CS-305

Formal Language & Automata Theory

Instructor: Dr. Ashish Phophalia

Asst. Prof., IIITV

ashish_p@iiitvadodara.ac.in

Reference Books:

1. Peter Linz
2. Micheal Sipser
3. K. L. Mishra & Chandrashekran
4. Kamala Kirtivasan,
5. John C. Martin
6. Aho, Ullman, Sethi
7. Dexter Kozen
8. Lewis & Papadimitriou
9. John Sevage
10. Vivek Kulkarni
11. …

Course Resources

- Offered to all major universities/colleges around the globe in CS stream
- NPTEL video lectures
- You are free to refer course website of other reputed universities/faculties

Video Lectures

1. Prof. Somnath Biswas, IIT Kanpur
2. Prof. Kamala K., IIT Madras
3. Prof. J. Ullman, Coursera/Stanford
4. Prof. Shai Simonson, ArsDigita University

Purpose of Course

- Historical Perceptive – Current Computation modeling
- Foundation course to computer science & research in relevant areas
- Major part in many competitive exams like GATE

Course Content

Mathematical Preliminaries: Set, Functions, Relation, Graph Theory, Mathematical Induction, Proof Techniques

Finite Automata: DFA, NDFA, Conversion b/w DFA & NDFA, Melay & Moore Machine, Minimization of automata

Languages & Grammars: Types and Properties of Chomsky classification

Regular Languages & Grammar, Pumpimg Leema

Context Free Language, Grammar & Pushdown Automata, Deterministic Context Free Language and Automatam, Pumping Leema

Context Sensitive Language, Grammar & Linear Bounded automata

Turning Machines & its variants, Undecideability & Reduceability

Computational Complexity: P, NP, NP Complete and Hard Problems, Post Correspondence Problem (PCP)

Course Evaluation

Course Structure: 3-1-0-4

Attendance – as per Institute norm for theory classes

A) Theory component – 75 Marks

1. Mid Semester Examination ~ **33% (25 Marks)**
2. End Semester Examination ~ **47% (35 Marks)**
3. Surprise Quizzes (at least 3) ~ **20% (15 Marks)**

B) Tutorial Component – 25 Marks

4. Assignments – **10 Marks**
5. Tutorial Submission (Random) – **10 Marks**
6. Tutorial attendance – **5 Marks**

Course Goals

Provide computation Models

Analyze power of Models

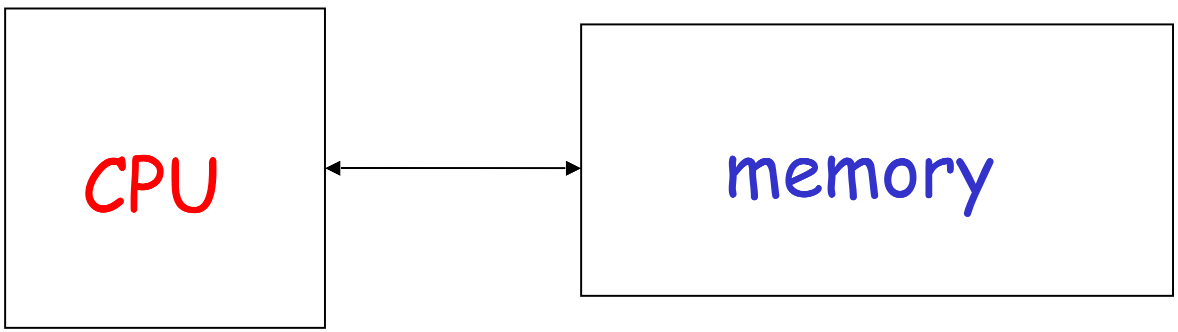
Answer Intractability questions:

What computational problems can each model solve?

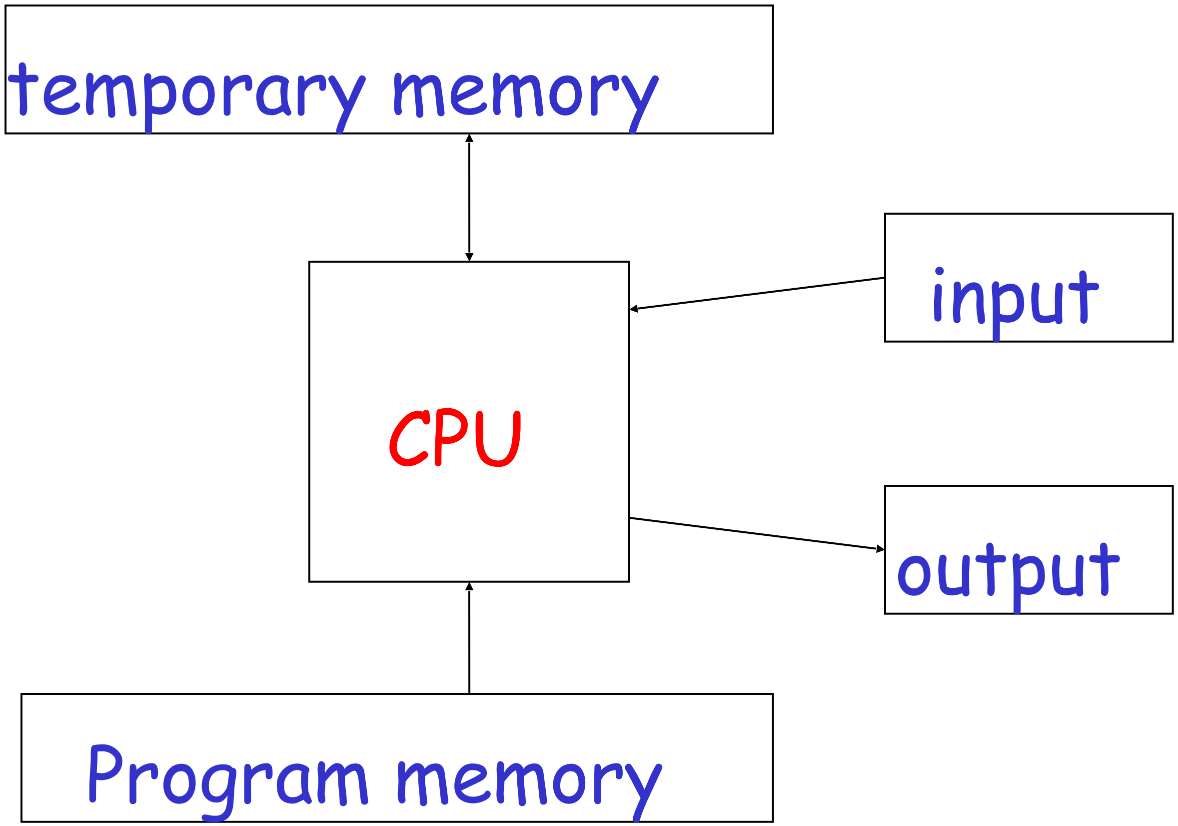
Answer Time Complexity questions:

How much time we need to solve the problems?

