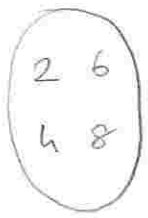
— Lecture 1.104 —

Set theory: It is a branch of mathematics that deals with properties of well-defined collection of objects. — Georg Cantor

- Set theory forms the basis of several other fields of study such as counting theory, relations, graph theory and finite state machines.
- Mathematicians use the term <u>set</u> to refer to a collection of any kind of objects: people, ideas or numbers, for example.

— A set is a well-defined collection of any kind of objects.



Set of positive even integers less-than 10

$E = \{2,4,6,8\}$

Set of vowels in the English alphabet

$V = \{a,e,i,o,u\}$

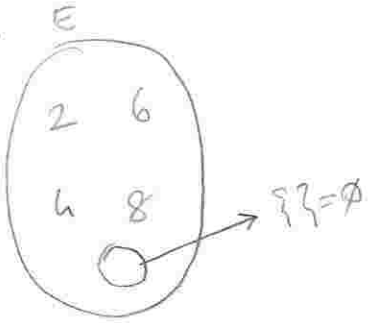
Empty set $= \in = \{\} = \emptyset$

- A set is an unordered collection of unique objects
- In mathematics we use the notation $\in$ to refer to elements of a set, and $\notin$ to refer to elements that are not in a set

→ Cardinality of a set

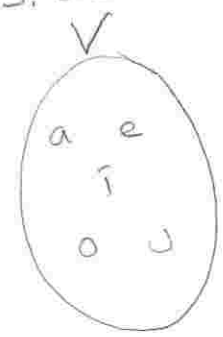
Def: Given a set S, the cardinality of S is the number of elements contained in S. We write it as $|S|$.

Ex)



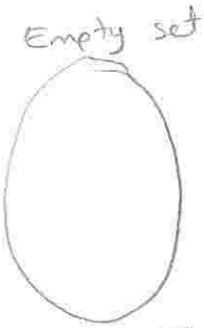
$E = \{2,4,6,8\}$

$|E| = 4$



$V = \{a,e,i,o,u\}$

$|V| = 5$



Empty set $= \{\} = \emptyset$

$|\{\}| = |\emptyset| = 0$

→ Subset of a set ($\subseteq$)

Def: A is said to be a subset of B if and only if every element of A is also an element of B. In this case we write $A \subseteq B$.

That is, we have the equivalence:

$A \subseteq B \iff$ if $x \in A$ then $x \in B$ (for all x)

$\emptyset \subseteq \{\emptyset\}$

$\emptyset \in \{\emptyset\}$

Ex)

$\{2,3\} \notin E$

$\{a,b\} \notin V$

$\{\} = \emptyset \subseteq E$

- Empty set is a subset of any set ($\emptyset \subseteq S$)

Special sets: $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}$

$\mathbb{N}$: set of natural numbers $= \{1, 2, 3, 4, \dots\}$

$\mathbb{Z}$: set of integers $= \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$

$\mathbb{Q}$: set of rational numbers (of form a/b where a & b are elements of $\mathbb{Z}$ and $b \neq 0$)

$\mathbb{R}$: set of real numbers

— Lecture 1.106 —

- Set representation:
  - Listing method
  - Set builder notation (rules of inclusion)

Listing method: Representing a set S using the listing method consists of listing all the elements of S.

Description:
S1 = set of all vowels in the English alphabet
$S1 = \{a, e, i, o, u\}$ (listing method)

Description:
S2 = set of all positive integers less 10
$S2 = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (listing method)

Description:
S3 = set of all positive even integers less than 10
$S3 = \{2, 4, 6, 8\}$ (listing method) $\longrightarrow \{2n \mid n \in \mathbb{Z}^+\}$

- Assume that we want to describe the set of rational numbers $\mathbb{Q}$.

$Q_1 = \{\dots, 1/1, 1/2, 1/3, 1/4, \dots\}$

$Q_1 = \{\frac{n}{m} \mid n, m \in \mathbb{Z} \text{ and } m \neq 0\}$

Ex) $S_1 = \{3n \mid n \in \mathbb{N} \text{ and } n < 6\} \longrightarrow S_1 = \{3, 6, 9, 12, 15\}$

$S_2 = \{2^n \mid n \in \mathbb{Z} \text{ and } 0 \leq n \leq 4\} \rightarrow S_2 = \{1, 2, 4, 8, 16\}$

$S_3 = \{2^{-n} \mid n \in \mathbb{Z} \text{ and } 0 \leq n < 8\} \rightarrow S_3 = \{1, \frac{1}{2}, \frac{1}{4}, \dots, \frac{1}{128}\}$

