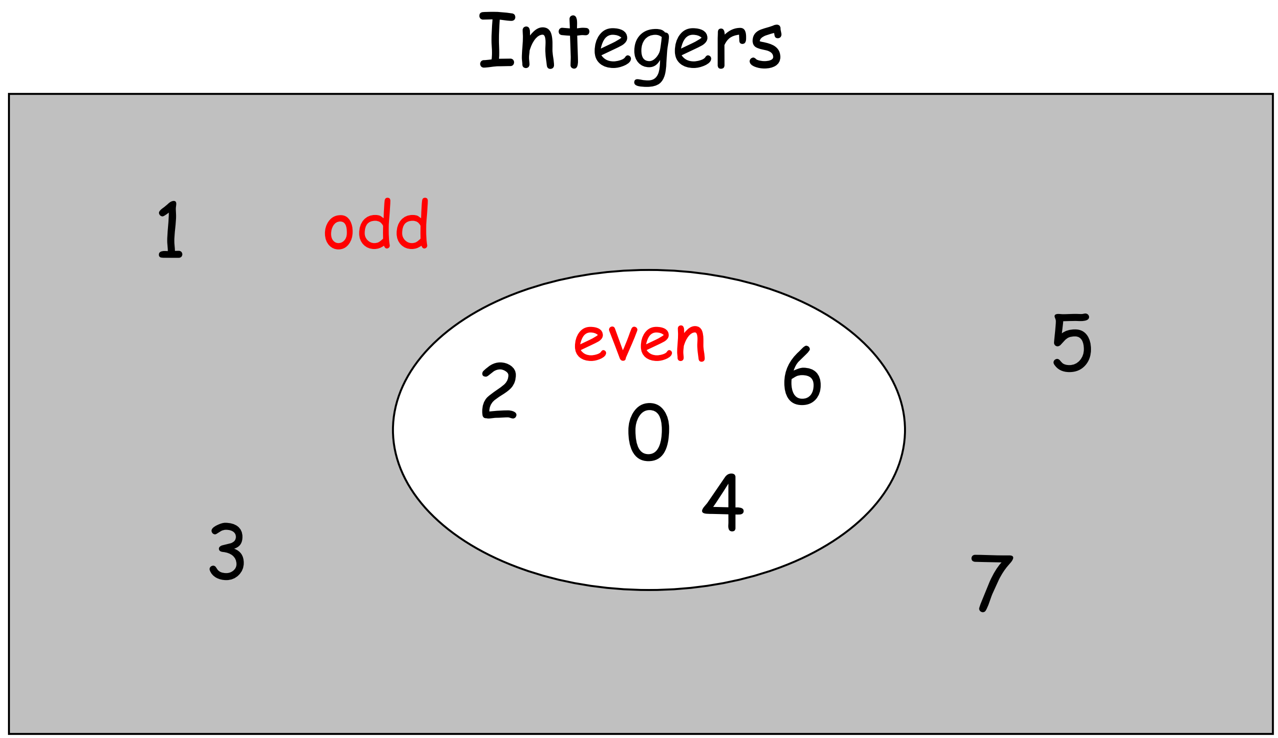
Universal set = {1, ..., 7}

$A = \{ 1, 2, 3 \}$ ⇨ $\overline{A} = \{ 4, 5, 6, 7\}$



$\overline{\overline{A}} = A$

$$\overline{\{\text{ even integers }\}} = \{\text{ odd integers }\}$$



DeMorgan's Laws

$$\overline{A \cup B} = \overline{A} \cap \overline{B}$$

$$\overline{A \cap B} = \overline{A} \cup \overline{B}$$

Empty, Null Set: $\emptyset$

$\emptyset = \{ \}$

$S \cup \emptyset = S$

$S \cap \emptyset = \emptyset$

$S - \emptyset = S$

$\emptyset - S = \emptyset$

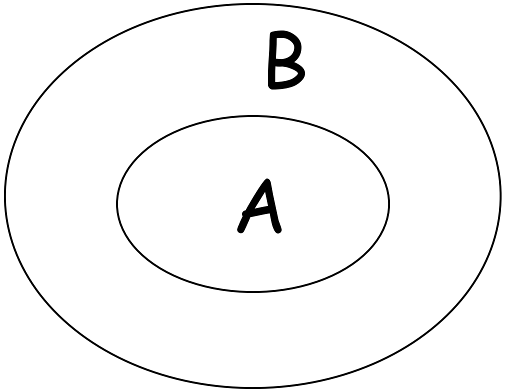
$\overline{\emptyset}$ = Universal Set

Subset

$A = \{1, 2, 3\}$ $B = \{1, 2, 3, 4, 5\}$

$A \subseteq B$

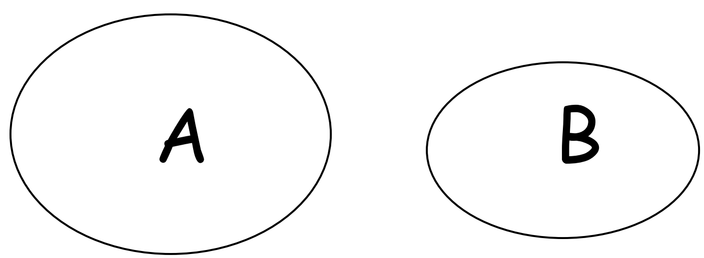
Proper Subset: $A \subset B$



Disjoint Sets

$A = \{1, 2, 3\}$ $\qquad$ $B = \{5, 6\}$

$$A \cap B = \emptyset$$



Set Cardinality

- For finite sets

$A = \{ 2, 5, 7 \}$

$|A| = 3$

(set size)

Powersets

A powerset is a set of sets

$S = \{ a, b, c \}$

Powerset of S = the set of all the subsets of S

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

Observation: $| 2^S | = 2^{|S|}$ $\quad (8 = 2^3)$

Cartesian Product

$A = \{ 2, 4 \}$ $B = \{ 2, 3, 5 \}$

$A \times B = \{ (2, 2), (2, 3), (2, 5), (4, 2), (4, 3), (4, 5) \}$

$|A \times B| = |A|\ |B|$

Generalizes to more than two sets

$A \times B \times \ldots \times Z$

SEQUENCES and TUPLES

A **sequence** of objects is a list of objects in some order.