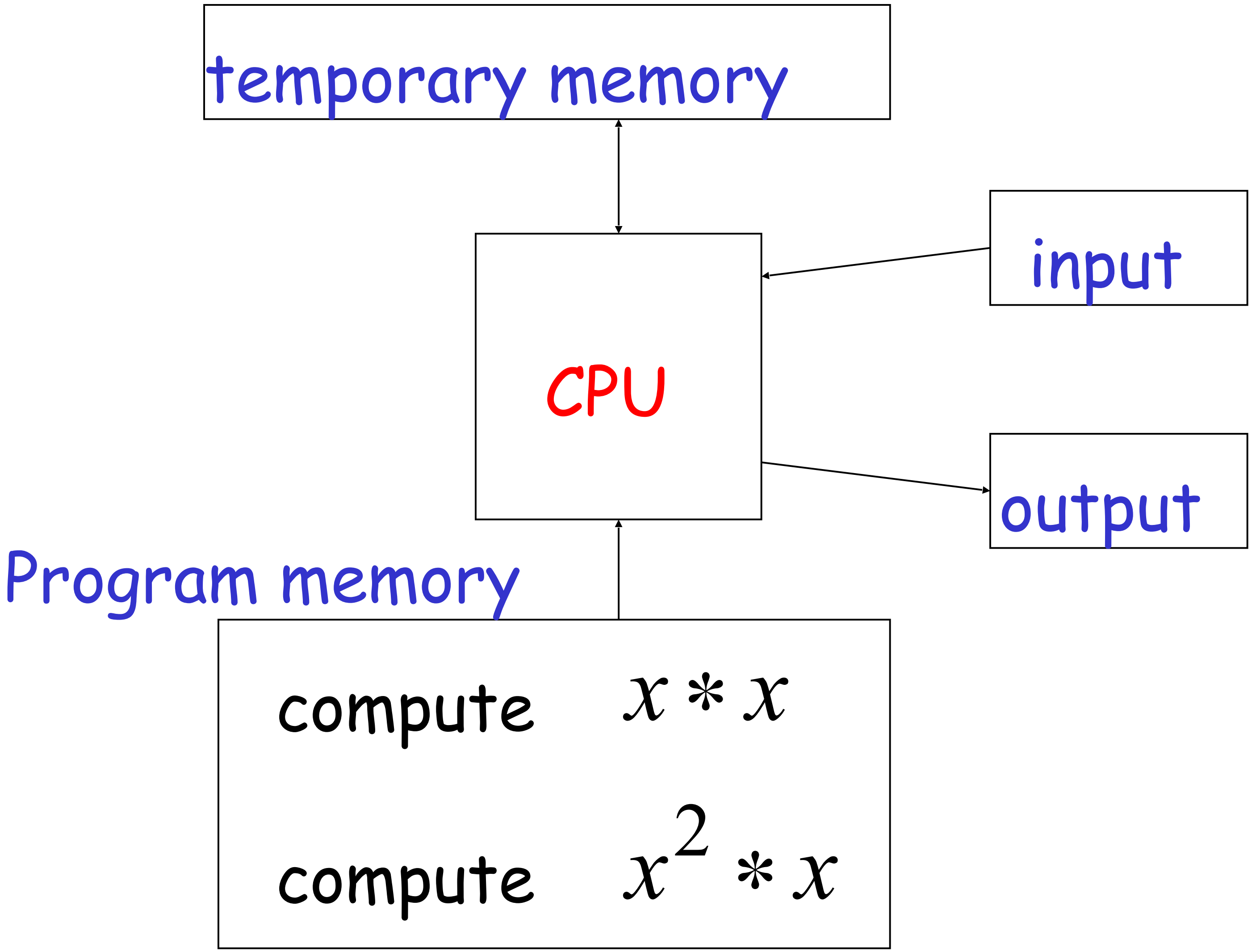
A widely accepted model of computation



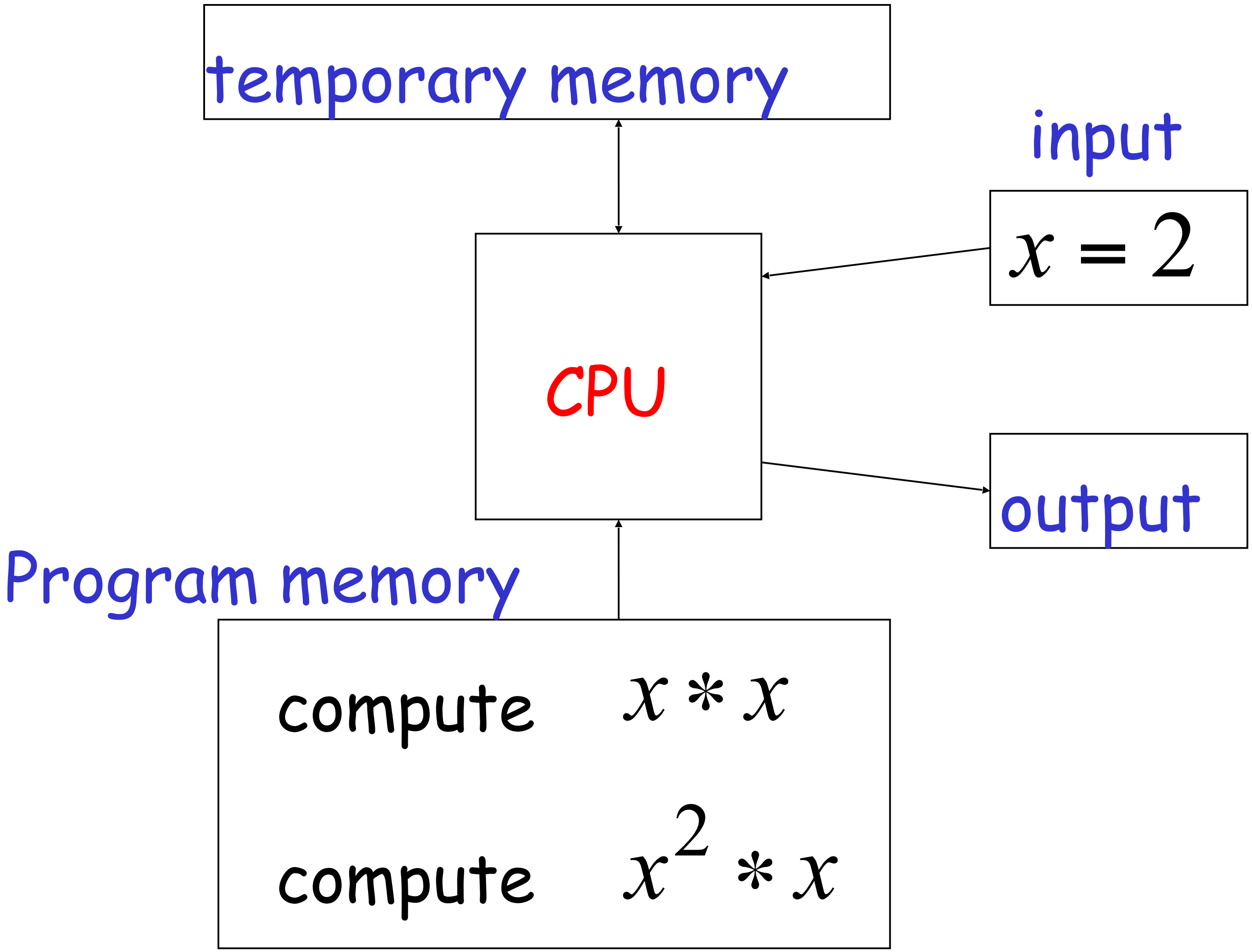
The different components of memory



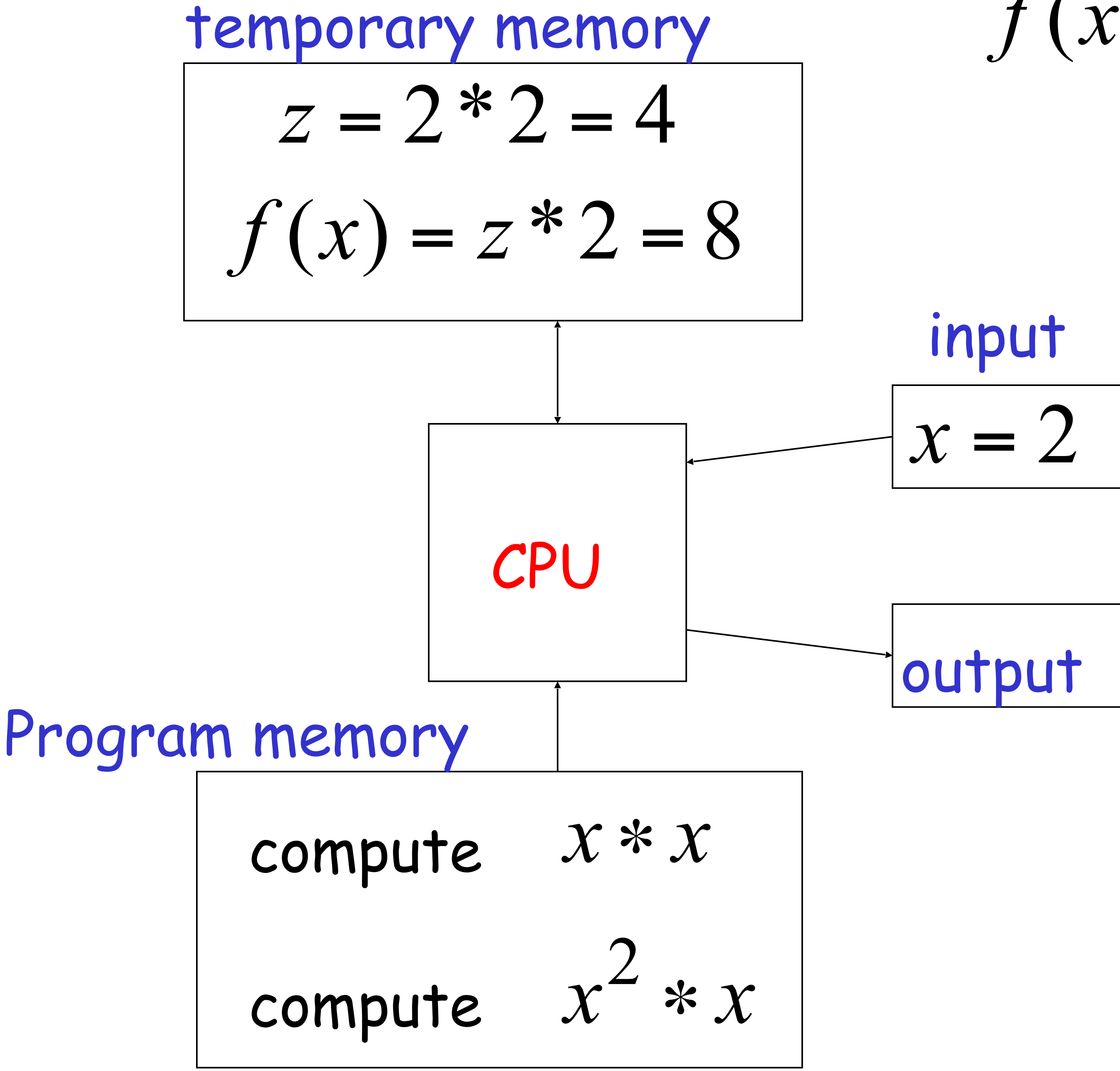
Example: $f(x) = x^3$



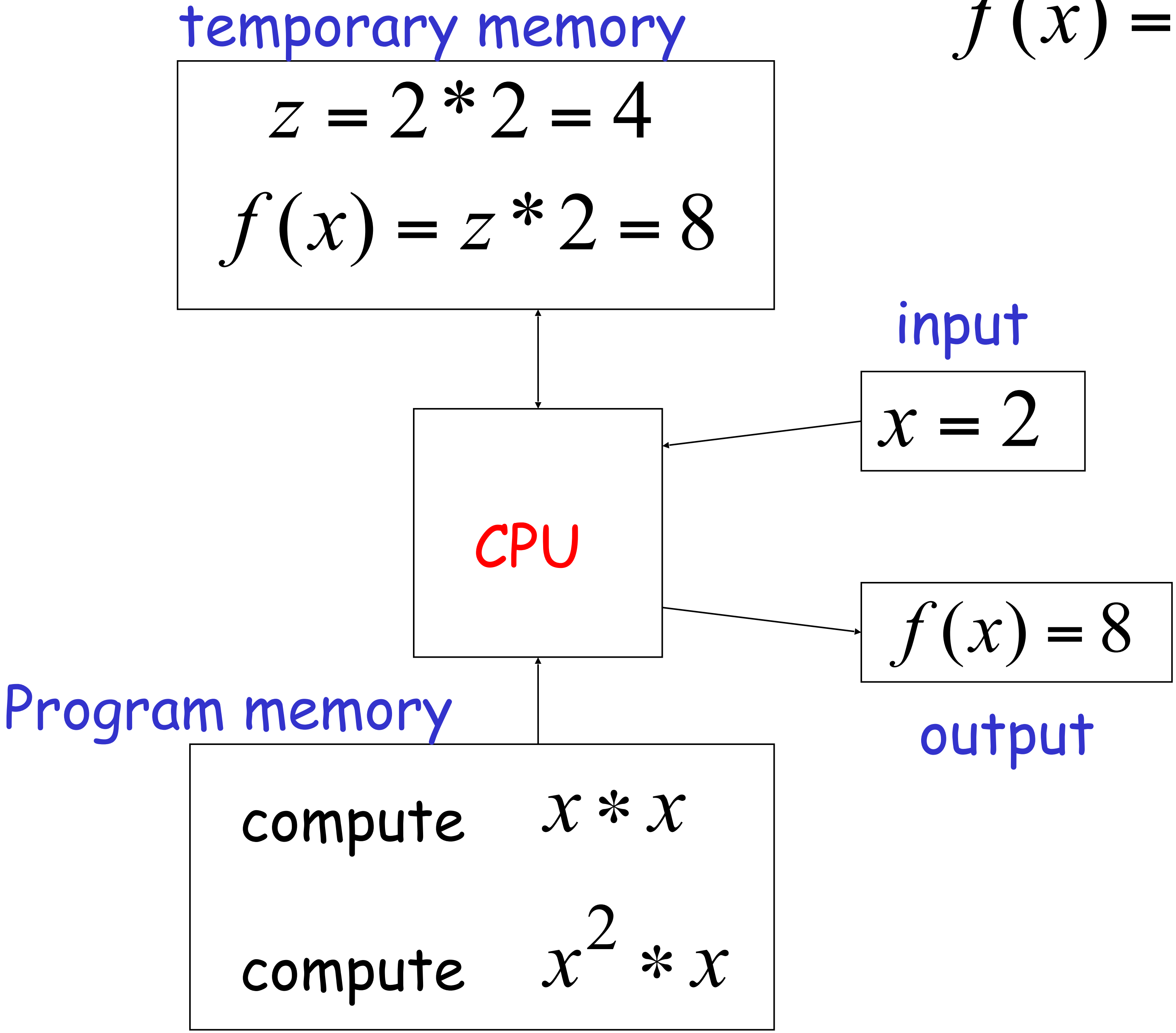
$$f(x) = x^3$$


temporary memory
input
$x = 2$
CPU
output
Program memory
compute $x * x$
compute $x^2 * x$

