Bilingual

Formal Languages and Automata Theory
-Teori Bahasa dan Otomata-

Minggu Ke-1

Departemen Informatika, Fakultas Sains dan Matematika
Universitas Diponegoro

Formal Languages and Automata Theory
-Teori Bahasa dan Otomata-

Target/Aims

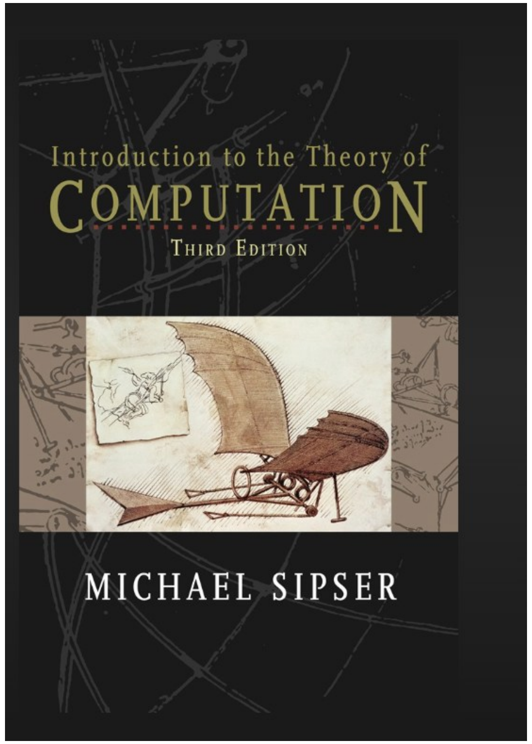
- **Sub CPMK05-2 (1):**
- Mampu menjelaskan (C2) konsep simbol, kata, tata bahasa, dan jenis-jenis otomata serta Formal Proof terhadap Regular Language
- **Bahan Kajian / Materi Pembelajaran:**
- Teori Finite Automata : konsep simbol, kata dan tata bahasa, dan jenis-jenis otomata
- **Indikator :**
- Ketepatan mendefinisikan simbol, kata, dan tata bahasa dalam Finite Automata

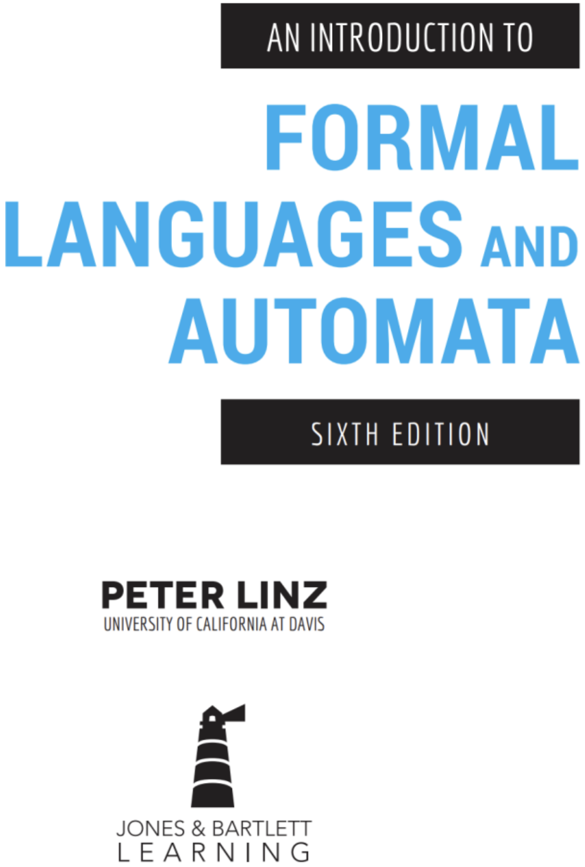
Overview/ Outline

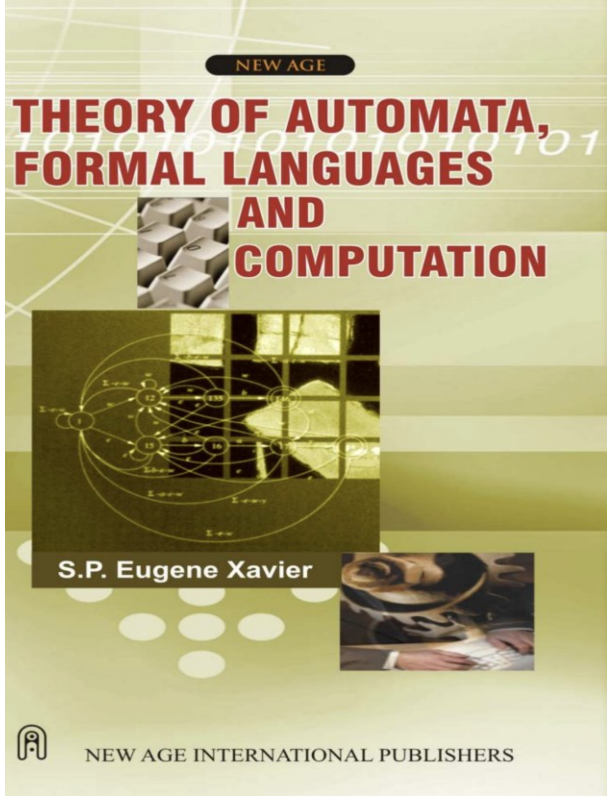
- Background– (Motivation and Various Type of Automata)
- Mathematical Preliminaries (Notation and Terminology)

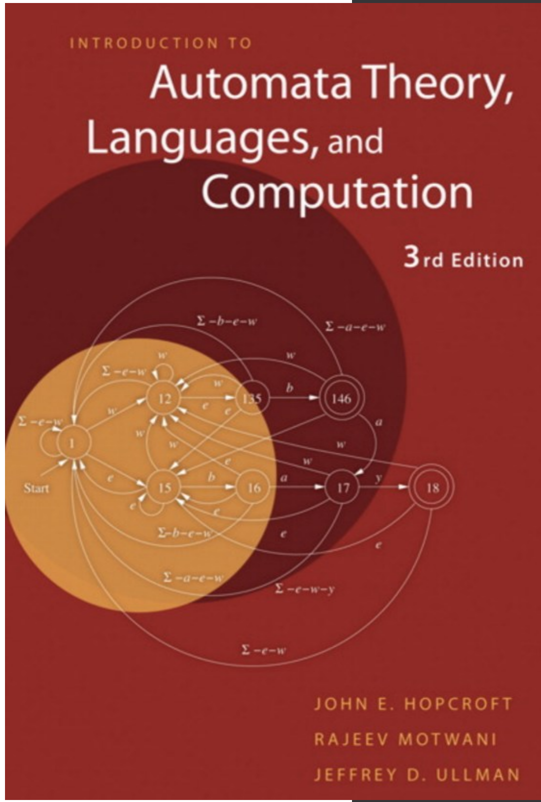
- Languages
- - String, Alphabet, Symbol, String Operations
- Overview Grammar

References









Silahkan cek di MS Teams, tab ‘Files’

Motivation
Why do we study TBO?

<u>Theory of Computation</u>

What are the fundamental capabilities and limitations of computers?

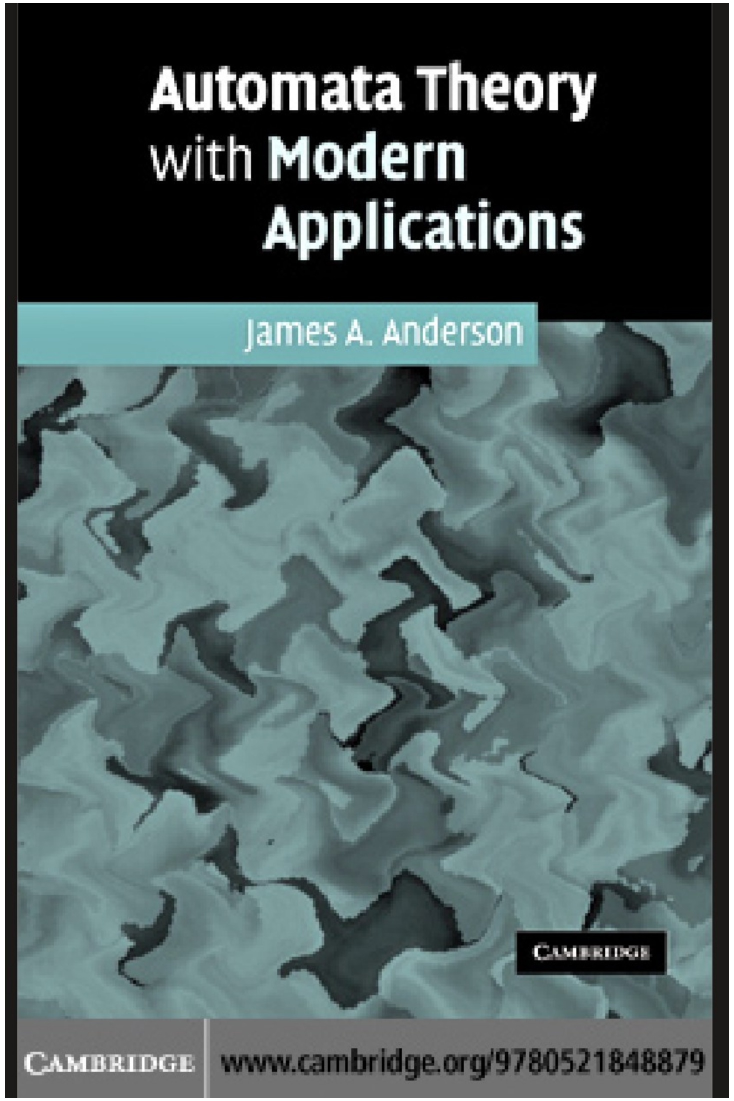
Computability theory	**Complexity theory**	**Automata theory**

Motivation
Why do we study TBO?

Automata is essential in programming language design and compiler construction.