Order and Repetition of elements matters.

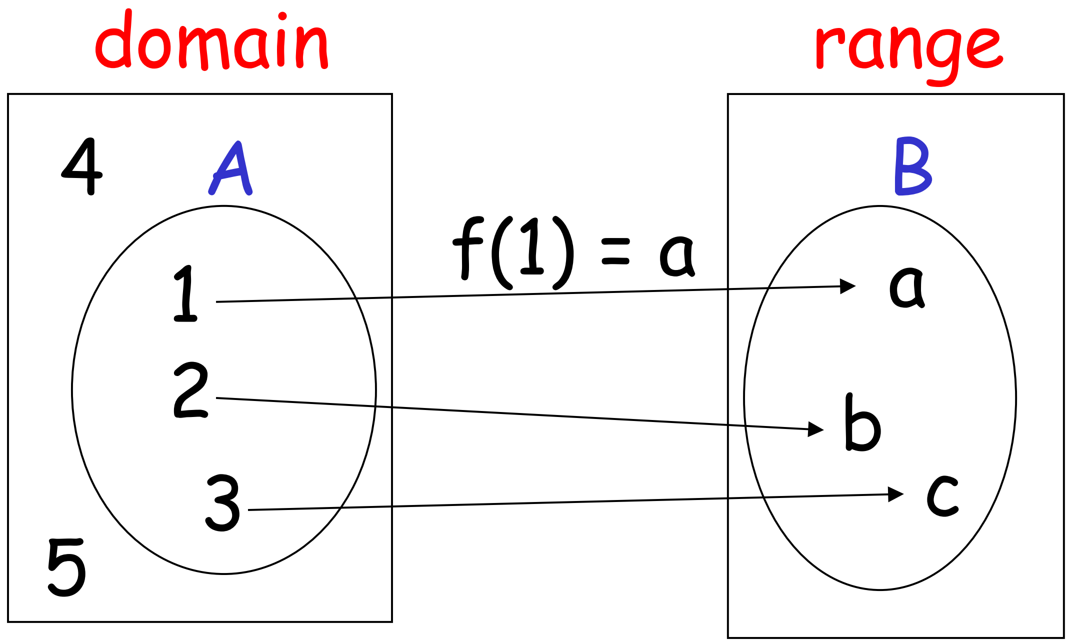
Tuples are finite sequences (k-tuple)

$$A = (1,\ \ 2,\ 3)$$

$$(1,\ \ 2,\ 3)\ \neq\ (2,\ \ 1,\ \ 3)$$

$$(1,\ \ 2,\ \ 2,\ 3)\ \neq\ (1,\ \ 2\ \ 3)$$

FUNCTIONS



$f : A \rightarrow B$

If A = domain

then f is a total function

otherwise f is a partial function

RELATIONS

$R = \{(x_1, y_1), (x_2, y_2), (x_3, y_3), \ldots\}$

$x_i \; R \; y_i$

e. g. if $R =$ '>': $2 > 1$, $3 > 2$, $3 > 1$

Equivalence Relations

- Reflexive: $x \, R \, x$
- Symmetric: $x \, R \, y \Rightarrow y \, R \, x$
- Transitive: $x \, R \, y$ and $y \, R \, z \Rightarrow x \, R \, z$

Example: $R =$ '$=$'

- $x = x$
- $x = y \Rightarrow y = x$
- $x = y$ and $y = z \Rightarrow x = z$

Equivalence Classes

For equivalence relation R

equivalence class of $x = \{y : x \, R \, y\}$

Example:

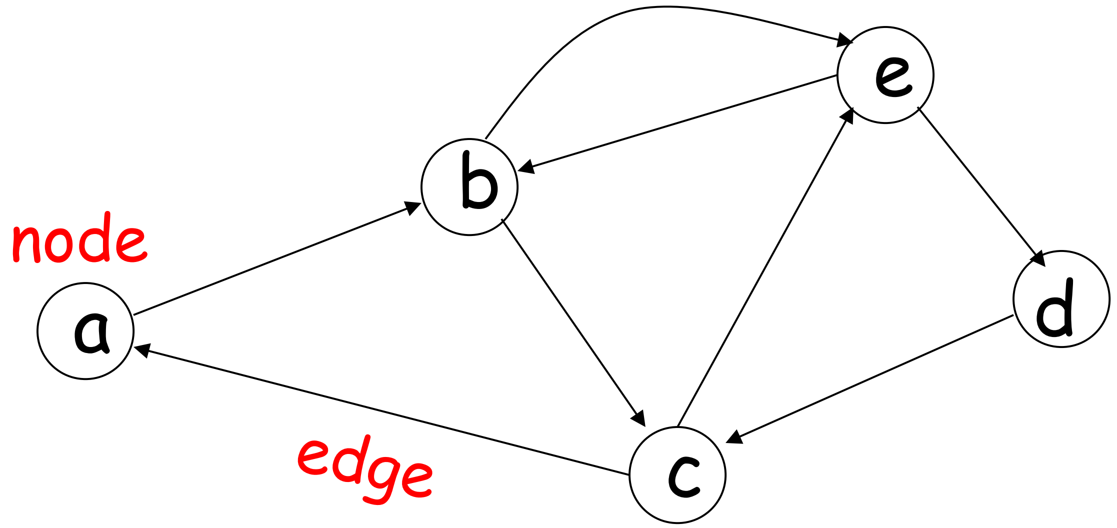
$$R = \{ (1, 1), (2, 2), (1, 2), (2, 1), (3, 3), (4, 4), (3, 4), (4, 3) \}$$