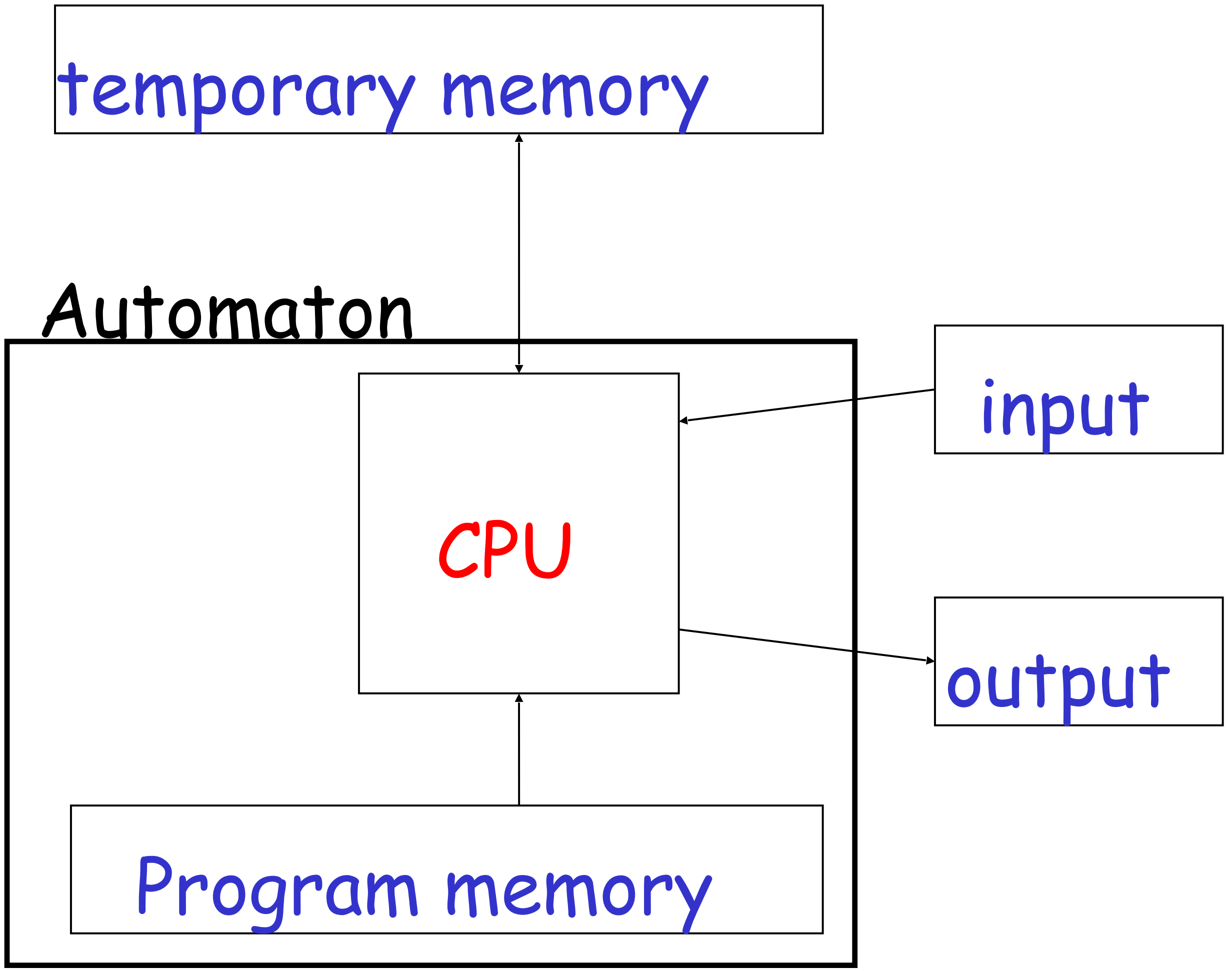
$$f(x) = x^3$$


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

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

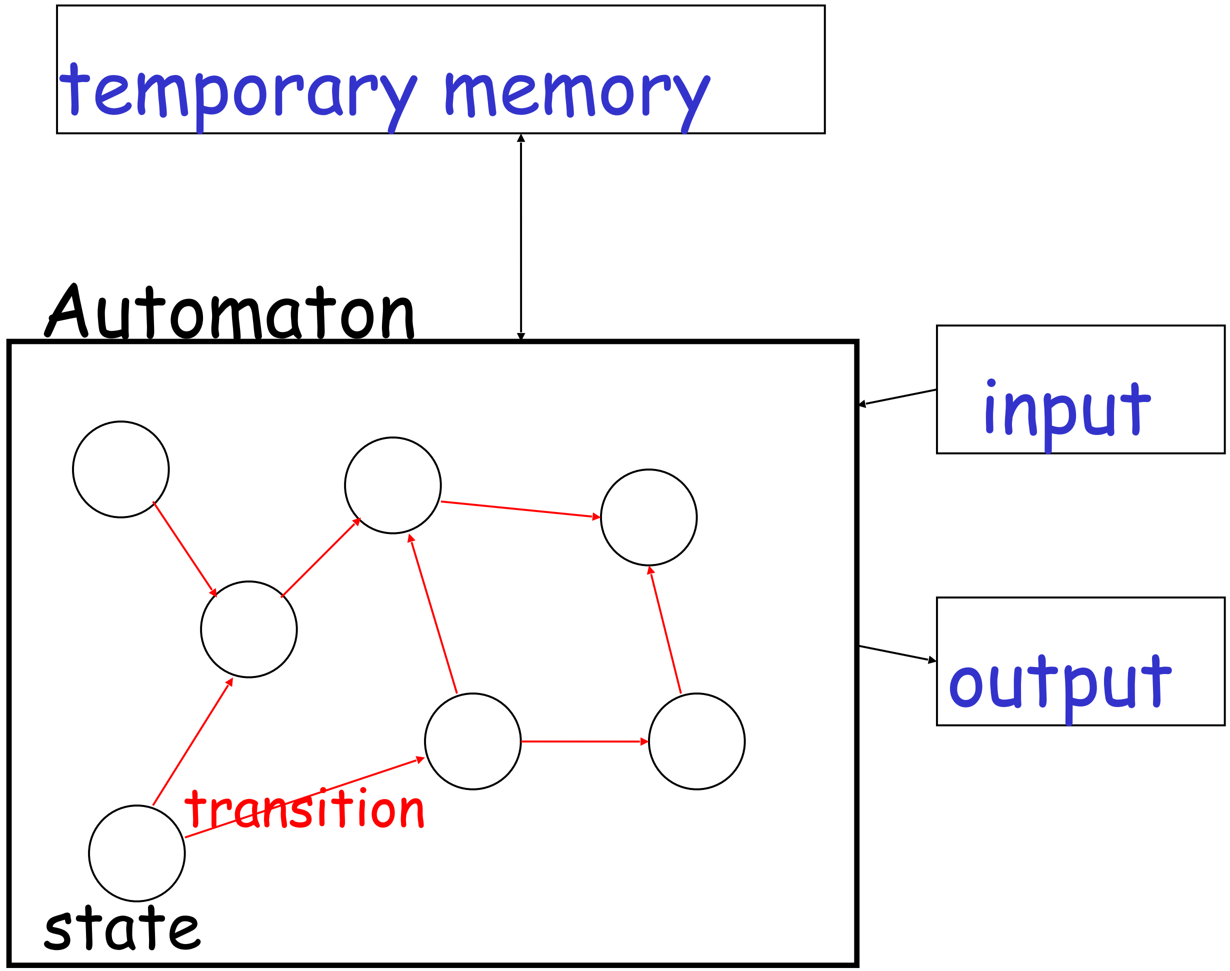

temporary memory
$z = 2 * 2 = 4$
$f(x) = z * 2 = 8$
input
$x = 2$
CPU
$f(x) = 8$
output
Program memory
compute $x * x$
compute $x^2 * x$

Automaton



temporary memory
Automaton
CPU
input
output
Program memory

Automaton



CPU+ProgramMem = States + Transitions

Different Kinds of Automata

Automata are distinguished by the temporary memory

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

Memory affects computational power:

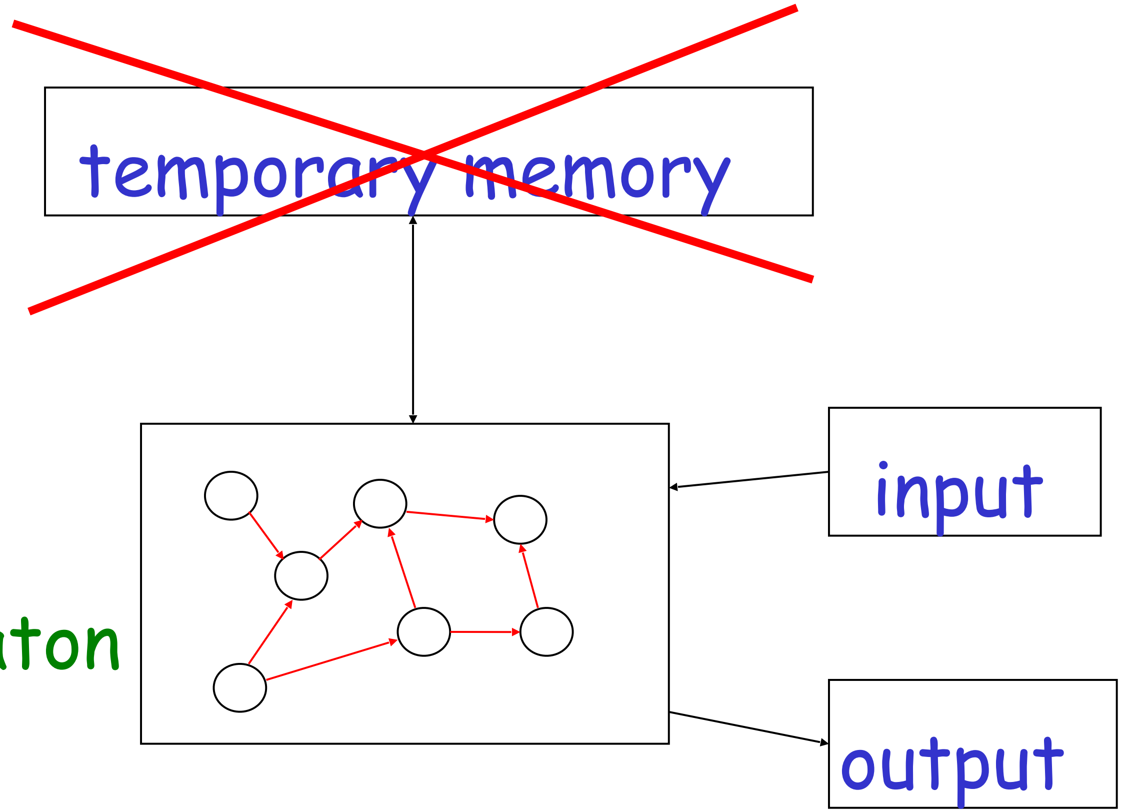
More flexible memory

results to

The solution of more computational problems

Finite Automaton

Finite Automaton

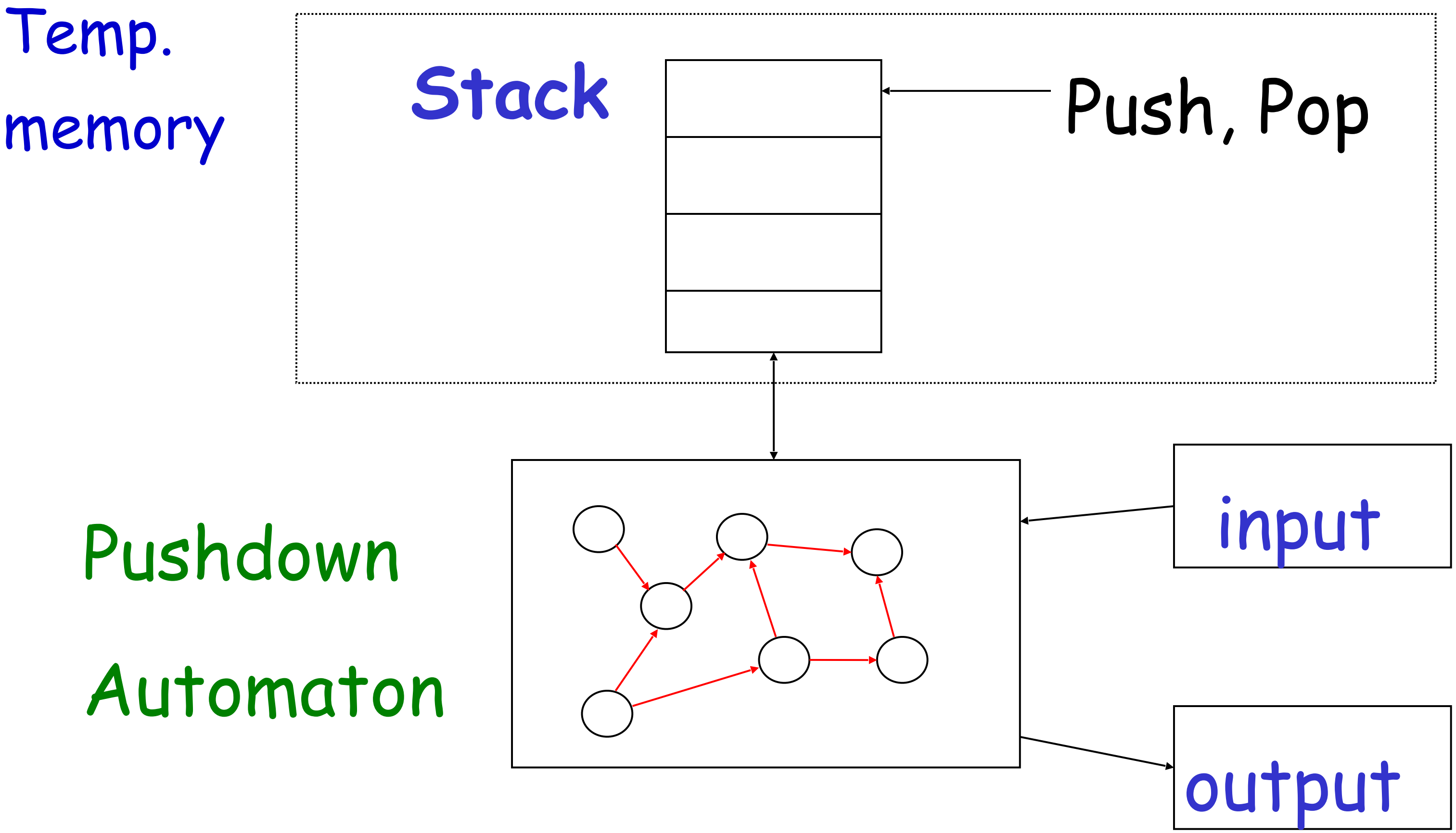


Example: Elevators, Vending Machines,

Lexical Analyzers

(small computing power)