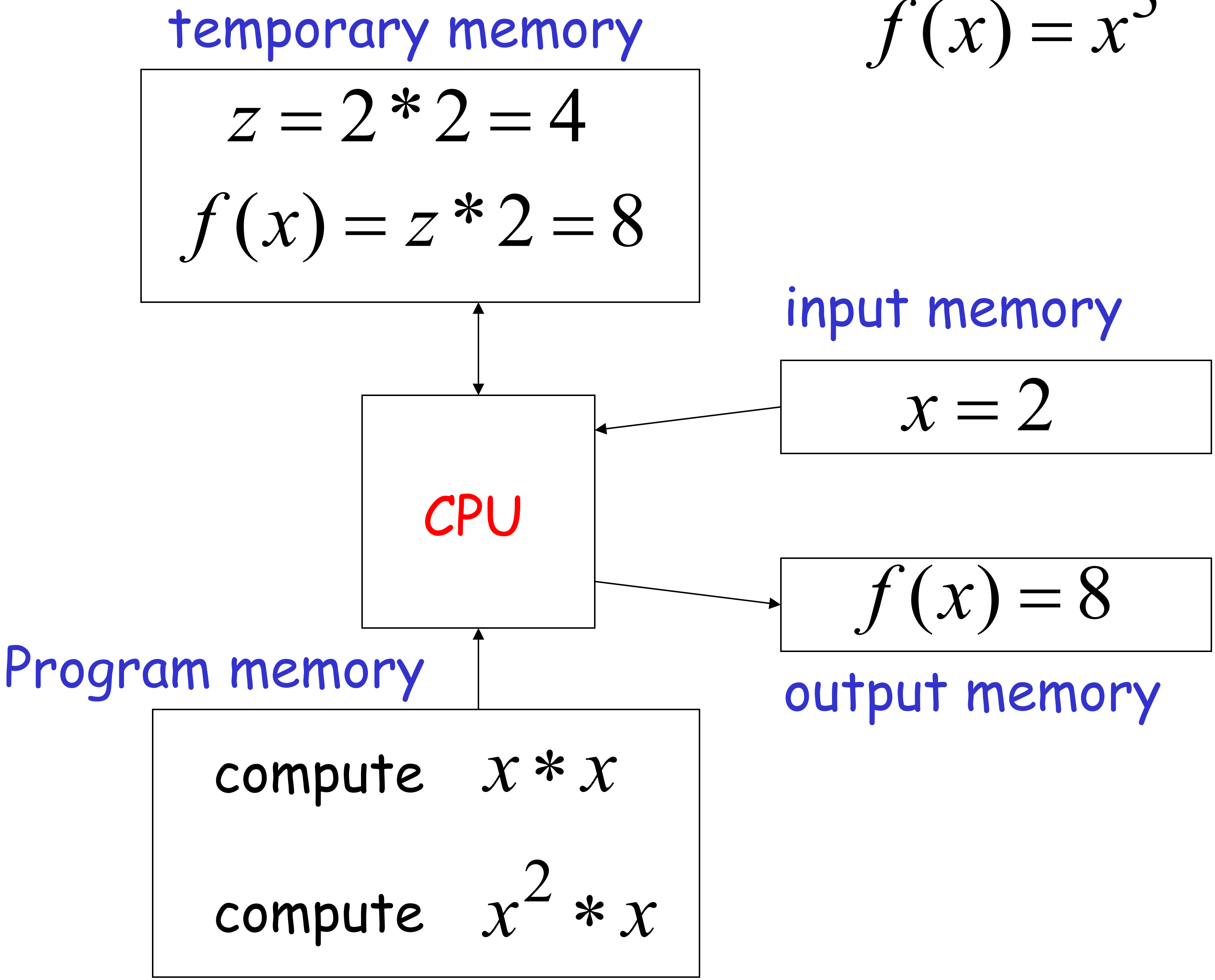
Automata theory is applied in NLP (natural language processing) and speech recognition for parsing and understanding language

In Bioinformatics, automata theory helps analyze DNA sequences and identify patterns



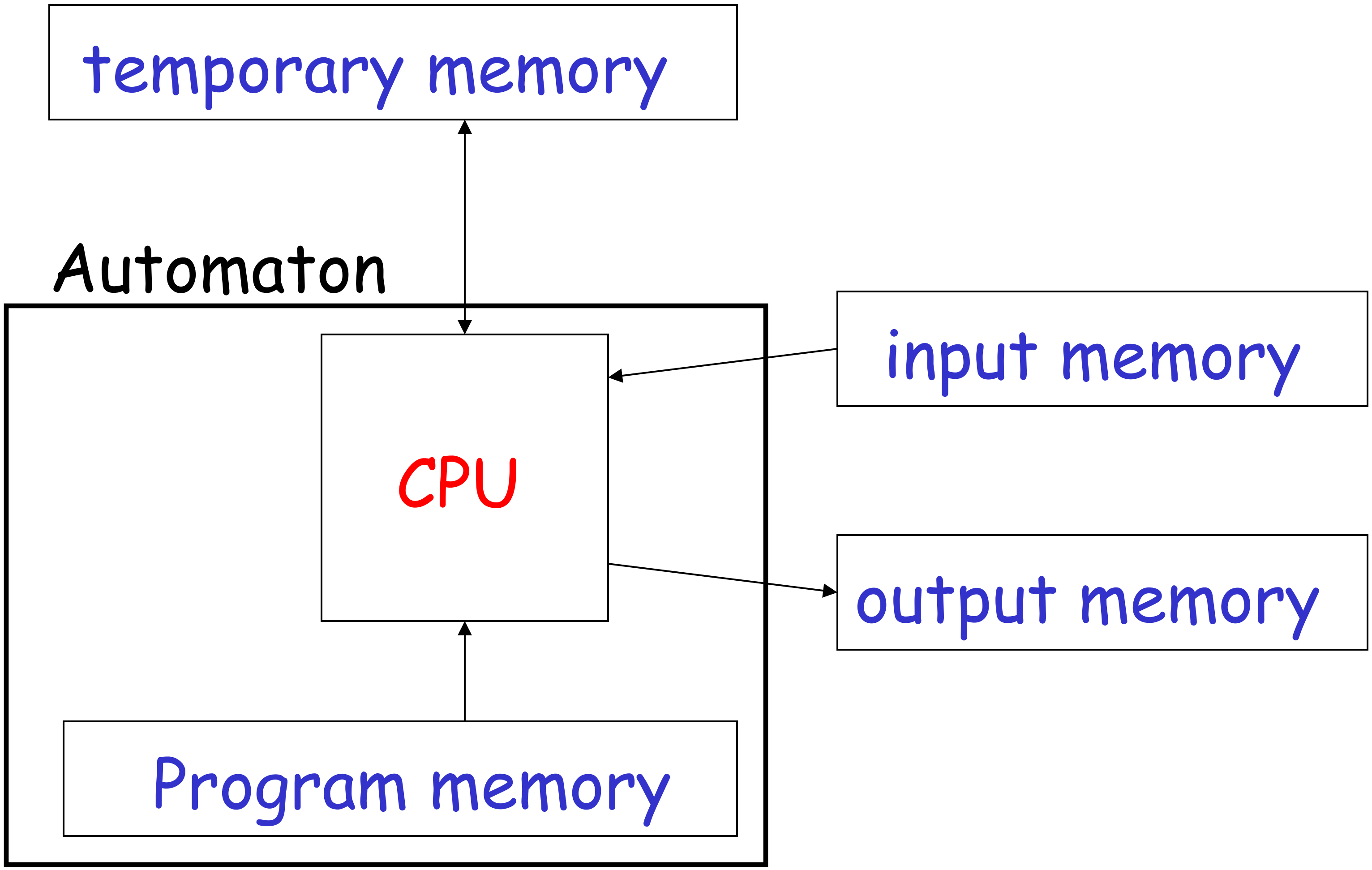
Models of Computation

$$f(x) = x^3$$



temporary memory
$z = 2*2 = 4$
$f(x) = z*2 = 8$
input memory
$x = 2$
CPU
$f(x) = 8$
output memory
Program memory
compute $x*x$
compute x^2*x

Automaton



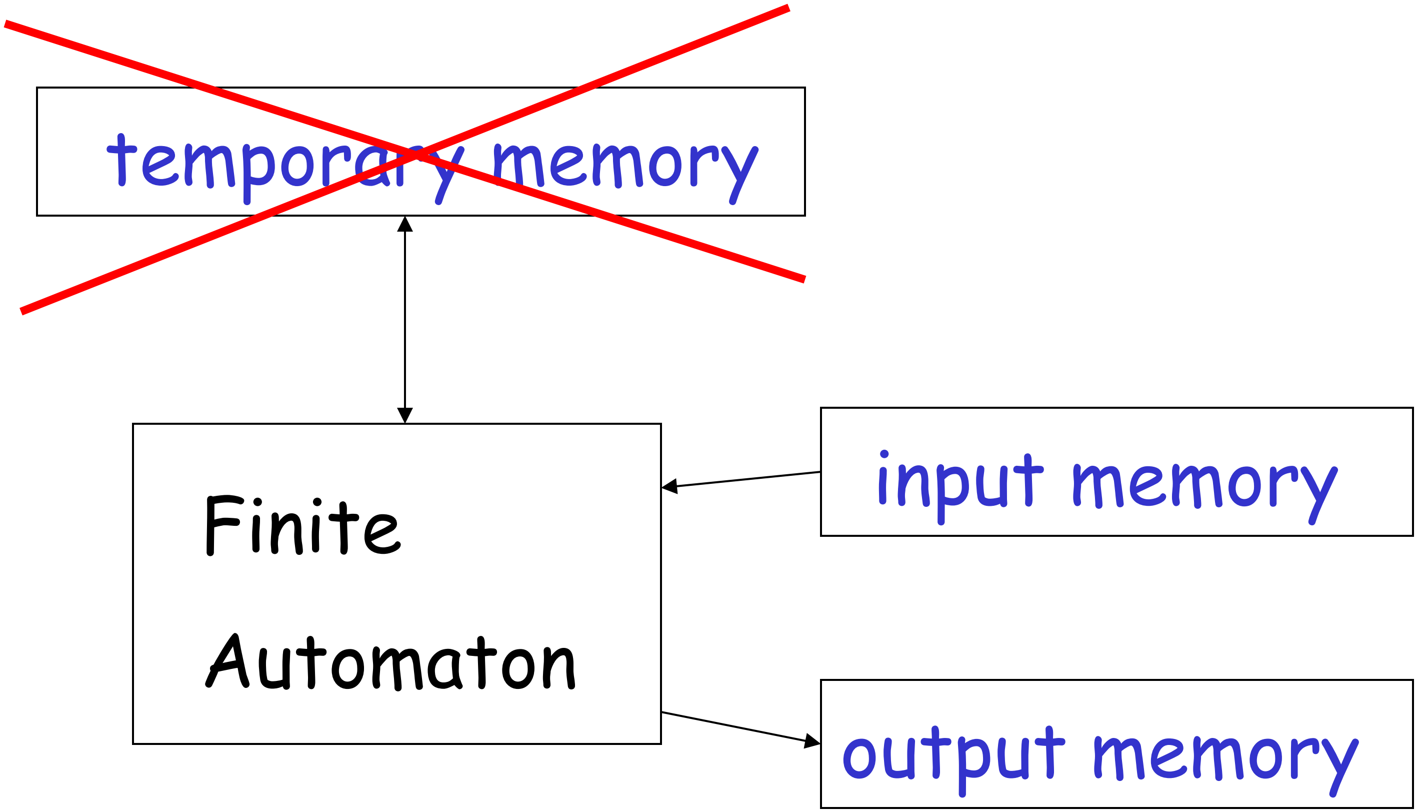
temporary memory
Automaton
CPU
input memory
output memory
Program memory