Equivalence class of $1 = \{1, 2\}$

Equivalence class of $3 = \{3, 4\}$

GRAPHS

A directed graph



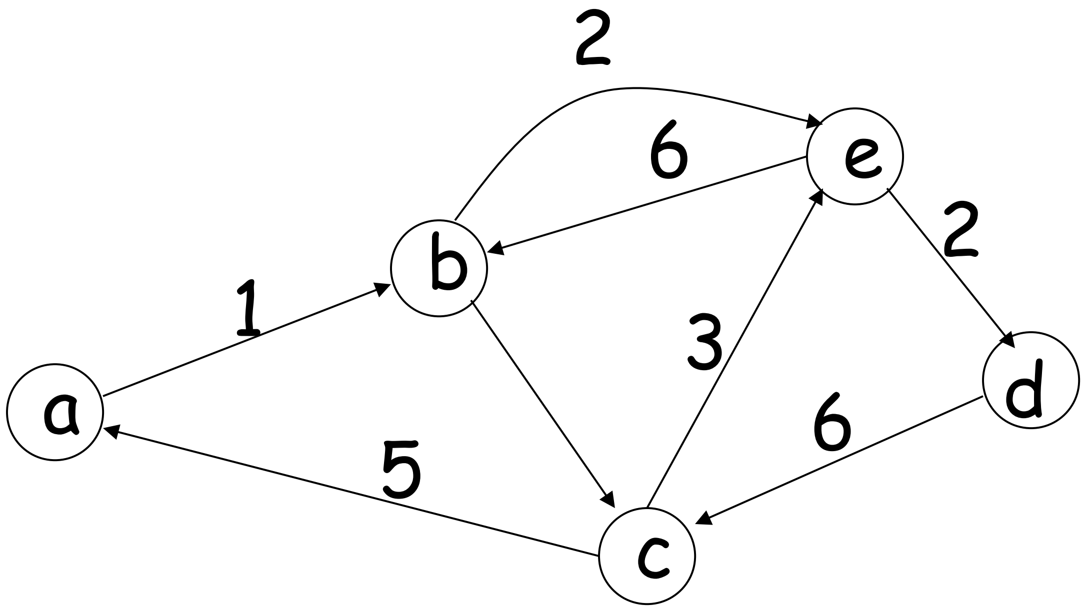
- Nodes (Vertices)

$$V = \{ a, b, c, d, e \}$$

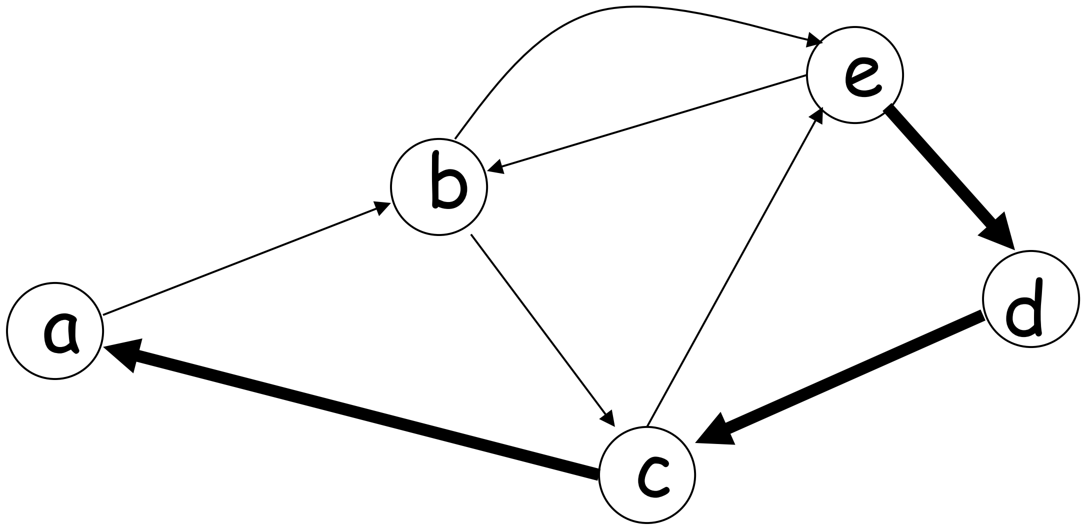
- Edges (arcs)

