TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Discrete Mathematics

Nguyễn Khánh Phương

Department of Computer Science
School of Information and Communication Technology
E-mail: phuongnk@soict.hust.edu.vn

Discrete Mathematics

Discrete Mathematics deals with

- "Separated" or discrete sets of objects (rather than continuous sets)
- Processes with a sequence of individual steps (rather than continuously changing processes)

Kind of problems solved by discrete mathematics

- How many ways are there to choose a computer password?
- What is the probability of winning a lottery?
- Is there a link between two users in a social network?
- What is the shortest path between two cities using a transportation system?
- How can a list of integers sorted in increasing order? How many steps are required to do such a sorting?

Importance of Discrete Mathematics

- Information is stored and manipulated by computers in a discrete fashion
- Applications in many different areas
- Discrete mathematics is a gateway to more advanced courses
- Develops mathematical reasoning skills
- Emphasizes the new role of mathematics

The new role of Mathematics

- Make the computer to solve the problem for you
- Modeling (vs. calculations)
- Using logic
  - to choose the right model
  - to write a correct computer program
  - to justify answers
- Efficiency
  - make the computer to solve the problem fast
  - choose the more efficient model

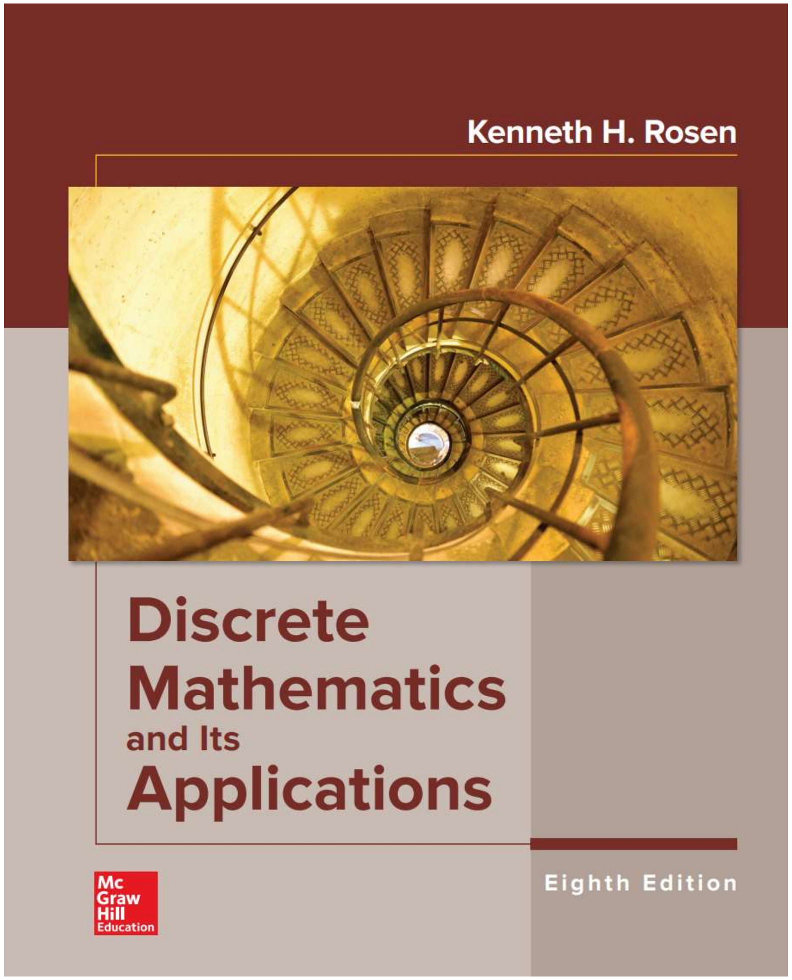
Goals of this course

- Study of standard facts of discrete mathematics

- Development of mathematical reasoning skills (emphasis on modeling, logic, efficiency)

- Discussion of applications

Text book



Rosen K.H. *Discrete Mathematics and its Applications (8th Editions)*. McGraw - Hill Book Company, 2019.

Use lecture notes as study guide.

Text book

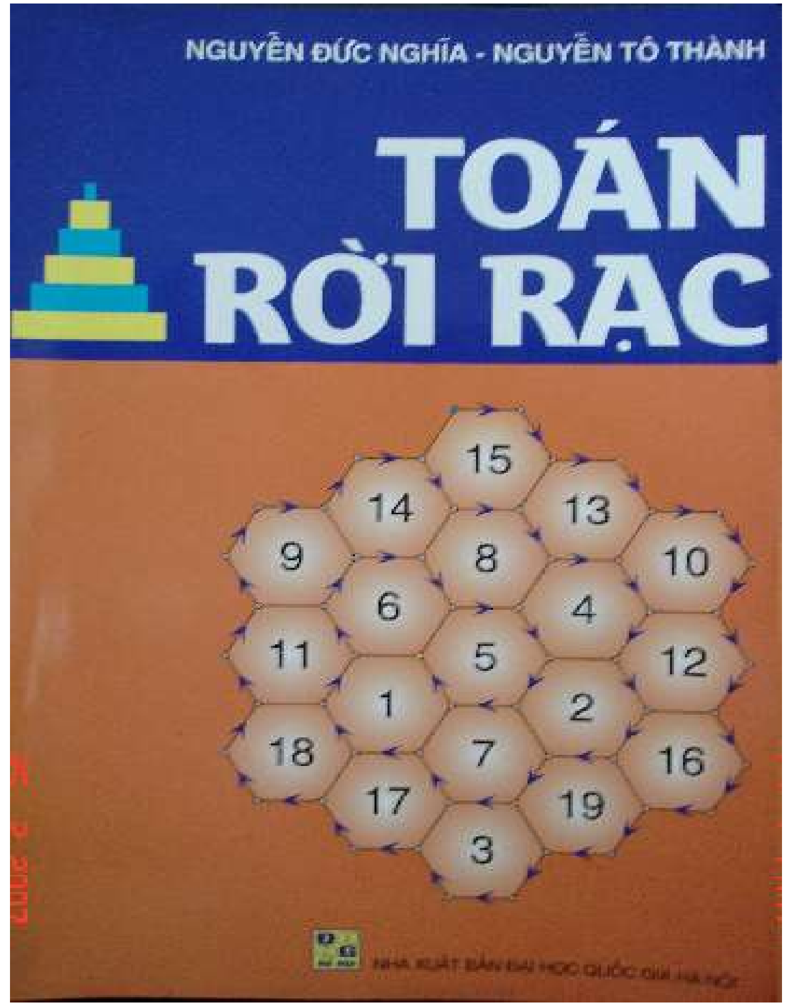
Nguyễn Đức Nghĩa,

Nguyễn Tô Thành

TOÁN RỜI RẠC

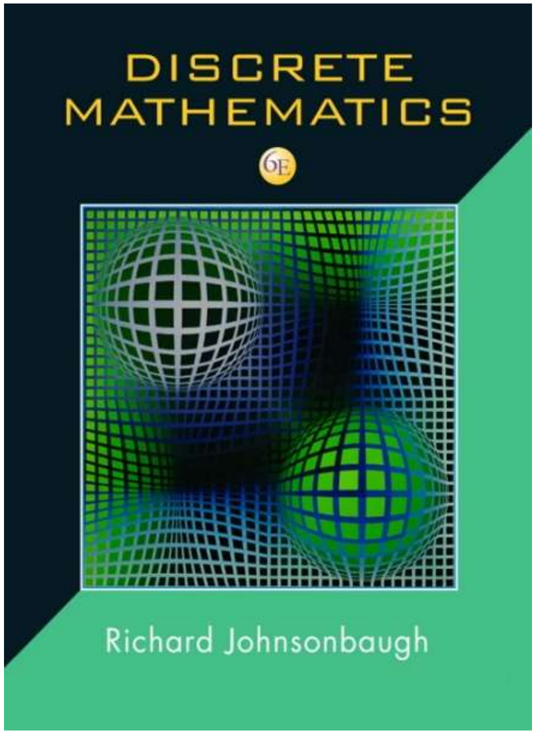
(in lần thứ ba)

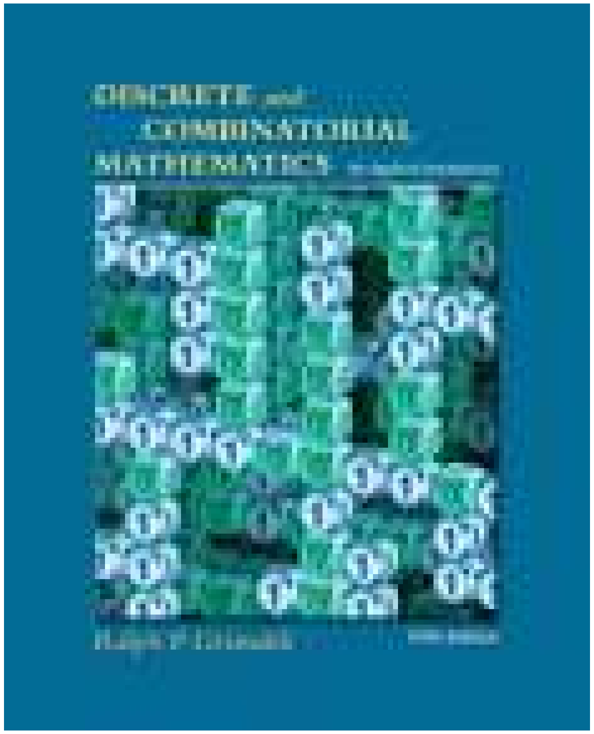
Nhà xuất bản Đại học Quốc gia Hà nội, 2003, 290 trang

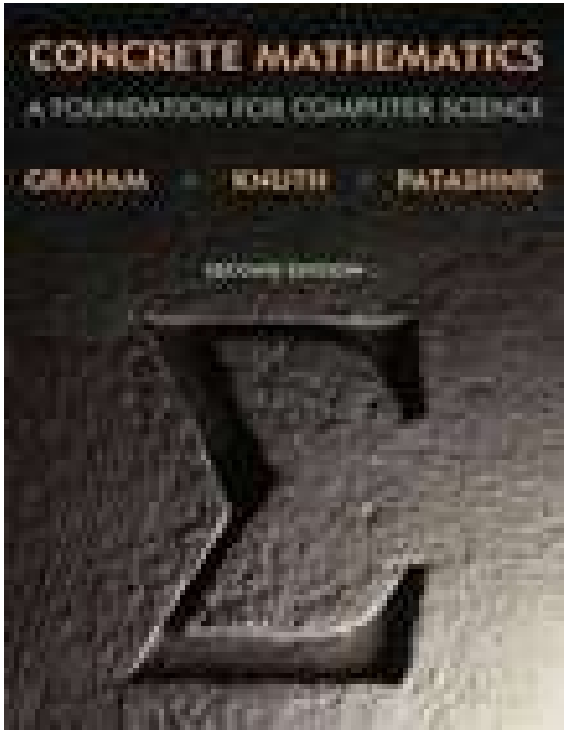


References

1. **Johnsonbaugh R.** *Discrete Mathematics.* Prentice Hall Inc., N. J., 1997.
2. **Grimaldi R.P.** *Discrete and Combinatorial Mathematics (an Applied Introduction),* Addison-Wesley, 5th edition, 2004.
3. **R. Graham, O. Patashnik, and D.E. Knuth.** *Concrete Mathematics*, Second Edition. Addison-Wesley, 1994.







References

4. Nguyễn Hữu Anh. *Toán rời rạc*, NXB Giáo dục,1999.

5. Nguyễn Xuân Quỳnh. *Cơ sở Toán rời rạc và ứng dụng*. NXB KHKT, Hà nội, 1996.

6. Đỗ Đức Giáo. *Toán rời rạc*. NXB KHKT, Hà nội, 2001.

PART 1

COMBINATORIAL THEORY

(Lý thuyết tổ hợp)

PART 2

GRAPH THEORY

(Lý thuyết đồ thị)

Chapter 1. Counting problem

- This is the problem aiming to answer the question: "How many ways are there that satisfy given conditions?" The counting method is usually based on some basic principles and some results to count simple configurations .
- Counting problems are effectively applied to evaluation tasks such as calculating the probability of an event, calculating the complexity of an algorithm (how long the algorithm will take to run), ...

Street art

Given N paintings in a row over a distance of M centimeters.

Each painting i ($1 \leq i \leq N$) will be drawn on a length of t_i cm, so $t_1+t_2+..+t_n = M$.

The K city's most famous artists have been selected to do this work, each artist will be assigned to draw at least one painting. To facilitate the artist's work, if someone is assigned to draw more than one painting, the paintings must be adjacent to each other on the street art

Contents of Part 1: Combinatorial Theory

Chapter 1. Counting problem

- This is the problem aiming to answer the question: “How many ways are there that satisfy given conditions?” The counting method is usually based on some basic principles and some results to count simple configurations .
- Counting problems are effectively applied to evaluation tasks such as calculating the probability of an event, calculating the complexity of an algorithm

Chapter 2. Existence problem

In the counting problem, configuration existence is obvious; in the existence problem, we need to answer the question: "Is there a combinatorial configuration that satisfies given properties ?”

Chapter 3. Enumeration problem

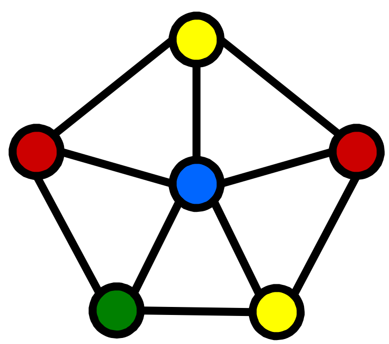
This problem is interested in giving all the configurations that satisfy given conditions.

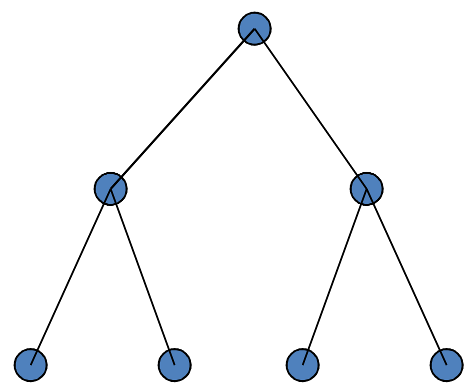
Chapter 4. Combinatorial optimization problem

- Unlike the enumeration problem, this problem only concerns the "best" configuration in a certain sense.
- In the optimization problems, each configuration is assigned a numerical value (which is the use value or the cost to construction the configuration), and the problem is that among the configurations that satisfy the given conditions, find the configuration with the maximum or minimum value assigned to it

Contents of Part 2: Graph Theory

- Graphs
- Degree sequence, Eulerian graphs, isomorphism
- Trees
- Matching
- Coloring



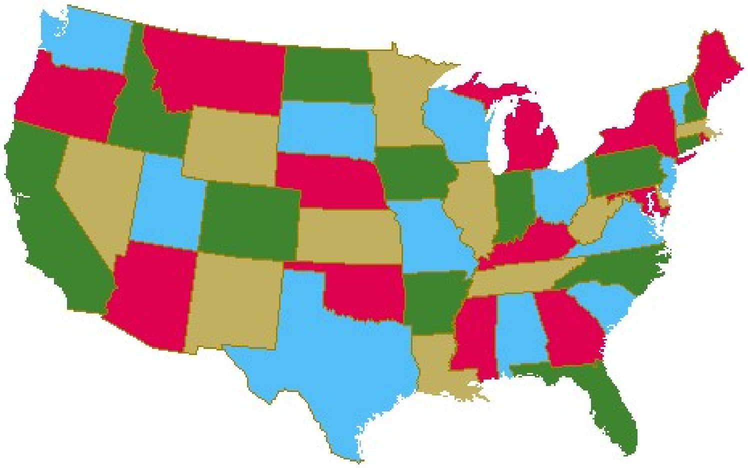


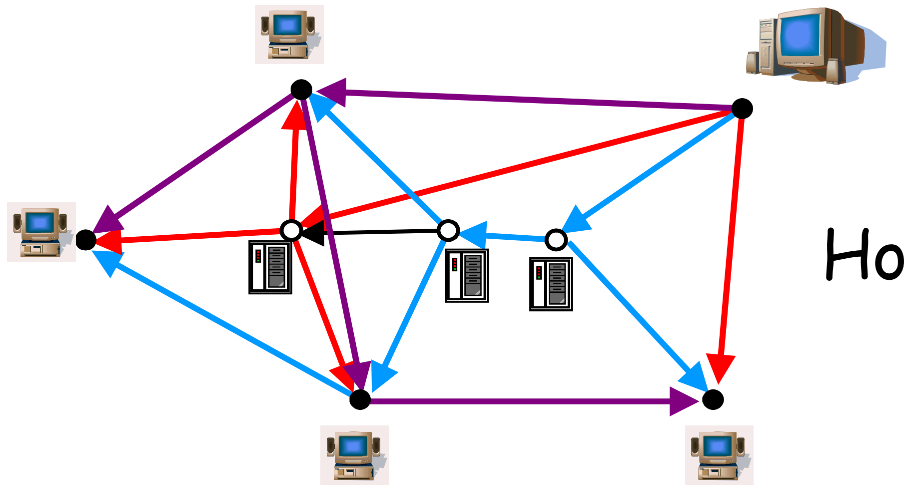
Computer networks, data structures

Contents of Part 2: Graph Theory

How to color a map?

How to schedule exams/projects?





How to send data efficiently?

Contents of Part 1

Chapter 0: Sets, Relations

Chapter 1: Counting problem

Chapter 2: Existence problem

Chapter 3: Enumeration problem

Chapter 4: Combinatorial optimization problem

Contents of Part 1

Chapter 0: Sets, Relations

Chapter 1: Counting problem

Chapter 2: Existence problem

Chapter 3: Enumeration problem

Chapter 4: Combinatorial optimization problem

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

Chapter 0

SETS, RELATIONS

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Contents

1.Definitions

- We have already implicitly dealt with sets
  - Integers (Z), rationals (Q), naturals (N), reals (R), etc.
- We will develop more fully
  - The definitions of sets
  - The properties of sets
  - The operations on sets

1.1 Set and element

1.2. Specification of set

1.1 Set and element

- **Definition**:
  - A set is an unordered collection of (unique) objects
  - The objects in a set are called elements or members of a set.

  A set is said to contain its elements.
- Notation, for a set A:
  - $x \in A$: x is an element of A
  - $x \notin A$: x is not an element of A

Example:

V={a, e, i, o, u} (vowels in English)

C = all students subscribed to IT3020E in Winter 2020

Note:

- We often denote sets with capitals
- Brackets are used to define the set. {.}

1.1 Set and element

- **Definition**: A multi-set is a set where you specify the number of occurrences of each element: $\{m_1 \cdot a_1, m_2 \cdot a_2, \ldots, m_r \cdot a_r\}$ is a set where
  - m_1 occurs a_1 times
  - m_2 occurs a_2 times
  - …
  - m_r occurs a_r times

1. Definitions

1.1 Set and element

1.2. Specification of set

1.2 Specification of set

Our first concern will be how to describe a set; that is, how do we most conveniently describe a set and the elements that are in it? Sets can be defined in various ways.

At first we consider two ways:

1. Set extension
2. Set intension

1.2 Specification of set

- A set is defined in **extension** when you enumerate all the elements:

$$O=\{0,2,4,6,8\}$$

- The **set-builder** notation

$$A = \{x \mid \text{conditions}(x)\}.$$

this could be read as "all x such that the conditions hold true".

Example: $O=\{\, x \mid (x\in Z) \wedge (x=2k) \text{ for some } k\in Z\}$

reads: O is the set that contains all x such that x is an integer and x is even

- A set is defined in **intension** when you give its set-builder notation

$$O=\{\, x \mid (x\in Z) \wedge (0 \le x \le 8) \wedge (x=2k) \text{ for some } k \in Z \,\}$$

1.2 Specification of set

Well-known sets in math:

- N = {0,1,2,3,...}
- Z = {...,-2,-1,0,1,2,...}
- Z^+ = {1,2,3,...}
- Q = {p/q | p in Z, q in Z, q is not 0}
- R = {x | x is a real number}.

{,...} is used to indicate the the rest of the sequence once it's clear how to proceed

Example: {1,2,3,4,...}

1.2 Specification of set

- There is a set with no elements. It is called the ***empty set*** (or *null set*) and denoted $\{\}$ or $\emptyset$.
- A set that has one element is called a ***singleton set***.
  - For example: $\{a\}$, with brackets, is a singleton set
  - a, without brackets, is an element of the set $\{a\}$
- Note the subtlety in $\emptyset \neq \{\emptyset\}$ **??why**
  - The left-hand side is the empty set
  - The right hand-side is a singleton set, and this set contains a set

1.2 Specification of set

- If there are exactly n distinct elements in a set S, with n is a nonnegative integer, we say that:
  - S is a finite set, and
  - The cardinality of S is n. Notation: $|S| = n$.
- **Definition.** A set is a ***finite set*** if it has a finite number of elements. A set that is not finite is an *infinite set*.
- Let A be a finite set. The number of different elements in A is called its *cardinality* and is denoted by $|A|$. Other notations commonly used for the cardinality of A are $N(A)$, $\#A$.