Different Kinds of Automata

Automata are distinguished by the temporary memory

- **Finite Automata:** no temporary memory
- **Pushdown Automata:** stack
- **Turing Machines:** random access memory

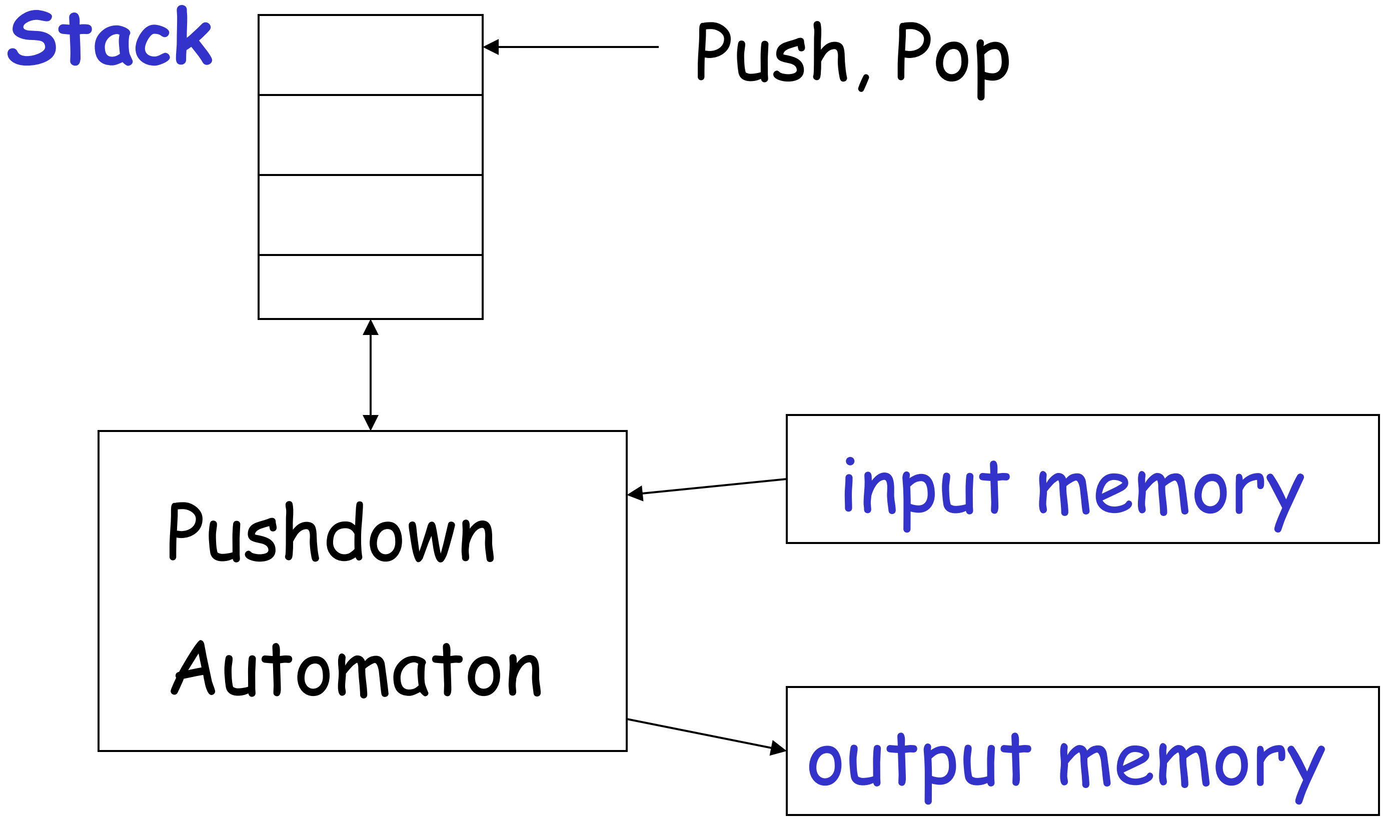
Finite Automaton



Example: Vending Machines

(small computing power)

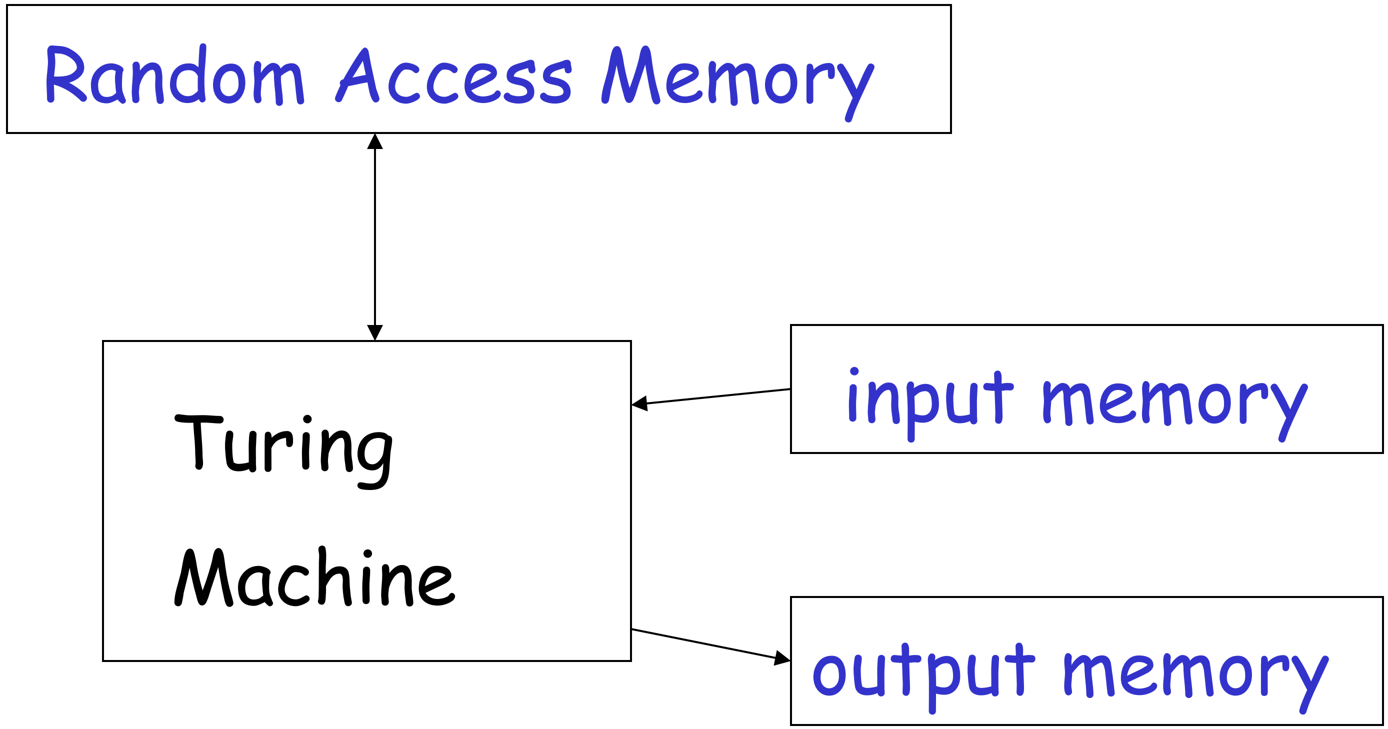
Pushdown Automaton



Example: Compilers for Programming Languages

(medium computing power)

Turing Machine



Examples: Any Algorithm

(highest computing power)

Power of Automata

Finite Automata < Pushdown Automata < Turing Machine

Less power → More power

Solve more computational problems

Mathematical Preliminaries

Mathematical Preliminaries

- Sets
- 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

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

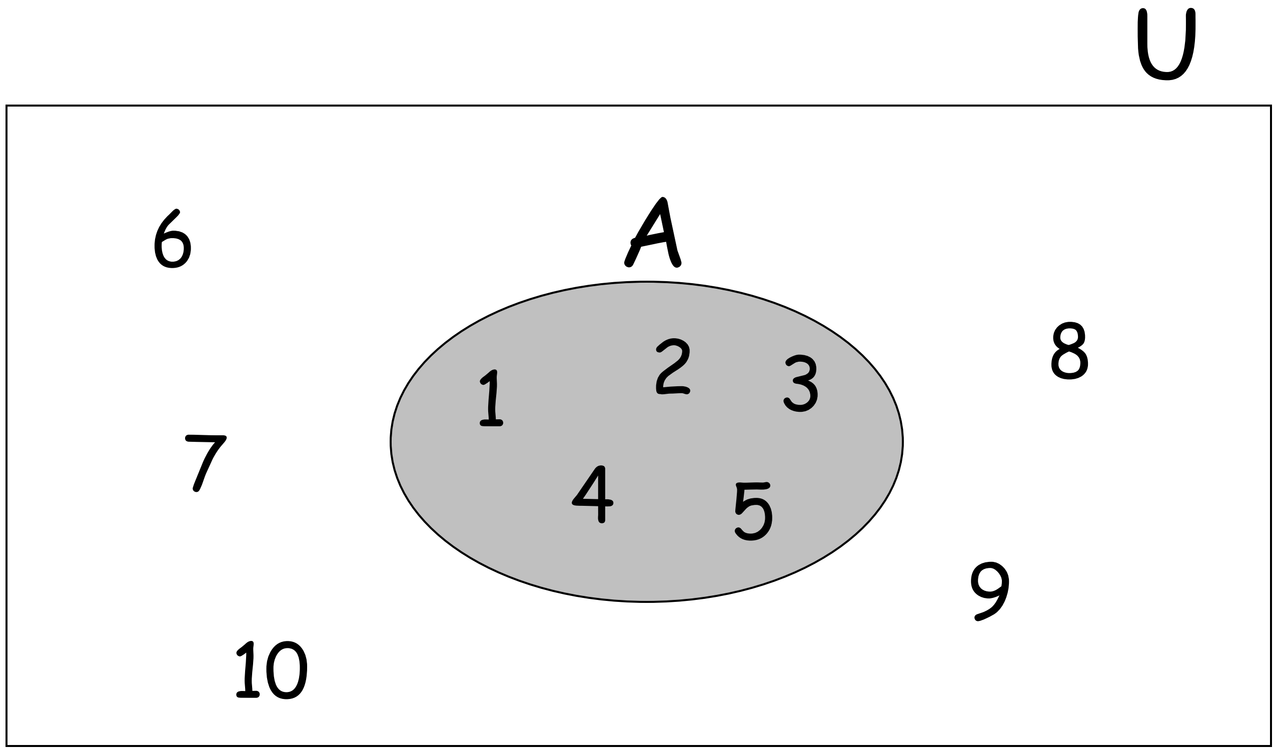
$C = \{ a, b, \dots, k \}$ → *finite set*