Pushdown Automaton

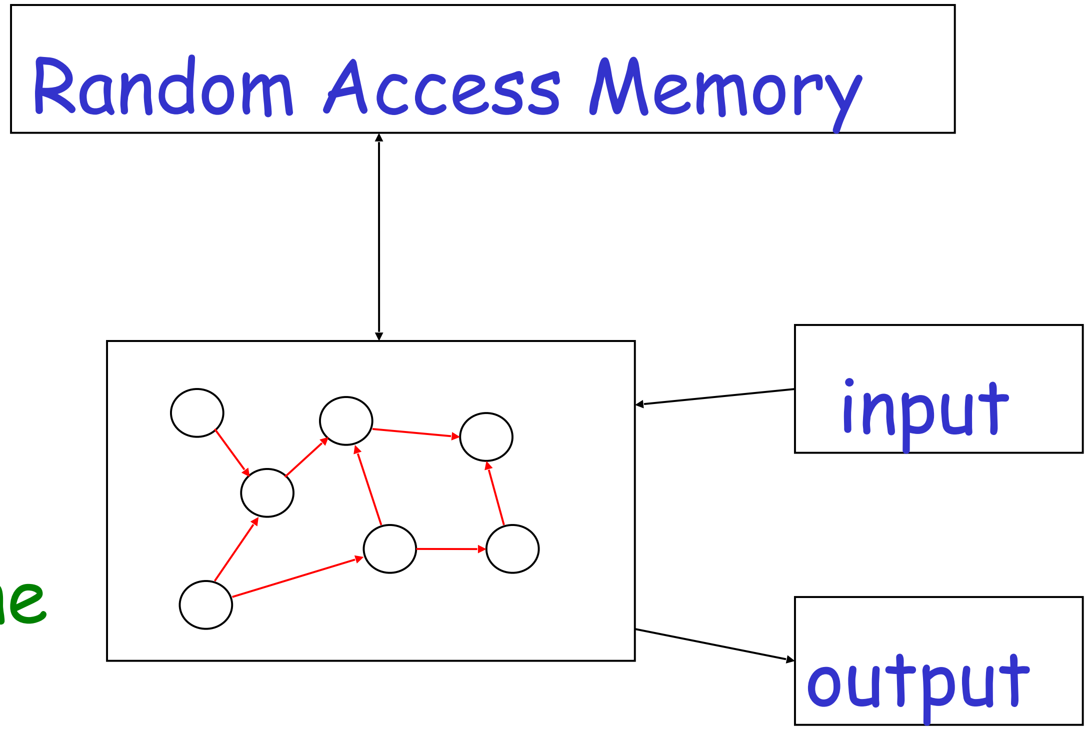


Example: Parsers for Programming Languages

(medium computing power)

Turing Machine

Temp. memory



Turing Machine

Examples: Any Algorithm

(highest known computing power)

Power of Automata

Simple problems		More complex problems		Hardest problems
Finite Automata	$<$	Pushdown Automata	$<$	Turing Machine

Less power $\longrightarrow$ More power

Solve more computational problems

Turing Machine is the most powerful known computational model

Question: can Turing Machines solve all computational problems?

Answer: NO
(there are unsolvable problems)

Time Complexity of Computational Problems:

P problems:

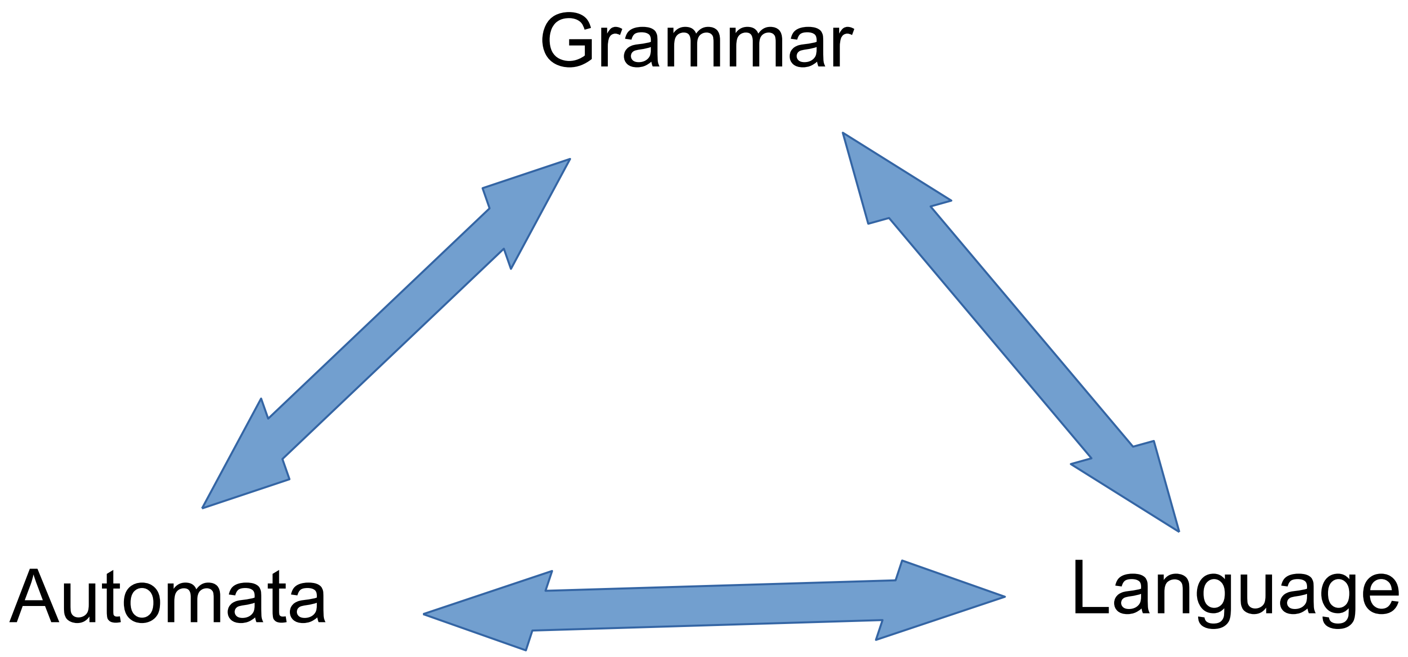
(Polynomial time problems)

Solved in polynomial time

NP-complete problems:

(Non-deterministic Polynomial time problems)

Believed to take exponential time to be solved


Grammar
Automata
Language

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

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

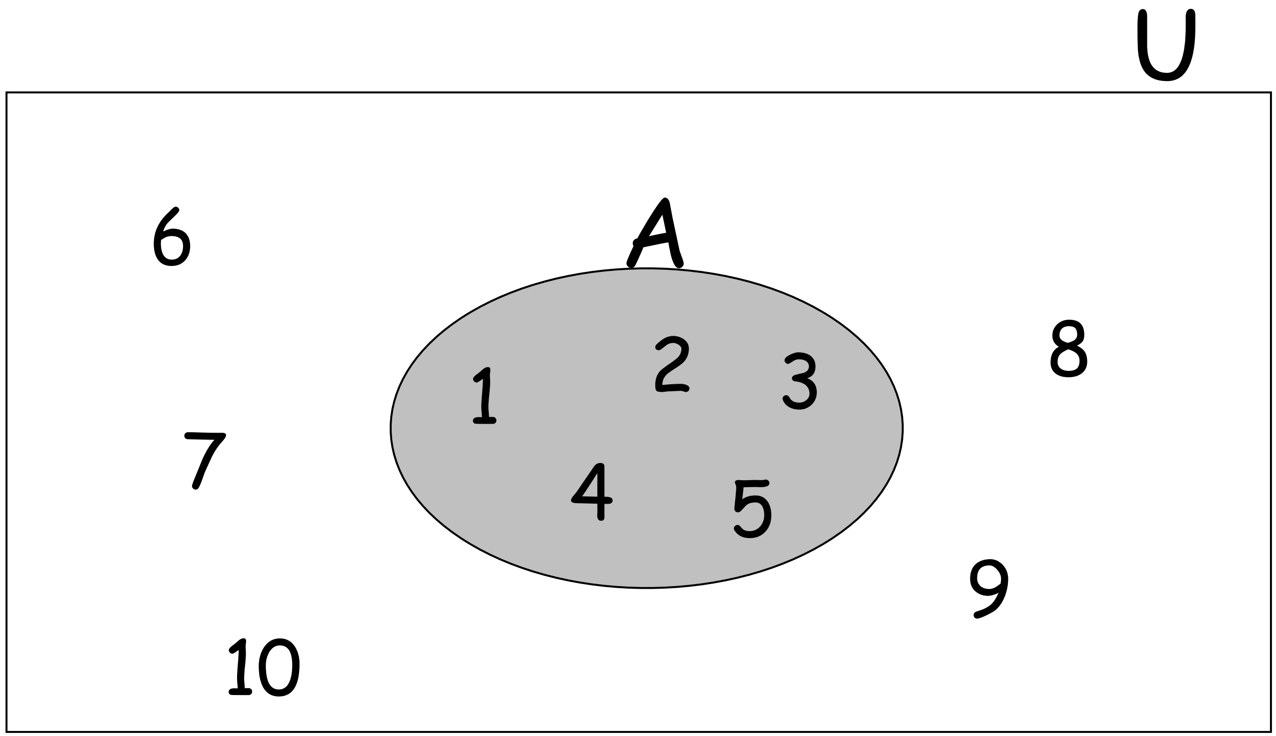
$C = \{ a, b, \ldots, k \}$ $\longrightarrow$ finite set