  If A be a infinite set, then we write $|A| = \infty$.

Example:

- $|\varnothing| = 0$ since $\varnothing$ contains no elements.
- $|\{\pi, 2, \text{Newton}\}| = 3$.
- If $N_n = \{0, 1, \ldots, n\}$ then $|N_n| = n + 1$.
- $|\{n: n \text{ is a prime number}\}| = \infty$.
- The sets N, Z, Q, R are all infinite

1.2 Specification of set

- Sets can be elements of other sets

Example:

- $S_1 = \{\varnothing, \{a\}, \{b\}, \{a,b\}, c\}$
- $S_2 = \{\{1\}, \{2,4,8\}, \{3\}, \{6\}, 4,5,6\}$

Example: What is the cardinality of the set ?

- $X = \{\{a, b\}\}$
- $A = \{1, 2, \{a, b\}\}$

1.2 Specification of set

– Let $B = \{x \mid (x \leq 100) \wedge (x \text{ is prime})\}$

the cardinality of B is $|B| = ?$

25 because there are 25 primes less than or equal to 100.

Contents

2. Set operations

2.1 Set comparison

2.2 Venn diagram

2.3 Set operations

2.4 Partition and cover

2.1. Set comparison

- **Definition**: Two sets, A and B, are equal if they contain the same elements. We write A=B.

Example:

- $\{2,3,5,7\}=\{3,2,7,5\}$, because a set is unordered
- Also, $\{2,3,5,7\}=\{2,2,3,5,3,7\}$ because a set contains unique elements
- However, $\{2,3,5,7\} \neq \{2,3\}$

2.1. Set comparison

if $P(x)$ and $Q(x)$ are propositional functions which are true for the same objects x, then the sets they define are equal, i.e.

$$\{x : P(x)\} = \{x : Q(x)\}.$$

Example: there are 2 sets

A = $\{x: (x - 4)^2 = 25\}$

B = $\{x: (x + 1)(x - 9) = 0\}$

Question: A = B ?

Yes: A = B, since the two propositional functions $P(x)$: $(x - 4)^2 = 25$ and $Q(x)$: $(x + 1)(x - 9) = 0$ are true for the same values of x, namely -1 and 5.

2.1. Set comparison

- **Definition**: A is said to be a **subset** of B, if and only if every element of A is also an element of B

that is: $\forall\, x\, (x \in A \Rightarrow x \in B)$

Denote: $A \subseteq B$ or $B \supseteq A$,

Example: $S = \{ 1, 2, 3, \ldots, 11, 12 \}$ and $T = \{ 1, 2, 3, 6 \}$ then $T \subseteq S$.

- **Theorem**: For any set S
  - $\varnothing \subseteq S$ and
  - $S \subseteq S$

2.1. Set comparison

- **Definition**: If A $\subseteq$ B and A $\neq$ B then set A is called a **proper subset** of set B.

(that is there is an element $x \in$ B such that $x \notin$ A)

Denote: A $\subset$ B

Example 1: $A = \{ 1, 2, 3 \}$, $B = \{ 2, 3, 1 \}$, $C = \{ 3 \}$. Then:

$$B = A,\ C \subset A,\ C \subset B.$$

Example 2:

$\{1, 4, 9, 16, ...\} \subseteq \{1, 2, 3, ...\} \subseteq \{0, 1, 2, ...\}$.

$\{1, 4, 9, 16, ...\} \subset \{1, 2, 3, ...\} \subset \{0, 1, 2, ...\}$. ?

Here, we could have used the proper subset symbol $\subset$ to link these three sets instead.

2.1. Set comparison

- You may be asked to show that a set is
  - a subset of,
  - proper subset of, or
  - equal to another set.
- To prove that A is a subset of B, use the equivalence discussed earlier $A \subseteq B \Leftrightarrow \forall x(x \in A \Rightarrow x \in B)$
  - To prove that $A \subseteq B$ it is enough to show that for an arbitrary (nonspecific) element x, $x \in A$ implies that x is also in B.
  - Any proof method can be used.
- To prove that A is a proper subset of B ($A \subset B$), you must prove
  - A is a subset of B **and**
  - $\exists x\ (x \in B) \wedge (x \notin A)$

2.1. Set comparison

- Finally to show that two sets are equal, it is sufficient to show independently (much like a biconditional) that
  - $A \subseteq B$ and
  - $B \subseteq A$
- Logically speaking, you must show the following quantified statements:

$$(\forall x\ (x \in A \Rightarrow x \in B)) \wedge (\forall x\ (x \in B \Rightarrow x \in A))$$

we will see an example later..

2.1. Set comparison

Examples:

- N = {0, 1, 2, 3, …} the set of *natural numbers*.
- Z = {…, −2, −1, 0, 1, 2, …} the set of *integers*.
- Z^+ : the set of positive integers
- Q = {p/q : p, $q \in$ Z and $q \neq 0$} the set of fractions or *rational numbers*.
- Q^+ : the set of positive rational numbers
- R = the set of *real numbers*;
- R^+ : the set of positive real numbers
- C = {$x + iy$: $x, y \in$ R and $i^2 = -1$} the set of *complex numbers*.

Clearly the following subset relations hold amongst these sets:

$$N \subseteq Z \subseteq Q \subseteq R \subseteq C.$$

Question: ??? N = Z^+

Note that N is *not* equal to Z^+ since 0 belongs to the first but not the second

2.1. Set comparison

We shall sometimes use E and O to denote the sets of even and odd integers respectively:

- $E = \{2n : n \in Z\} = \{\ldots, -4, -2, 0, 2, 4, \ldots\}$
- $O = \{2n+1 : n \in Z\} = \{\ldots, -3, -1, 1, 3, 5, \ldots\}$

Universal set **(U)** : contains as subsets all sets relevant to the current task or study. Anything outside the universal set is simply not considered. The universal set is not something fixed for all time -we can change it to suit different contexts.

2. Set operations

2.1 Set comparison

2.2 Venn diagram

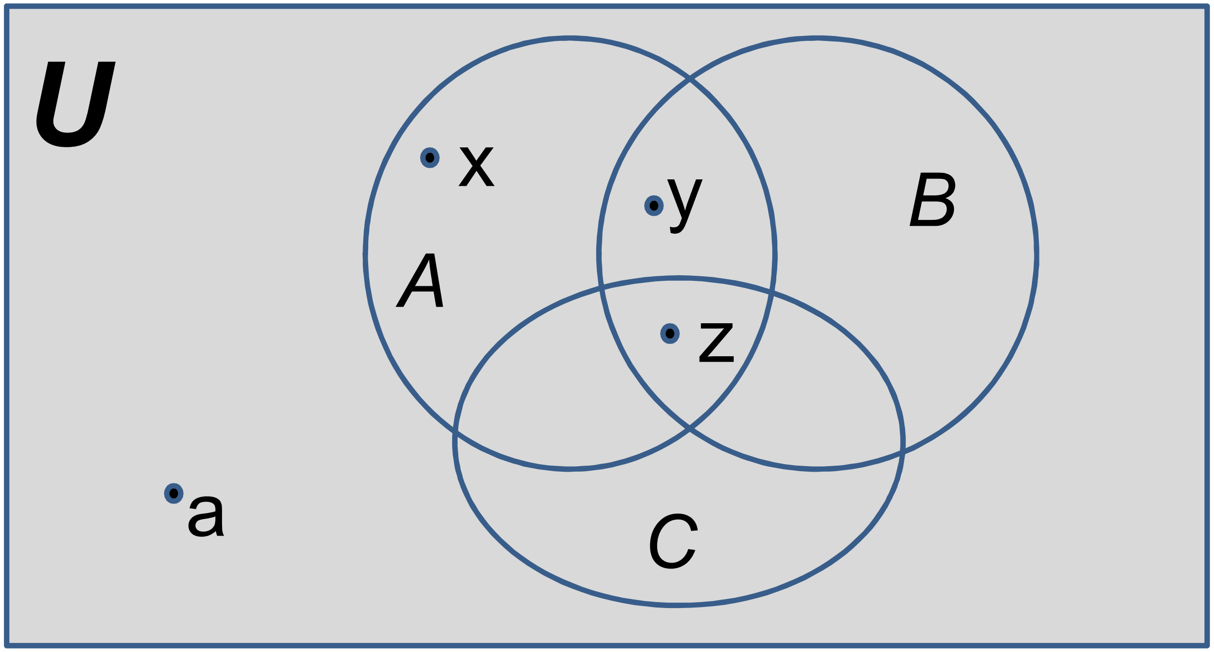
2.3 Set operations

2.4 Partition and cover

2.2. Venn diagram

John Venn
1834-1923

- A set can be represented graphically using a Venn Diagram

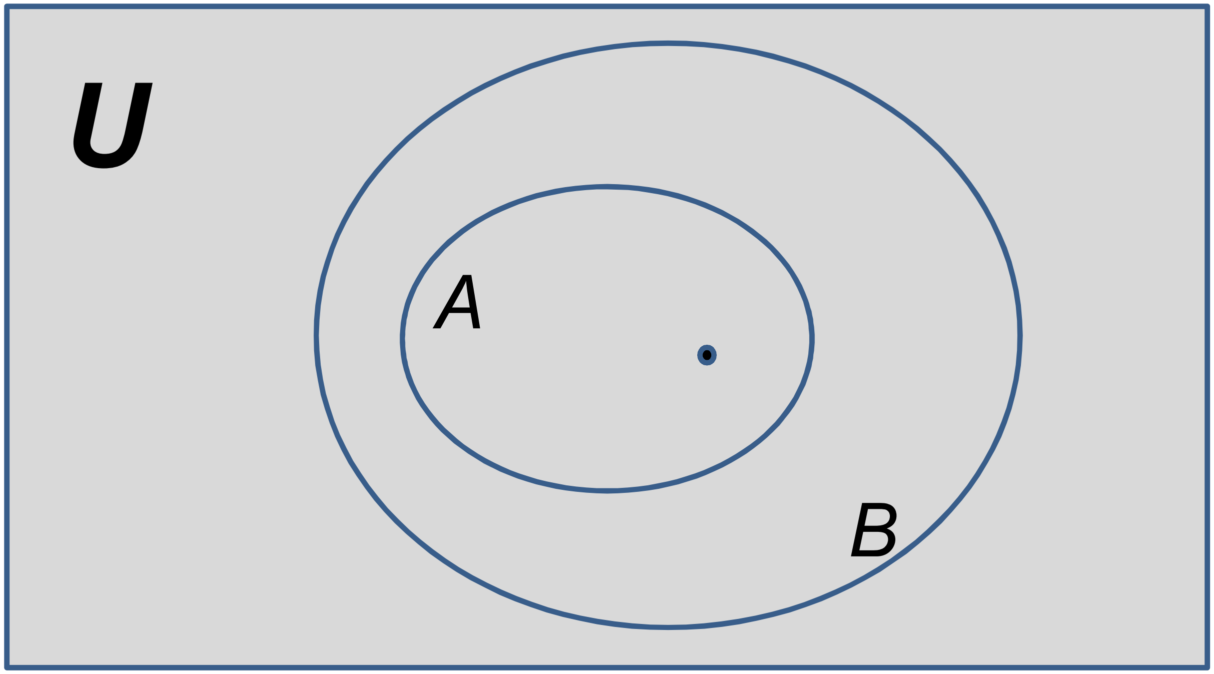


- The universal set U is represented by the interior of a rectangle
- The sets by region inside the rectangle and elements which belong to a given set are placed inside the region representing it.
- If an element belongs to more than one set in the diagram, the two regions representing the sets concerned must overlap and the element is placed in the overlapping region.

In this way the picture represents the relationships between the sets concerned.

2.2. Venn diagram

Example:



if $A \subseteq B$: the region representing A may be enclosed inside the region representing B to ensure that every element in the region representing A is also inside that representing B

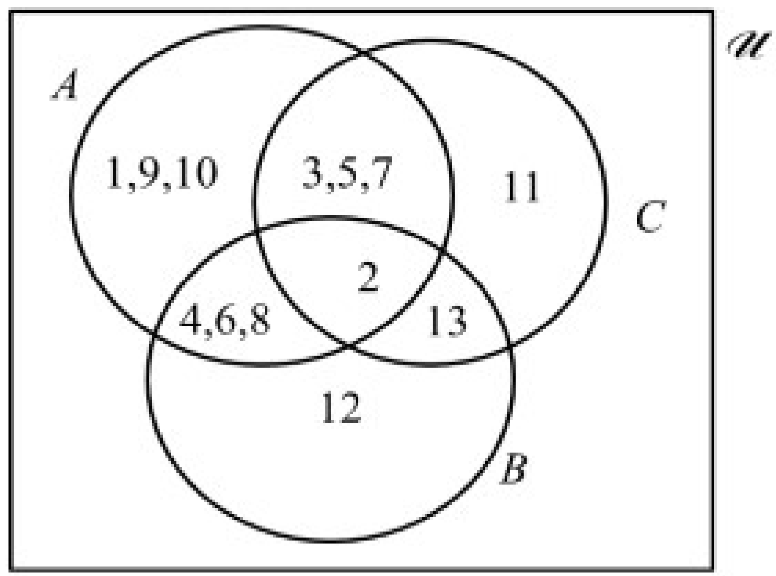
2.2. Venn diagram

Example: Draw the Venn diagram that represents 3 sets:

$A = \{1, 2, \ldots, 10\}$,

$B = \{2, 4, 6, 8, 12, 13\}$,

$C = \{2, 3, 5, 7, 11, 13\}$



2. Set operations

2.1 Set comparison

2.2 Venn diagram

2.3 Set operations

2.4 Partition and cover

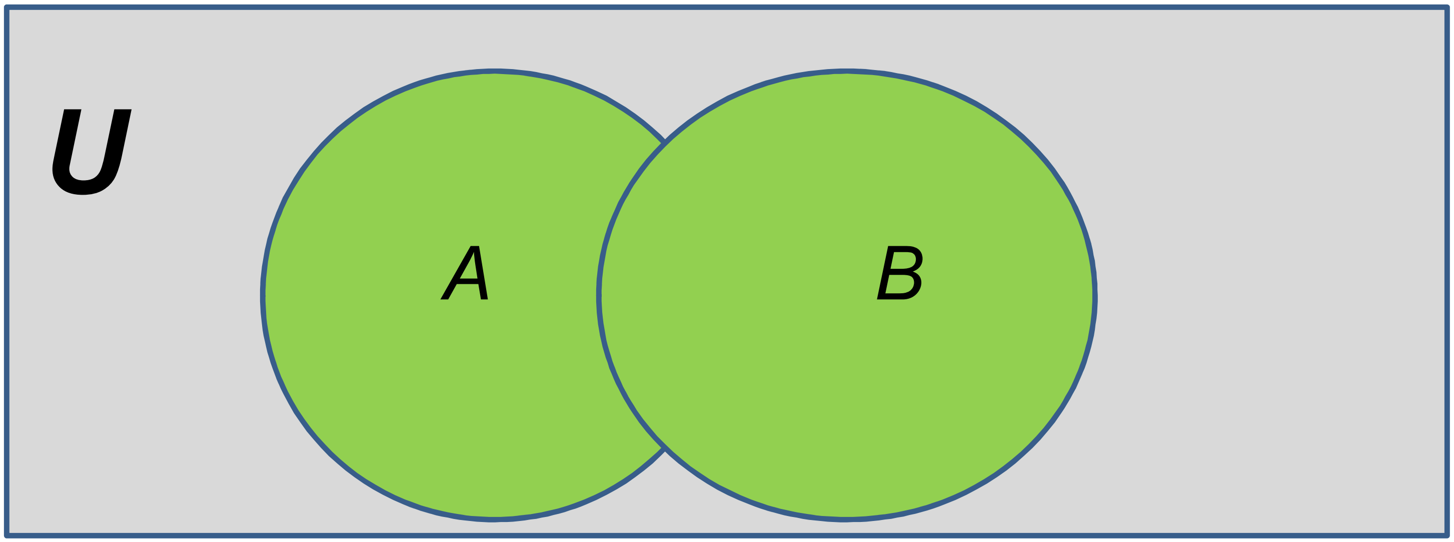
2.3. Set operations

- Arithmetic operators (+,-, × ,÷) can be used on pairs of numbers to give us new numbers
- Similarly, set operators exist and act on two sets to give us new sets:
  1. Union
  2. Generalized union
  3. Intersection
  4. Generalized intersection
  5. Set difference
  6. Set complement

Set Operators: Union

- **Definition**: The union of two sets A and B is the set that contains all elements in A, B, or both. We write:

$$A \cup B = \{\, x \mid (x \in A) \vee (x \in B) \,\}$$



Set Operators: Generalized Union

- **Definition**: The union of a collection of sets is the set that contains those elements that are members of at least one set in the collection

Let A_1, A_2, ..., A_n be sets. Their union is:

$$\bigcup_{i=1}^{n} A_i = A_1 \cup A_2 \cup \ldots \cup A_n$$

$$= \{x \mid x \in A_1 \text{ or } x \in A_2 \text{ or } \ldots \text{ or } x \in A_n\}$$

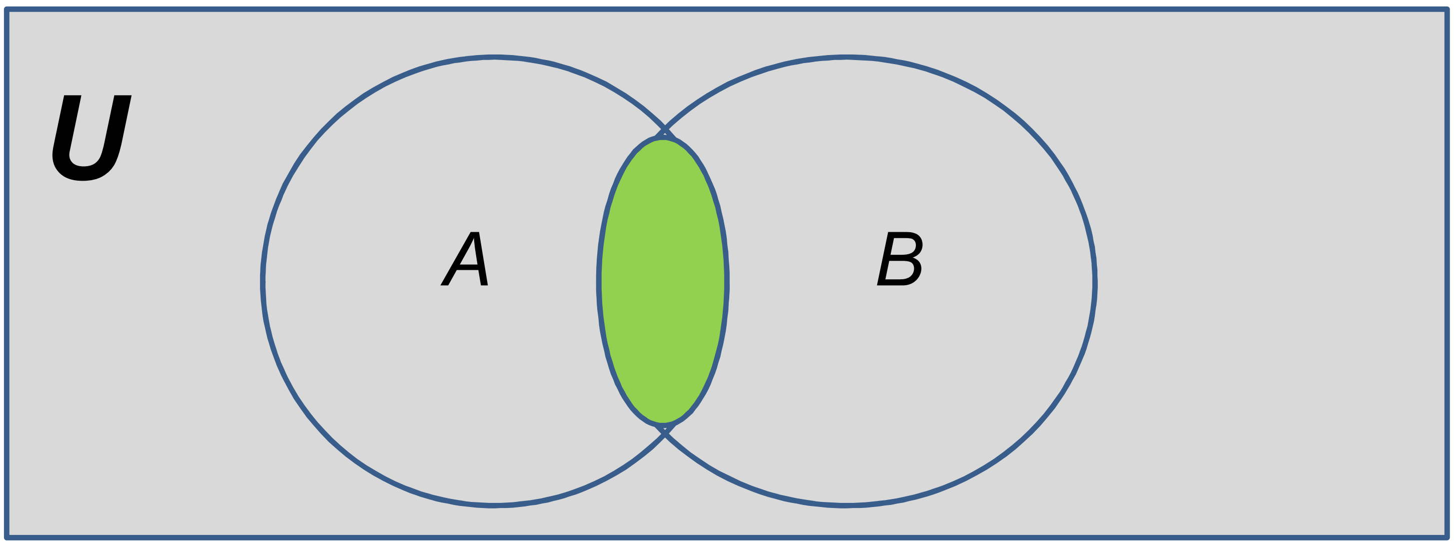
$$= \{x \mid x \text{ belongs to at least one set } A_i, i = 1, \ldots, n\}$$

Set Operators: Intersection

- **Definition**: The intersection of two sets A and B is the set that contains all elements that are element of both A and B.

We write:

$$A \cap B = \{x \mid (x \in A) \wedge (x \in B)\}$$



Two sets A and B are *disjoint* if $A \cap B = \varnothing$.