$S = \{ 2, 4, 6, \dots \}$ → *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

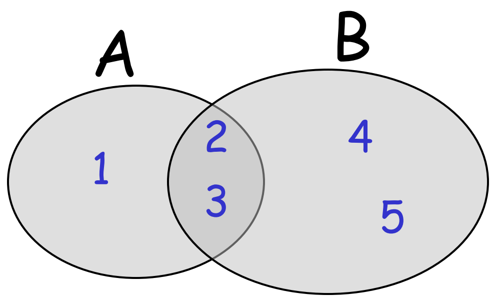
$U = \{ 1 , \ldots , 10 \}$

Set Operations

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

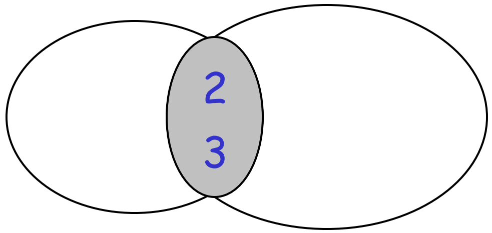
- Union

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



- Intersection

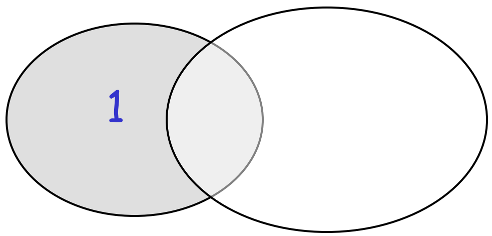
$$A \cap B = \{2, 3\}$$



- Difference

$$A - B = \{1\}$$

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

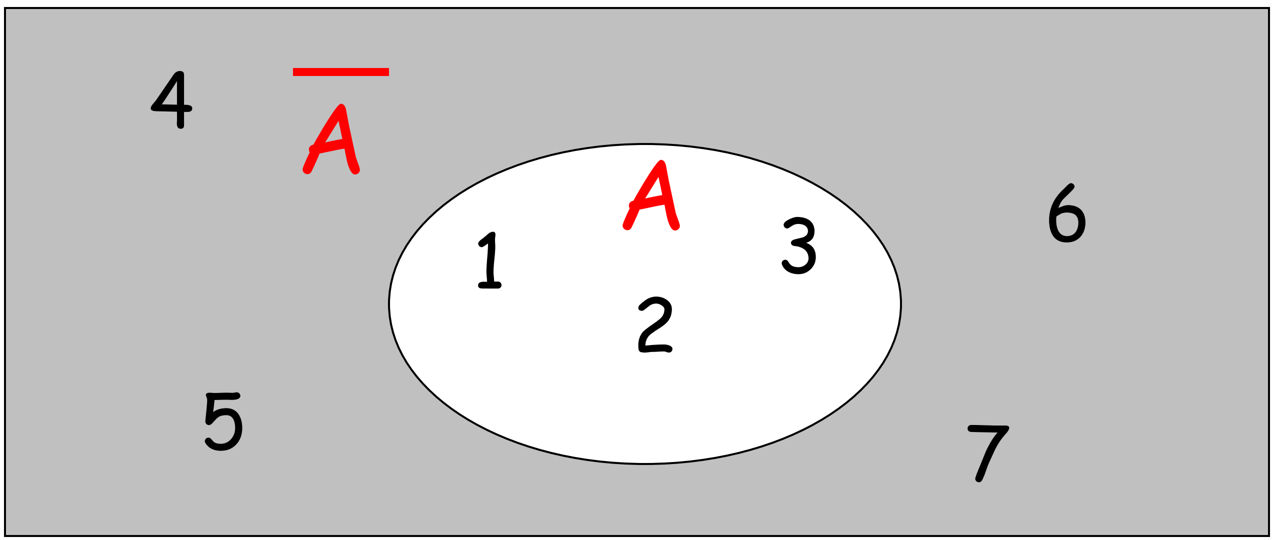


Venn diagrams

- Complement

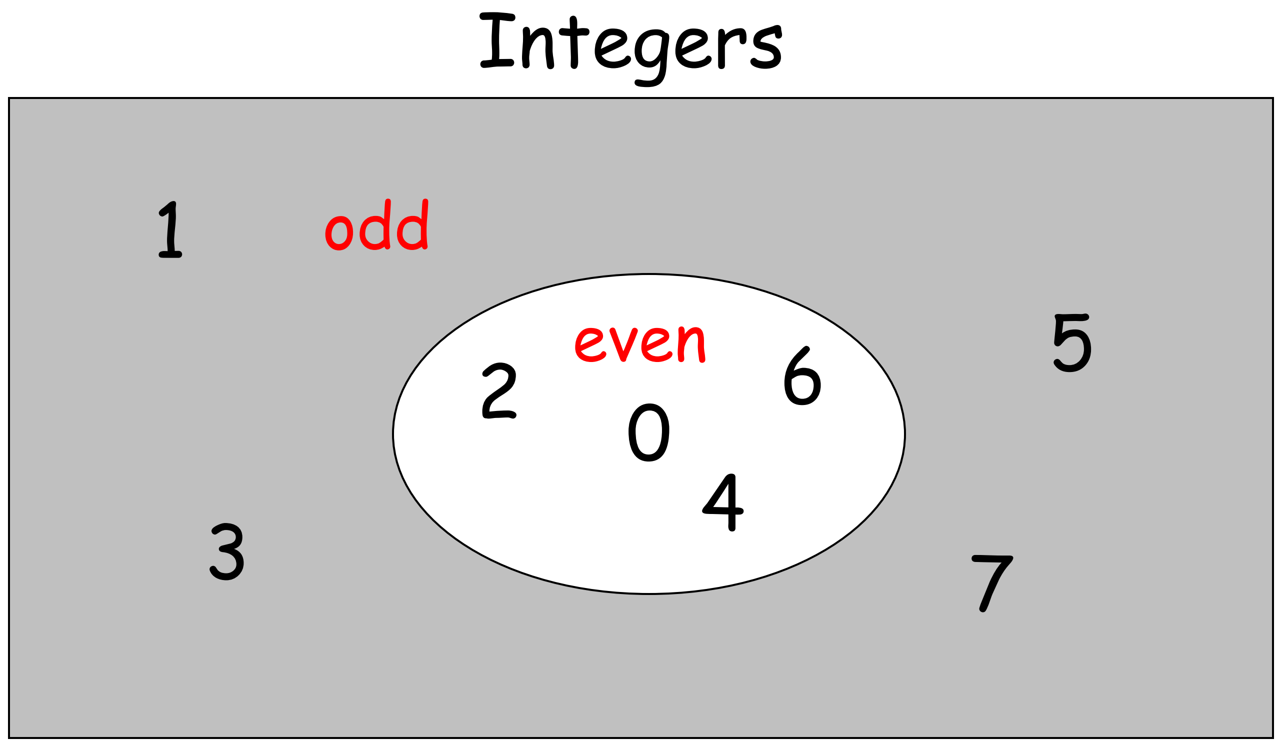
Universal set = {1, ..., 7}

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



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

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



Exercise (1-2)

1. With $S1 = \{2, 3, 5, 7\}$, $S2 = \{2, 4, 5, 8, 9\}$, and $U = \{1 : 10\}$, compute $\neg S1 \cup S2$.
2. With $S1 = \{2, 3, 5, 7\}$ and $S2 = \{2, 4, 5, 8, 9\}$, compute $S1 \times S2$ and $S2 \times S1$.
3. For $S = \{2, 5, 6, 8\}$ and $T = \{2, 4, 6, 8\}$, compute $|S \cap T| + |S \cup T|$.

DeMorgan's Laws

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

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

Exercise (2-2)

4. Prove DeMorgan's laws, Equations (1.2) and (1.3), by showing that if an element x is in the set on one side of the equality, then it must also be in the set on the other side of the equality.
5. Show that for all sets S and T, $S - T = S \cap \neg T$.