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

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

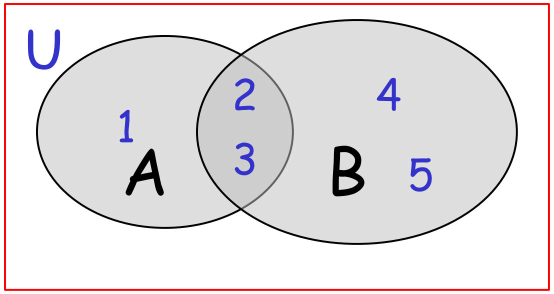
- Intersection

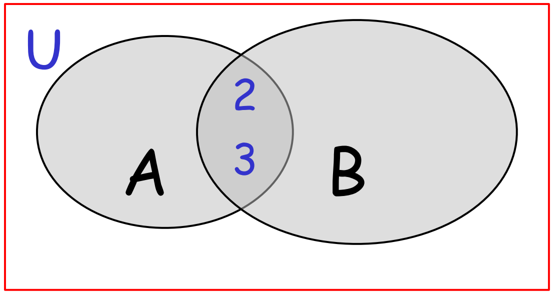
  $A \cap B = \{ 2, 3 \}$

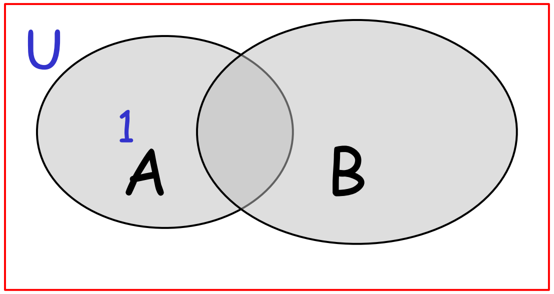
- Difference

  $A - B = \{ 1 \}$

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





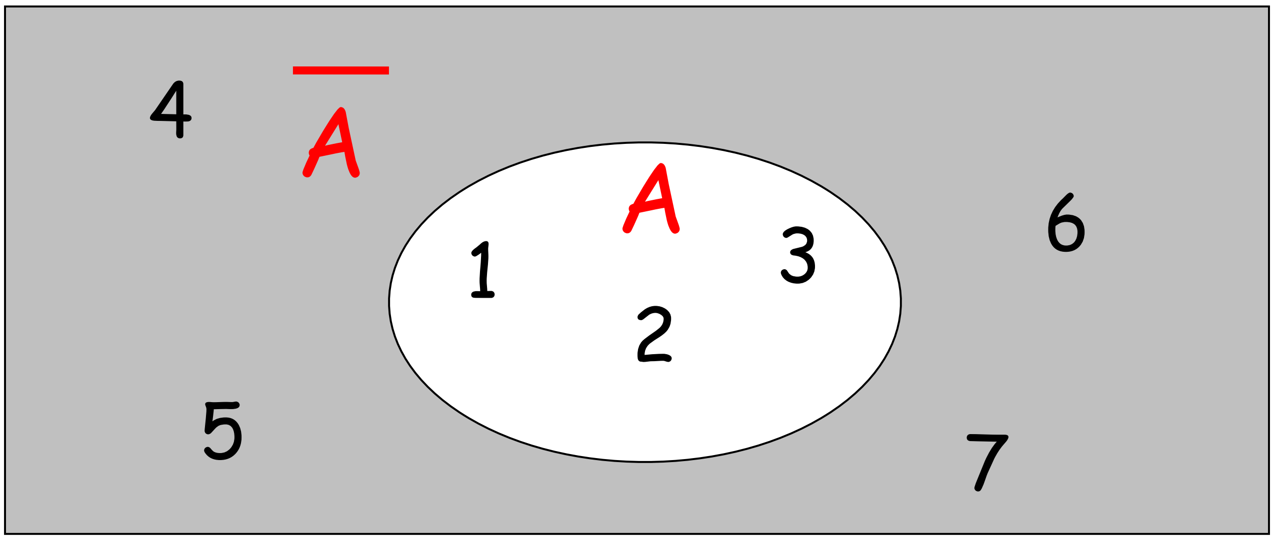


Venn diagrams

- Complement

Universal set = {1, …, 7}

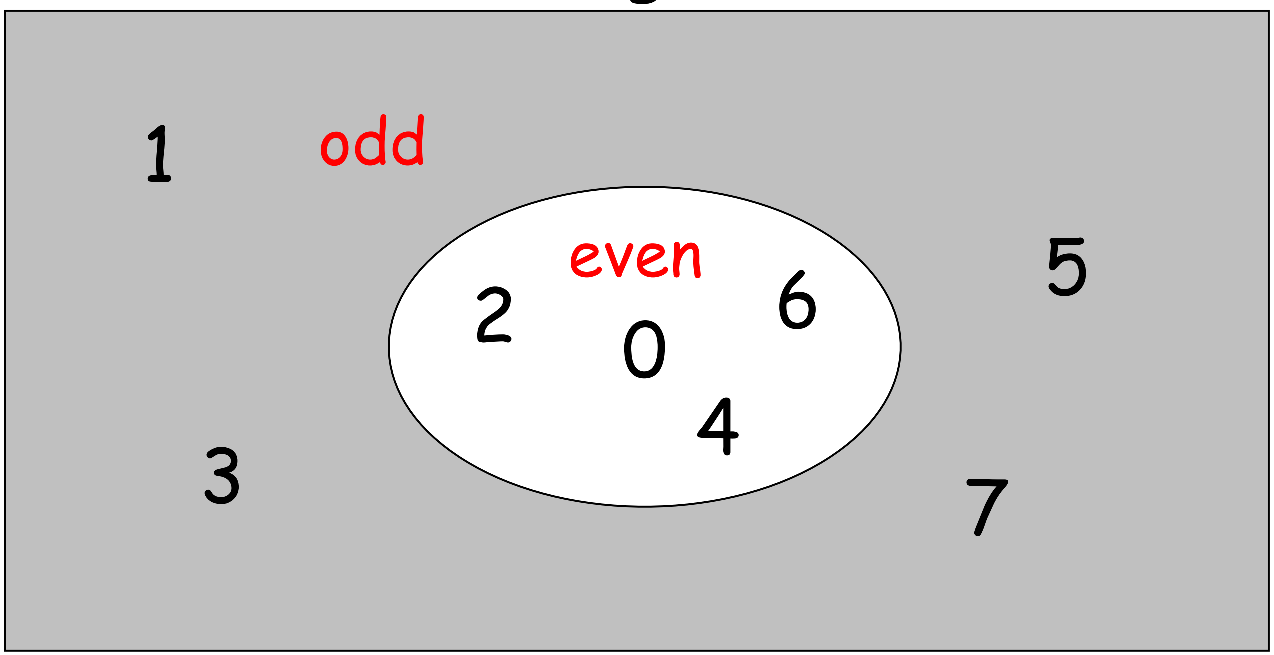
$A = \{ 1, 2, 3 \} \Rightarrow \overline{A} = \{ 4, 5, 6, 7 \}$



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

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

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: $\emptyset$

$\emptyset = \{\ \}$

$S \cup \emptyset = S$

$S \cap \emptyset = \emptyset$

$\overline{\emptyset}$ = Universal Set

$S - \emptyset = S$

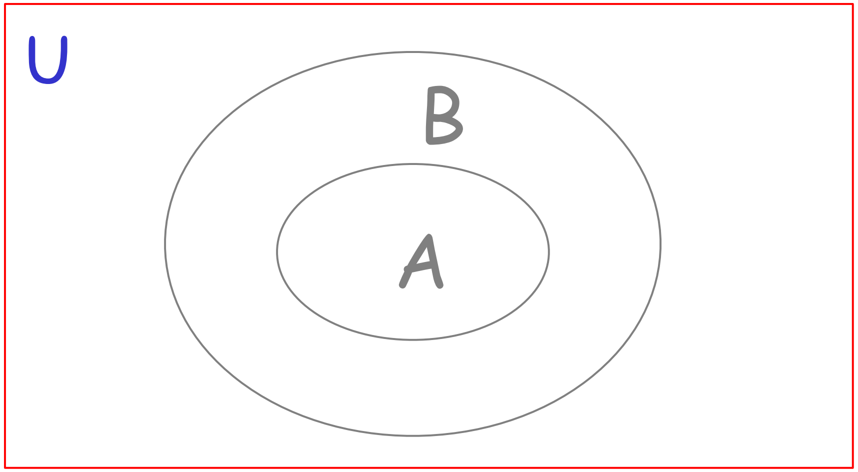
$\emptyset - S = \emptyset$

Subset

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

$A \subseteq B$

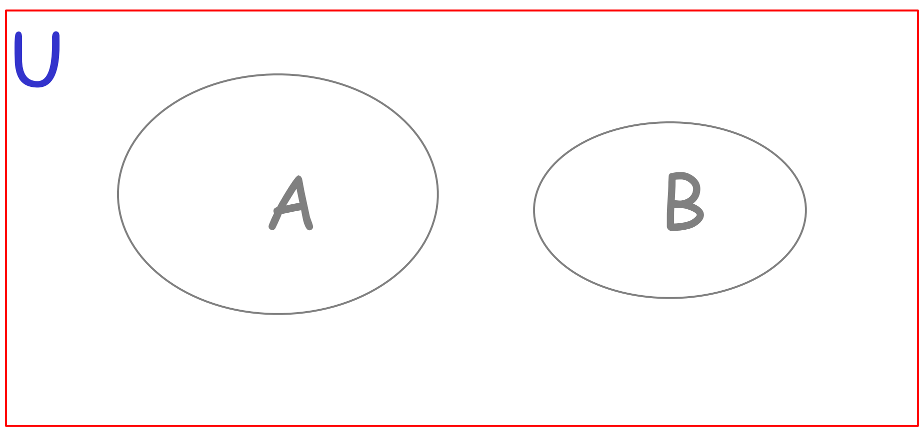
Proper Subset: $A \subset B$



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

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

Power set 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

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

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

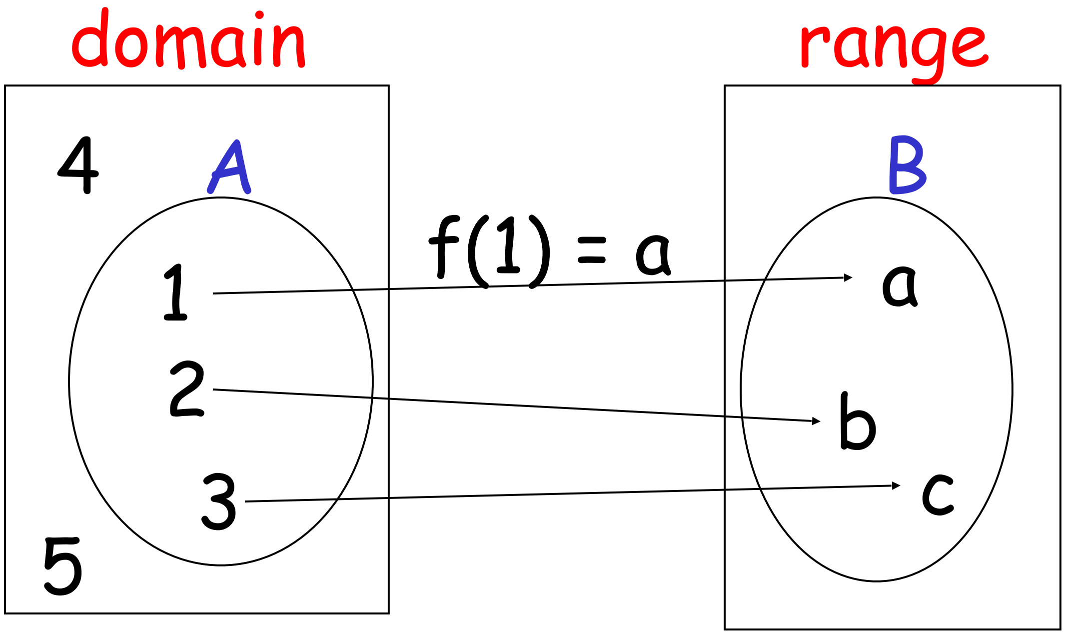
$(4, 2), (4, 3), (4, 5) \}$

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

Generalizes to more than two sets

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

FUNCTIONS



$f : A \to B$

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:

$R = \{ (1, 1), (2, 2), (1, 2), (2, 1),$

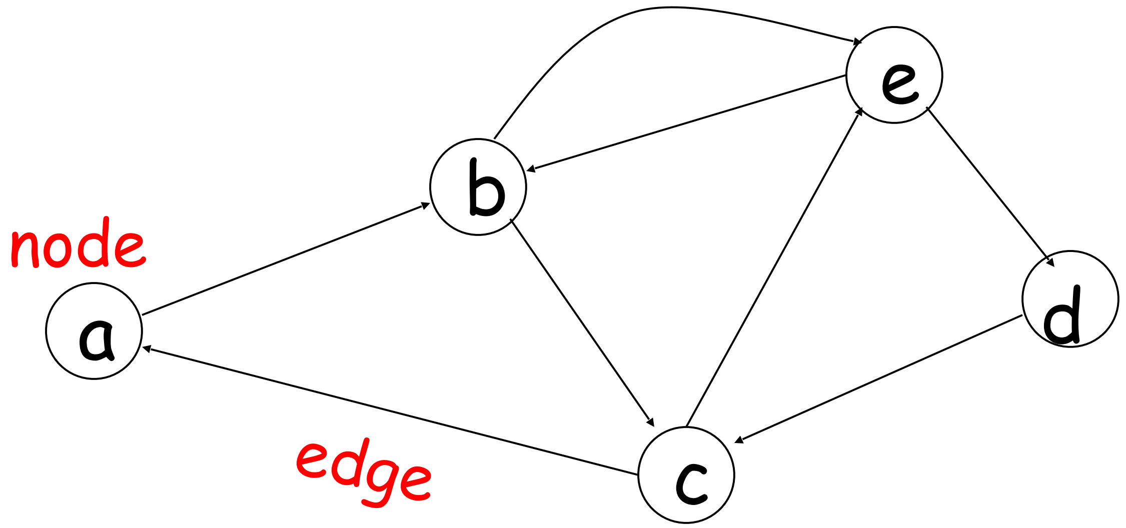
$(3, 3), (4, 4), (3, 4), (4, 3) \}$

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

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

GRAPHS

A directed graph



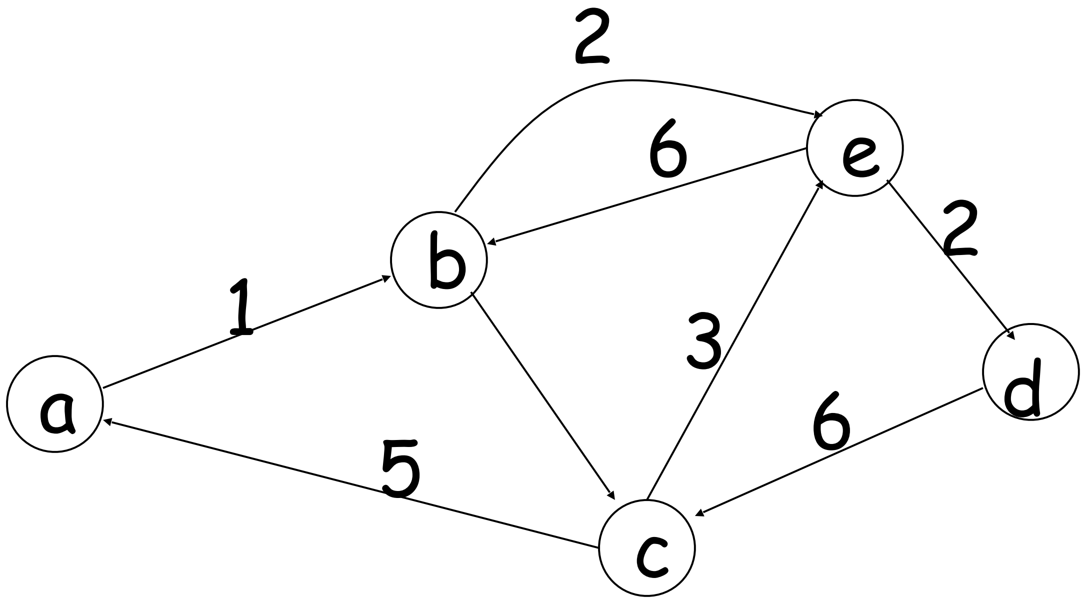
- Nodes (Vertices)