Set Operators: Generalized Intersection

- **Definition**: The intersection of a collection of sets is the set that contains those elements that are members of every set in the collection

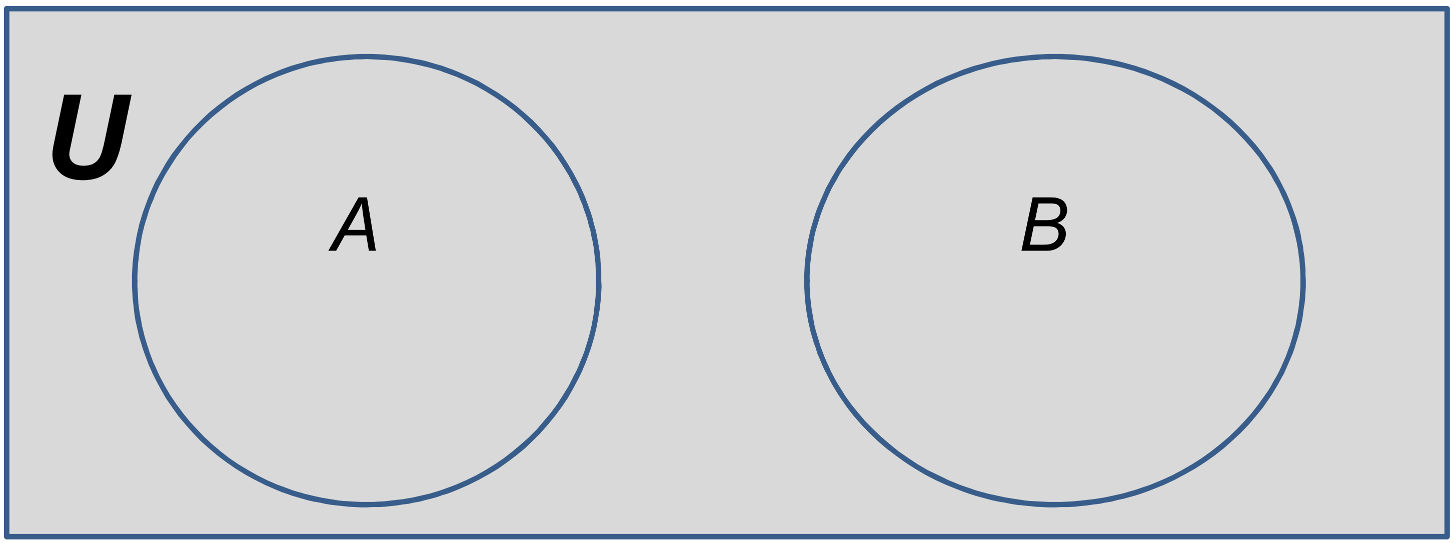
$$\bigcap_{i=1}^{n} A_i = A_1 \cap A_2 \cap \ldots \cap A_n$$

$$= \{x | x \in A_i \; for \; all \; i = 1, 2, .., n\}$$

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Disjoint Sets

- **Definition**: Two sets are said to be disjoint if their intersection is the empty set: $A \cap B = \varnothing$



Set Operators: Generalized Intersection

- $\bigcap_{i \in I} A_i = \{x | x \in A_i \text{ for all } i \in I\}$ is used for the intersection of the family of sets A_i indexed by the set I.
- A collection of sets $\{A_i \mid i \in I\}$ is *disjoint* if

$$\bigcap_{i \in I} A_i = \emptyset$$

A collection of sets is *pairwise disjoint* (or *mutually disjoint*) if every pair of sets in the collection are disjoint.

Example:

$A = A_1 \cup A_2 \cup A_3 \cup A_4$

$A_1 = \{1, 3, 5, 7\}$

$A_2 = \{2, 4, 6\}$

$A_3 = \{-1, -3, -5, -7\}$

$A_4 = \{-2, -4, -6\}$

$A_5 = \{2\}$

$B = A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5$

? A is not disjoint

? A is disjoint

? A is pairwise disjoint

? B is not disjoint

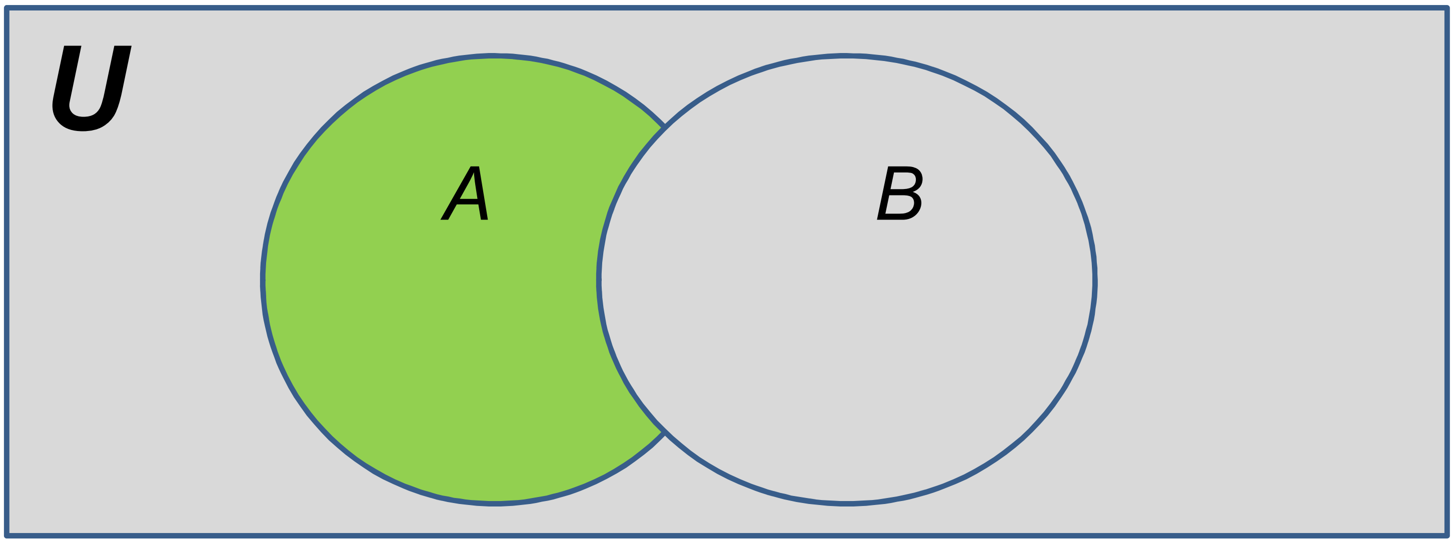
? B is disjoint

? B is pairwise disjoint

Set Operators: Set Difference

- **Definition**: The difference of two sets A and B is the set containing those elements that are in A but not in B.

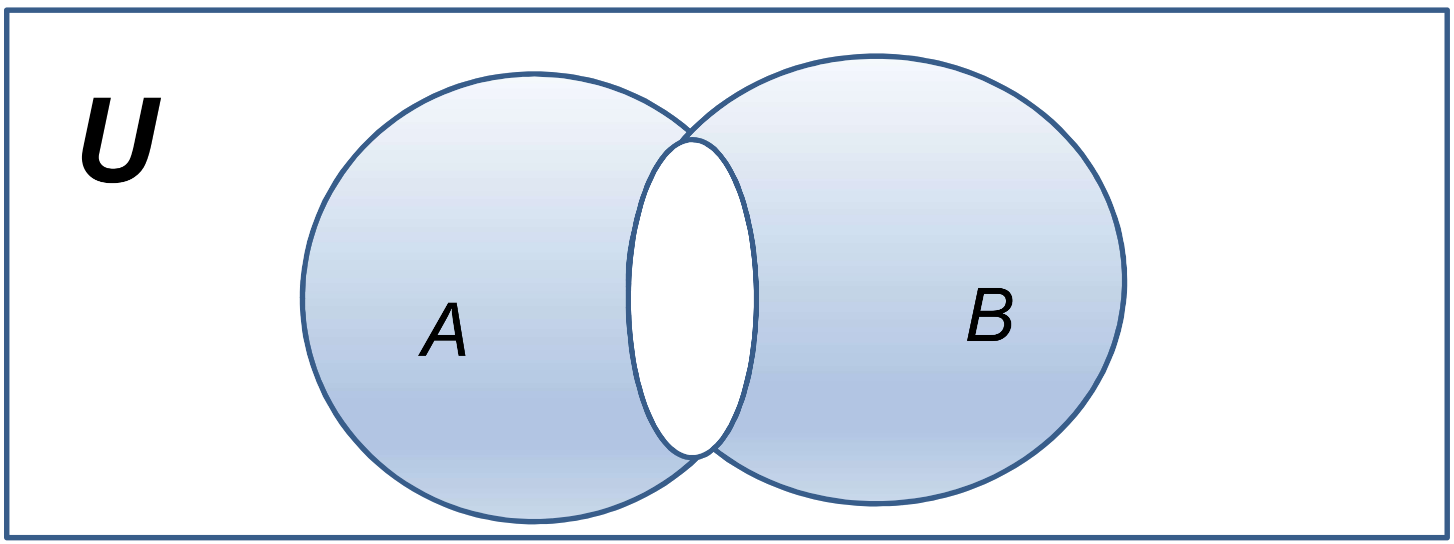
Denote: A\B or A−B



Set Operators: Set Difference

- The *symetric difference* of two sets A and B, denoted $A \oplus B$, is defined as follows

$$A \oplus B = (A \cup B) \setminus (A \cap B)$$

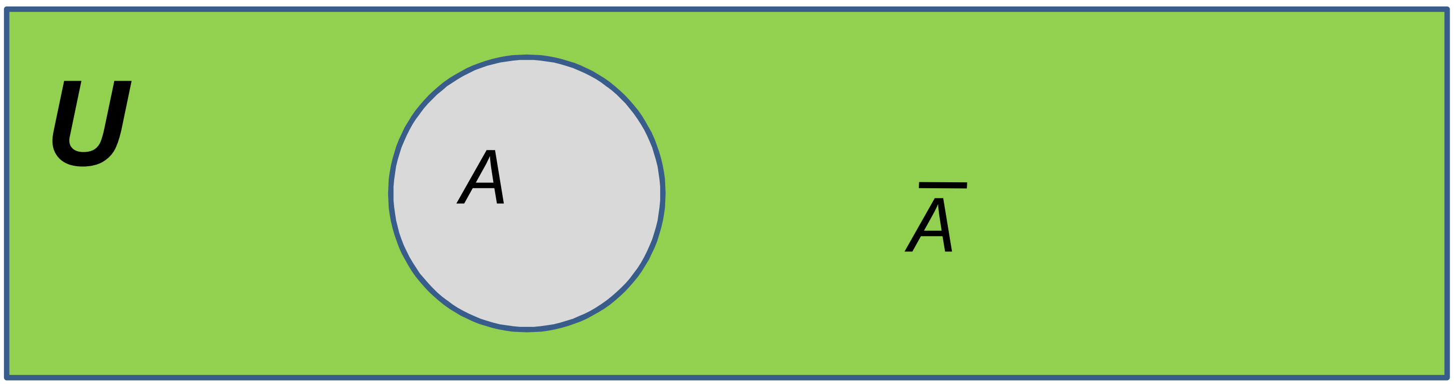


NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Set Operators: Set Complement

- **Definition**: The complement of a set A, denoted $\bar{A}$, or A^c or ¬ A consists of all elements not in A. That is the difference of the universal set and U: U\A

$$\bar{A} = A^C = \{x \mid x \notin A\}$$



$$A - B = \{x \mid x \in A \text{ and } x \notin B\} = A \cap B^c.$$

Set Complement

Examples:

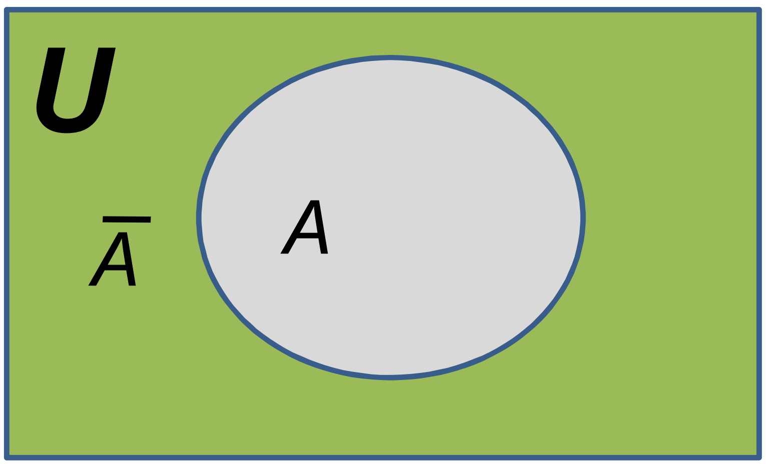
Let A=$\{1, 2, 3\}$, $B = \{3, 4, 5\}$. Then

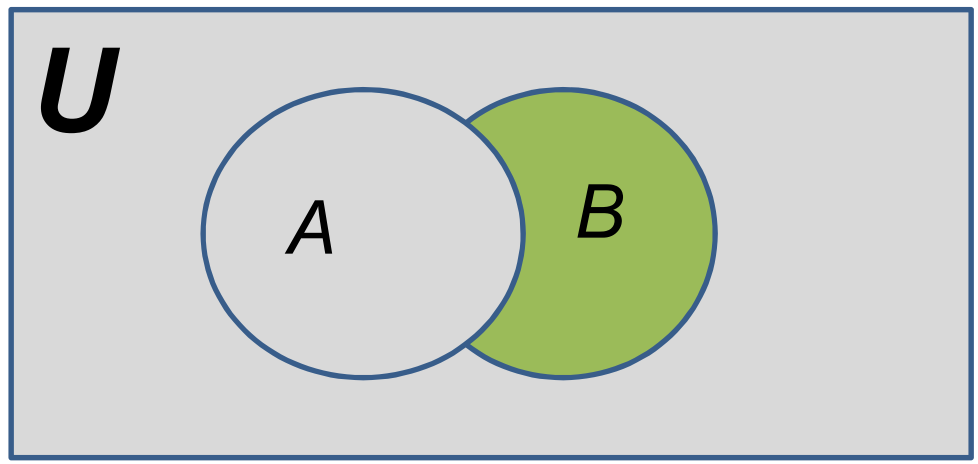
- $A \cup B =$
- $A \cap B =$
- $A \setminus B =$
- $A \oplus B =$

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Set Complement: Absolute & Relative

- Given the Universe U, and A,B $\subset$ U.
- The (absolute) complement of A is $\bar{A} = U \backslash A$
- The (relative) complement of A in B is B\A





2. Set operations

2.1 Set comparison

2.2 Venn diagram

2.3 Set operations

2.4 Partition and cover

2.4. Partition and cover

Let $\mathcal{E} = \{E_i\}_{i \in I}$ be a collection of subsets of the set M, $E_i \subseteq M$.

Collection $\mathcal{E}$ will be called a cover of M if each element of M must be an element of at least one of the sets of $\mathcal{E}$:

$$M \subset \bigcup_{i \in I} E_i \Leftrightarrow \forall x \in M \ \exists i \in I \ x \in E_i .$$

The disjoint cover $\mathcal{E}$ of M is called a partition of M, i.e.

$$\mathcal{E} \text{ is a partition of } M \Leftrightarrow M = \cup_{i \in I} E_i,\ E_i \cap E_j = \varnothing,\ i \neq j.$$

Example: M = {1, 2, 3, 4}

$\mathcal{E}_1 = \{\{1, 2\}, \{3, 4\}\}$ is a partition of M

$\mathcal{E}_2 = \{\{1, 2, 3\}, \{3, 4\}\}$ is not a partition of M

2.4. Partition and cover

Example: M= {11, 12, 13, 14}

$\mathcal{E}_1$ = {{11, 12}, {11, 13}, {12, 14}}

$\mathcal{E}_2$ = {{11, 12}, {13, 14}}

$\mathcal{E}_3$ = {{11, 12}, {13}}

$\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$ is cover / partition of M ?

Contents

Contents

3.1. Power Set

- **Definition**: The power set of a set A, denoted P(A), is the set of all subsets of A.

Examples

Let A = $\{\varnothing\}$ → P (A) = $\{\varnothing\}$

Let A = {a} → P (A) = $\{\varnothing, \{a\}\}$

Let A = $\{a, b\}$ → P(A)= $\{\varnothing, \{a\}, \{b\}, \{a, b\}\}$

Let A = $\{a, b, c\}$ → P(A) = ?

$\{$
$\varnothing,$
$\{a\}, \{b\}, \{c\},$
$\{a, b\}, \{a, c\}, \{b, c\},$
$\{a, b, c\}$
$\}$

- Note: the empty set ∅ and the set itself are always elements of the power set.

3.1. Power Set

- **Theorem**: Let A be a set such that $|A|=n$, then

$$|P(A)| = 2^n$$

Proof: Let A be the set $\{a_1, a_2, ..., a_n\}$.

- We can form a subset of A by considering each element a_i in turn and either including it or not in the subset.
- For each element there are two choices (either include it or don't) and the choice for each element is independent of the choices for the other elements, so there are 2^n choices altogether.
- Each of these 2^n choices gives a different subset and every subset of A can be obtained in this way.

Let $A = \{a, b, c\} \rightarrow P(A) = ?$

0 1 0 1 0 1

$\{$
$\emptyset,$
$\{a\}, \{b\}, \{c\},$
$\{a, b\}, \{a, c\}, \{b, c\},$
$\{a, b, c\}$
$\}$

3.1. Power Set

Let $A = \{a, b, c\}$ and $B = \{a, b\}$. Determine whether each of the following is true or false and give a brief justification.

1. $B \in \mathrm{P}(A)$
2. $B \in A$
3. $A \in \mathrm{P}(A)$
4. $A \subseteq \mathrm{P}(A)$
5. $B \subseteq \mathrm{P}(A)$
6. $\{\{a\}, B\} \subseteq \mathrm{P}(A)$
7. $\varnothing \in \mathrm{P}(A)$
8. $\varnothing \subseteq \mathrm{P}(A)$.

3.1. Power Set

Theorem.

For all sets A and B:

1. $A \subseteq B$ if and only if $P(A) \subseteq P(B)$.
2. $P(A) \cap P(B) = P(A \cap B)$.
3. $P(A) \cup P(B) \subseteq P(A \cup B)$.

3.1. Power Set

Theorem.

For all sets A and B:

1. $A \subseteq B$ if and only if $P(A) \subseteq P(B)$.

We prove the two statements:

$A \subseteq B \Rightarrow P(A) \subseteq P(B)$ and

$P(A) \subseteq P(B) \Rightarrow A \subseteq B$.

- Firstly, suppose $A \subseteq B$. Let $X \in P(A)$. This means $X \subseteq A$. Since $A \subseteq B$, it follows that $X \subseteq B$, which means that $X \in P(B)$. Since $X \in P(A)$ implies $X \in P(B)$, we conclude that $P(A) \subseteq P(B)$, which completes the proof of the first statement.
- To prove the converse statement, suppose $P(A) \subseteq P(B)$. Since $A \in P(A)$, it follows that $A \in P(B)$. This means that $A \subseteq B$, which completes the proof.

Contents

3.2 Properties of set operations

Let A, B and C be any sets. The following laws hold

Equality	**Name**
$A \cup \varnothing = A$ $A \cap U = A$	Identity laws (Đồng nhất)
$A \cup U = U$ $A \cap \varnothing = \varnothing$	Domination laws (Trội)
$A \cup A = A$ $A \cap A = A$	Idempotent laws (Lũy đẳng)
$A \cup \bar{A} = U$ $A \cap \bar{A} = \varnothing$ $\overline{\varnothing} = U$ $\overline{U} = \varnothing$	Complementation laws (Bù)
$\overline{(\overline{A})} = A$	Involution laws (Bù kép)

3.2 Properties of set operations

Equality	**Name**
$A \cup B = B \cup A$ $A \cap B = B \cap \mathrm{A}$	Commutative laws (Giao hoán)
$A \cup (B \cup C) = (A \cup B) \cup C$ $A \cap (B \cap C) = (A \cap B) \cap C$	Associative laws (Kết hợp)
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	Distributive laws (Phân phối)
$\overline{A \cup B} = \overline{A} \cap \overline{B}$ $\overline{A \cap B} = \overline{A} \cup \overline{B}$	De Morgan's laws (Luật De Morgan)
$A \cap (A \cup B) = A$ $A \cup (A \cap B) = A.$	Absorption laws

Proving Set Equivalences

To prove set equivalence

$$A = B,$$

We could use common techniques:

1. Proof $A \subseteq B$ and $B \subseteq A$.
2. Using definitions and equivalence of logical propositions that define the set.
3. Use the truth table.

Proving Set Equivalences

To prove set equivalence

$$A = B,$$

We could use common techniques:

1. **Proof $A \subseteq B$ and $B \subseteq A$.**
2. Using definitions and equivalence of logical propositions that define the set.
3. Use the truth table.