$$E = \{ (a,b), (b,c), (b,e),(c,a), (c,e), (d,c), (e,b), (e,d) \}$$

Labeled Graph



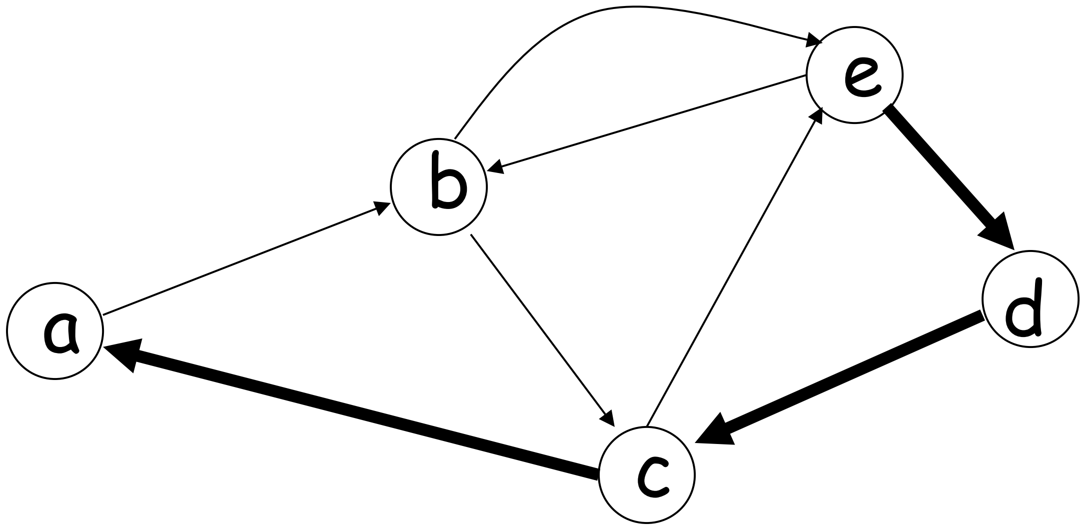
Walk



Walk is a sequence of adjacent edges

(e, d), (d, c), (c, a)