Empty, Null Set: $\emptyset$

$\emptyset = \{\ \}$

$S \cup \emptyset = S$

$S \cap \emptyset = \emptyset$

$S - \emptyset = S$

$\emptyset - S = \emptyset$

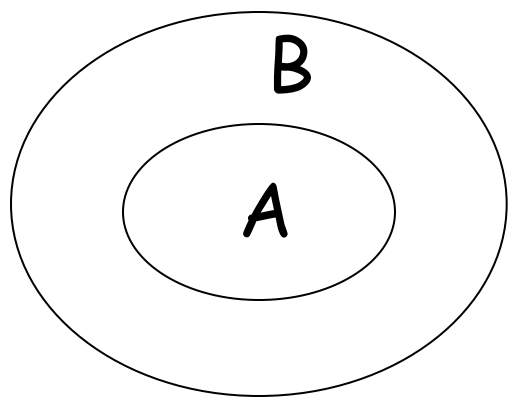
$\overline{\emptyset}$ = Universal Set

Subset

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

$$A \subseteq B$$

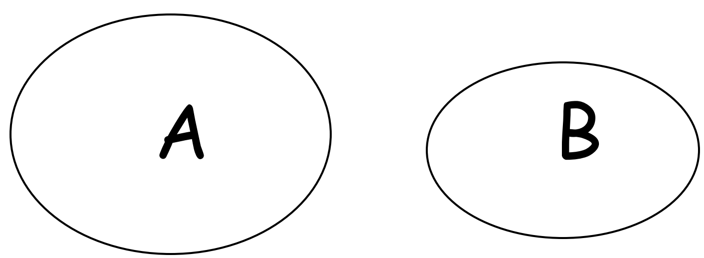
Proper Subset: $A \subset B$



Disjoint Sets

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

$$A \cap B = \emptyset$$



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

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

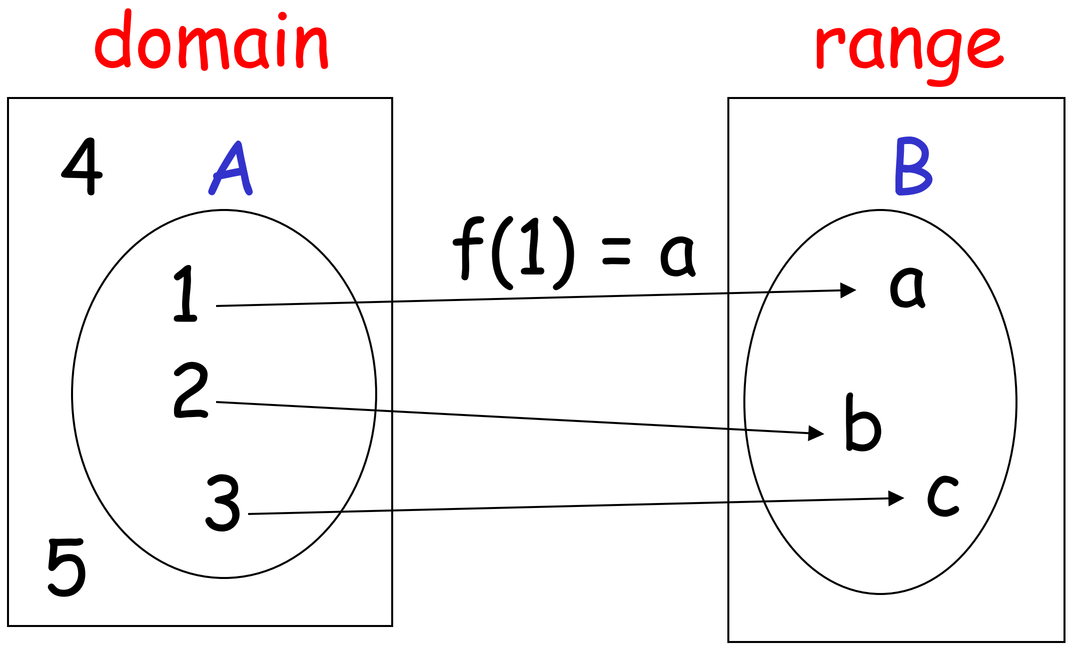
$A \times B = \{ (2, 2), (2, 3), (2, 5), (4, 2), (4, 3), (4, 5) \}$

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

Generalizes to more than two sets

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

FUNCTIONS



f : A -> 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:

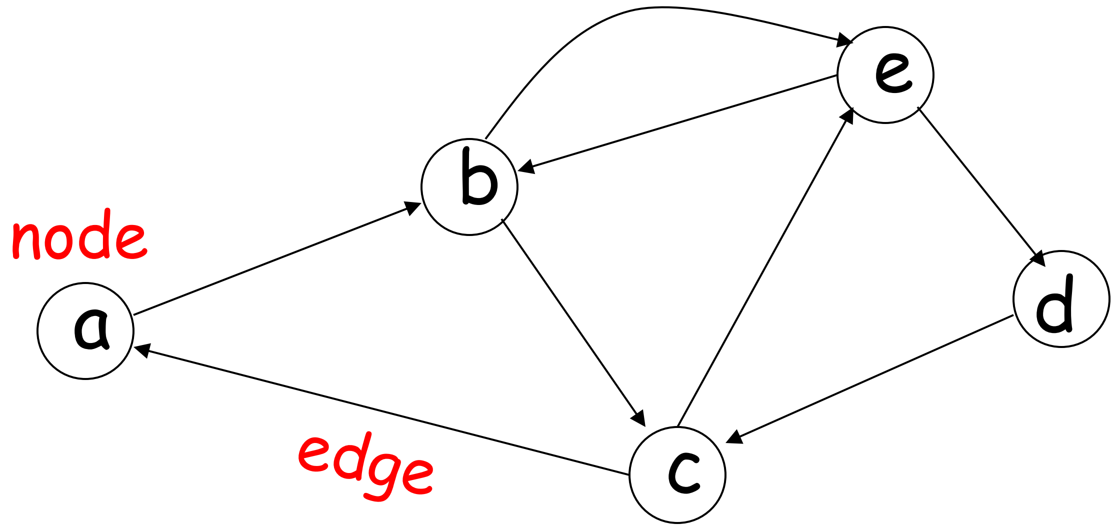
$$R = \{ (1, 1), (2, 2), (1, 2), (2, 1),$$
$$(3, 3), (4, 4), (3, 4), (4, 3) \}$$

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

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

GRAPHS

A directed graph



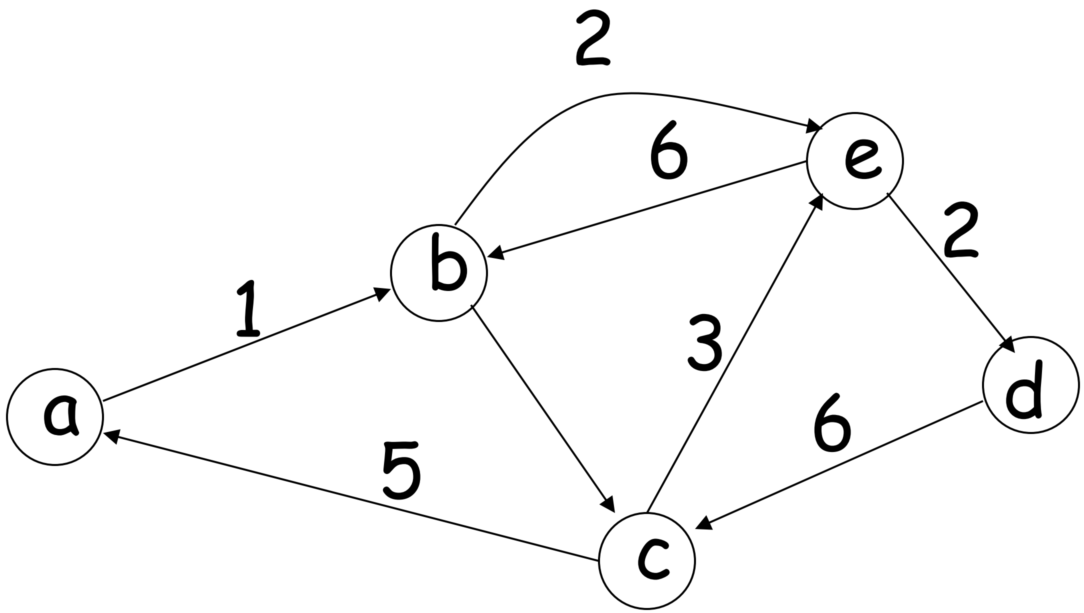
- Nodes (Vertices)