Proving Set Equivalences

- Recall that to prove such identity A = B, we must show that:
  1. The left-hand side is a subset of the right-hand side
  2. The right-hand side is a subset of the left-hand side
  3. Then conclude that the two sides are thus equal

Example: Let

- A=$\{x \mid x$ is even$\}$
- B=$\{x \mid x$ is a multiple of 3$\}$
- C=$\{x \mid x$ is a multiple of 6$\}$

- Show that $A \cap B = C$

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Show that $A \cap B = C$

Example: Let

- $A = \{x \mid x \text{ is even}\}$
- $B = \{x \mid x \text{ is a multiple of } 3\}$
- $C = \{x \mid x \text{ is a multiple of } 6\}$

- Show that $A \cap B = C$

- The left-hand side is a subset of the right-hand side

$A \cap B \subseteq C$: $\forall x \in A \cap B$

$\Rightarrow x$ is a multiple of 2 and x is a multiple of 3

$\Rightarrow$ we can write $x = 2*3*k$ for some integer k

$\Rightarrow x = 6k$ for some integer $k \Rightarrow x$ is a multiple of 6

$\Rightarrow x \in C$

- The right-hand side is a subset of the left-hand side

$C \subseteq A \cap B$: $\forall x \in C$

$\Rightarrow x$ is a multiple of 6 $\Rightarrow x = 6k$ for some integer k

$\Rightarrow x = 2(3k) = 3(2k) \Rightarrow x$ is a multiple of 2 and of 3

$\Rightarrow x \in A \cap B$

Proving Set Equivalences

Example 2: Proof that: $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$.

- **Part 1:** Proof $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.
  - Assume $x \in A \cap (B \cup C)$, need to proof $x \in (A \cap B) \cup (A \cap C)$.
  - As $x \in A$, and either $x \in B$ or $x \in C$.
    - Case1: $x \in B$. Then $x \in A \cap B$, therefore $x \in (A \cap B) \cup (A \cap C)$.
    - Case2: $x \in C$. Then $x \in A \cap C$, therefore $x \in (A \cap B) \cup (A \cap C)$.
  - Thus, $x \in (A \cap B) \cup (A \cap C)$.
  - So $A \cap (B \cup C) \subseteq (A \cap B) \cup (A \cap C)$.
- **Part 2:** Proof $(A \cap B) \cup (A \cap C) \subseteq A \cap (B \cup C)$.

Proving Set Equivalences

To prove set equivalence

$$A = B,$$

We could use common techniques:

1. Proof $A \subseteq B$ and $B \subseteq A$.
2. **Using definitions and equivalence of logical propositions that define the set.**
3. Use the truth table.

Proving Set Equivalences

Example 3: Proof: $\overline{A \cap B} = \overline{A} \cup \overline{B}$

$\overline{A \cap B} = \{x \mid x \notin A \cap B\}$	According to complementation definition
$= \{x \mid \neg(x \in (A \cap B))\}$	According to definition of $\notin$
$= \{x \mid \neg(x \in A \wedge x \in B)\}$	According to definition of intersection
$= \{x \mid x \notin A \vee x \notin B)\}$	According to De Morgan law
$= \{x \mid x \in \overline{A} \vee x \in \overline{B})\}$	According to complementation definition
$= \{x \mid x \in \overline{A} \cup \overline{B})\}$	According to union definition

Proving Set Equivalences

To prove set equivalence

$$A = B,$$

We could use common techniques:

1. Proof $A \subseteq B$ and $B \subseteq A$.
2. Using definitions and equivalence of logical propositions that define the set.
3. **Use the truth table.**

Truth table

- Building tables:
  - The columns correspond to set expressions.
  - The rows correspond to all possible combinations of membership in the set.
- Fill in the table: Use "1" to indicate a member, "0" to indicate non-member.
- Equality is proven if two columns corresponding to two expressions on both sides are **identical**.

Example 4: Proof: $(A \cup B) - B = A - B$.

A	B	$A \cup B$	$(A \cup B) - B$	$A - B$
0	0	0	0	0
0	1	1	0	0
1	0	1	1	1
1	1	1	0	0

Proving Set Equivalences

Example 5: Using the truth table, proof that

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

A B C	$B \cup C$	$A \cap (B \cup C)$	$A \cap B$	$A \cap C$	$(A \cap B) \cup (A \cap C)$
1 1 1	1				
1 1 0	1				
1 0 1	1				
1 0 0	0				
0 1 1	1				
0 1 0	1				
0 0 1	1				
0 0 0	0				

Contents

Contents

4.1. Characteristic vector

Suppose that we have U = $\{u_1, u_2, \ldots, u_n\}$, where n is not too large. Then each subset $M \subset$ U can be represented by a vector $b = (b_1, b_2, \ldots, b_n)$ where

$$b_i = 1 \leftrightarrow u_i \in M,\ i = 1, 2, \ldots, n.$$
$$= 0 \text{ otherwise}$$

- Vector b constructed by this rule is called characteristic vector of the set M.
- It is clear that each subset $M \subset$ U corresponds to unique characteristic vector b, and on the other hand, each binary n-vector b corresponds to unique subset of U

0 1 1 0 1 0 1 0 0 0 1

Example:

Suppose that U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Consider the subsets $S, Q \subseteq$ U.

- $S = \{2, 3, 5, 7, 11\} \leftrightarrow 01101010001$
- $Q = \{1, 2, 4, 11\} \leftrightarrow 11010000001$

4.1. Characteristic vector

Note that all the set operation $\cup$ (Union), $\cap$ (Intersection), $\neg$(complement) can be done by correspondently logic operation OR, AND, NOT

Example: Suppose that U = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}.

Consider the subsets S, $Q \subseteq$ U.

- $S = \{2, 3, 5, 7, 11\} \leftrightarrow 01101010001$
- $Q = \{1, 2, 4, 11\} \leftrightarrow 11010000001$
- $S \cup Q \leftrightarrow 01101010001 \vee 11010000001$

  ➔ $S \cup Q \leftrightarrow 11111010001$
- $S \cap Q \leftrightarrow 01101010001 \wedge 11010000001$

  ➔ $S \cap Q \leftrightarrow 01001000001$
- $S = 01101010001$

  ➔ $\neg S \leftrightarrow 10010101110$

Contents

4.2. Subset enumeration

In many practical situation, we have to examine all the subsets of a given set U = $\{u_1, u_2, \ldots, u_n\}$.

For example: Enumerate all subsets of U = {1, 2, 3}

1. $\emptyset$
2. {1}
3. {2}
4. {3}
5. {1, 2}
6. {1, 3}
7. {2, 3}
8. {1, 2, 3}

4.2. Subset enumeration

In many practical situation, we have to examine all the subsets of a given set U = $\{u_1, u_2, ..., u_n\}$. How to do it?

- Answer: Each subset of U ~ a characteristic vector

→Enumeration of all the subsets of U ~ enumeration of all binary *n*-vector.

1 2

Since each binary *n*-vector can be considered as the binary representation of a nonnegative integer $\alpha(b) = b_1 b_2 \ldots b_n$, $0 \leq \alpha(b) \leq 2^n$-1

→ enumeration of all binary *n*-vector 2

~ enumeration of binary representation for all nonnegative integer from 0 to 2^n-1.

3

1.4.2. Subset enumeration

For example: Enumerate all subsets of U = {1, 2, 3}

0) 000 $\varnothing$

1) 001 {3}

2) 010 {2}

3) 011 {2,3}

4) 100 {1}

5) 101 {1,3}

6) 110 {1,2}

7) 111 {1,2,3}

Enumeration of all the subsets of U ❶
~ enumeration of all binary n-vector ❷
~ enumeration of binary representation for all nonnegative integer from 0 to 2^n-1. ❸

4.2. Subset enumeration

Subset Enumeration Algorithm:

- **Step** $k = 0, 1, ..., 2^n\text{-}1$: Output binary representation of the number k.
- Clearly, if we have the binary representation of the number k ($b_1b_2...b_n$) then the binary representation of the number k+1 can be obtained by binary addition $b_1b_2...b_n$ to 1.

For example: Enumerate all subsets of U = {1, 2, 3}

0)	000	$\varnothing$
1)	001	{3}
2)	010	{2}
3)	011	{2,3}
4)	100	{1}
5)	101	{1,3}
6)	110	{1,2}
7)	111	{1,2,3}

$$\begin{array}{r} 101 \\ +\quad 1 \\ \hline 110 \end{array}$$

4.2. Subset enumeration

Subset Enumeration Algorithm

- **Step** $k = 0, 1, ..., 2^n-1$: Output binary representation of the number k.
- Clearly, if we have the binary representation of the number k $(b_1b_2...b_n)$ then the binary representation of the number $k+1$ can be obtained by binary addition $b_1b_2...b_n$ to 1.

Algorithm: BINARY_INCREMENT($b_1, b_2, ..., b_n$)

$i = n$;

while (n>=1) **and** (b_i==1)

    $b_i = 0$;

    $i = i - 1$;

endwhile

    $b_i = 1$;

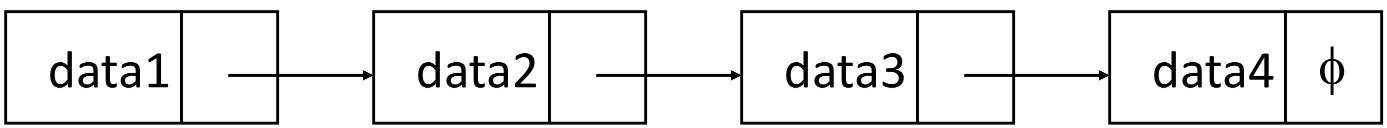
$$
\begin{array}{r}
1001011111111 \\
+ \quad\quad\quad\quad\quad\quad 1 \\
\hline
1001100000000
\end{array}
$$

Contents

4.3. List of elements

- When the set U contain a large number of elements, but considered subset U have a small cardinality, binary representation is not reasonable. In this case we can represent the subset by list of all its elements.
- This list is usually implemented as the **linked list structure**. Each element of list is a record that consists of two fields, one of which contains the information of the element and the other one is a pointer to the next element:

```
class ListNode {
    Object data ; //element information
    ListNode next; //pointer to the next element
}
```



Contents

1. Definitions
2. Set operations
3. The algebra of sets
4. Computer representation of sets
5. **Relations**
6. Functions
7. Recursion

5. Relations

5.1. Ordered pair

5.2. Cartesian product

5.3. Binary relation

5.4. Relation representation

5.5. Operations on relations

5.6. Properties of relations

5.1. Ordered pair

- An *ordered pair* is a set of a pair of objects with an order associated with them.
- In general (x, y) is different from (y, x).
- **Definition (equality of ordered pairs):** Two ordered pairs (a, b) and (c, d) are equal if and only if $a = c$ and $b = d$.

Example: if the ordered pair (a, b) is equal to $(1, 2)$, then a=1, and b=2. $(1, 2)$ is not equal to the ordered pair $(2, 1)$.

5.2. Cartesian product

René Descartes
(1596-1650)

- Let $A_1, A_2, \ldots, A_n$ be any sets, where $n \in \mathbb{Z}^+$ and $n \geq 3$.

Cartesian product of n sets $A_1, A_2, \ldots, A_n$ is defined as follows:

$$A_1 \times A_2 \times \ldots \times A_n \equiv_{\text{def}} \{(a_1, a_2, \ldots, a_n) \mid a_i \in A_i, 1 \leq i \leq n\}.$$

When $A_1 = A_2 = \ldots = A_n = A$, it is usually to denote $A \times A \times \ldots \times A$ by A^n

An element of $A_1 \times A_2 \times \ldots \times A_n$ is called an ordered n-tuple.

When n=3, we have a *triple*.

Example: $A = \{1, 2\}$, $B = \{a, b\}$ and $C = \{\alpha, \beta\}$ then

$A \times B \times C = \{(1, a, \alpha), (1, a, \beta), (1, b, \alpha), (1, b, \beta), (2, a, \alpha), (2, a, \beta), (2, b, \alpha), (2, b, \beta)\}$.

5. Relations

5.1. Ordered pair

5.2. Cartesian product

5.3. Binary relation

5.4. Relation representation

5.5. Operations on relations

5.6. Properties of relations

5.2. Cartesian product

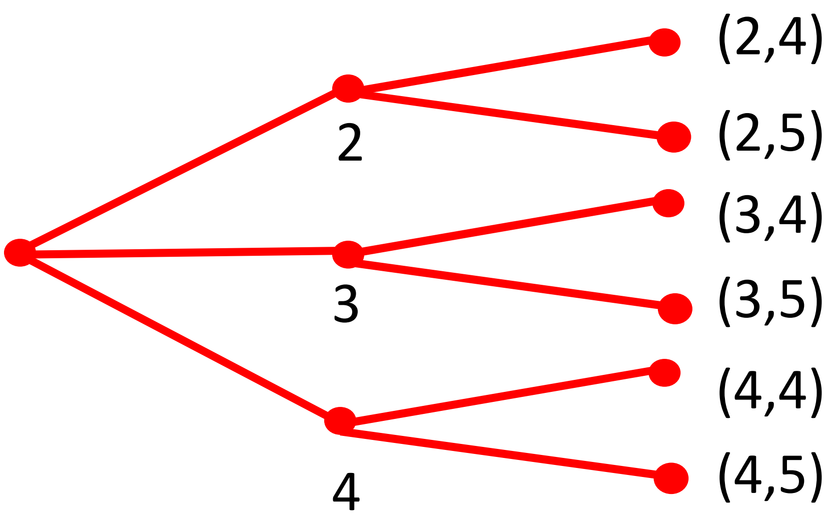
Theorem : If X_1, X_2, ..., X_n are finite sets then

$$|X_1 \times X_2 \times \ldots \times X_n| = |X_1| \times |X_2| \times \ldots \times |X_n|$$

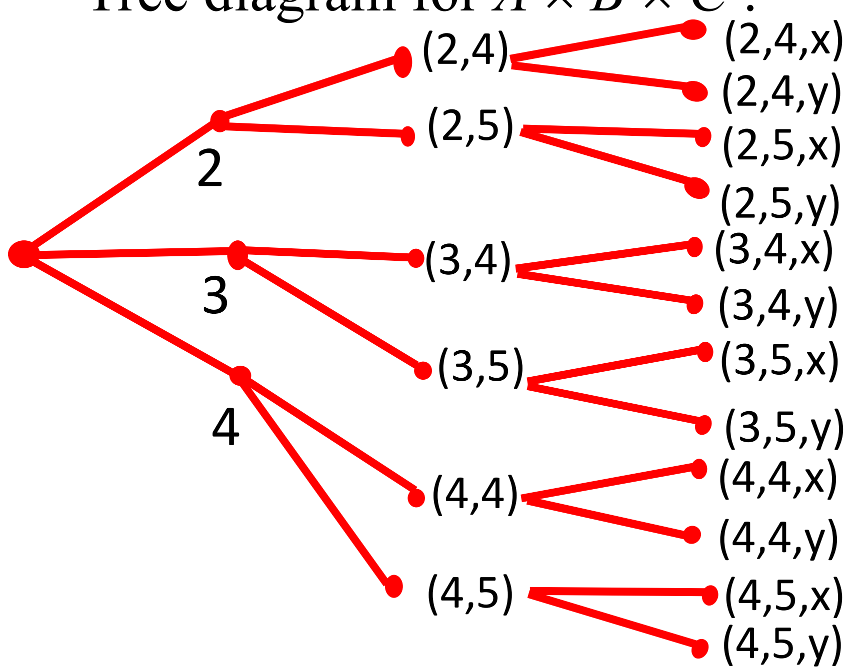
Enumeration: To enumerate all elements of Cartesian product of the sets we can use *tree diagram*.

Example: $A = \{2, 3, 4\}$, $B = \{4, 5\}$, and $C = \{x, y\}$.

Tree diagram for $A \times B$:



Tree diagram for $A \times B \times C$:



5.2. Cartesian product

How the Cartesian product operation behaves with respect to the other set theory operations such as intersection and union?

Theorem : For any three sets A, B, C:

$$A \times (B \cap C) = (A \times B) \cap (A \times C);$$

$$A \times (B \cup C) = (A \times B) \cup (A \times C);$$

$$(A \cap B) \times C = (A \times C) \cap (B \times C);$$

$$(A \cup B) \times C = (A \times C) \cup (B \times C).$$

5. Relations

5.1. Ordered pair

5.2. Cartesian product

5.3. Binary relation

5.4. Relation representation

5.5. Operations on relations

5.6. Properties of relations

5.3. Binary Relation

Let A and B be sets:

- **Definition (binary relation):** A binary relation from a set A to a set B is a set of ordered pairs (a, b) where a is an element of A and b is an element of B.
  - A binary relation from A to B is a subset R $\subseteq A \times B$]
  - A relation on a set A is a relation from A to A, i.e., a subset R $\subseteq A \times A$
- Notation: When an ordered pair (a, b) is in a relation R, we write $a\ R\ b$, or $(a, b) \in R$. It means that element a is related to element b in relation R. We will write $a\ \overline{R}\ b$ when a element a is not related to element b in relation R.

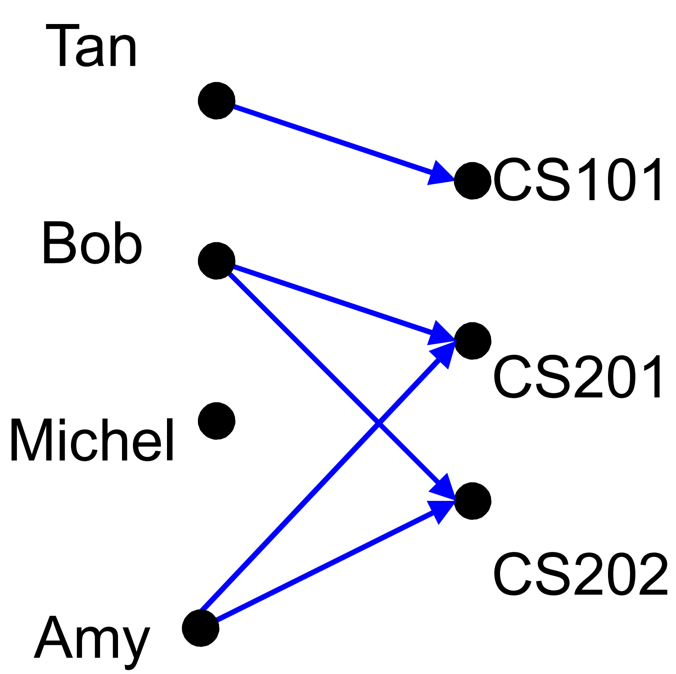
5.3. Binary Relation

Example:

- Let A be the students in a the CS major
  - A = {Tan, Bob, Michel, Amy}
- Let B be the courses the department offers
  - B = {CS101, CS201, CS202}
- We specify relation $R = A \times B$ as the set that lists all students $a \in A$ enrolled in class $b \in B$
  - R = { (Tan, CS101), (Bob, CS201), (Bob, CS202),
    (Amy, CS201), (Amy, CS202) }

Representing relations

We can represent relations graphically:



We can represent relations in a table:

	CS101	CS201	CS202
Tan	X		
Bob		X	X
Michel			
Amy		X	X

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Relations *on* a Set

- A relation *on* a set A is a relation from A to A.

Examples of relations on Z^+: $R_<$, $R_\geq$, $R_>$:

- $R_< = \{(x, y) | \, x < y\}$ ($R_<$ is relation "strictly less than").
- $R_\geq = \{(x, y) | \, x \geq y\}$ ($R_\geq$ is relation "greater or equal").
- $R_> = \{(x, y) | \, x > y\}$ ($R_>$ is relation "strictly greater than").

Relations *on* a Set

Consider the following relations on Z:

- $R_1 = \{(a, b) \mid a \leq b\}$
- $R_2 = \{(a, b) \mid a > b\}$
- $R_3 = \{(a, b) \mid a = b \text{ or } a = -b\}$
- $R_4 = \{(a, b) \mid a = b\}$
- $R_5 = \{(a, b) \mid a = b+1\}$
- $R_6 = \{(a, b) \mid a + b \leq 3\}$

	(1,1)	(1,2)	(2,1)	(1,-1)	(2,2)
R1	√	√			√
R2			√	√	
R3	√			√	√
R4	√				√
R5			√		
R6	√	√	√	√	

For each the following ordered pairs

(1,1), (1,2), (2,1), (1,-1), and (2,2)

show which relation it belongs to.

5.3. Binary relation

A binary relation is a set of ordered pairs (x, y)

The **domain** is the set of all x values in the relation

domain = {-1,0,2,4,9}

These are the x values written in a set from smallest to largest

{(2,3), (-1,5), (4,-2), (9,9), (0,-6)}

This is a relation

These are the y values written in a set from smallest to largest

range = {-6,-2,3,5,9}

The **range** is the set of all y values in the relation

5. Relations

5.4. Relation representation

1. Set of ordered pairs
2. Mapping
3. Table
4. Grid graph
5. Binary matrix
6. Directed graph

For relations on a set

5.4. Relation representation: set of ordered pairs

A __**relation**__ is a set of __**ordered pairs (x, y)**__.

Relation = { (3,5),(-2,4),(-3,4),(0,-4) }

The __**domain**__ is the __**set**__ of __**x**__ values.

The __**range**__ is the __**set**__ of __**y**__ values.