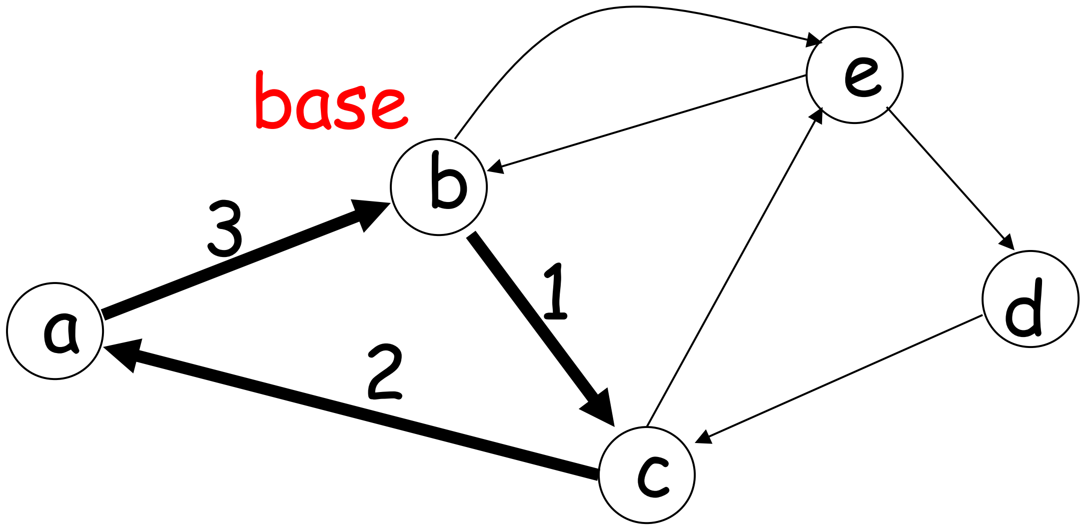
Path



Path is a walk where no edge is repeated

Simple path: no node is repeated

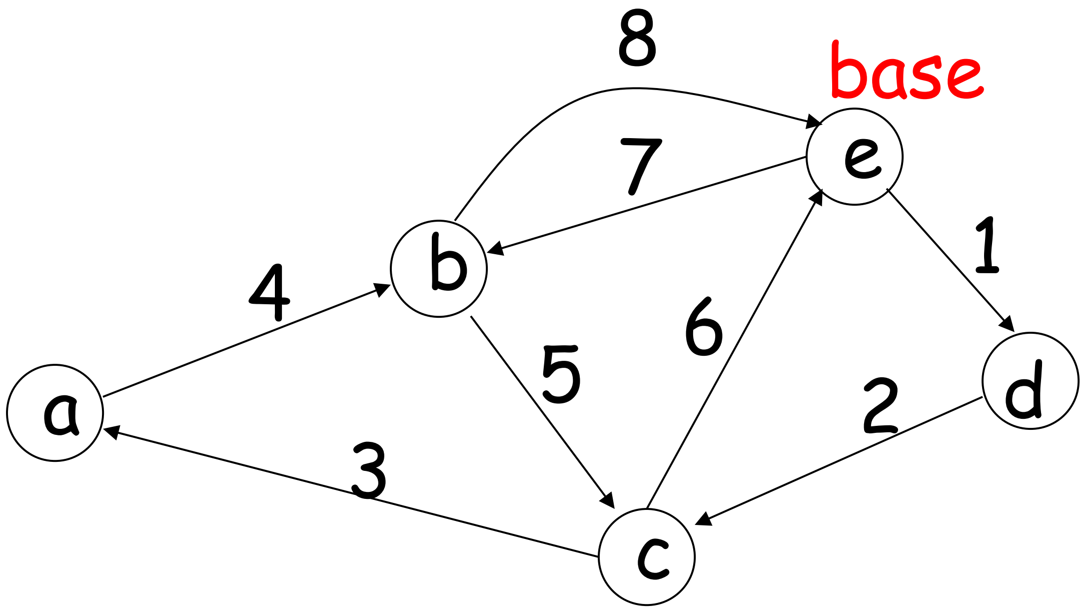
Cycle



Cycle: a walk from a node (base) to itself

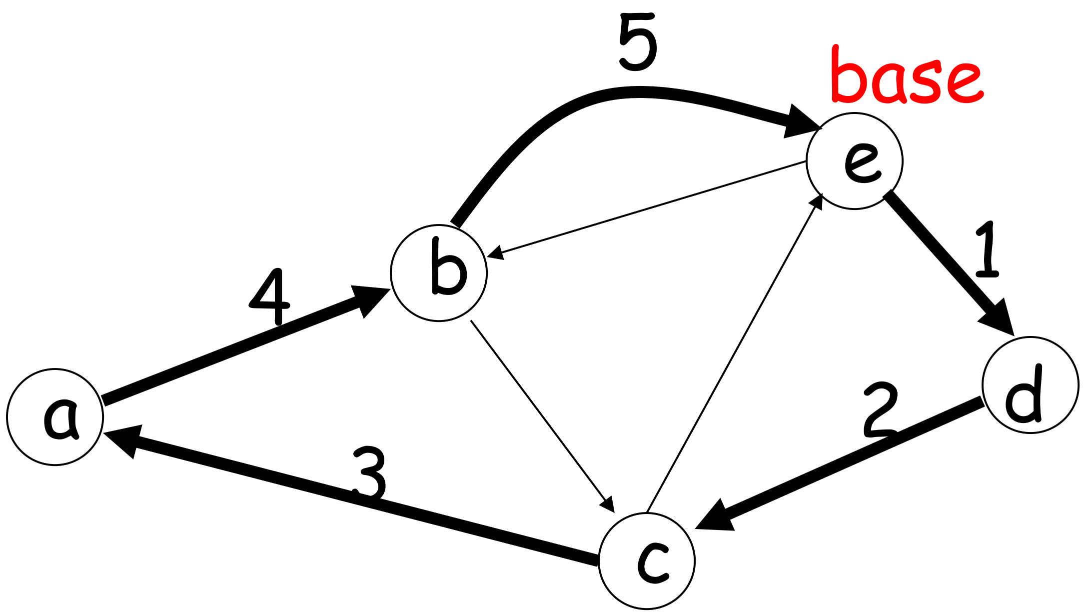
Simple cycle: only the base node is repeated