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

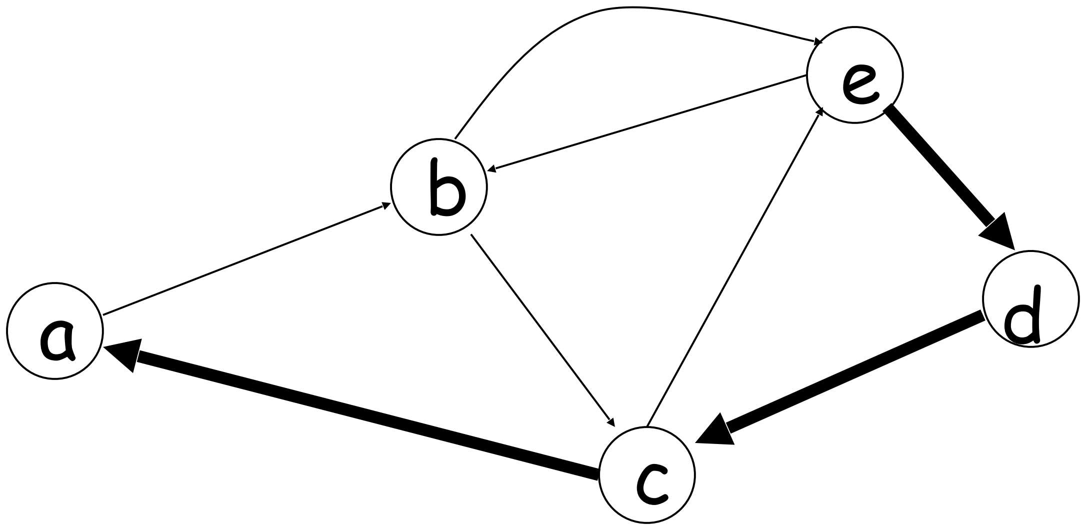
- Edges

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

Labeled Graph



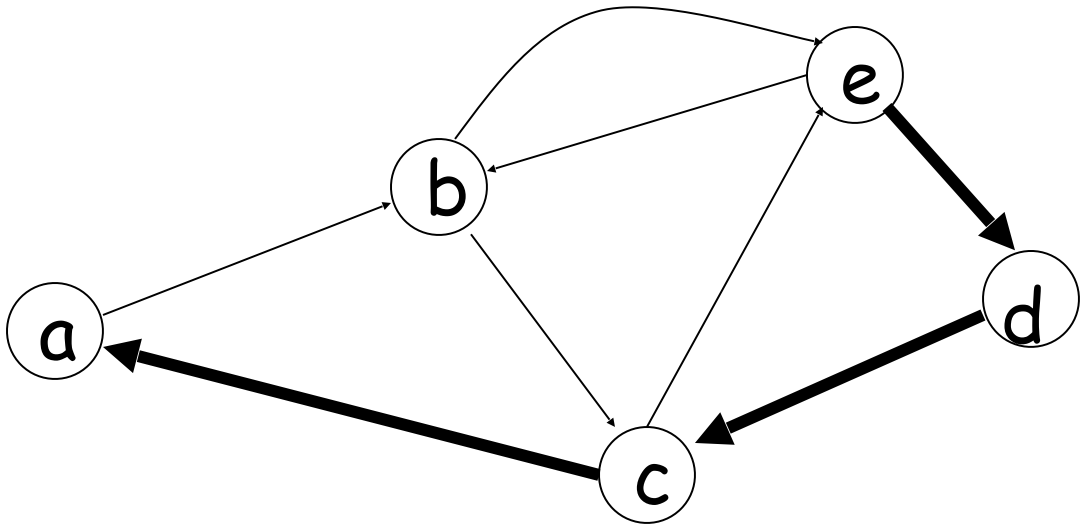
Walk



Walk is a sequence of adjacent edges

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

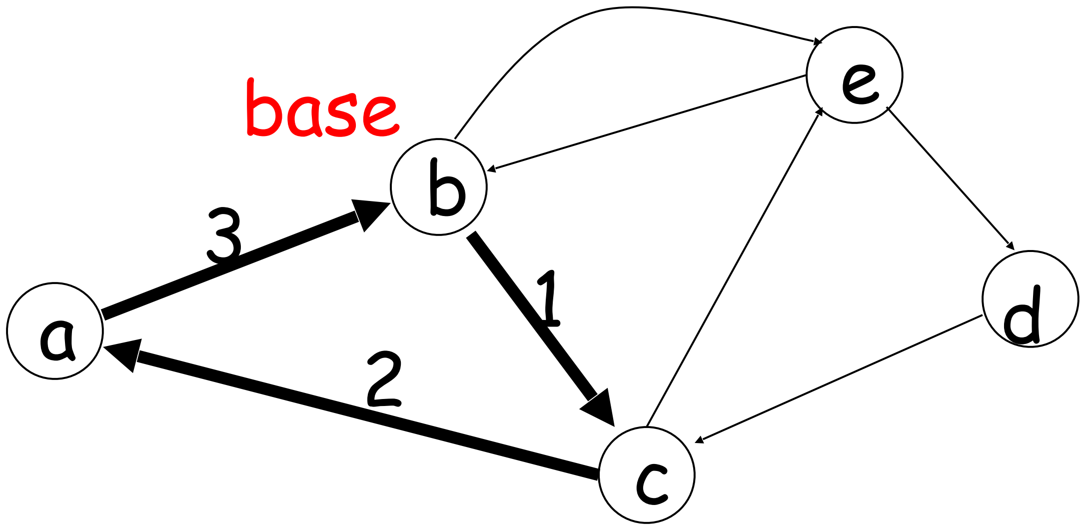
Path



Path is a walk where no edge is repeated

Simple path: no node is repeated

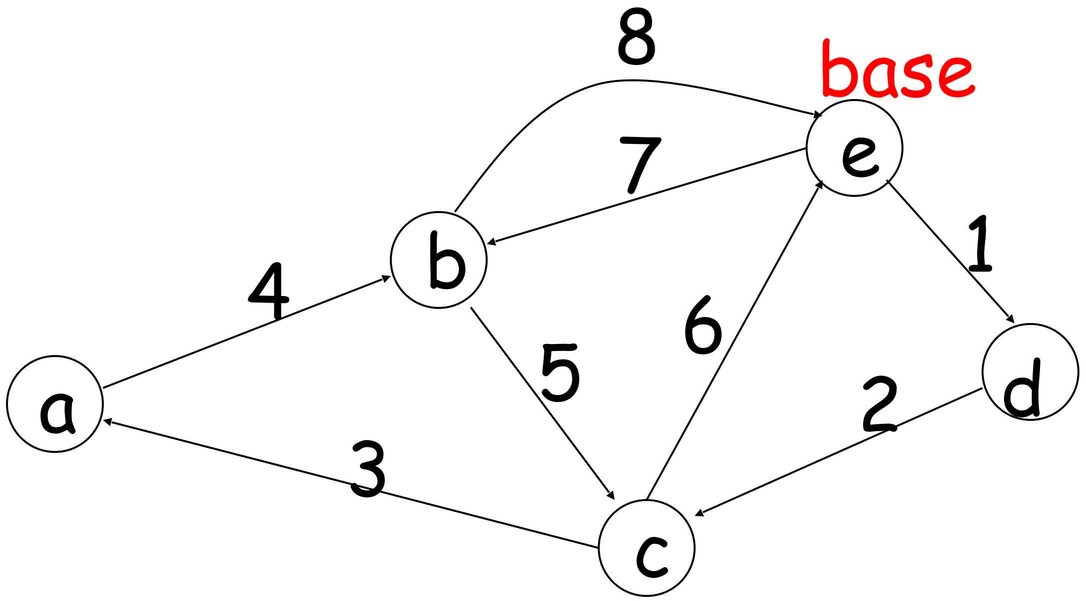
Cycle



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

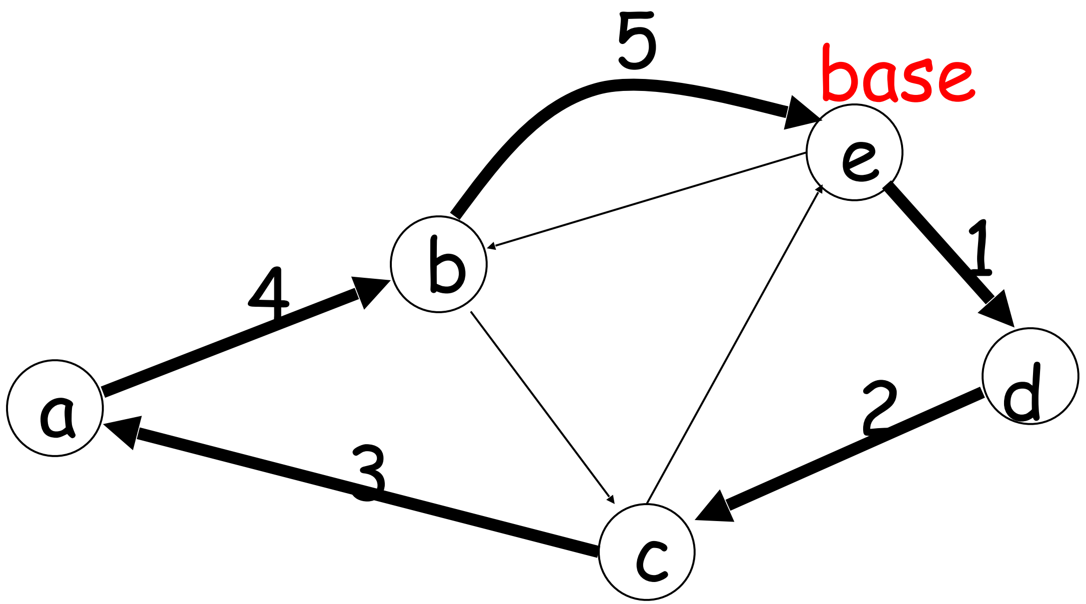
Simple cycle: only the base node is repeated