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

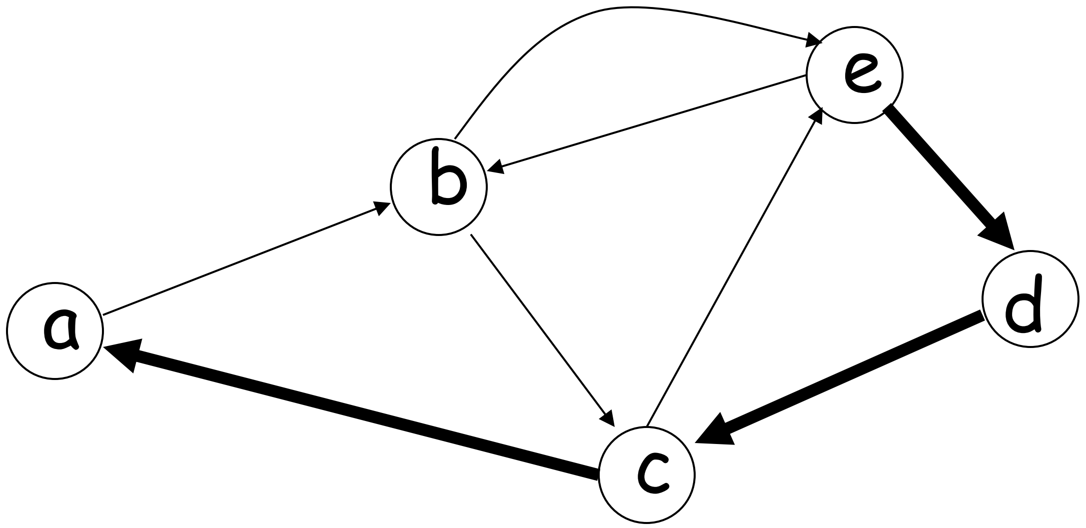
- Edges

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

Labeled Graph



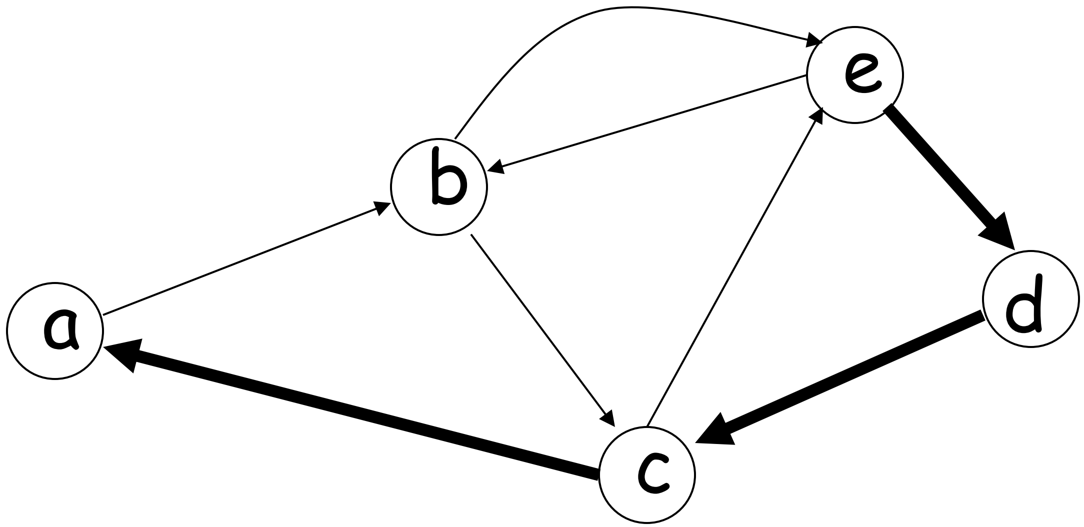
Walk



Walk is a sequence of adjacent edges

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

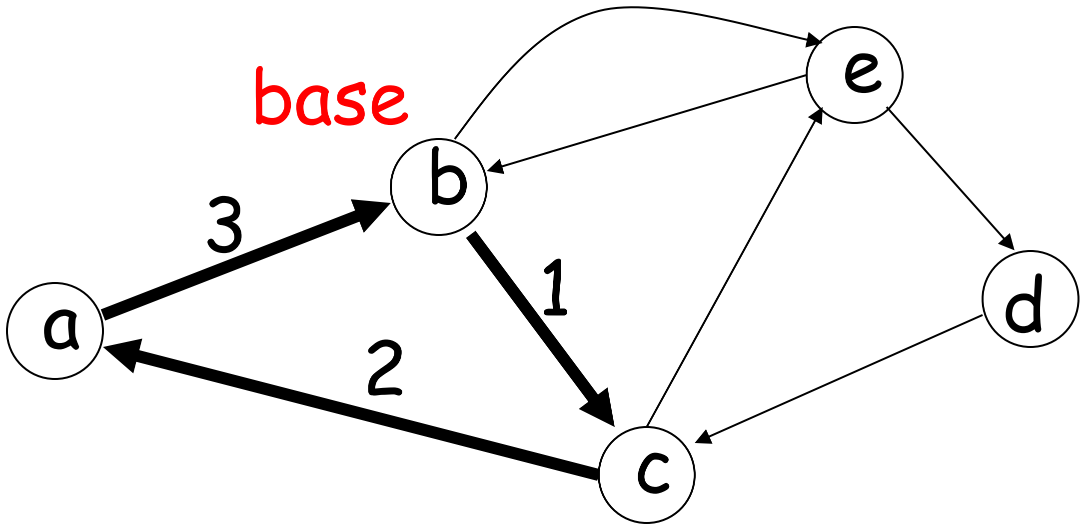
Path



Path is a walk where no edge is repeated

Simple path: no node is repeated

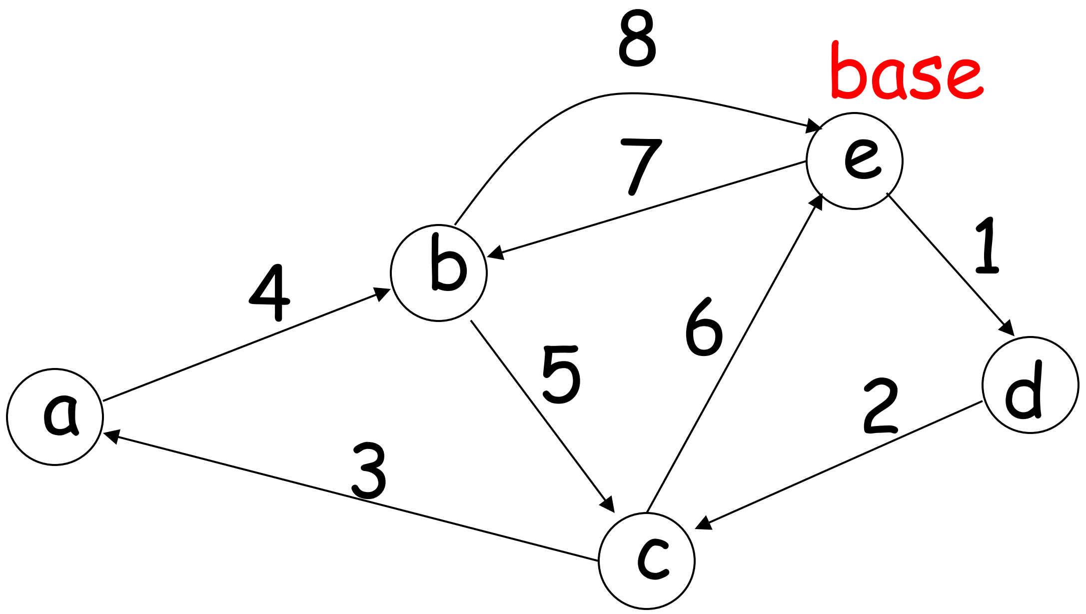
Cycle



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

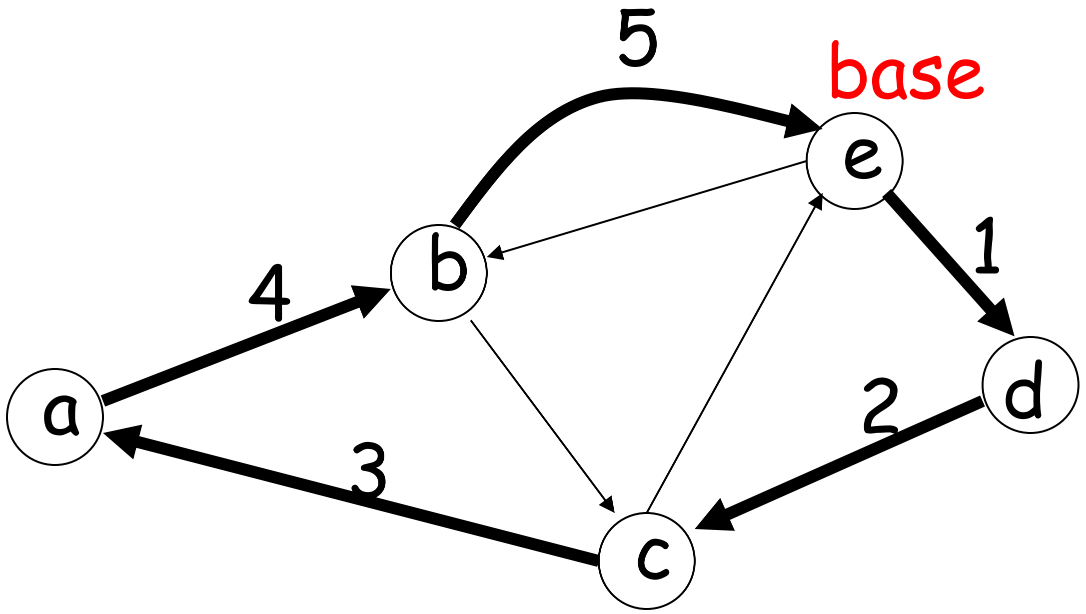
Simple cycle: only the base node is repeated