Euler Tour



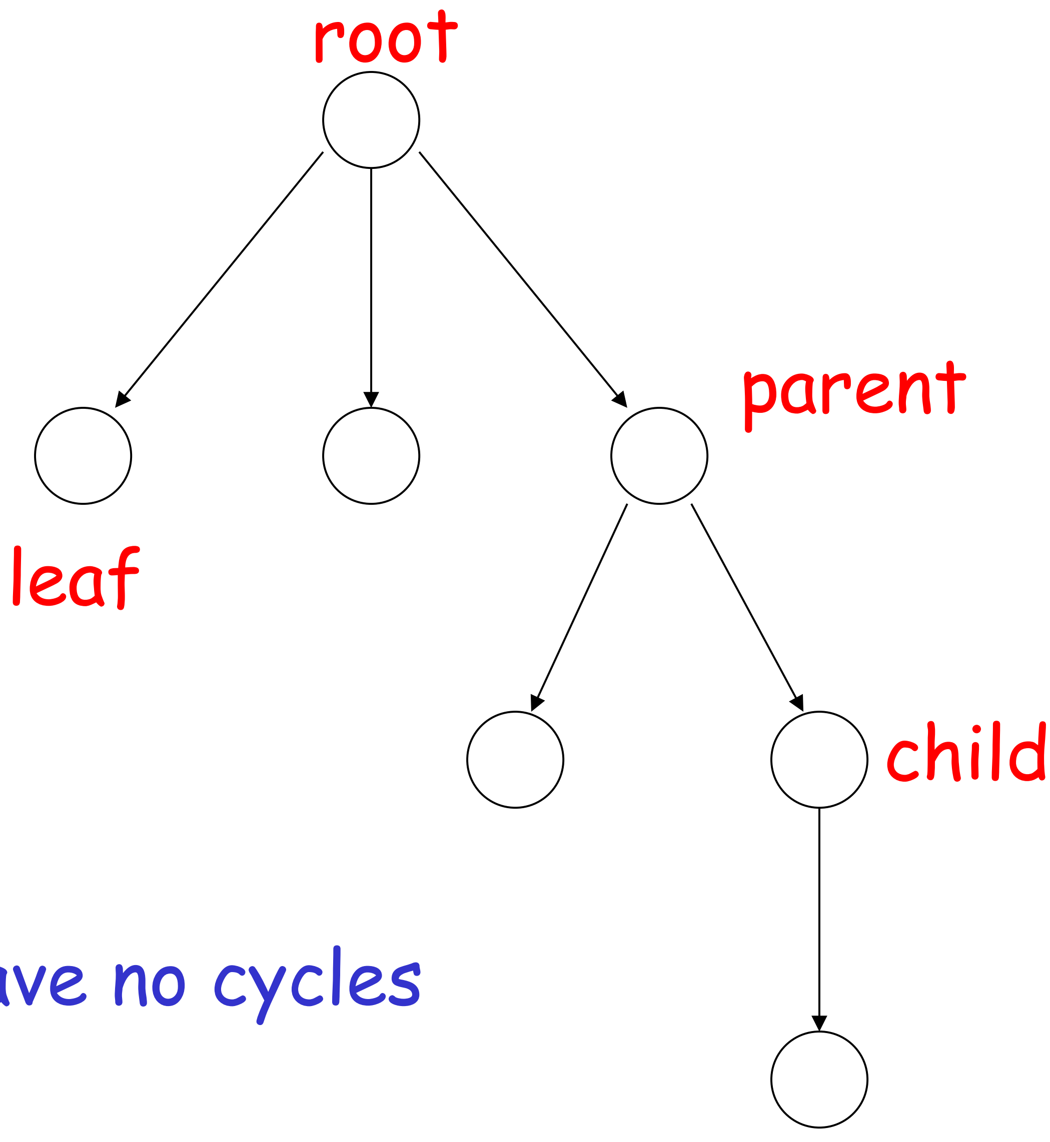
A cycle that contains each edge once

Hamiltonian Cycle

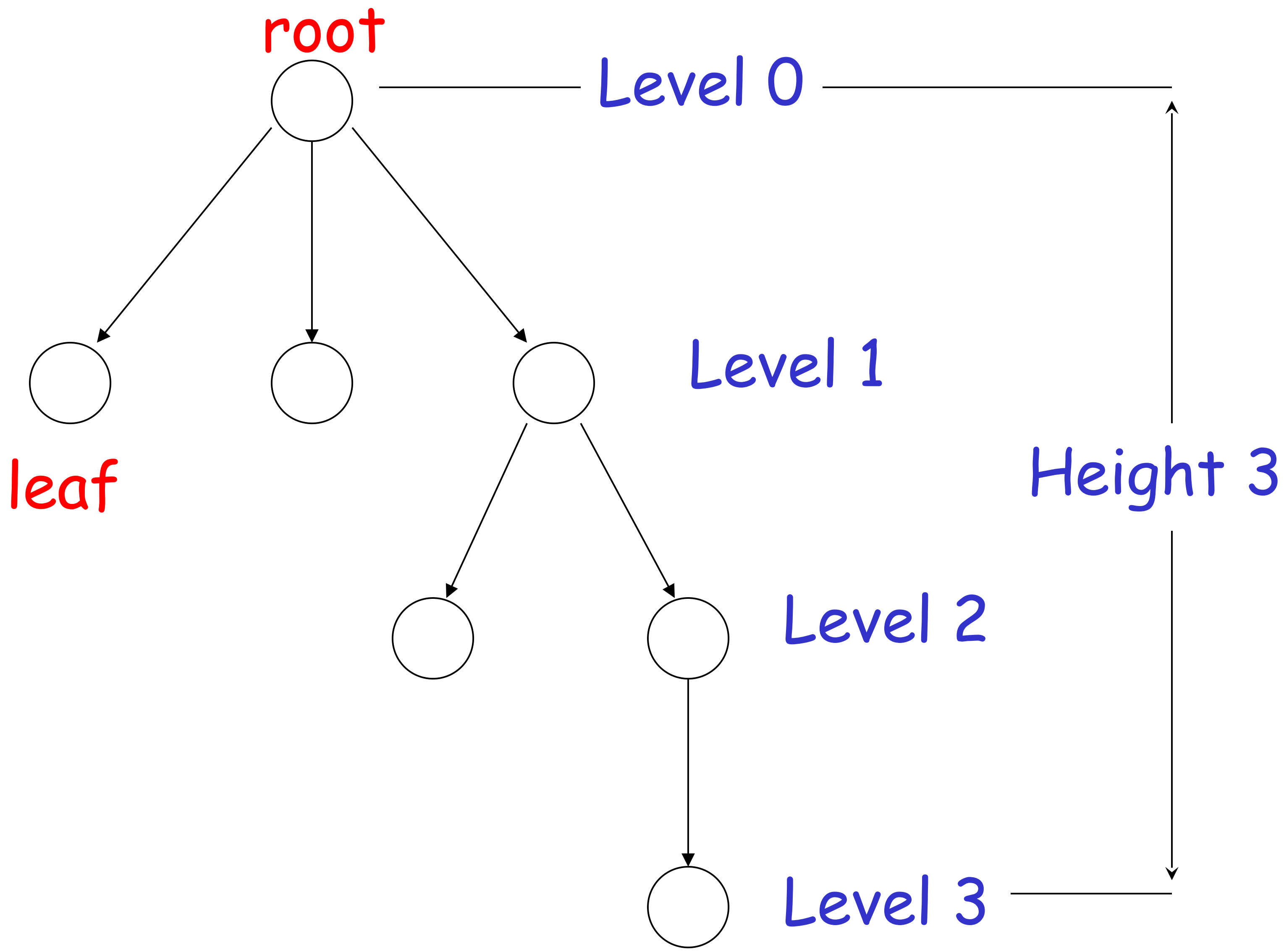


A simple cycle that contains all nodes

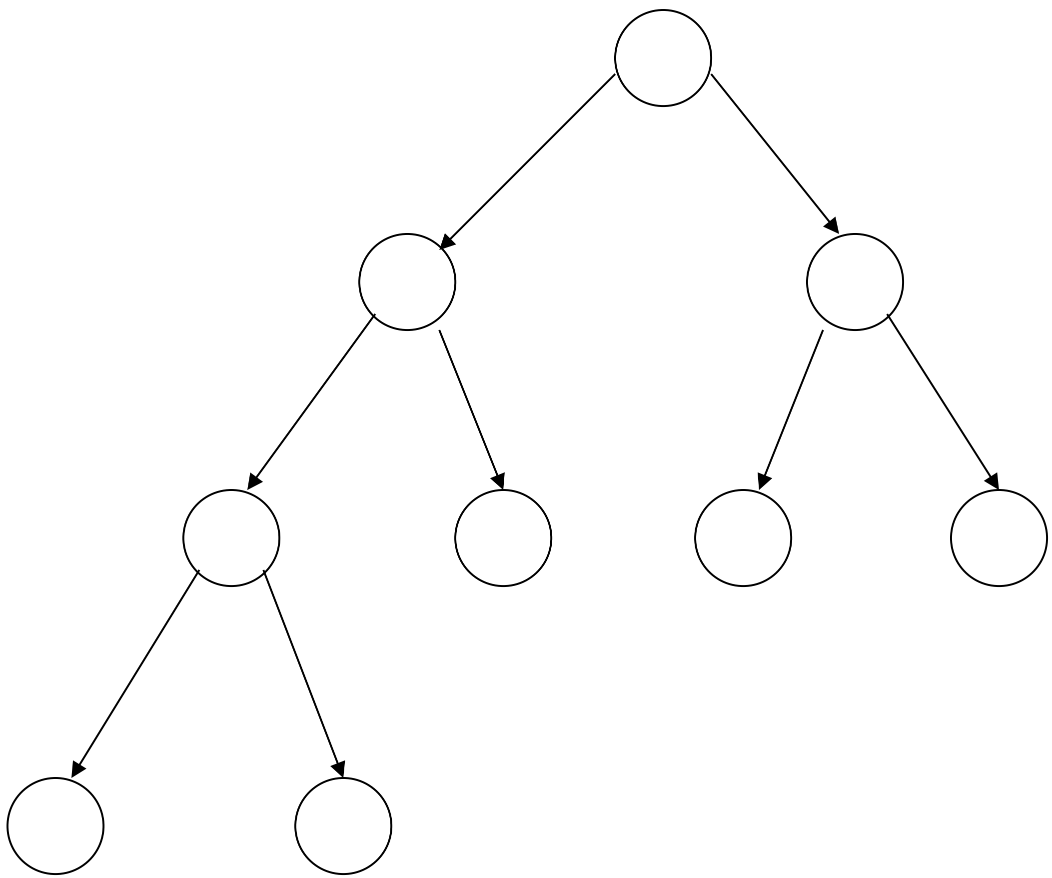
Trees



Trees have no cycles


root
Level 0
Level 1
leaf
Height 3
Level 2
Level 3

Binary Trees



PROOF TECHNIQUES

- Proof by induction
- Proof by contradiction
- Proof by construction

Induction

We have statements $P_1, P_2, P_3, \ldots$