Euler Tour



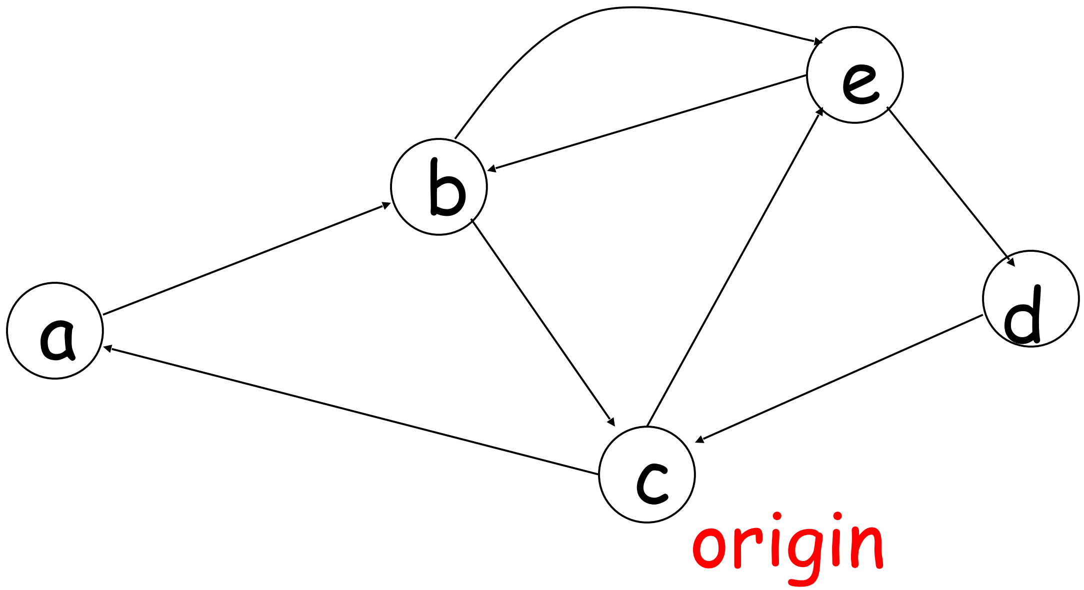
A cycle that contains each edge once

Hamiltonian Cycle

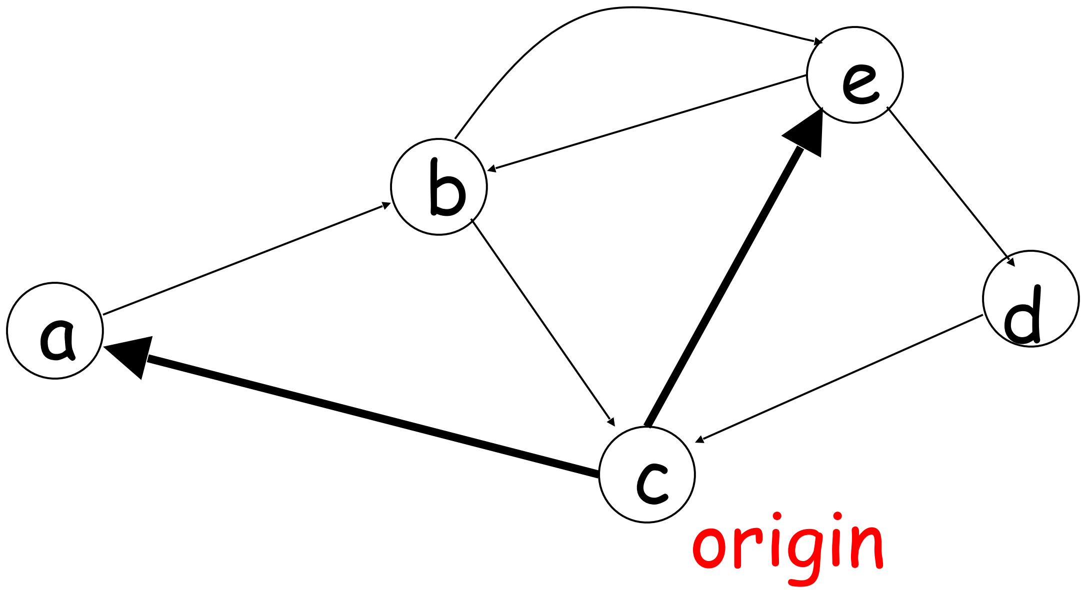


A simple cycle that contains all nodes

Finding All Simple Paths



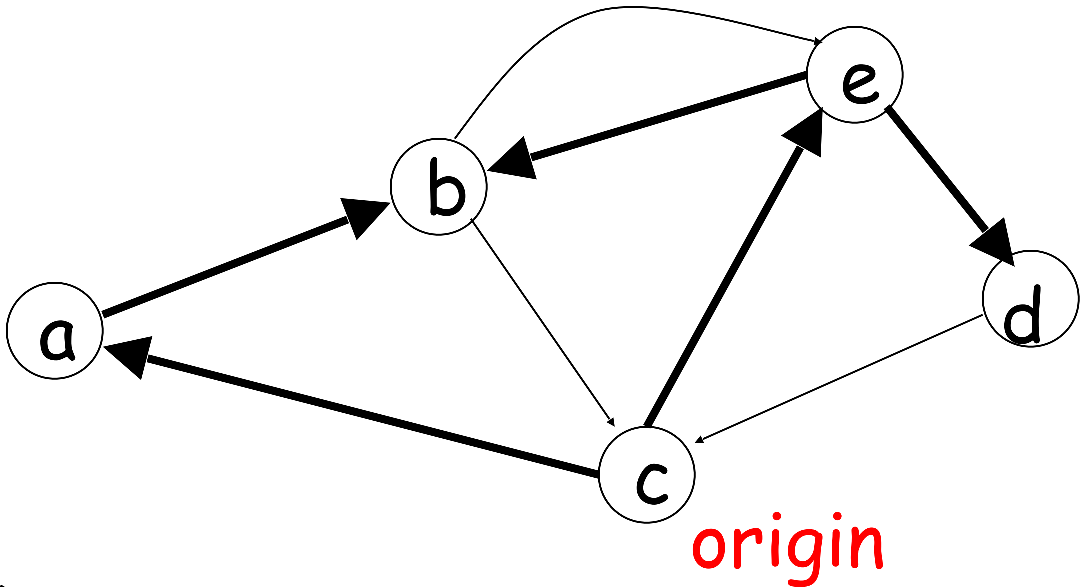
Step 1



(c, a)

(c, e)

Step 2



(c, a)

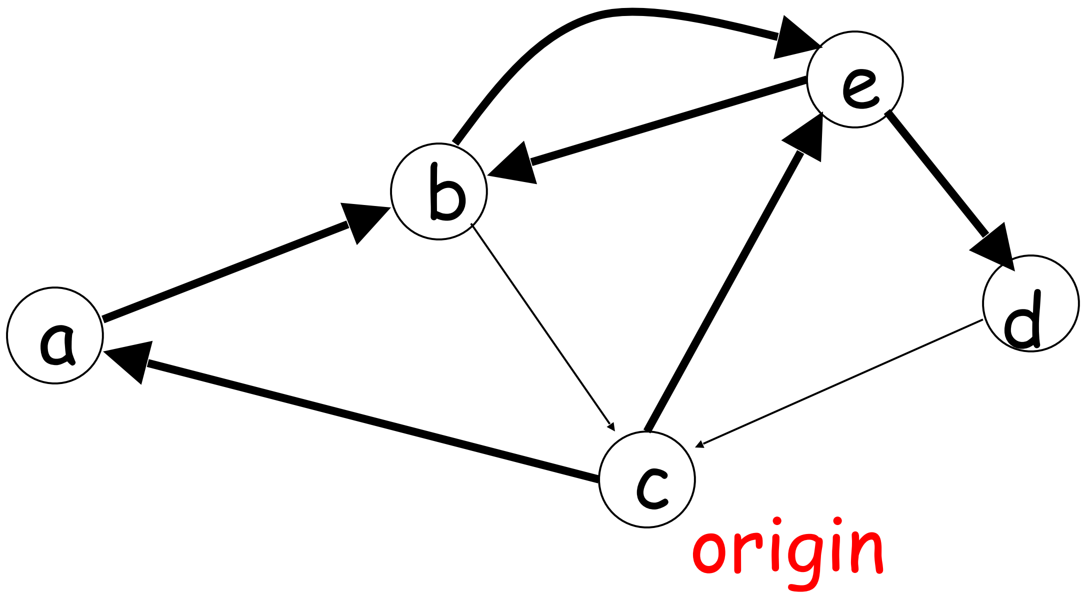
(c, a), (a, b)

(c, e)

(c, e), (e, b)

(c, e), (e, d)

Step 3



(c, a)

(c, a), (a, b)

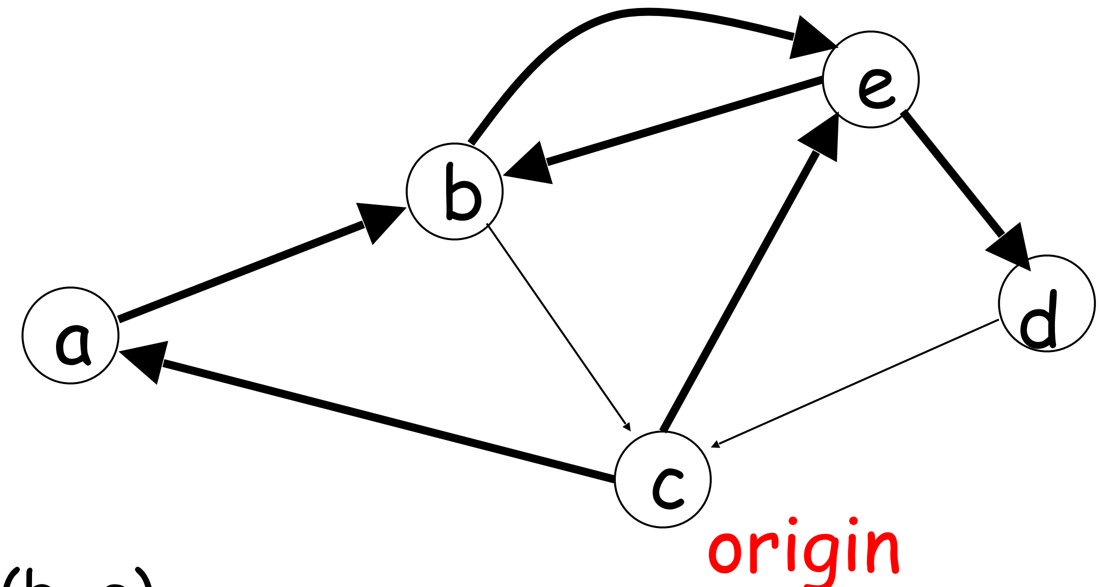
(c, a), (a, b), (b, e)

(c, e)

(c, e), (e, b)

(c, e), (e, d)

Step 4



(c, a)

(c, a), (a, b)

(c, a), (a, b), (b, e)

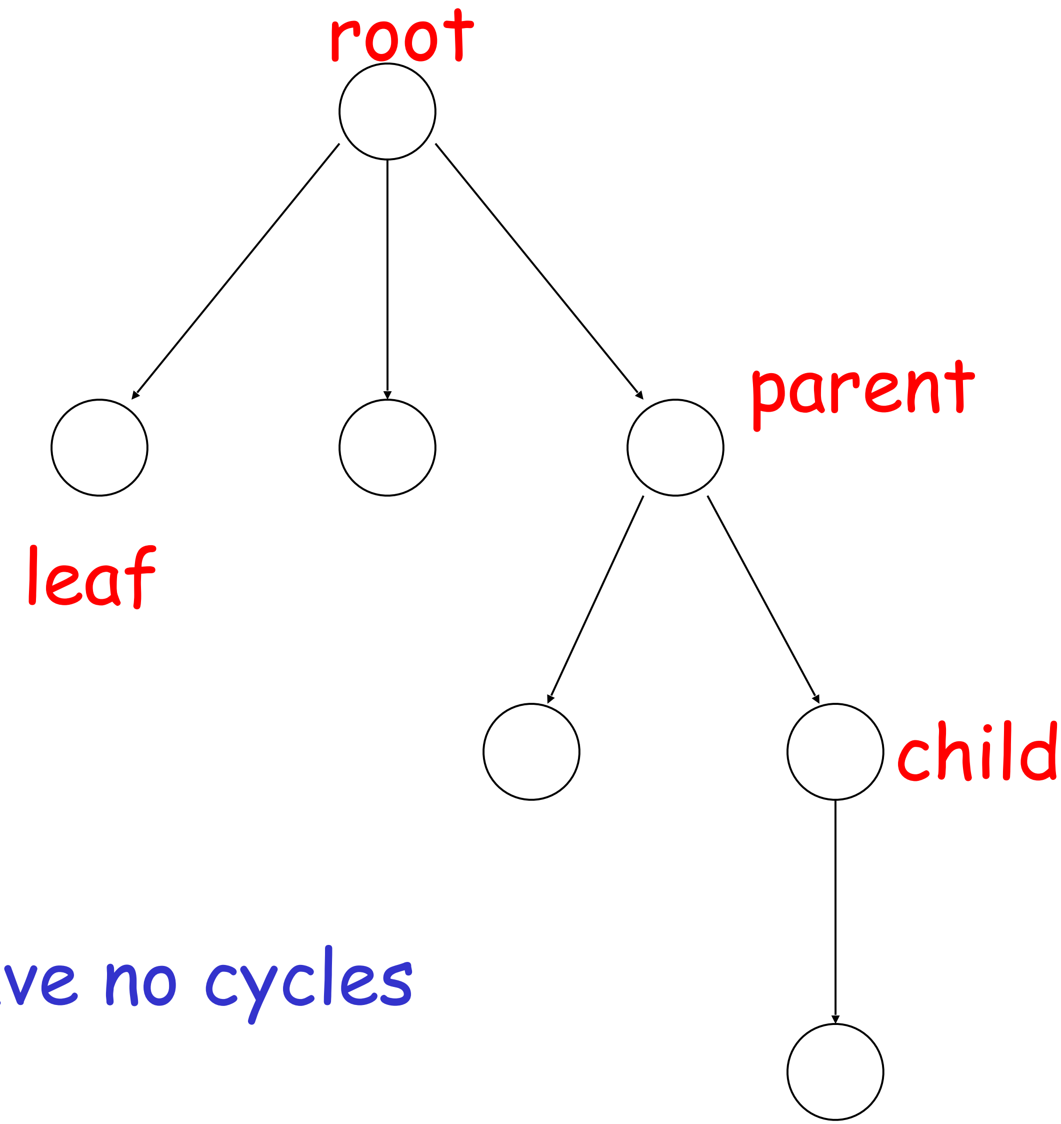
(c, a), (a, b), (b, e), (e,d)

(c, e)