Euler Tour



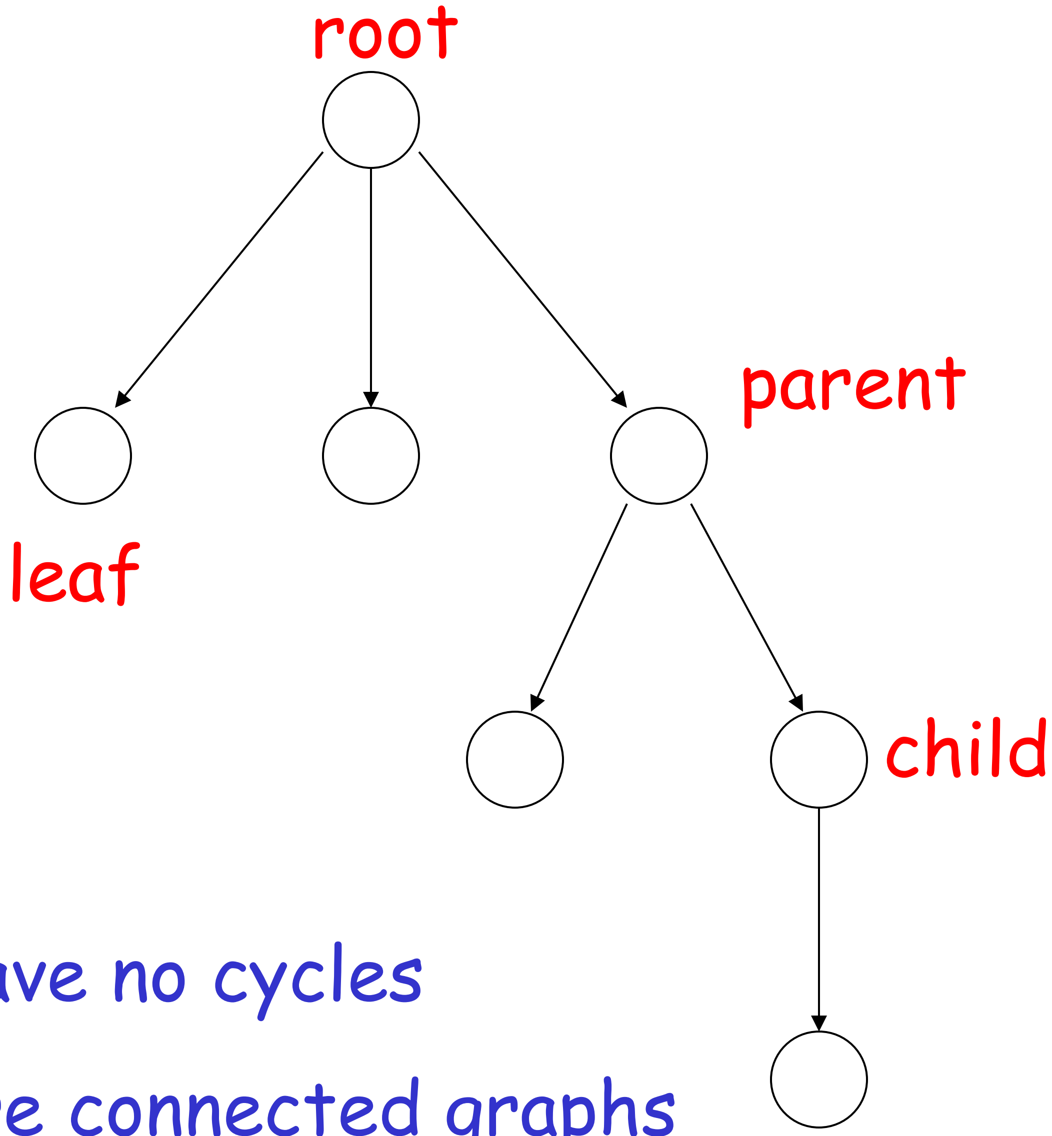
A cycle that contains each edge once

Hamiltonian Cycle



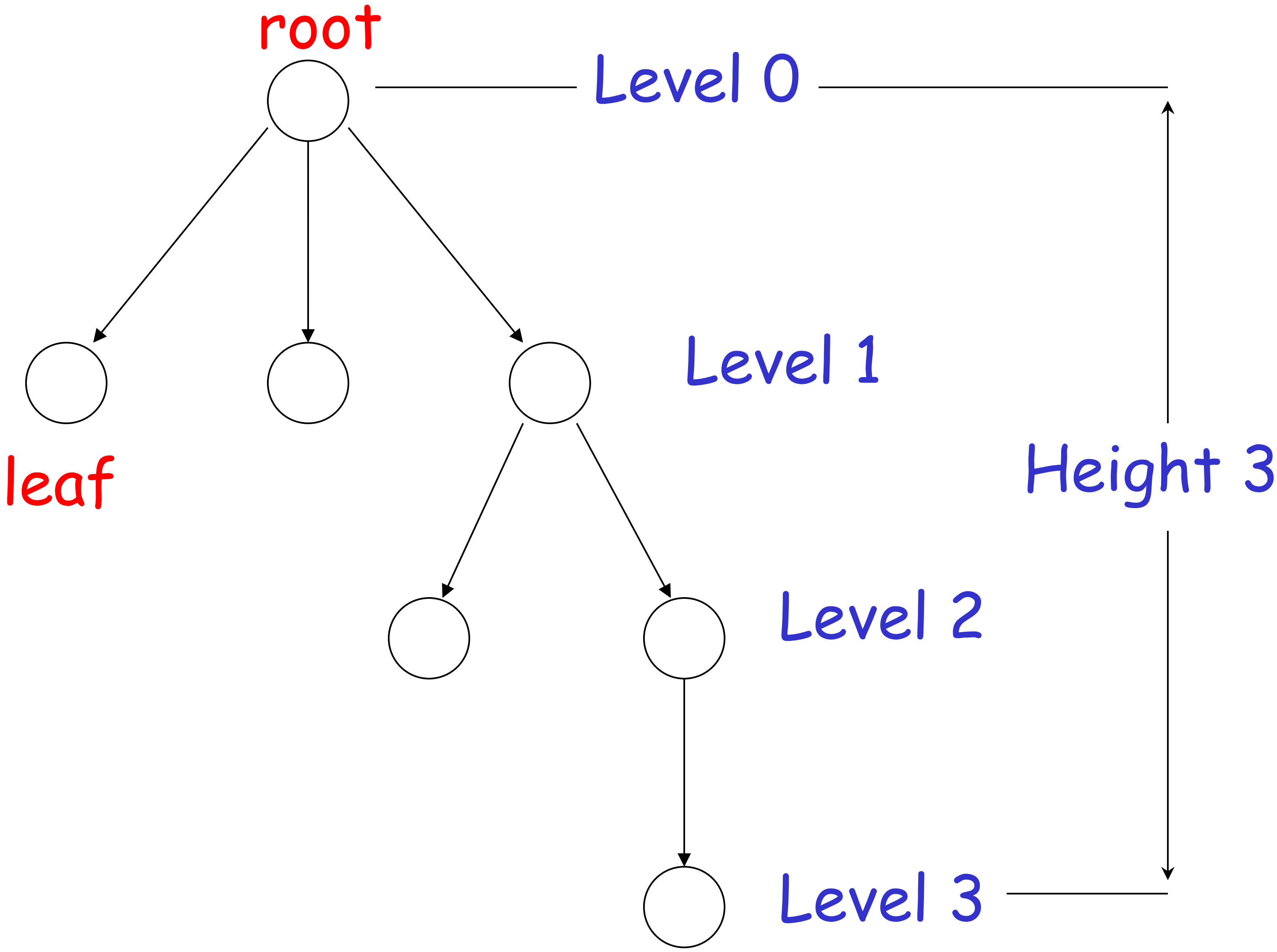
A simple cycle that contains all nodes

Trees

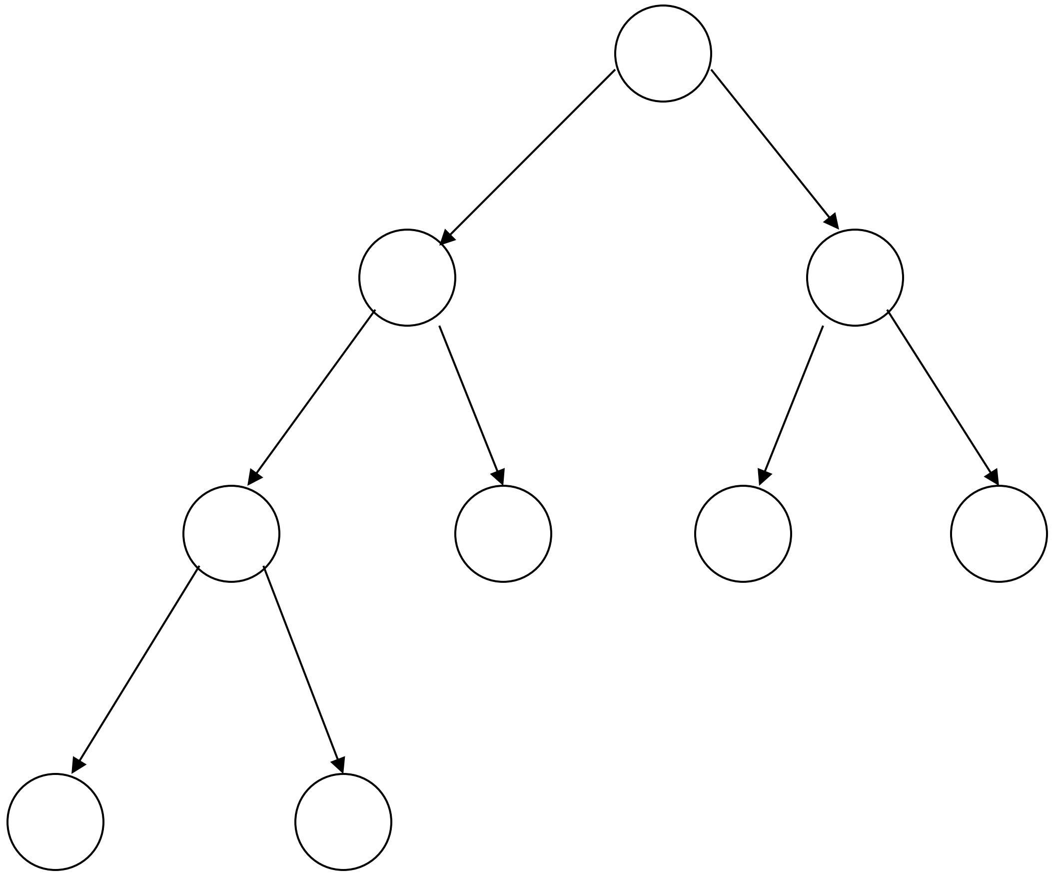


Trees have no cycles

Trees are connected graphs


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

Binary Trees



PROOF TECHNIQUES

- Proof by induction

- Proof by contradiction

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 $P_1, P_2, \ldots, P_b$ which are true

- Inductive hypothesis

  Let's assume $P_1, P_2, \ldots, P_k$ are true,

  for any $k >= b$

- Inductive step

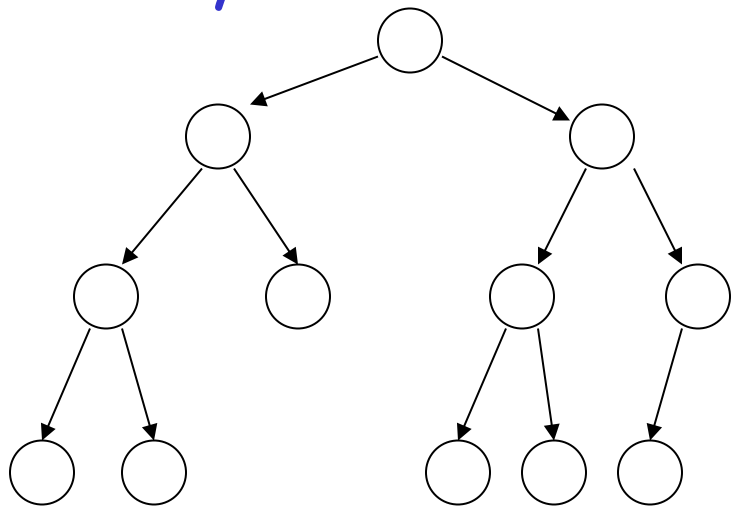
  Show that P_{k+1} is true

Example

Theorem: A binary tree of height n has at most 2^n leaves.