If we know

- for some b that $P_1, P_2, \ldots, P_b$ are true
- for any $k >= b$ that

  $P_1, P_2, \ldots, P_k$ imply P_{k+1}

Then

Every P_i is true

Proof by Induction

- Inductive basis

  Find P_1, P_2, …, P_b which are true

- Inductive hypothesis

  Let's assume P_1, P_2, …, P_k are true,

  for any k >= b

- Inductive step

  Show that P_{k+1} is true

Example

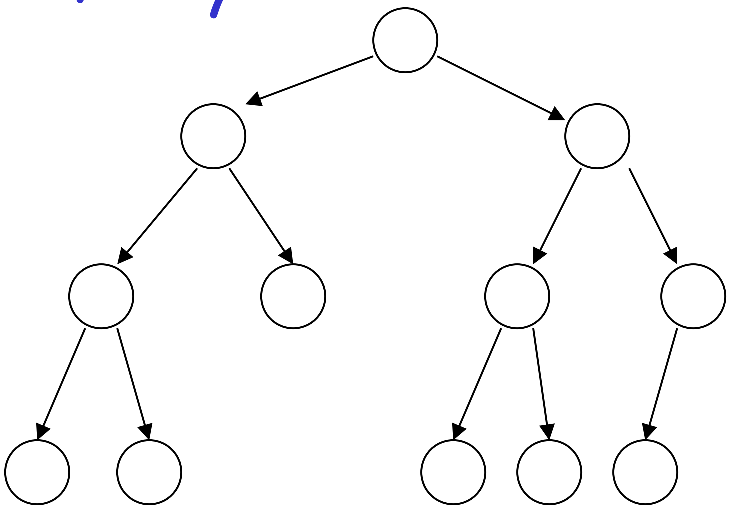
Theorem: A binary tree of height n

has at most 2^n leaves.