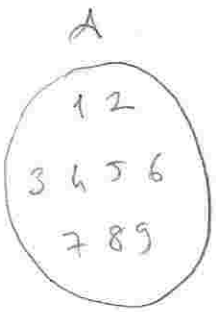
$S_4 = \{\frac{1}{2n} \mid n \in \mathbb{Z} \text{ and } 0 < n \leq 5\} \rightarrow S_4 = \{\frac{1}{2\times1}, \frac{1}{2\times2}, \dots, \frac{1}{2\times5}\}$

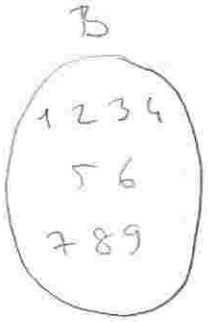
$S_5 = \{2^n \mid n \in \mathbb{Z} \text{ and } 0 \leq n \leq 7\} \rightarrow S_5 = \{2^0, 2^1, 2^2, \dots, 2^7\}$

— LECTURE 1.108 —

- Subset (recap)

A set can have another set as its element.





$A = \{1,2,3,4,5,6,7,8,9\}$ $\quad B = \{\{1,2,3,4\},\{5,6\},\{7,8,9\}\}$

$\{1,2,3,4\} \subseteq A$ but $\{1,2,3,4\} \in B$

- Powerset of a set

Def: Given a set S, the powerset of S, $P(S)$, is the set containing all the subsets of S.

Given a set $S = \{1,2,3\}$

→ The subsets of S are:

$\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}$

$P(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

Ex) $P(\emptyset) = ?$, $P(P(\emptyset)) = ?$

$\emptyset \subseteq \emptyset$, $P(\emptyset) = \{\emptyset\}$ therefore $\emptyset \subseteq P(\emptyset)$, $\{\emptyset\} \subseteq P(\emptyset)$

Therefore : $P(P(\emptyset)) = \{\emptyset, \{\emptyset\}\}$

- Cardinality of a powerset

Def: Given a set S, then $|P(S)| = 2^{|S|}$

Ex) $S = \{1,2,3\}$, $|S| = 3$

$P(S) = \{\emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\}$

$|P(S)| = 8 = 2^3 = 2^{|S|}$

Ex) Given a set A, if $|A| = n$, find $|P(P(P(A)))|$

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

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

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

– LECTURE 1.110

→ Set operations and membership tables:

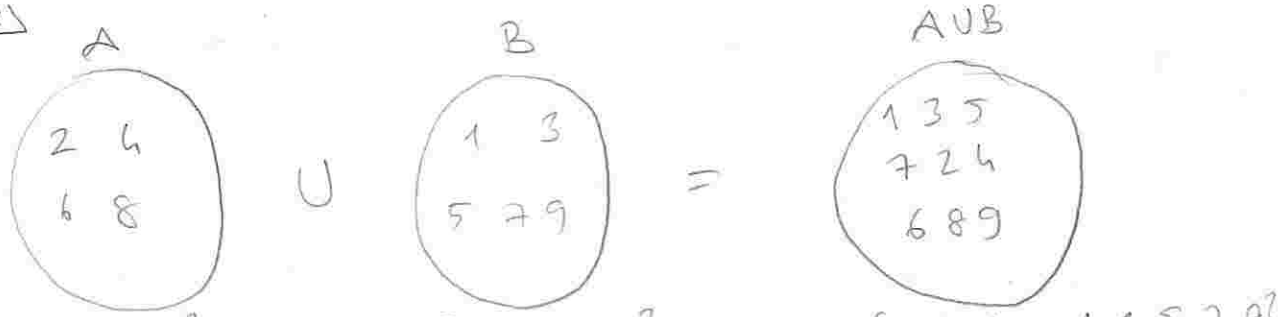
- Set intersection
- Set union
- Set difference
- Symmetric difference

- Union ($A \cup B$)

Def: Given two sets A and B, the union of A & B, $A \cup B$, contains all the elements in either A or B.