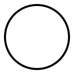
Proof by induction:

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) 

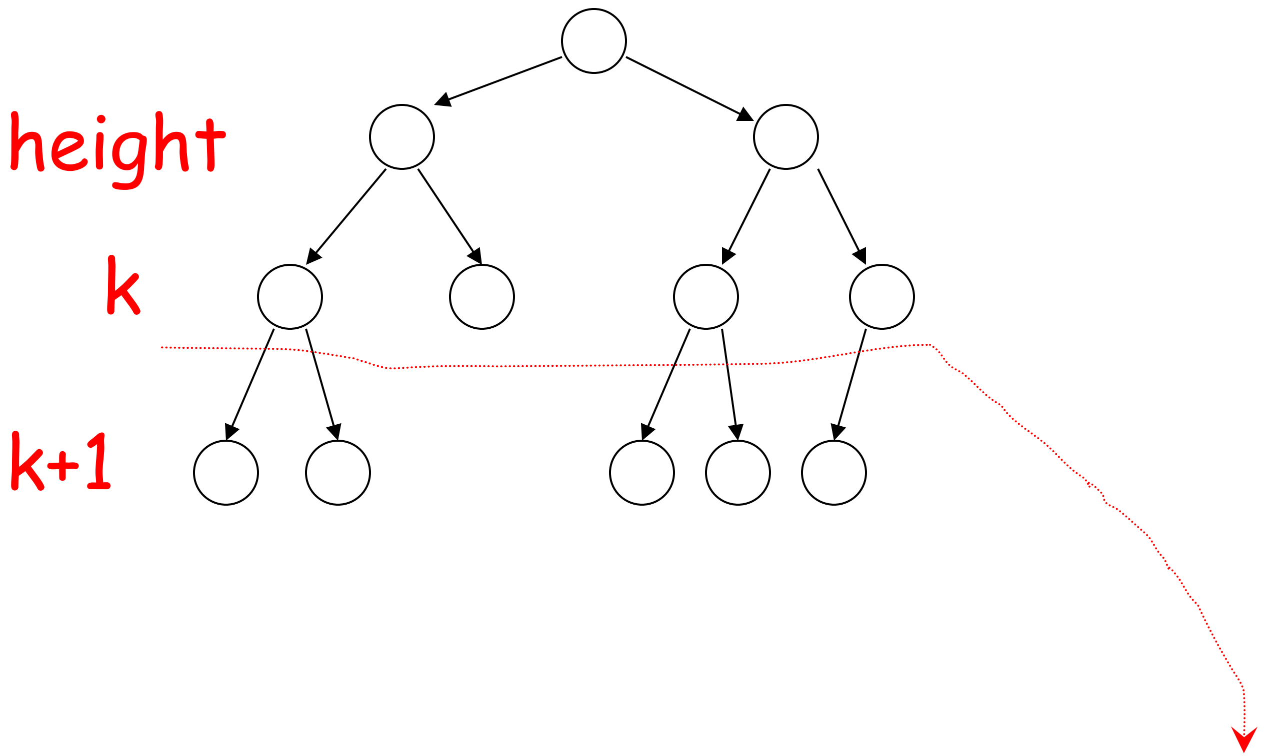
- Inductive hypothesis

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

- Induction step

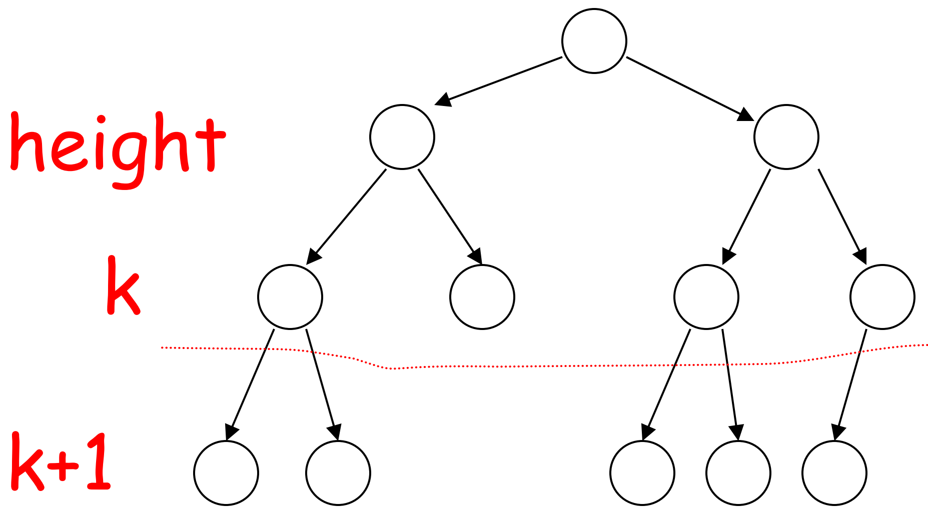
  we need to show that $L(k + 1) <= 2^{k+1}$

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!

Now you try!

Prove that $1+2+3+\ldots+n = n(n+1)/2$

More Exercises

1. Show that if $S1 \subseteq S2$, then $S2 \subseteq S1$.
2. Show that $S1 = S2$ if and only if $S1 \cup S2 = S1 \cap S2$.
3. Show that $S1 \cup S2 - S1 \cap S2 = S2$.
4. Show that the distributive law $S1 \cap (S2 \cup S3) = (S1 \cap S2) \cup (S1 \cap S3)$ holds for sets.
5. Show that $S1 \times (S2 \cup S3) = (S1 \times S2) \cup (S1 \times S3)$.

More Exersices

6. Draw a picture of the graph with vertices {v1, v2, v3} and edges {(v1, v1) , (v1, v2) , (v2, v3) , (v2, v1) , (v3, v1)}. Enumerate all cycles with base v1.
7. Construct a graph with five vertices, ten edges, and no cycles.

More Exersices

8. Show that

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1.$$

9. Show that

   √8 is not a rational number.

10. Show that

    √3 is irrational.

Languages

Alphabet, String Language

Alphabet is a finite, nonempty set of symbol, denoted by Σ.

A language is a set of strings

String is a sequence of letters/symbols

A language is a set of strings

String: A sequence of letters

Examples: "cat", "dog", "house", ...

Defined over an alphabet:

$$\Sigma = \{a, b, c, \ldots, z\}$$

Alphabets and Strings

We will use small alphabets:

$$\Sigma = \{a, b\}$$

Strings

a

ab $\qquad u = ab$

$abba$ $\qquad v = bbbaaa$

$baba$ $\qquad w = abba$

$aaabbbaabab$

String Operations

$w = a_1 a_2 \cdots a_n$ $\qquad$ $abba$

$v = b_1 b_2 \cdots b_m$ $\qquad$ $bbbaaa$

Concatenation

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

$$w = a_1 a_2 \cdots a_n \qquad ababaaabbb$$

Reverse

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

String Length

$$w = a_1 a_2 \cdots a_n$$

Length:

$$|w| = n$$

Examples:

$$|abba| = 4$$

$$|aa| = 2$$

$$|a| = 1$$

Length of Concatenation

$$|uv| = |u| + |v|$$

Example:

$$u = aab, \quad |u| = 3$$

$$v = abaab, \quad |v| = 5$$

$$|uv| = |aababaab| = 8$$

$$|uv| = |u| + |v| = 3 + 5 = 8$$

Empty String

A string with no letters: λ

Observations: $|\lambda| = 0$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

Empty String

A string with no letters: λ

Observations: $|\lambda| = 0$

$$\lambda w = w\lambda = w$$

$$\lambda abba = abba\lambda = abba$$

Substring

Substring of string:

a subsequence of consecutive characters

String	Substring
$\underline{ab}bab$	ab
$\underline{abba}b$	$abba$
$ab\underline{b}ab$	b
$a\underline{bbab}$	$bbab$

Prefix and Suffix

$abbab$

Prefixes	Suffixes
λ	$abbab$
a	$bbab$
ab	bab
abb	ab
$abba$	b
$abbab$	λ

$$w = uv$$

prefix

suffix

Another Operation

$$w^n = \underbrace{ww\cdots w}_{n}$$

Example:

$$(abba)^2 = abbaabba$$

Definition:

$$w^0 = \lambda$$

$$(abba)^0 = \lambda$$

Another Operation

$$w^n = \underbrace{ww\cdots w}_{n}$$

Example: $$(abba)^2 = abbaabba$$

Definition: $$w^0 = \lambda$$

$$(abba)^0 = \lambda$$

The * Operation

Σ^*: the set of all possible strings from alphabet Σ

$$\Sigma = \{a, b\}$$

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

The + Operation

Σ^+ : the set of all possible strings from alphabet Σ except λ

$$\Sigma = \{a, b\}$$

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

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

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

Languages

A language is any subset of Σ^*

Example: $\Sigma = \{a, b\}$

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

Languages: $\{\lambda\}$

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

$$\{\lambda, abba, baba, aa, ab, aaaaaa\}$$

Note that:

Sets $\qquad \emptyset = \{\,\} \neq \{\lambda\}$

Set size $\qquad |\{\,\}| = |\emptyset| = 0$

Set size $\qquad |\{\lambda\}| = 1$

String length $\qquad |\lambda| = 0$

Another Example

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

$$\left.\begin{array}{l} \lambda \\ ab \\ aabb \\ aaaaabbbbb \end{array}\right\} \in L \qquad abb \notin L$$

Operations on Languages

The usual set operations

$$\{a, ab, aaaa\} \bigcup \{bb, ab\} = \{a, ab, bb, aaaa\}$$

$$\{a, ab, aaaa\} \bigcap \{bb, ab\} = \{ab\}$$

$$\{a, ab, aaaa\} - \{bb, ab\} = \{a, aaaa\}$$

Complement: $\overline{L} = \Sigma^* - L$

$$\overline{\{a, ba\}} = \{\lambda, b, aa, ab, bb, aaa, \ldots\}$$

Reverse

Definition: $$L^R = \{w^R : w \in L\}$$

Examples: $$\{ab, aab, baba\}^R = \{ba, baa, abab\}$$

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

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

Concatenation

Definition: $$L_1 L_2 = \{xy : x \in L_1, y \in L_2\}$$

Example: $$\{a, ab, ba\}\{b, aa\}$$

$$= \{ab, aaa, abb, abaa, bab, baaa\}$$

Another Operation

Definition: $L^n = \underbrace{LL \cdots L}_{n}$

$$\{a,b\}^3 = \{a,b\}\{a,b\}\{a,b\} =$$
$$\{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

Special case: $L^0 = \{\lambda\}$

$$\{a, bba, aaa\}^0 = \{\lambda\}$$

More Examples

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

$$L^2 = \{a^n b^n a^m b^m : n, m \geq 0\}$$

$$aabbaaabbb \in L^2$$

Star-Closure (Kleene *)

Definition: $L^* = L^0 \cup L^1 \cup L^2 \cdots$

Example:

$$\{a,bb\}^* = \left\{ \begin{array}{l} \lambda, \\ a,bb, \\ aa,abb,bba,bbbb, \\ aaa,aabb,abba,abbbb,\ldots \end{array} \right\}$$

Positive Closure

Definition:
$$L^+ = L^1 \cup L^2 \cup \cdots$$
$$= L^* - \{\lambda\}$$

$$\{a, bb\}^+ = \left\{ \begin{array}{l} a, bb, \\ aa, abb, bba, bbbb, \\ aaa, aabb, abba, abbbb, \ldots \end{array} \right\}$$

Exercise

1. Use induction on n to show that $|u^n| = n\,|u|$ for all strings u and all n.

2. Let $\Sigma = \{a, b\}$ and $L = \{aa, bb\}$. Use set notation to describe L.

Grammar

Grammar

Grammar 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 variables
P is a finite set of productions

The detail information about Grammar will be provide in the session 5th