Proof by induction:

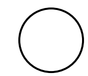
let $L(i)$ be the maximum number of

leaves of any subtree at height i



We want to show: $L(i) <= 2^i$

- Inductive basis

  $L(0) = 1$ (the root node) 

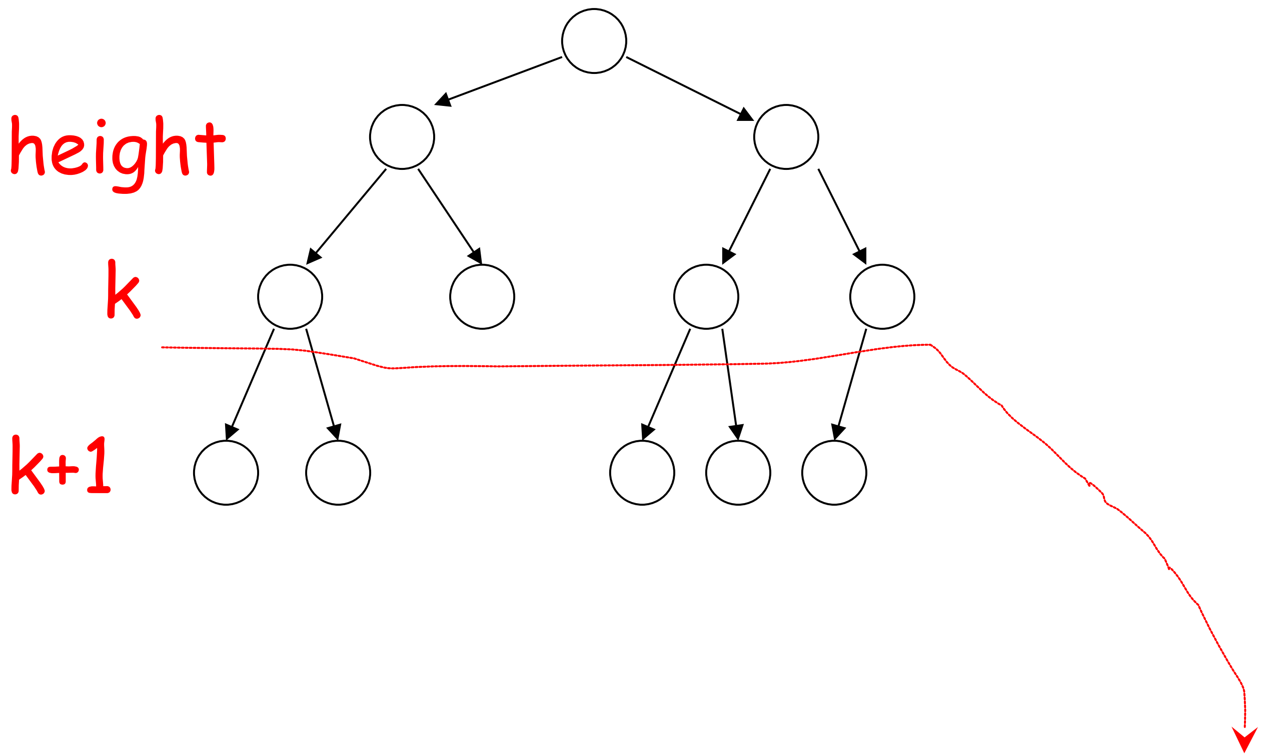
- Inductive hypothesis

  Let's assume $L(i) <= 2^i$ for all $i = 0, 1, \ldots, k$

- Induction step

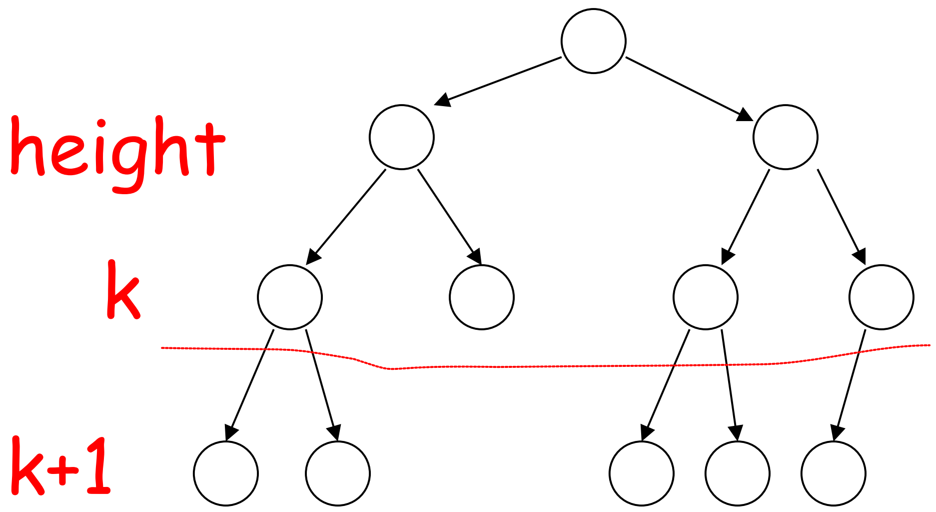
  we need to show that $L(k + 1) <= 2^{k+1}$

Induction Step



From Inductive hypothesis: $L(k) <= 2^k$

Induction Step



$L(k) <= 2^k$

$$L(k+1) <= 2 * L(k) <= 2 * 2^k = 2^{k+1}$$

(we add at most two nodes for every leaf of level k)

Proof by Contradiction

We want to prove that a statement P is true

- we assume that P is false
- then we arrive at an incorrect conclusion
- therefore, statement P must be true

Example

Theorem: $\sqrt{2}$ is not rational

Proof:

Assume by contradiction that it is rational

$\sqrt{2} = n/m$

n and m have no common factors

We will show that this is impossible

$\sqrt{2} = n/m \Rightarrow 2m^2 = n^2$

Therefore, n^2 is even $\Rightarrow$ n is even

$n = 2k$

$2m^2 = 4k^2 \Rightarrow m^2 = 2k^2 \Rightarrow$ m is even

$m = 2p$

Thus, m and n have common factor 2

Contradiction!