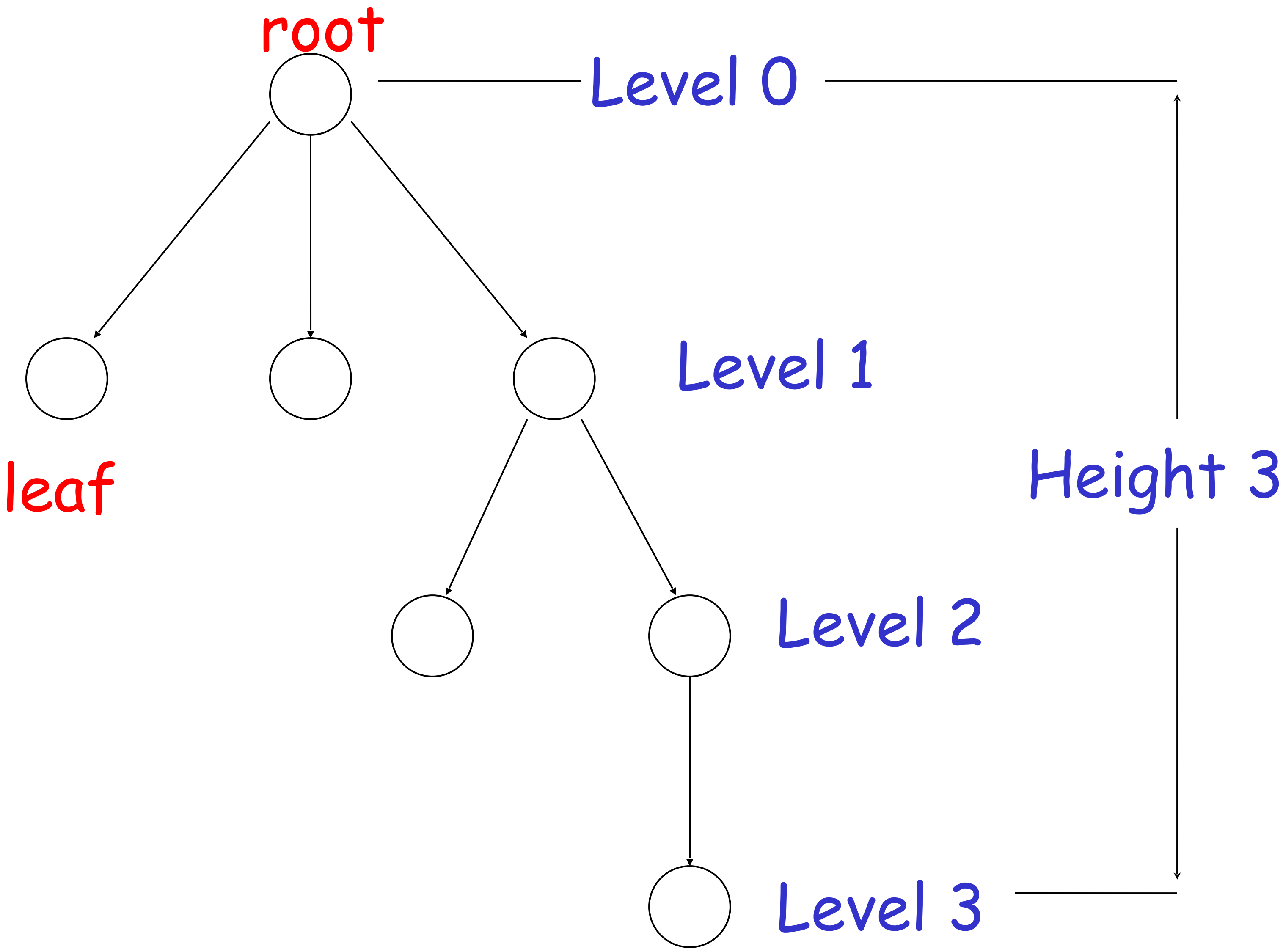
(c, e), (e, b)

(c, e), (e, d)

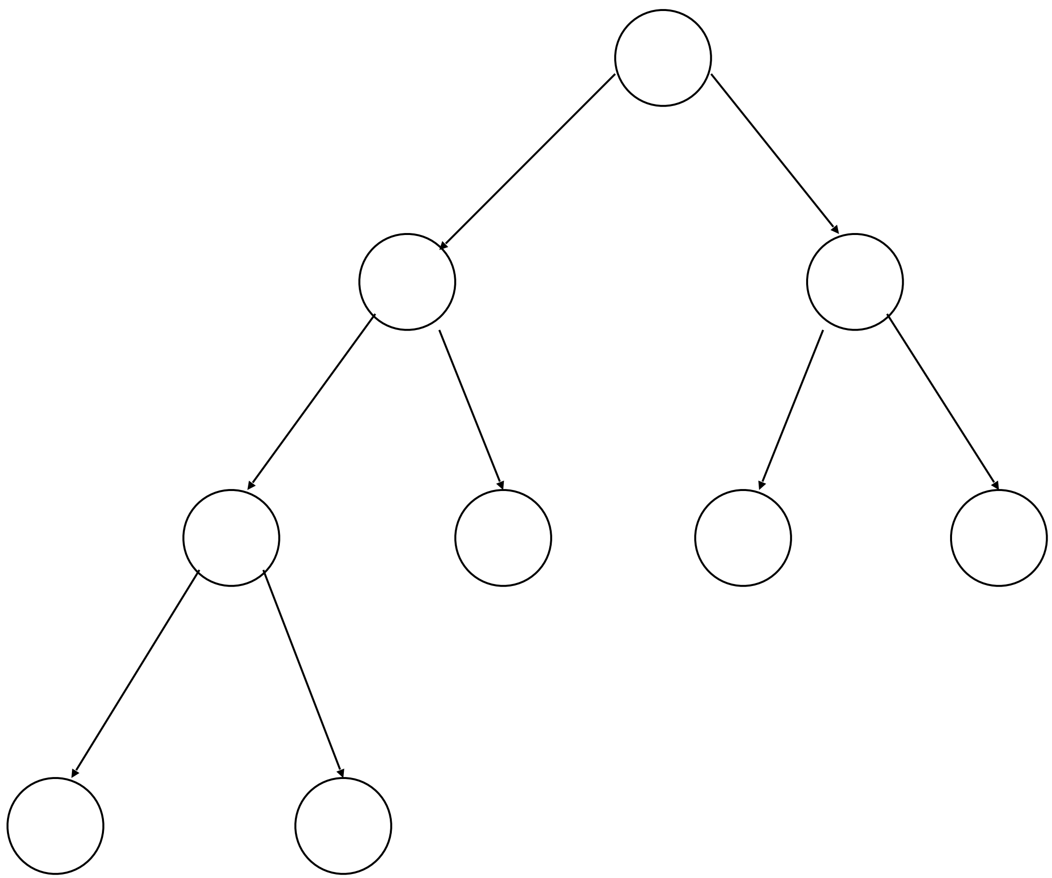
Trees



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

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

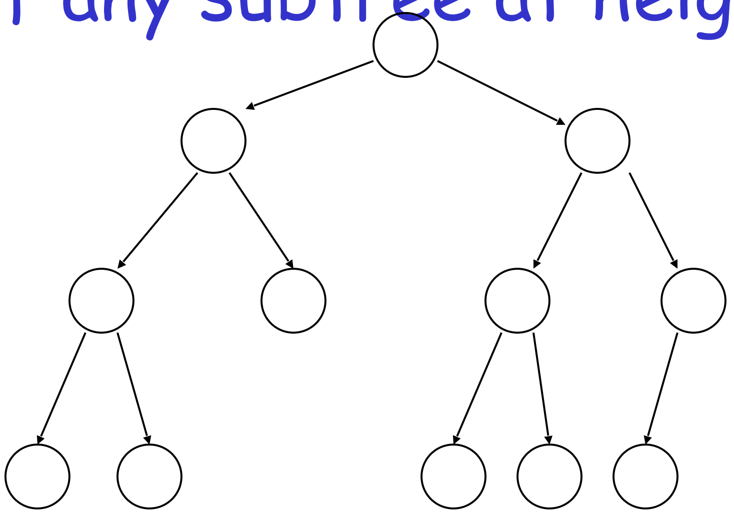
  Show that P_{k+1} is true

Example

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

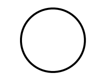
Proof by induction:

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) 

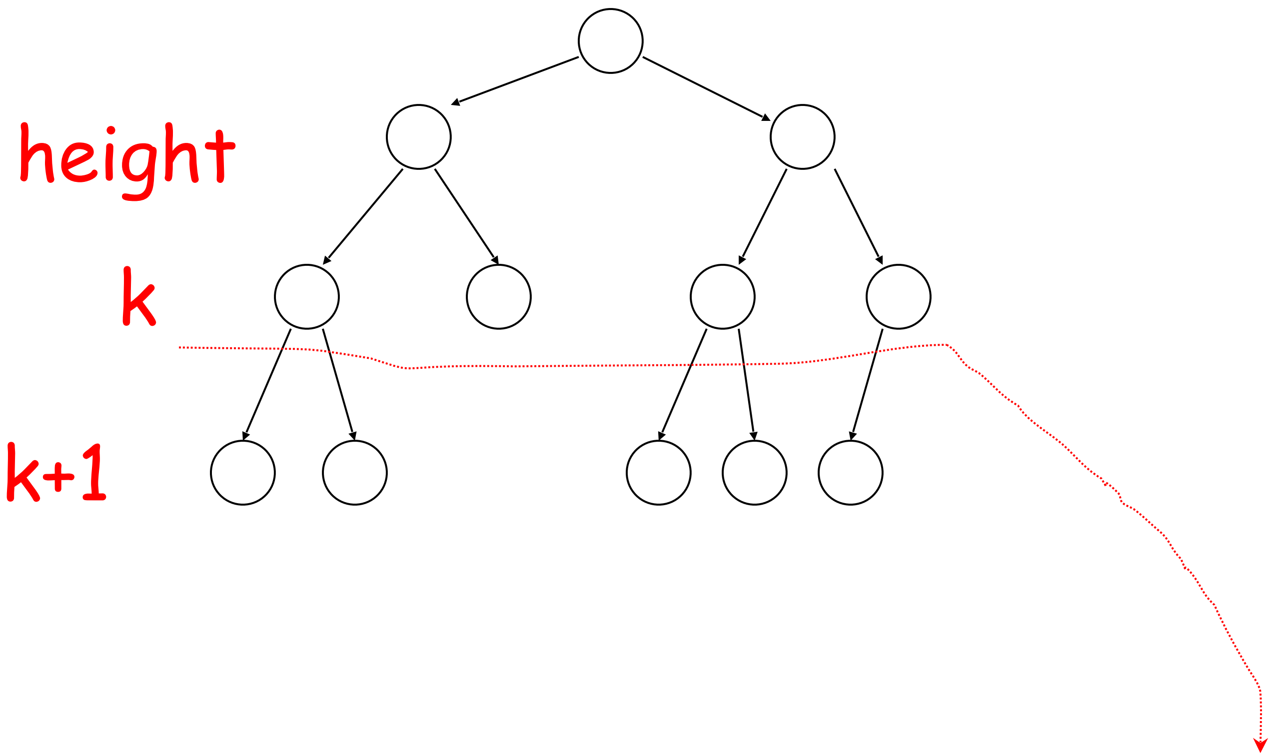
  - Inductive hypothesis

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

  - Induction step

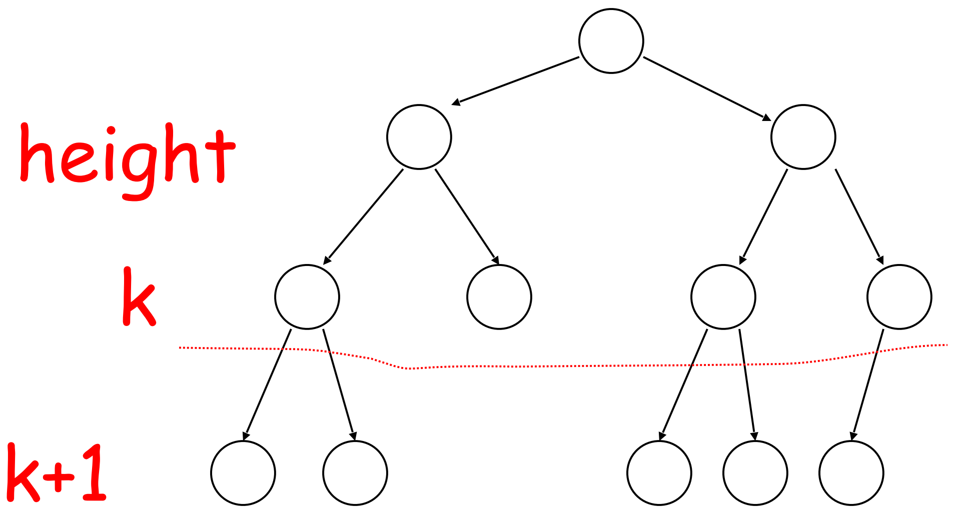
  we need to show that $L(k + 1) <= 2^{k+1}$

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)

Remark

Recursion is another thing

Example of recursive function:

$$f(n) = f(n-1) + f(n-2)$$

$$f(0) = 1, \quad f(1) = 1$$

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!