EXAMPLE

Relation = { (3,5),(-2,4),(-3,4),(0,-4) }

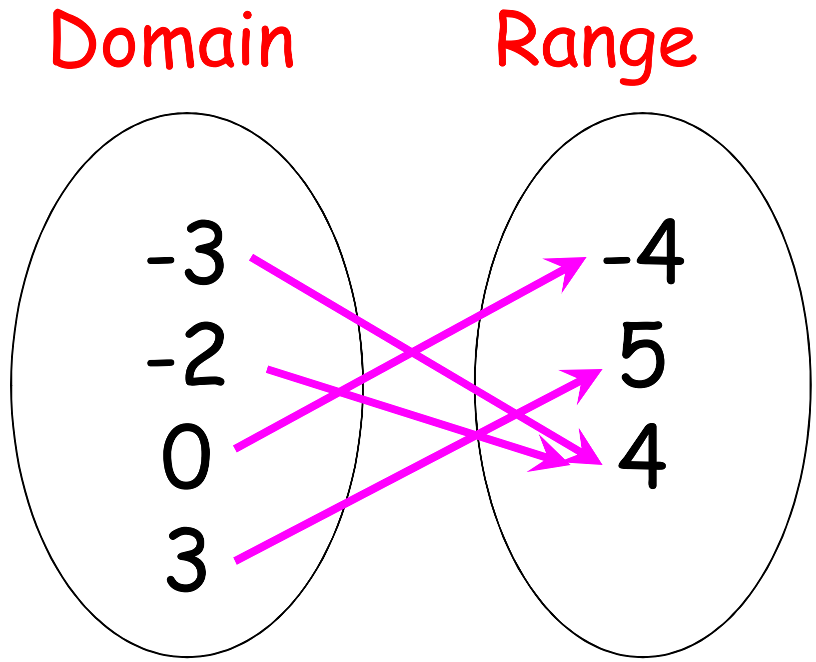
Domain: { -3 , -2 , 0 , 3 }

Range: { -4 , 5 , 4 }

NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

Relation = { (3,5),(-2,4),(-3,4),(0,-4) }

2. Mapping:



NGUYỄN KHÁNH PHƯƠNG
Bộ môn KHMT – ĐHBK HN

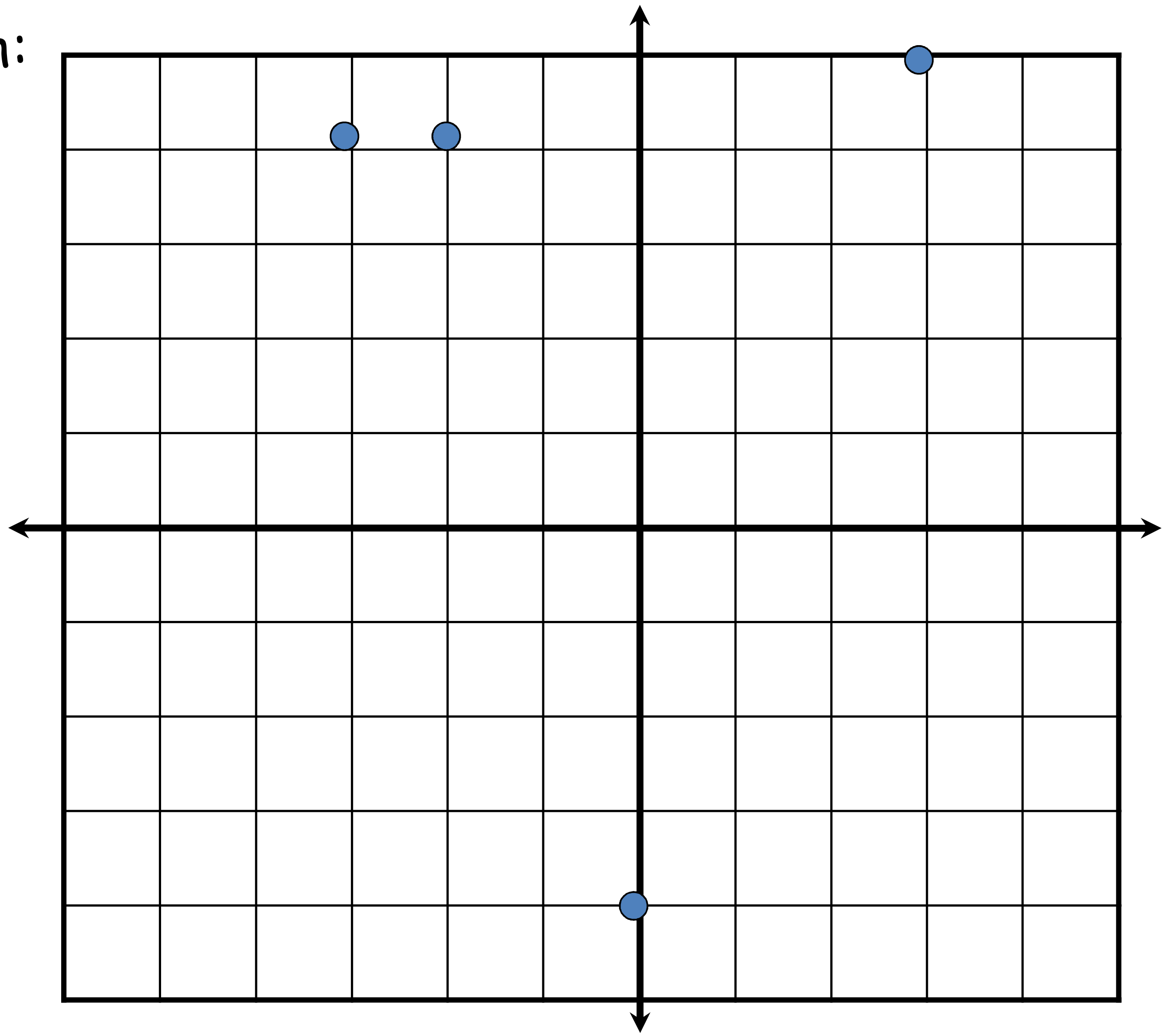
Relation = { (3,5),(-2,4),(-3,4),(0,-4) }

3. Table:

domain	range
x	y
-3	4
-2	4
0	-4
3	5

Relation = { (3,5),(-2,4),(-3,4),(0,-4) }

4. Grid graph:



Relation = { (3,5),(-2,4),(-3,4),(0,-4) }

5. Binary matrix

Domain a: { -3, -2, 0, 3 }

Range b: { -4, 5, 4 }

$M_R = [m_{ij}]_{n \times m}$ of zeros and ones with n (=4) rows and m (=3)columns

$$m_{ij} = \begin{cases} 1, & \text{if } a_i \; R \; b_j \\ 0, & \text{if } a_i \; \overline{R} \; b_j \end{cases}$$

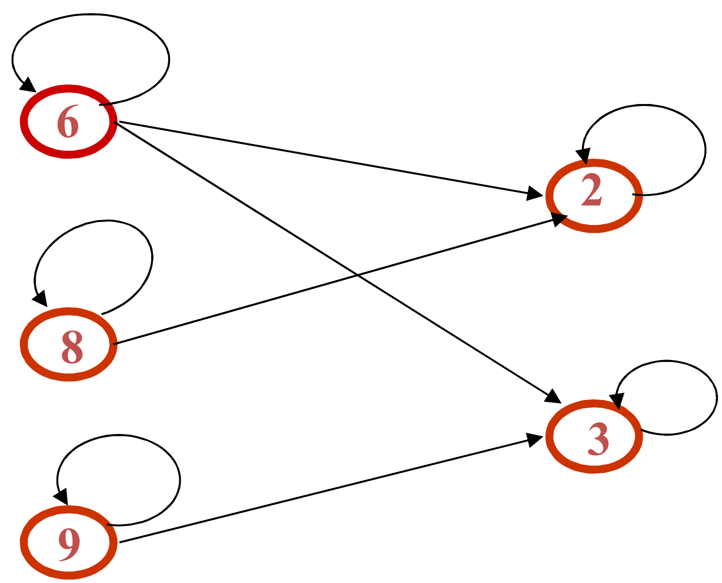
$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

5.4. Relation representation

6. Directed graph

- Let $A = \{2, 3, 6, 8, 9\}$. Consider relation R: *a* is related to *b* iff *a is divisible by b*. We have:

$R = \{(2,2), (3,3), (6,2), (6,3), (6,6), (8,2), (8,8), (9,3), (9,9)\}$.



5.4. Relation representation

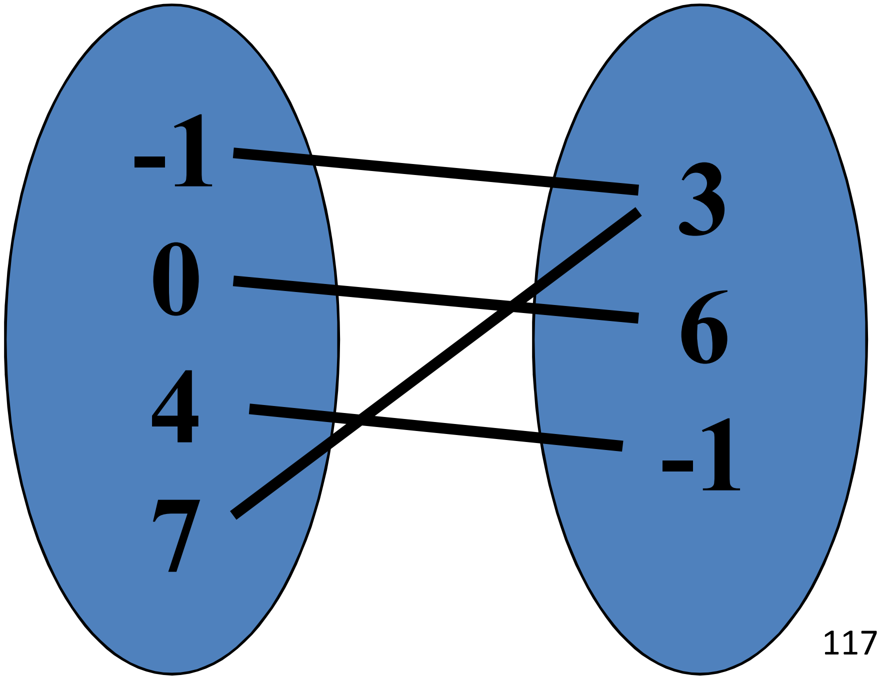
Exercise: Given the following table, show the relation as set of ordered pairs, domain, range and mapping

X	-1	0	4	7
Y	3	6	-1	3

Relation = **{(-1,3), (0,6), (4,-1), (7,3)}**

Domain = **{-1, 0, 4, 7}**

Range = **{-1, 3, 6}**



5. Relations

5.5. Operations on relations

- A relation is a set. It is a set of ordered pairs if it is a binary relation. Thus all the set operations apply to relations such as union, intersection and complementing.

Example:

- The union of the "less than" and "equality" relations on the set of integers is the "less than or equal to" relation on the set of integers.
- The intersection of the "less than" and "less than or equal to" relations on the set of integers is the "less than" relation on the same set.
- The complement of the "less than" relation on the set of integers is the "greater than or equal to" relation on the same set.

1. **Complementary Relations**
2. **Inverse Relations**
3. **Identity relation**
4. ***n*-ary Relations**
5. **Composite Relation**

5.5. Operations on relations

1. Complementary Relations

Let $R \subseteq A \times B$ be a binary relation. *Complementary relation* $\overline{R}$ of R is defined as the set

$$\overline{R} \equiv_{\text{def}} \{(a,b) \mid (a,b) \notin R\} = (A \times B) - R .$$

Example:

$$\overline{R_<} = \{(a,b) \mid (a,b) \notin R_<\} = \{(a,b) \mid \text{a} \geq b\} = R_\geq .$$

2. Inverse Relations

Each binary relation $R \subseteq A \times B$ has inverse relation R^{-1}, which defined by

$$R^{-1} = \{(b,a) \mid (a,b) \in R\}.$$

Example1:

$$(R_<)^{-1} = \{(b,a) \mid a<b\} = \{(b,a) \mid b>a\} = R_>.$$

Example 2: Let B be the set of jobs, and A be the set of the workers. Consider R as relation from A to B that defined as follows

$aRb \Leftrightarrow a$ perform b.

Then $b\, R^{-1}\, a \Leftrightarrow b$ is performed by a.

Note: We have $(R^{-1})^{-1} = R$.

5.5. Operations on relations

3. Identity Relation

Identity relation I_A on the set A is defined as

$$I_A = \{(a, a) \mid a \in A\}.$$

***4. n*-ary Relations**

An *n-ary* relation on sets $A_1, \ldots, A_n$ is a set of ordered n-tuples $(a_1, \ldots, a_n)$ where a_i is an element of A_i for all i, $1 \le i \le n$. Thus an n-ary relation R on sets $A_1, \ldots, A_n$ is a subset of Cartesian product $A_1 \times A_2 \times \ldots \times A_n$:

$$R \subseteq A_1 \times A_2 \times \ldots \times A_n.$$

Note that the sets $A_1, \ldots, A_n$ are not to be different.

Example: Application of n-ary Relations

Example: Teaching Assignments

Professor	*Department*	*Course Number*
Cruz	Chemistry	335
Cruz	Chemistry	412
Farber	Psychology	501
Farber	Psychology	617
Grammar	Physics	544
Grammar	Physics	551
Rosen	Computer Science	518
Rosen	Mathematics	575

- A relational database models a database as a relation.
- The relation's domains are called its attributes.
- How many attributes in the Teaching Assignment table?

The relation's primary key is an attribute whose value uniquely determines an element in the relation.

In general, a primary key may consist of > 1 attribute.

What single attribute could serve as the primary key in the Teaching Assignment table?

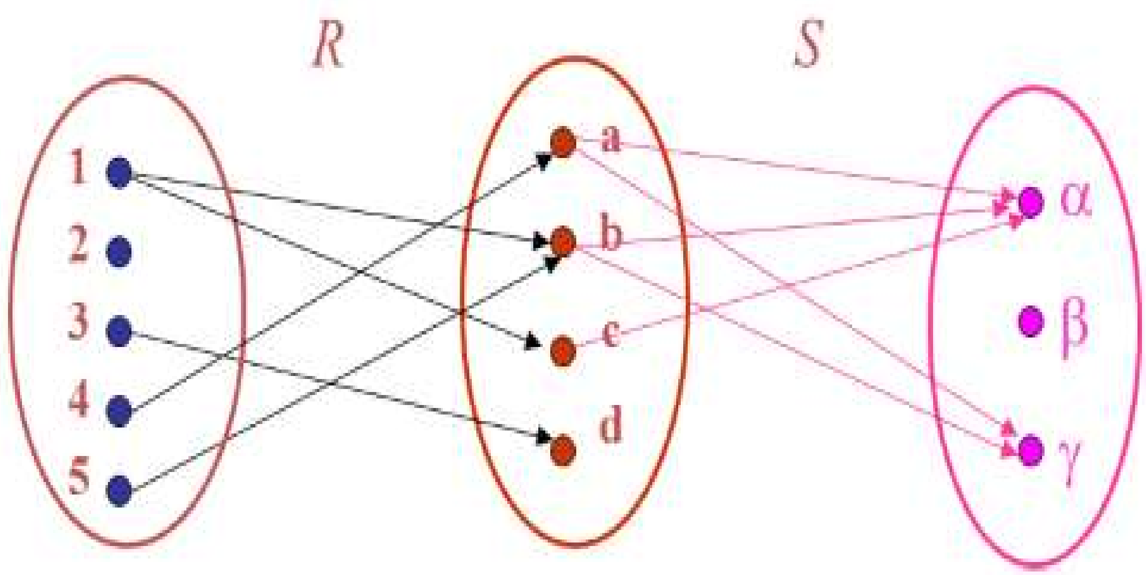
5.5. Operations on relations

5. Composite Relations

Suppose $R \subseteq A \times B$ and $S \subseteq B \times C$. Composite (or product) relation $R \circ S$ of two relations R and S is the following

$$R \circ S = \{(a,c) \mid aRb \wedge bSc\}$$

Example: Let $A = \{1, 2, 3, 4, 5\}$, $B = \{\text{a, b, c, d}\}$, $C = \{\alpha, \beta, \gamma\}$. Consider the relations R, S which are displayed in the following diagram



$R \circ S = ?$

We have: $R \circ S = \{(1, \alpha), (1, \gamma), (4, \alpha), (4, \gamma), (5, \alpha), (5, \gamma)\} \subseteq A \times C$

Example

- Let S be a set of students:

  { Bill, Jill, Will }.

- Let C be a set of courses:

  { 16, 24, 32, 40, 48, 56 }

- Let

  R = { (s, c) | student s has taken course c }.

  - Many students may have taken the same course.
  - A student may have taken many courses.

Matrix representation of relation R

Relation R: Student *s* has taken course *c*

COURSE

S
T
U
D
E
N
T

	16	24	32	40	48	56
Bill	1	1	0	0	0	0
Jill	1	1	1	0	0	1
Will	1	0	0	1	0	0

(row i, column j) = 1 $\Longleftrightarrow$ student i has taken course j.

Matrix representation of relation S

Relation S: Course c has been taught by teacher t

TEACHER

COURSE

	Mike	Diana	Pete
16	1	1	0
24	1	1	1
32	0	1	0
40	0	0	0
48	1	0	1
56	1	0	0

$S \circ R = M_R \times M_S$ "Boolean" matrix product

$S \circ R = \{ (s, t) \mid \exists c\, (sRc \wedge cSt) \}.$

Describe in English what $S \circ R$ represents ?

$$\begin{array}{c} \text{STUDENT} \end{array} \overset{\text{COURSE}}{\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}} \quad \begin{array}{c} \text{COURSE} \end{array} \overset{\text{TEACHER}}{\begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix}} = \begin{array}{c} \text{STUDENT} \end{array} \overset{\text{TEACHER}}{\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}}$$

5. Relations

5.1. Ordered pair

5.2. Cartesian product

5.3. Binary relation

5.4. Relation representation

5.5. Operations on relations

5.6. Properties of relations

5.6. Properties of relations

Six properties of relations we will study:

1. Reflexive
2. Irreflexive
3. Symmetric
4. Asymmetric
5. Anti-symmetric
6. Transitive

5.6. Properties of relations

Let R be a relation on set A. We say that R is:

1. *reflexive* if and only if $a\ R\ a$ for every $a \in A$;
2. *irreflexive* if and only if its complementary relation is reflexive.
3. *symmetric* if and only if $a\ R\ b$ implies $b\ R\ a$ for every $a, b \in A$;
4. *asymmetric* if and only if $(a, b) \in R \Rightarrow (b, a) \notin R$ (Asymmetry is the opposite of symmetry)
5. *anti-symmetric* if and only if $a\ R\ b$ and $b\ R\ a$ implies $a = b$ for every $a, b \in A$ (Antisymmetry is *not* the opposite of symmetry)
6. *transitive* if and only if $a\ R\ b$ and $b\ R\ c$ implies $a\ R\ c$ for every $a, b, c \in A$.