$A \cup B = \{x \mid x \in A \text{ or } x \in B\}$

Ex)



$A = \{2,4,6,8\}$ $B = \{1,3,5,7,9\}$ $A \cup B = \{2,4,6,8,1,3,5,7,9\}$

→ Membership table ($A \cup B$)

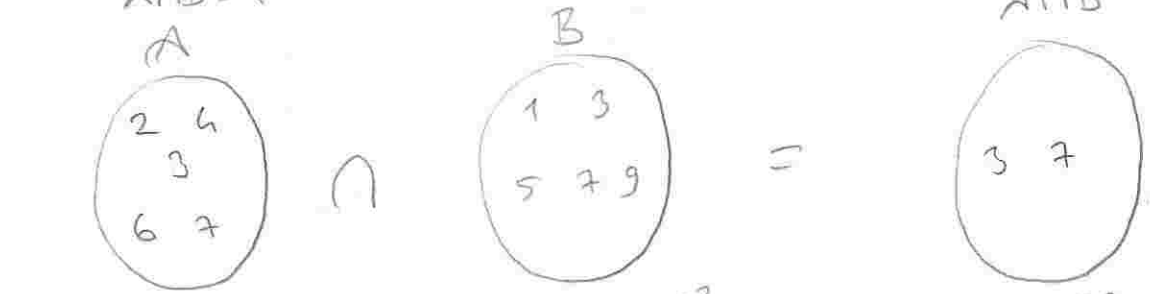
- 1 means the element belongs to the set
- 0 means it does not belong to the set

A	B	A ∪ B
0	0	0
0	1	1
1	0	1
1	1	1

- Intersection

Def: Given two sets A and B, the intersection of A and B, $A \cap B$ contains all the elements in both A & B.

$A \cap B = \{x \mid x \in A \text{ and } x \in B\}$



$A = \{2,3,4,6,7\}$ $B = \{1,3,5,7,9\}$ $A \cap B = \{3,7\}$

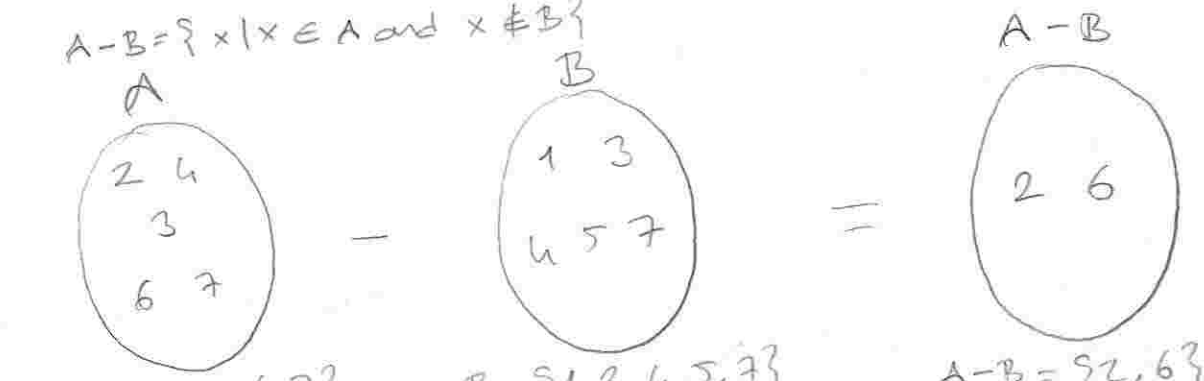
→ Membership table ($A \cap B$)

A	B	$A \cap B$
0	0	0
0	1	0
1	0	0
1	1	1

- Set difference ($A-B$)

Def: Given two sets A and B, the set difference, $A-B$, contains the elements that are in A but not in B.

$A-B = \{x \mid x \in A \text{ and } x \notin B\}$



$A = \{2,3,4,6,7\}$ $B = \{1,3,4,5,7\}$ $A-B = \{2,6\}$

→ Membership table

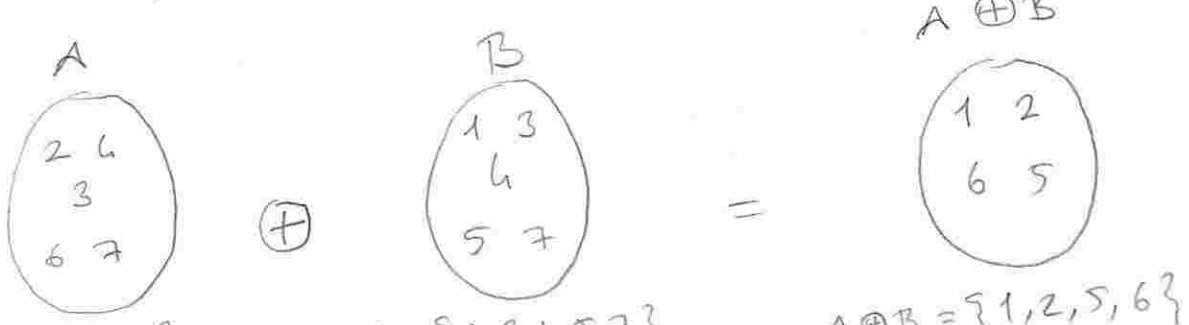
A	B	$A-B$
0	0	0
0	1	0
1	0	1
1	1	0

- Symmetric difference ($A \oplus B$)

Def: Given two sets A and B, the symmetric difference, $A \oplus B$, contains the elements that are in A or B but not in both.