Examples of reflexive relations: $=$, $\leq$, $\geq$

Examples of irreflexive relations (relations that are not reflexive): $<$, $>$

Examples of symmetric relations: $=$

Examples of asymmetric relations: $<$, $>$

Examples of anti-symmetric relations: $=$, $\leq$, $\geq$

Examples of transitive relations:

- If $a < b$ and $b < c$, then $a < c$ ➔ Thus, $<$ is transitive
- If $a = b$ and $b = c$, then $a = c$ ➔ Thus, $=$ is transitive

Notes on *symmetric relations

Let R be a relation on set A. We say that R is:

1. *symmetric* if and only if $a\ R\ b$ implies $b\ R\ a$ for every $a, b \in A$;
2. *asymmetric* if and only if $(a, b) \in R \Rightarrow (b, a) \notin R$ (Asymmetry is the opposite of symmetry)
3. *anti-symmetric* if and only if $a\ R\ b$ and $b\ R\ a$ implies $a = b$ for every $a, b \in A$ (Antisymmetry is *not* the opposite of symmetry)

Example: A relation can be neither symmetric or asymmetric

$$R = \{ (a,b) \mid a=|b| \}$$

- This is not symmetric

  -4 is not related to itself
- This is not asymmetric

  4 is related to itself
- Note that it is antisymmetric

Properties of relations summary

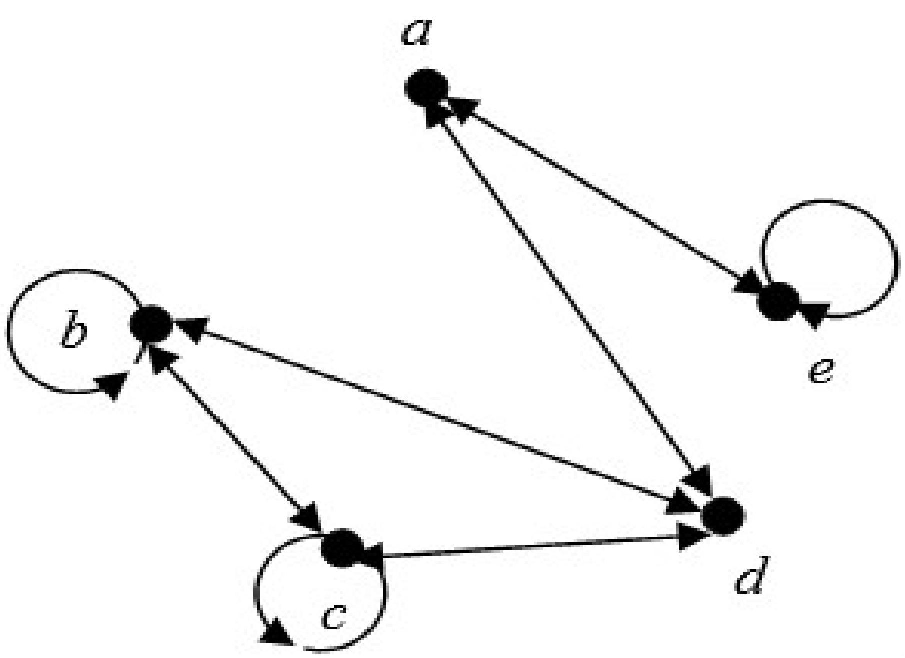
	$=$	$<$	$>$	$\leq$	$\geq$
Reflexive					
Irreflexive					
Symmetric					
Asymmetric					
Antisymmetric					
Transitive					

Properties of relations summary

	$=$	$<$	$>$	$\leq$	$\geq$
Reflexive	X			X	X
Irreflexive		X	X		
Symmetric	X				
Asymmetric		X	X		
Antisymmetric	X			X	X
Transitive	X	X	X	X	X

5.6. Properties of relations

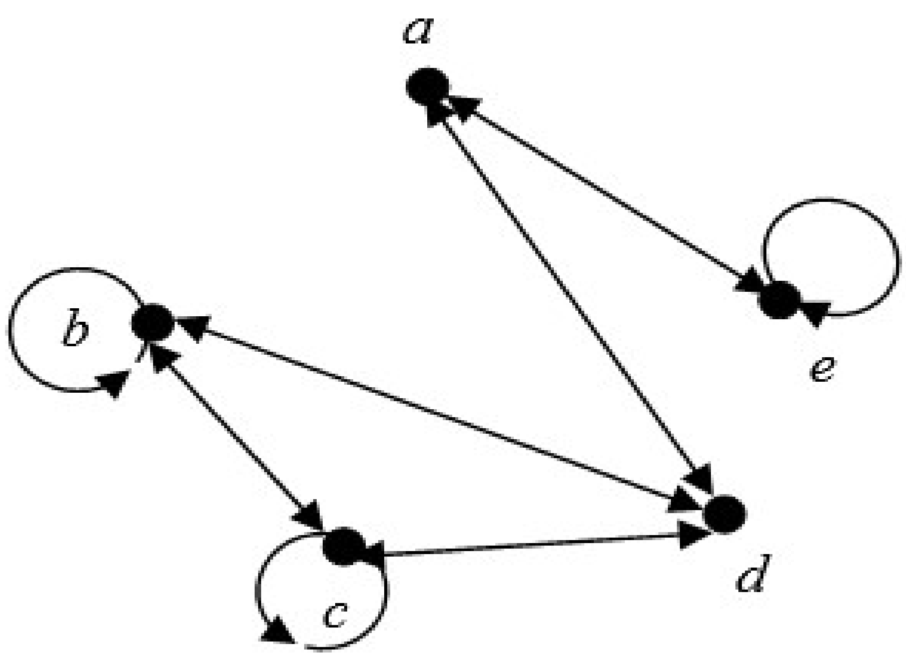
Example: Consider the directed graph of a relation R on the set $A = \{a, b, c, d, e\}$. Which of the properties does R satisfy?



1. *reflexive* if and only if $a\ R\ a$ for every $a \in A$;
2. *irreflexive* if and only if its complementary relation is reflexive.
3. *symmetric* if and only if $a\ R\ b$ implies $b\ R\ a$ for every $a, b \in A$;
4. *asymmetric* if and only if $(a, b) \in R \Rightarrow (b, a) \notin R$ (Asymmetry is the opposite of symmetry)
5. *anti-symmetric* if and only if $a\ R\ b$ and $b\ R\ a$ implies $a = b$ for every $a, b \in A$ (Antisymmetry is *not* the opposite of symmetry)
6. *transitive* if and only if $a\ R\ b$ and $b\ R\ c$ implies $a\ R\ c$ for every $a, b, c \in A$.

5.6. Properties of relations

Example: Consider the directed graph of a relation R on the set $A = \{a, b, c, d, e\}$. Which of the properties does R satisfy?



- R is not reflexive, since there is no arrow from d to itself, for example.
- R is symmetric, but not anti-symmetric, since every arrow connecting distinct points is bidirectional.
- R is not transitive since, for instance, there are arrows from a to d, and from d to b, but not from a to b.

Equivalence relation

Let R be a binary relation on the set S.

- (1) R is reflexive if sRs $\forall s \in S$
- (2) R is symmetric if s_1Rs_2 ➔ s_2Rs_1 $\forall s_1, s_2 \in S$
- (3) R is transitive if s_1Rs_2 and s_2Rs_3 ➔ s_1Rs_3 $\forall s_1, s_2, s_3 \in S$
- (4) R is equivalence relation if it is reflexive, symmetric, and transitive

➔ A binary relation is an equivalence relation on a non-empty set S if and only if the relation is reflexive(R), symmetric(S) and transitive(T).

Example 1: S = {All people}. Define `xRy` if x has the same parents as y
➔ R is equivalence relation on S.

Example 2: S = $\mathbb{R}$. Define `xRy` if x <y

As $x \not< x$ ➔ R is not reflexive ➔ R is not equivalence relation on S.

Equivalence relation

Let R be a binary relation on the set S.

- (1) R is reflexive if sRs $\forall s \in S$
- (2) R is symmetric if s_1Rs_2 ➔ s_2Rs_1 $\forall s_1, s_2 \in S$
- (3) R is transitive if s_1Rs_2 and s_2Rs_3 ➔ s_1Rs_3 $\forall s_1, s_2, s_3 \in S$
- (4) R is equivalence relation if it is reflexive, symmetric, and transitive

➔ A binary relation is an equivalence relation on a non-empty set S if and only if the relation is reflexive(R), symmetric(S) and transitive(T).

➔ A binary relation is a **partial order** if and only if the relation is reflexive(R), antisymmetric(A) and transitive(T).

Exercise

Determine whether or not each of the following binary relations on the given set is reflexive, symmetric, antisymmetric, or transitive. If a relation has a certain property, prove this is so; otherwise, provide a counterexample to show that it does not.

a. Let $S = \mathbb{R}$. Define a relation R on S by xRy iff $x = y$

b. Let S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}. Define a relation R on set A = SxS by $(a, b)R(c,d)$ iff $10a + b \leq 10c+d$

c. Let $A = \mathbb{Z} \setminus \{0\}$. Define a relation R on set A, by aRb iff $ab > 0$

Exercise

a. Let S = $\mathbb{R}$. Define a relation R on S by xRy iff x = y

1. Yes R is reflexive.

Proof:

Let $a \in \mathbb{R}$.

Then $a = a$.

Hence R is reflexive.$\square$

2. Yes R is symmetric.

Proof:

Let $(a, b) \in \mathbb{R}$.

If $a = b$, clearly $b = a$.

Hence R is symmetric.$\square$

4. Yes R is transitive.

Proof:

Let $a, b, c \in \mathbb{R}$ s.t. $a = b$ and $b = c$.

We shall show that aRc.

Since $a = b$ and $b = c$ it follows that $a = c$

Thus aRc.$\square$

3. Yes R is antisymmetric.

Proof:

Let $a, b \in \mathbb{R}$ s.t. $a = b$ and $b = a$ then clearly $a = b$ $\forall a, b \in \mathbb{R}$.$\square$

Contents

6. Functions

6.1. Definitions

6.2. Properties of function

6.3. Injective, surjective and bijective function

6.4. Function representation

6. Functions

- **Definition**: A function f from a set A to a set B, denote it by $f: A \to B$, is a relation from A to B that satisfies:
  - for each element a in A, there is an element b in B such that (a, b) is in the relation, and
  - if (a, b) and (a, c) are in the relation, then $b = c$. ➔ **1 to 1**

A function is also called a *mapping* or a *transformation*.

The set A in the above definition is called the *domain* of the function and B its *codomain*.

Thus, f is a function if it *covers* the domain (maps every element of the domain) and it is *single valued*.

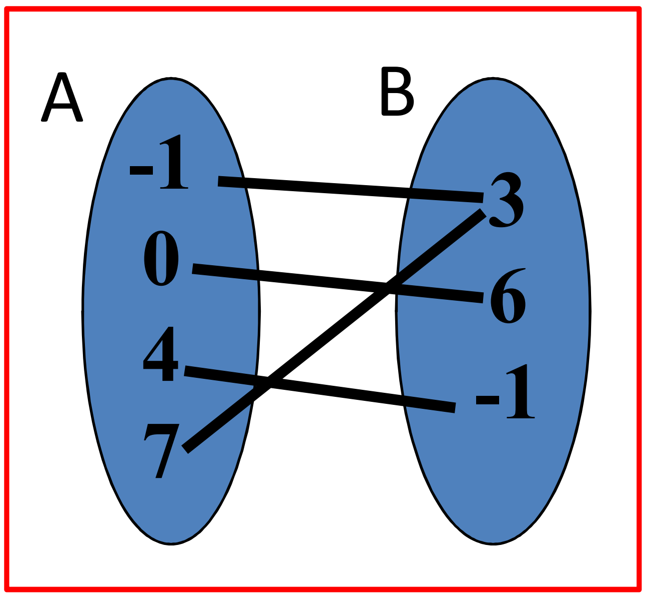
The relation given by f between a and b represented by the ordered pair (a, b) is denoted as $f(a) = b$, and b is called the image of a under f.

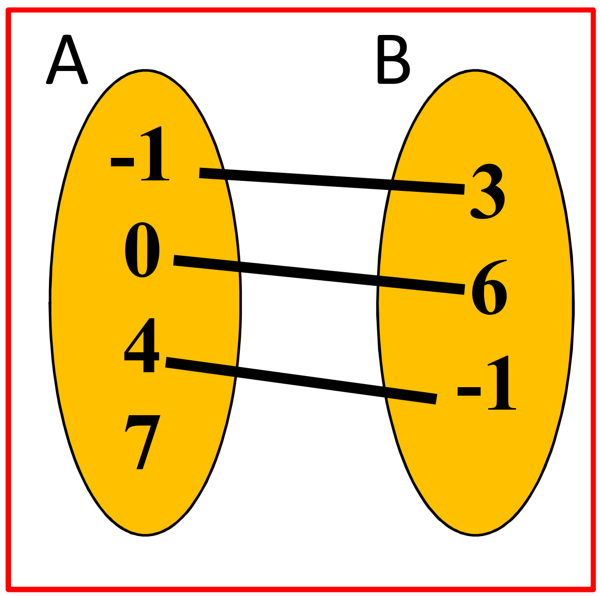
- **Proposition**: If $|A| = m$, $|B| = n$, then the number of possible functions from A to B is n^m

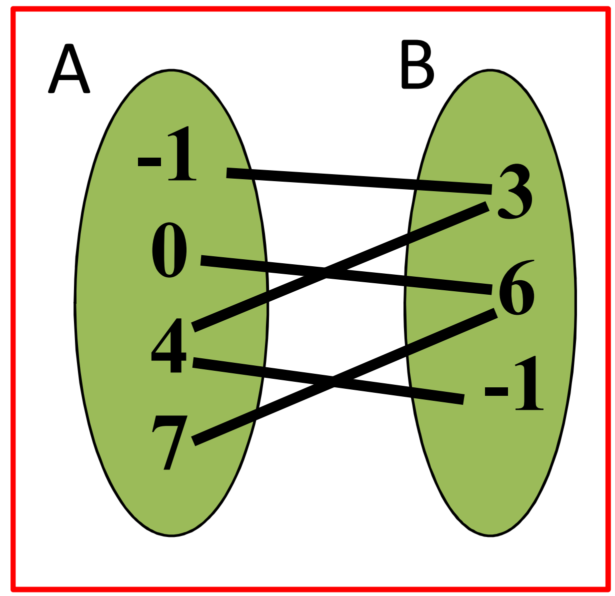
6. Functions

Thus, *f*: *A*→*B* is a function if it *covers* the domain (maps every element of the domain) and it is *single valued.*

- Single valued: each element in the domain is used only once
- Not allowed: 1 – many and 1 to empty







Which relation is function?

6. Functions

- The image of the set S under function $f : A \to B$, denoted by $f(S)$ is:

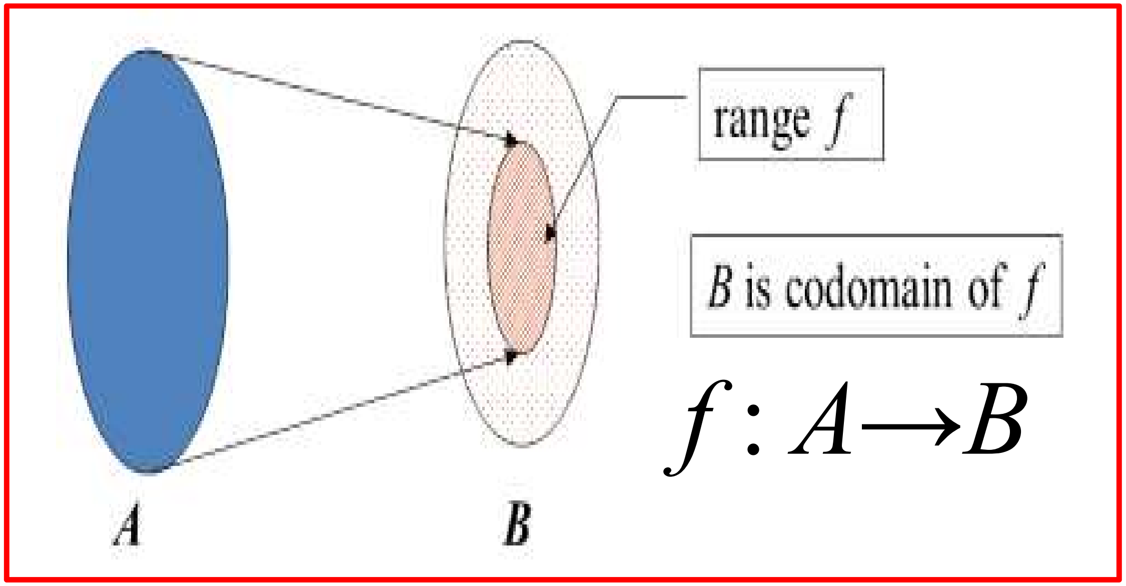
$$f(S) = \{ f(a) \mid a \in S \}$$

- The image of the domain under function $f : A \to B$, denoted by range f is:

$$\text{range } f = f(A)$$

(is also called the range of f)

In general case: range $f = f(A) \subseteq B$.



6. Functions

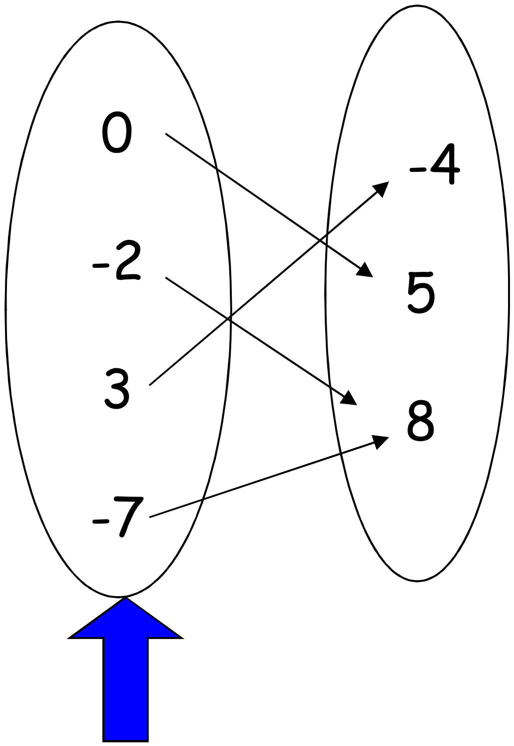
Example 1: Let $A = \{a, b\}$, $B = \{1, 2, 3\}$. Which following relations from A to B are functions from A to B ?

- $P = \{(a,1), (b,1)\}$
- $Q = \{(a,2), (b,3)\}$
- $S = \{(a,1)\}$
- $T = \{(a,2), (b,1), (b,3)\}$

Relation is function if:

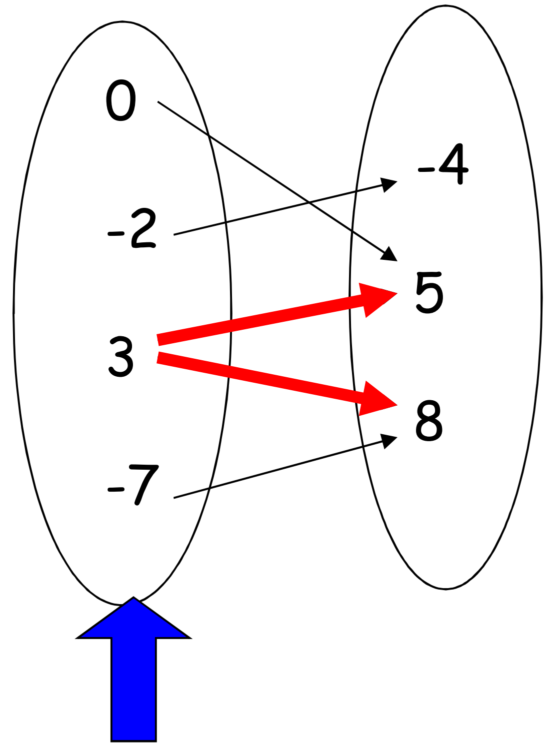
- Single valued: each element in the domain is used only once
- Not allowed: 1 – many and 1 to empty

Example 2: Are the following relations also functions?



Each element of the domain is only used once.

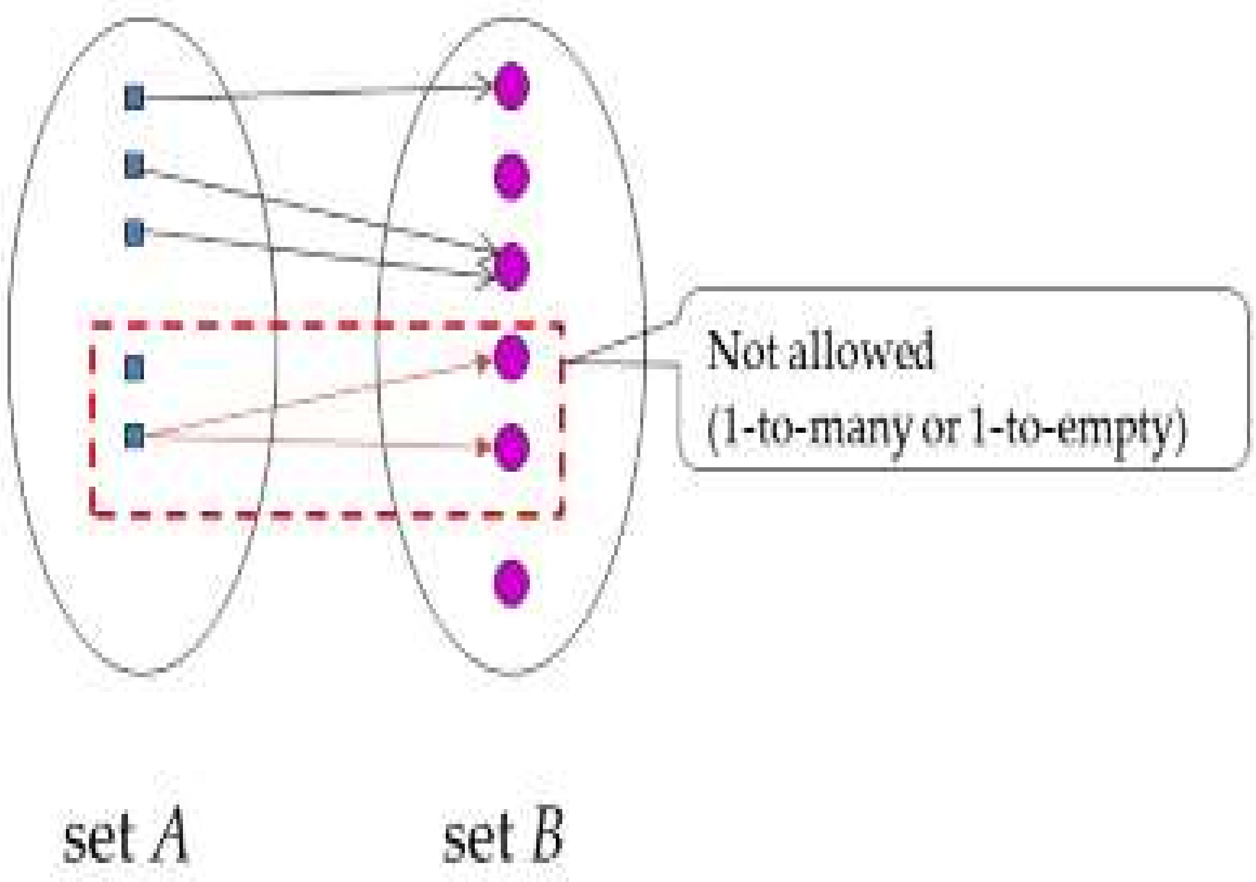
FUNCTION



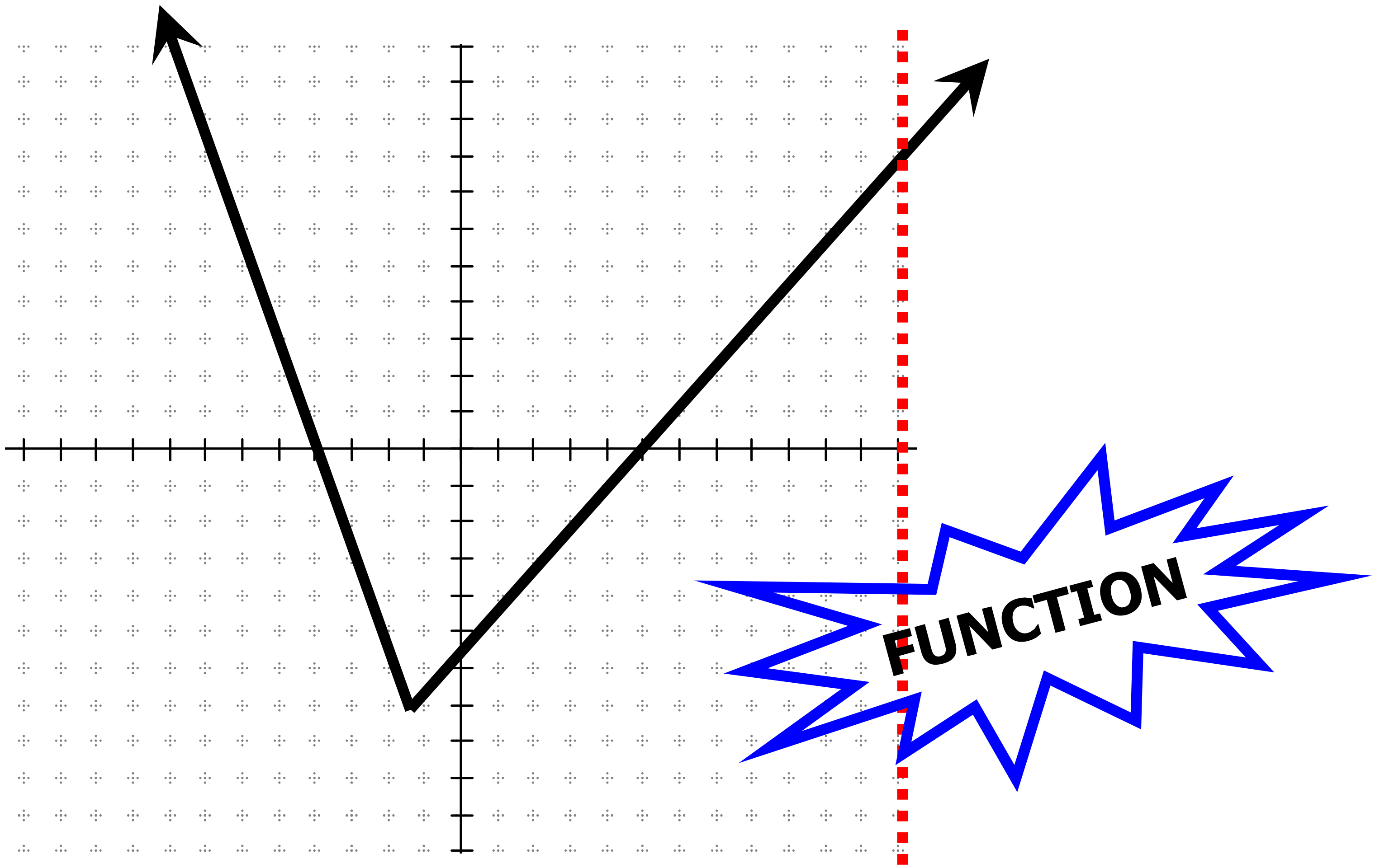
3 is used twice.

NOT A FUNCTION

Example 2: Are the following relations also functions?



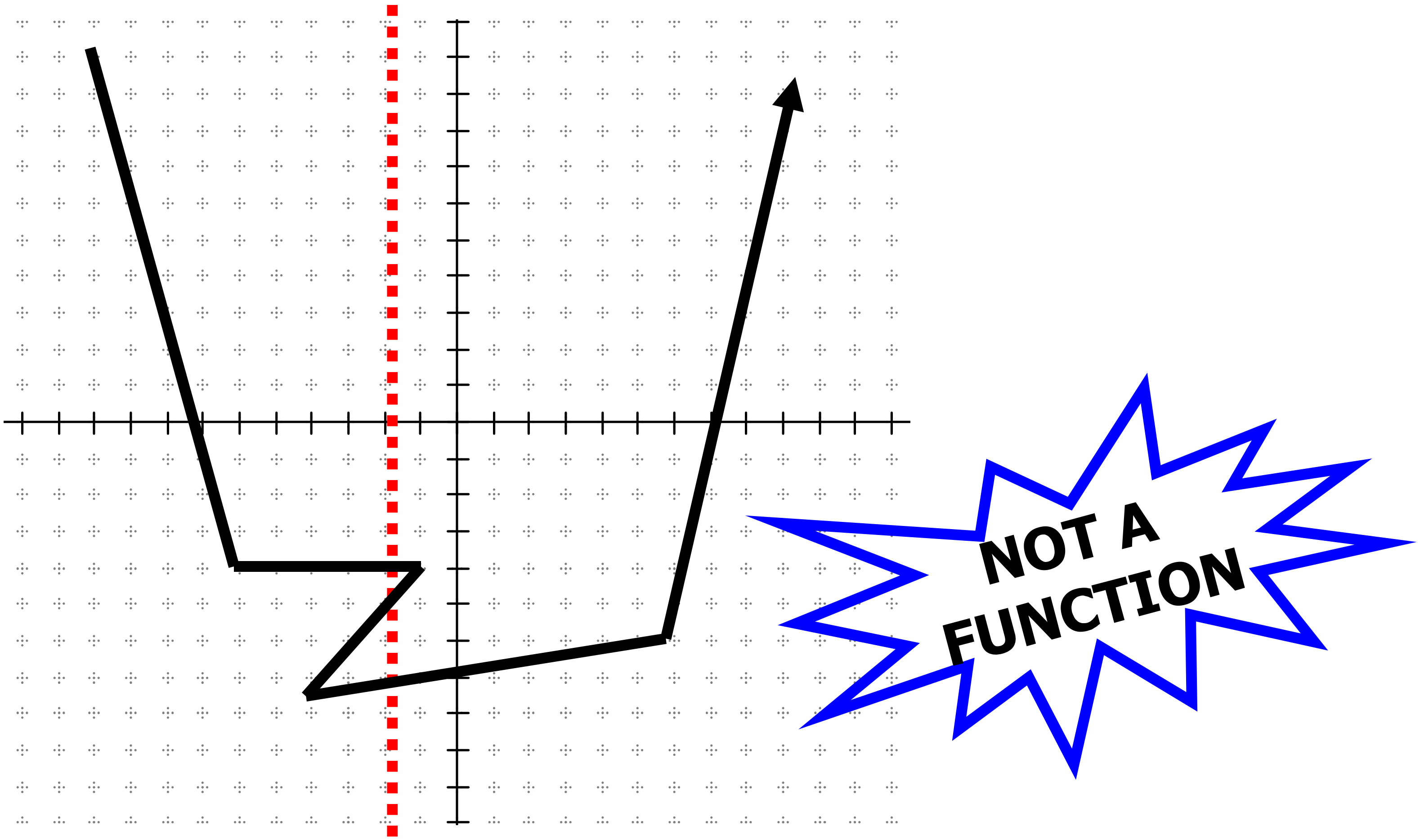
Example 2: Are the following relations also functions?



Will all vertical lines cross the graph at **only** one point?

YES. Each x is only used once.

Example 2: Are the following relations also functions?



Will all vertical lines cross the graph at only one point?

NO. x =-2 is used three times.

Example 2: Are the following relations also functions?

x	y
-4	0
-2	-3
2	3
-4	1

NOT A FUNCTION

Are all x values used only once?

NO. $x = -4$ is used twice.

6. Functions

6.1. Definitions

6.2. Properties of function

6.3. Injective, surjective and bijective function

6.4. Function representation

6.2. Properties of function

$f: A \rightarrow B$ is a function from a set A to a set B, $S \subseteq A$, and $T \subseteq B$.

- **Property 1:** $f(S \cup T) = f(S) \cup f(T)$

1. **Proof for $f(S \cup T) \subseteq f(S) \cup f(T)$:**
   - Let y be an arbitrary element of $f(S \cup T)$. Then there is an element x in $S \cup T$ such that $y = f(x)$. If x is in S, then y is in $f(S)$. Hence y is in $f(S) \cup f(T)$.
   - Similarly y is in $f(S) \cup f(T)$ if x is in T.
   - Hence if $y \in f(S \cup T)$, then $y \in f(S) \cup f(T)$.
2. **Proof for $f(S) \cup f(T) \subseteq f(S \cup T)$:**
   - Let y be an arbitrary element of $f(S) \cup f(T)$. Then y is in $f(S)$ or in $f(T)$. If y is in $f(S)$, then there is an element x in S such that $y = f(x)$. Since $x \in S$ implies $x \in S \cup T$, $f(x) \in f(S \cup T)$.
   - Hence $y \in f(S \cup T)$.
   - Similarly $y \in f(S \cup T)$ if $y \in f(T)$.

Property 1 has been proven.

6.2. Properties of function

f: $A \rightarrow B$ is a function from a set A to a set B, $S \subseteq A$, and $T \subseteq B$.

- **Property 2:** $f(S \cap T) \subseteq f(S) \cap f(T)$

Proof.

- Let y be an arbitrary element of $f(S \cap T)$. Then there is an element x in $S \cap T$ such that $y = f(x)$, that is there is an element x which is in S and in T, and for which $y = f(x)$ holds. Hence $y \in f(S)$ and $y \in f(T)$, that is $y \in f(S) \cap f(T)$.
- Note here that the converse of Property 2 does not necessarily hold. For example let $S = \{1\}$, $T = \{2\}$, and $f(1) = f(2) = \{3\}$. Then $f(S \cap T) = f(\varnothing) = \varnothing$, while $f(S) \cap f(T) = \{3\}$. Hence $f(S) \cap f(T)$ cannot be a subset of $f(S \cap T)$ giving a counterexample to the converse of Property 2.

6. Functions

6.1. Definitions

6.2. Properties of function

6.3. Injective, surjective and bijective function

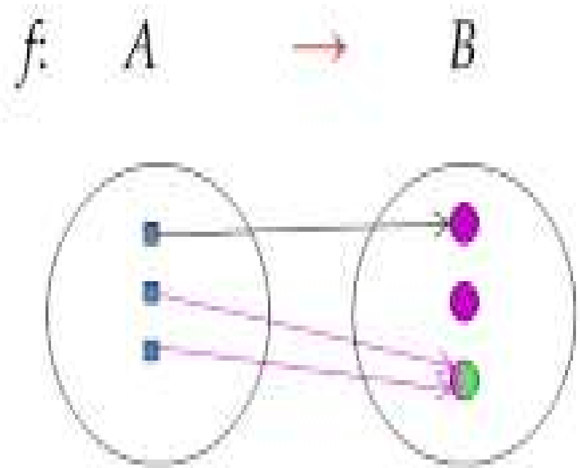
6.4. Function representation

6.3. Injective, surjective and bijective function

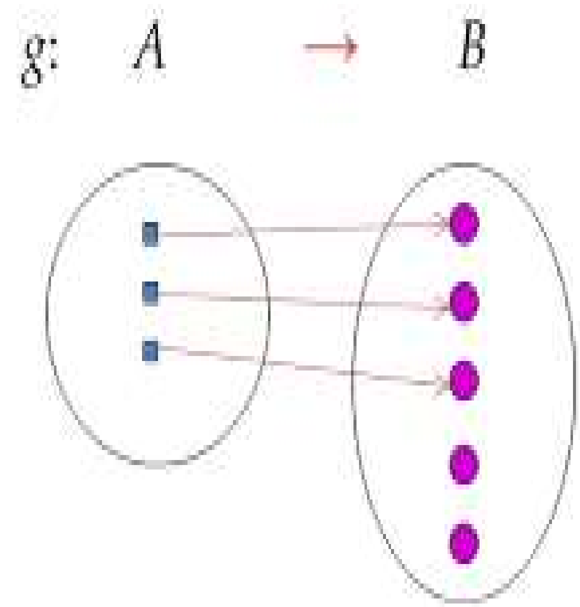
- A function f from a set A to a set B is said to be *injective* (one-to-one) if and only if:

  for all elements $a_1, a_2 \in A$

  if $f(a_1) = f(a_2)$ then $a_1 = a_2$



The function f is not injective



The function g is injective

$$a_1, a_2 \in A,\ a_1 \neq a_2 \Rightarrow f(a_1) \neq f(a_2)$$

6.3. Injective, surjective and bijective function

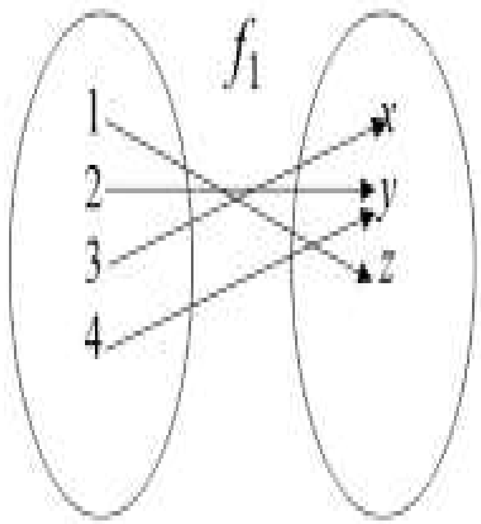
- A function f from a set A to a set B is said to be *surjective* (onto), if and only if: $\forall b \in B, \exists\, a \in A: b = f(a)$.

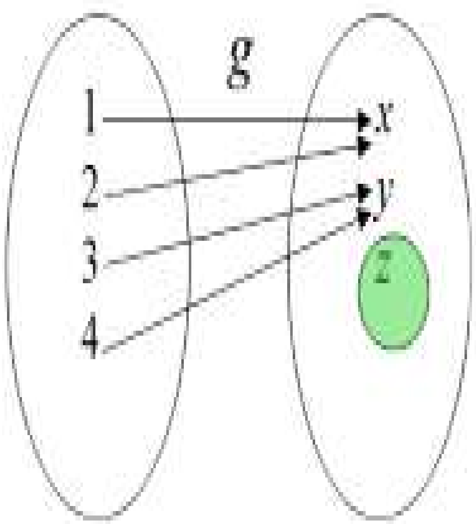
 (read: for any element $b \in B$ there is an element $a \in A$ *such that* $f(a) = b$)

that is: f is onto if and only if $f(A) = B$.

Example: $A = \{1, 2, 3, 4\}$ and $B = \{x, y, z\}$. Then the functions:

$f_1 = \{(1, z), (2, y), (3, x), (4, y)\}$; $g=\{(1, x), (2, x), (3, y), (4, y)\}$



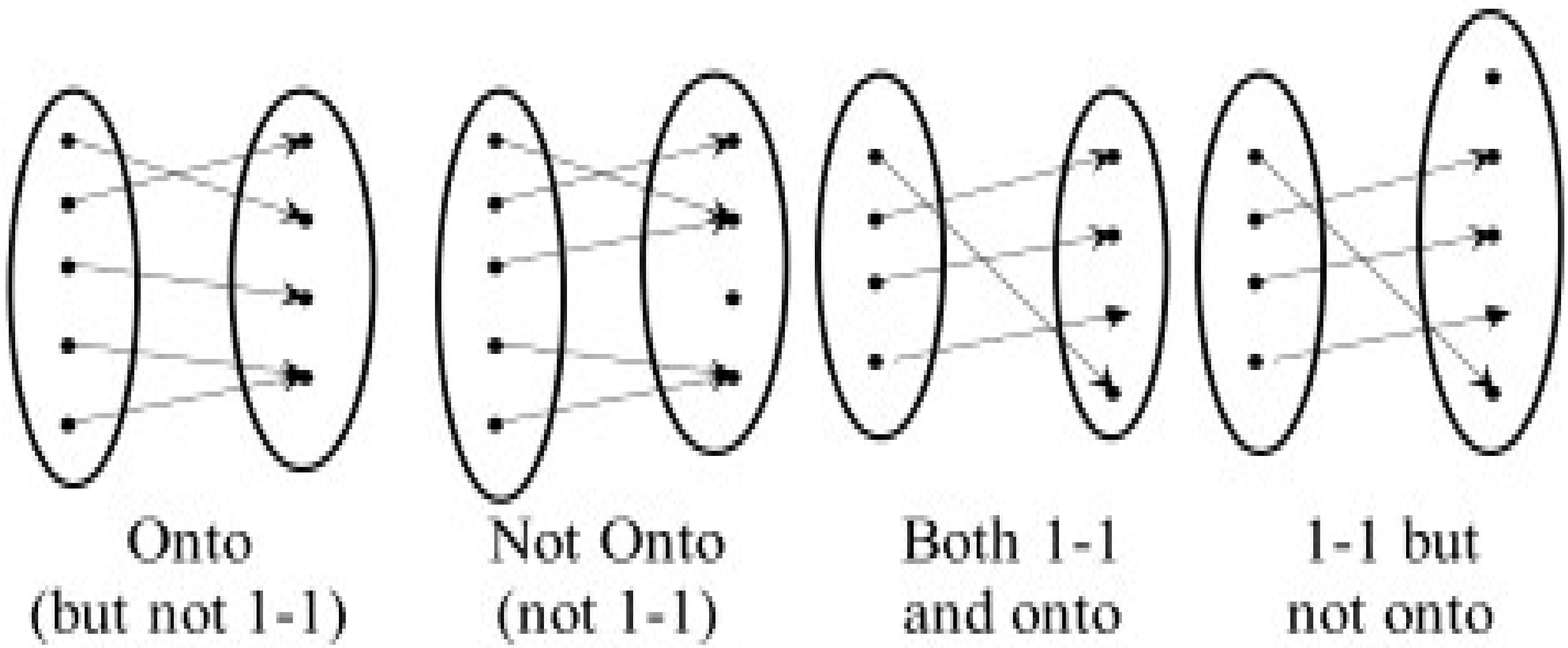


The function f_1 is onto

The function g is not onto because $g(A)=\{x,y\}\neq B$

6.3. Injective, surjective and bijective function

- A function is called a *bijection*, if it is injective (1-1) and surjective (onto).



6.3. Injective, surjective and bijective function

- A function is called a *bijection*, if it is injective (1-1) and surjective (onto).

Examples:

1) Linear functions: $f(x)=ax+b$ when $a\neq 0$
(with domain and co-domain **R**)
2) Exponential functions: $f(x)=b^x$ $(b>0, b\neq 1)$
(with domain **R** and co-domain $\mathbf{R}^+$)
3) Logarithmic functions: $f(x)=\log_b x$ $(b>0, b\neq 1)$
(with domain $\mathbf{R}^+$ and co-domain **R**)

6. Functions

6.1. Definitions

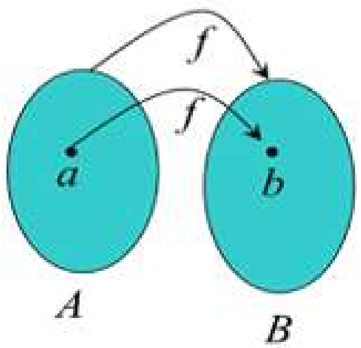
6.2. Properties of function

6.3. Injective, surjective and bijective function

6.4. Function representation

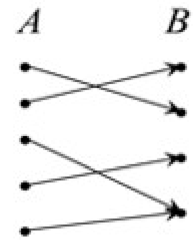
6.4. Function representation

Functions can be represented four different ways:

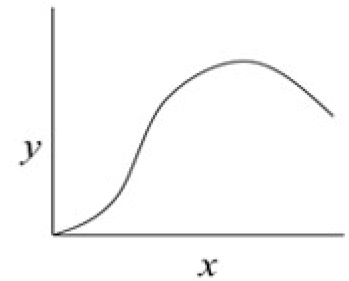


Like Venn diagrams

1. mapping
2. graph
3. table
4. matrix



Bipartite Graph



Plot

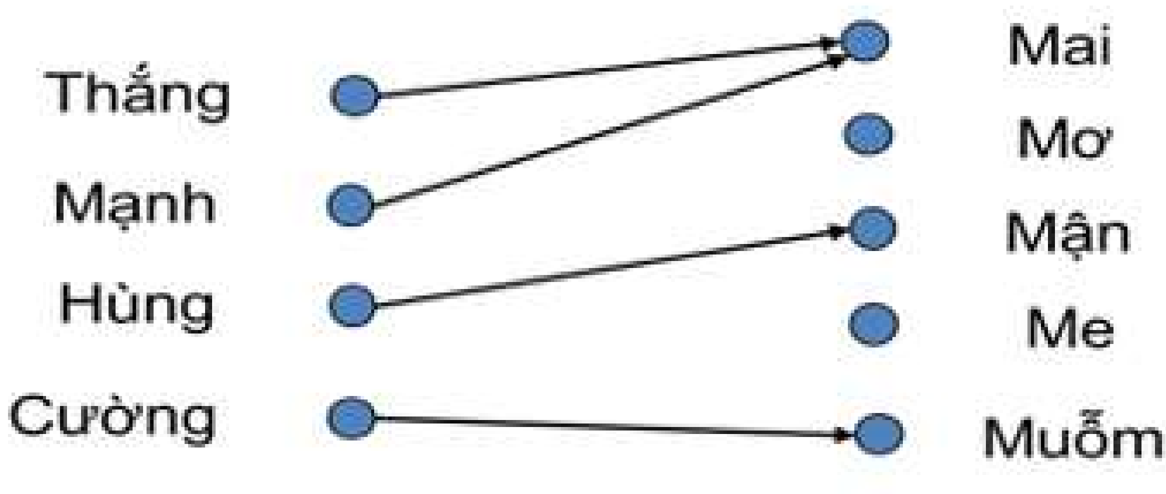
a	a_1	a_2	...	a_m
$f(a)$	$f(a_1)$	$f(a_2)$	...	$f(a_m)$

The function $f: A \to B$ can be defined by a matrix $A_f = \{a_{ij}\}$ size of $m \times n$ where

$$a_{ij} = \begin{cases} 1, \text{ if } b_j = f(a_i) \\ 0, \text{ otherwise} \end{cases}, i = 1, \ldots, m; j = 1, \ldots, n.$$

6.4. Function representation

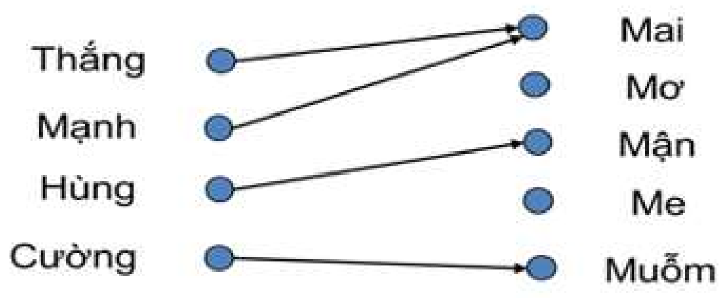
- Let A = {Thắng, Mạnh, Hùng, Cường} and B = {Mai, Mơ, Mận, Me, Muỗm}. Consider the function $f : A \rightarrow B$ defined by the following diagram:



Represent this function by table and matrix

6.4. Function representation

- Represent this function by table and matrix



$$A_f = \begin{array}{c c} \begin{array}{ccccc} \text{Mai} & \text{Mơ} & \text{Mận} & \text{Me} & \text{Muỗm} \end{array} & \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{l} \text{Thắng} \\ \text{Mạnh} \\ \text{Hùng} \\ \text{Cường} \end{array} \end{array}$$

x	Thắng	Mạnh	Hùng	Cường
$y=f(x)$	Mai	Mai	Mận	Muỗm

Contents

7. Recursion

- In recursive definitions, we define a function, a sequence or a set over an infinite number of elements by defining the function or predicate value or set-membership of larger elements in terms of that of smaller ones.
  - Recursion is a process of defining an object (functions, sequences, sets, algorithms) in terms of itself (or of part of itself).
  - There are:
    1. recursive function
    2. recursive set
    3. recursive structure
    4. recursive algorithm

7. Recursion

- In induction, we prove all members of an infinite set have some property P by proving the truth for larger members in terms of that of smaller members.
- In recursive definitions, we similarly define a function, a sequence or a set over an infinite number of elements by defining the function or predicate value or set-membership of larger elements in terms of that of smaller ones.
  - Recursion is a process of defining an object (functions, sequences, sets, algorithms) in terms of itself (or of part of itself).
  - There are:
    1. recursive function
    2. recursive set
    3. recursive structure
    4. recursive algorithm

Recursively Defined Functions

Definition: A recursive definition of a function consists of two steps.

- Basis step: Specify the value of the function at zero.
- Recursive step: Give a rule for finding its value at an integer from its values at smaller integers.

Example: Give a recursive definition of the factorial function n!

$$f(0) = 1$$

$$f(n + 1) = (n + 1) \cdot f(n)$$

Example: Give a recursive definition of Fibonacci numbers.

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

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

7. Recursion

- In induction, we prove all members of an infinite set have some property P by proving the truth for larger members in terms of that of smaller members.
- In recursive definitions, we similarly define a function, a sequence or a set over an infinite number of elements by defining the function or predicate value or set-membership of larger elements in terms of that of smaller ones.
  - Recursion is a process of defining an object (functions, sequences, sets, algorithms) in terms of itself (or of part of itself).
  - There are:
    1. recursive function
    2. **recursive set**
    3. recursive structure
    4. recursive algorithm

Recursively Defined Sets

Definition: A recursive definition of a set consists of two steps.

- Basis step: Specify an initial collection of elements
- Recursive step: Give the rules for forming new elements in the set from those already known to be in the set.

Sometimes the recursive definition has an exclusion rule, which specifies that the set contains nothing other than those elements specified in the basis step and generated by applications of the rules in the recursive step.

Example1: A subset of Integers S:

- Basis step:

  $3 \in S$.

- Recursive step:

  If $x \in S$ and $y \in S$, then $x + y \in S$

➔ $S=\{3,6,9,12,15,\ldots\}$

Initially 3 is in S, then $3 + 3 = 6$, then $3 + 6 = 9$, etc.

Recursively Defined Sets

Example 2: Give recursive definition for the natural numbers N ={0,1,2,..}

- Basis step:

  $0 \in N$

- Recursive step:

  If $n \in N$ then $n + 1 \in N$

Exercise:

Assume the alphabet $\Sigma = \{a, b, c, d\}$

A set of all strings containing symbols in Σ :

$$\Sigma^* = \{“”, a, aa, aaa, aaa…, ab, …b, bb, bbb, …\}$$

Give recursive definition of Σ^*

- Basis step:

  - Empty string $\alpha \in \Sigma^*$

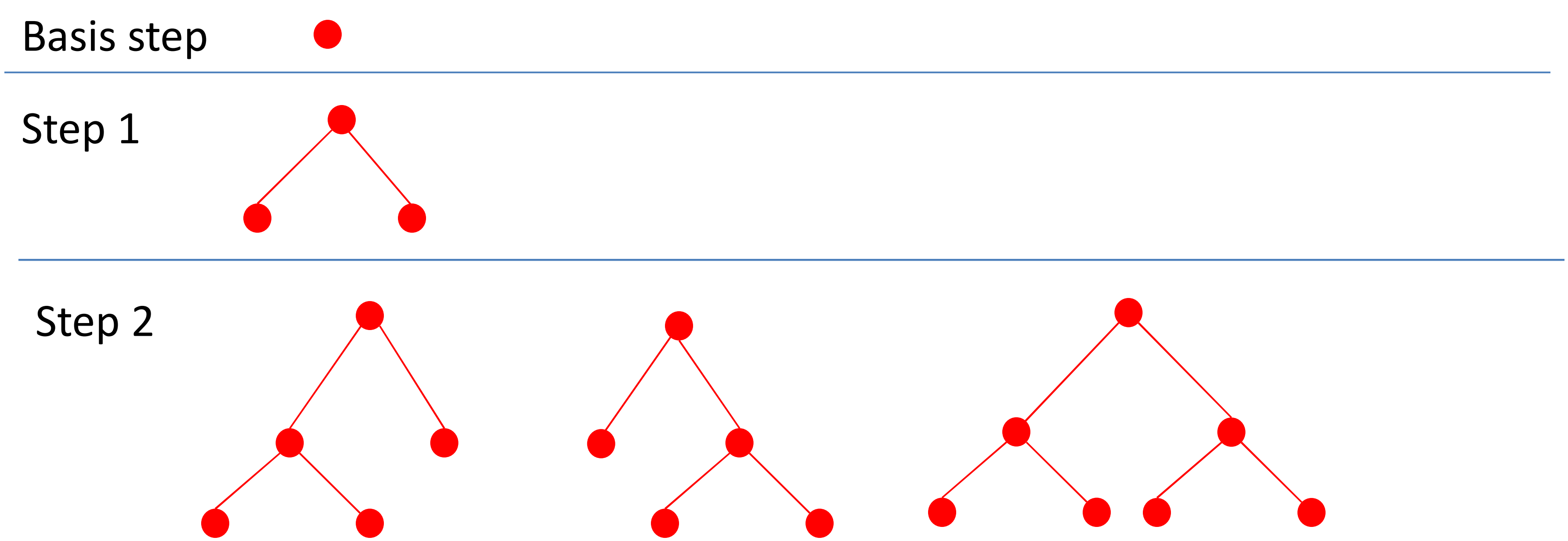
- Recursive step:

  – If $w \in \Sigma^*$ and $x \in \Sigma$ then $wx \in \Sigma^*$

Recursively Defined Sets

Example Full Binary Tree: The set of full binary trees can be defined recursively by these steps:

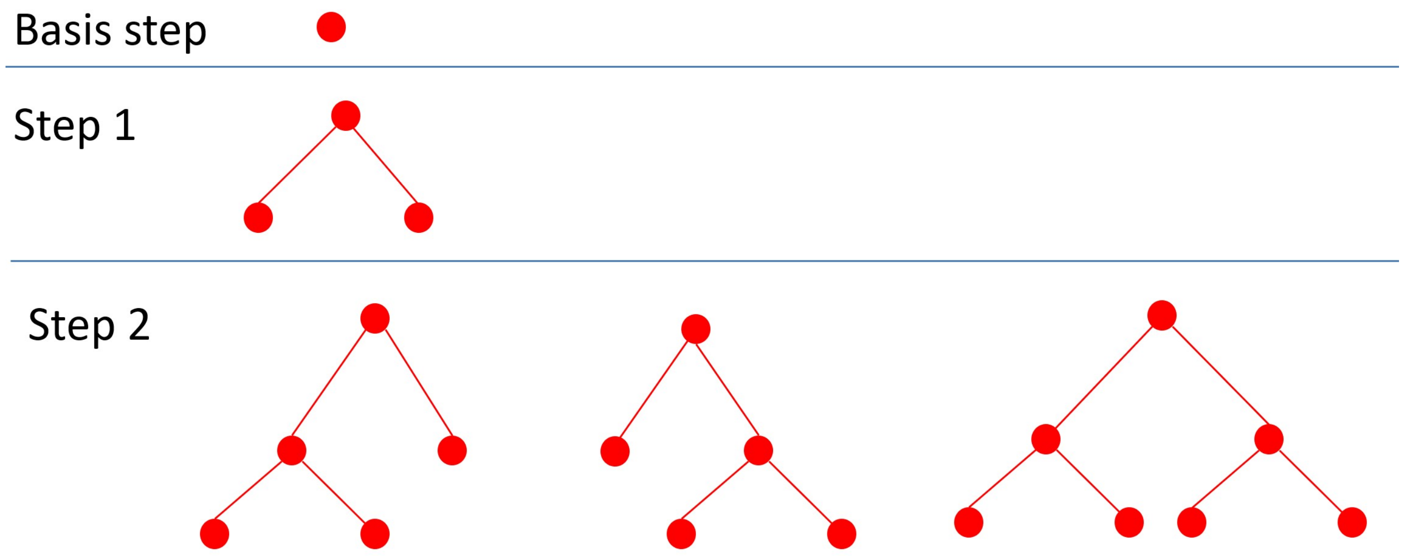
- Basis step: There is a full binary tree consisting of only a single vertex r
- Recursive step: If T_1 and T_2 are disjoint full binary trees, there is a full binary tree consisting of a root r together with edges connecting the root to each of the roots of the left subtree T_1 and the right subtree T_2.



Functions on Full Binary Tree

The height $h(T)$ of a full binary tree T is defined recursively as follows:

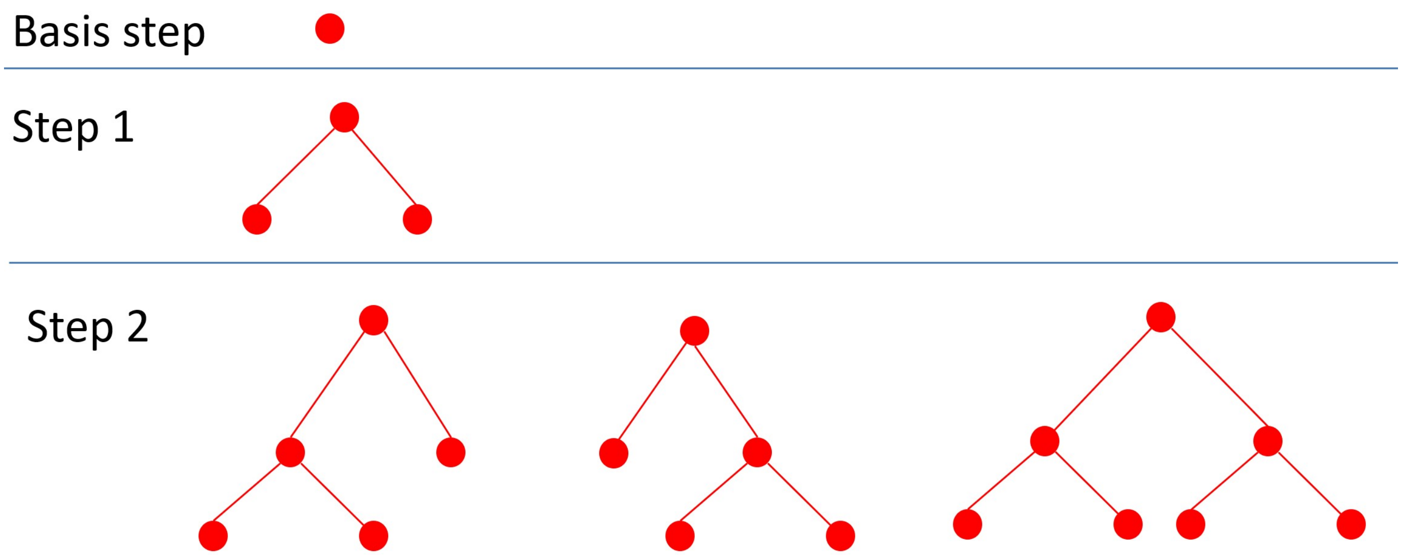
- Basis step: The height of a full binary tree T consisting of only a root r is $h(T) = 0$.
- Recursive step: If T_1 and T_2 are full binary trees, then the full binary tree T has height $h(T) = 1 + \max(h(T_1), h(T_2))$



Functions on Full Binary Tree

The number of vertices $n(T)$ of a full binary tree T is defined recursively as follows:

- Basis step: The number of vertices $n(T)$ of a full binary tree T consisting of only a root r is $n(T) = 1$.
- Recursive step: If T_1 and T_2 are full binary trees, then the full binary tree T has number of vertices $n(T) = 1 + n(T_1) + n(T_2)$.



7. Recursion

- In induction, we prove all members of an infinite set have some property P by proving the truth for larger members in terms of that of smaller members.
- In recursive definitions, we similarly define a function, a sequence or a set over an infinite number of elements by defining the function or predicate value or set-membership of larger elements in terms of that of smaller ones.
  - Recursion is a process of defining an object (functions, sequences, sets, algorithms) in terms of itself (or of part of itself).
  - There are:
    1. recursive function
    2. recursive set
    3. **recursive structure**
    4. recursive algorithm

Recursively Defined Structures

To prove a property of the elements of a recursively defined set, we use "Recursively defined structures" (structural induction).

Definition: A recursive definition of a structure consists of two steps.

- Basis step: Show that the property holds for all elements specified in the basis step of the recursive definition.
- Recursive step: Show that if the property is true for each of the parts used to construct new elements in the recursive step of the definition, then the property also holds for these new elements.

Example: Structural Induction on Binary Trees

Theorem: If T is a full binary tree, then $n(T) \leq 2^{h(T)+1} - 1$.

Recursively Defined Structures

Example: Structural Induction on Binary Trees

Theorem: If T is a full binary tree, then $n(T) \leq 2^{h(T)+1} - 1$.

Proof by structural induction:

Basis step: Show that the property holds for all elements specified in the basis step of the recursive definition.

The result holds for a full binary tree consisting only of a root:

$n(T) = 1$ and $h(T) = 0$. Hence, $n(T) = 1 \leq 2^{0+1} - 1 = 1$.

Recursive step: Show that if the property is true for each of the parts used to construct new elements in the recursive step of the definition, then the property also holds for these new elements.

By induction hypothesis we assume $n(T_1) \leq 2^{h(T_1)+1} - 1$ and also $n(T_2) \leq 2^{h(T_2)+1} - 1$ whenever T_1 and T_2 are full binary trees.

We need to proof that full binary tree built by T_1 and T_2 as left and right subtree, respectively, has $n(T) \leq 2^{h(T)+1} - 1$

Recursively Defined Structures

Recursive step: Show that if the property is true for each of the parts used to construct new elements in the recursive step of the definition, then the property also holds for these new elements.

By induction hypothesis we assume $n(T_1) \leq 2^{h(T_1)+1} - 1$ and also $n(T_2) \leq 2^{h(T_2)+1} - 1$ whenever T_1 and T_2 are full binary trees.

We need to proof that full binary tree built by T1 and T2 as left and right subtree, respectively, has $n(T) \leq 2^{h(T)+1} - 1$

$n(T) = 1 + n(T_1) + n(T_2)$ (by the recursive formula of n(T))

$\leq 1 + (2^{h(T1)+1} - 1) + (2^{h(T2)+1} - 1)$ (by inductive hypothesis)

$\leq 2 * \max(2^{h(T1)+1}, 2^{h(T2)+1}) - 1$

$= 2 * 2^{\max(h(T1),h(T2))+1} - 1$

$= 2 * 2^{h(T)} - 1$ (by the recursive definition of h(T): $h(T) = 1 + \max(h(T_1), h(T_2))$)

$= 2^{h(T)+1} - 1$

7. Recursion

- In induction, we prove all members of an infinite set have some property P by proving the truth for larger members in terms of that of smaller members.
- In recursive definitions, we similarly define a function, a sequence or a set over an infinite number of elements by defining the function or predicate value or set-membership of larger elements in terms of that of smaller ones.
  - Recursion is a process of defining an object (functions, sequences, sets, algorithms) in terms of itself (or of part of itself).
  - There are:
    1. recursive function
    2. recursive set
    3. recursive structure
    4. **recursive algorithm**

Recursive algorithm

- **Definition:** A recursive algorithm solves a problem by reducing it to an instance of the same problem with smaller input(s).

Example 1: Write a recursive algorithm for computing a^n, where $a \in R \setminus \{0\}$, $n \in N$

Recursive definition of a^n :

- initial value : $a^0 = 1$
- recursive definition: $a^n = a \cdot a^{n-1}$.

Example (A procedure to compute a^n)

procedure *power* $(a \neq 0 : \text{real}, n \in \mathbb{N})$
    if $n = 0$ **then return** 1
    else return $a \cdot \mathit{power}\,(a, n-1)$

Recursive algorithm

Example 2: Write a recursive algorithm for computing Fibonacci number $F(n)$

Recursive definition of $F(n)$:

- initial value : $F(n=0) = 0;\ F(n=1) = 1$
- recursive definition: $F(n) = F(n-1) + F(n-2)$

It can be defined recursively

procedure F(n: $n \in \mathbb{N}_0$)

if $n = 0$ **then** F(0) := 0
else if $n = 1$ **then** F(1) := 1
else F(n) := F(n-1) + F(n-2)

it can be defined iteratively

procedure F(n: $n \in \mathbb{N}_0$)

```
if n = 0 then y := 0
else {
    x := 0
    y := 1
    for i := 1 to n-1
    {    z := x+y
         x := y
         y := z
    }
}
F := y   //y is F(n)
```