$A \oplus B = \{x \mid (x \in A \text{ or } x \in B) \text{ and } x \notin A \cap B\}$

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



$A = \{2,3,4,6,7\}$ $B = \{1,3,4,5,7\}$ $A \oplus B = \{1,2,5,6\}$

→ Membership table ($A \oplus B$)

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

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

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

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

$A - B = \{1,2\}$

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

★ Essential Reading

- Discrete Mathematics & its applications 7th Edition
  pp. 115-118
- Exercises: 1 to 12 on p. 125 & 15 to 35 on p. 126

— LECTURE 1.201 —

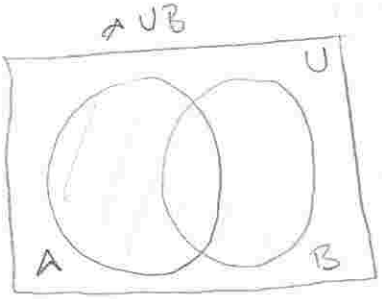
WEEK 2

- Universal set: it is a set that contains everything. We note the universal set with the letter U.
- Venn diagrams: A venn diagram is used to visualise the possible relations among a collection of sets. Venn $\subseteq U$
- Complement of a set A, $\bar{A}$, contains all the elements in the universal set U but not in A.

$\bar{A} = U - A$ & $\bar{A} \cup A = U$

Ex) $U = \{1,2,3,4,5,6,7,8,9,10\}$ and $A = \{2,4,6,8,10\}$

then $\bar{A} = \{1,3,5,7,9\}$



Venn-diagram for $A \cup B$

-LECTURE 1.205-

- Commutativity
  - Set union is commutative
    $A \cup B = B \cup A$
  - Set interaction is commutative
    $A \cap B = B \cap A$
  - Symmetric difference is commutative
    $A \oplus B = B \oplus A$
  - Set difference is <u>not</u> commutative
    $A - B \neq B - A$

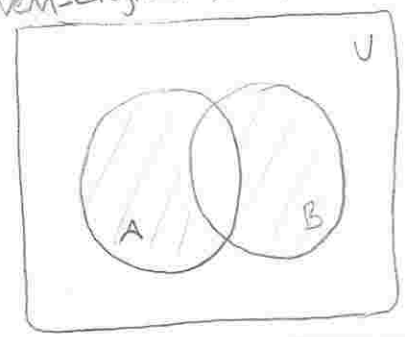
- Associativity
  Def: It concerns the grouping of elements in an operation

Ex) The addition of numbers is associative.
Given three numbers a, b, and c, we have:
$(a+b)+c = a+(b+c)$

- Set union is associative
  $(A \cup B) \cup C = A \cup (B \cup C)$
- Set intersection is associative
  $(A \cap B) \cap C = A \cap (B \cap C)$
- Symmetric difference is associative
  $(A \oplus B) \oplus C = A \oplus (B \oplus C)$
- Set difference is not associative
  $(A - B) - C \neq A - (B - C)$

- Distributivity
  Def: The distributive property is sometimes called the distributive law of multiplication and division
  $a(b+c) = ab + ac$
  We say that the multiplication is distributive over the addition.
  - The set union is distributive over the set intersection:
    $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
  - The set intersection is distributive over the set union:
    $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

Venn-diagram for $A \oplus B$



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

— LECTURE 1.2.3 —

De Morgan's Law

Def: The complement of the union of two sets A and B is equal to the intersection of their complements.

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

The complement of the intersection of two sets A and B is equal to the union of their complements

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

⟶ Proof using membership tables of $\overline{A \cup B} = \overline{A} \cap \overline{B}$

A	B	$\overline{A}$	$\overline{B}$	$A \cup B$	$\overline{A \cup B}$	$\overline{A} \cap \overline{B}$
0	0	1	1	0	1	1
0	1	1	0	1	0	0
1	0	0	1	1	0	0
1	1	0	0	1	0	0

$A \cap B$	$\overline{A \cap B}$	$\overline{A} \cup \overline{B}$
0	1	1
0	1	1
0	1	1
1	0	0

Ex) Given $U = \{1,2,3,4,5,6,7,8,9\}$ and A, B two subsets of U

$A = \{1,2,3,4\}$ and $B = \{4,5,6,7\}$

$A \cup B = \{1,2,3,4,5,6,7\}$ and

$A \cap B = \{4\}$

$\overline{A \cup B} = \{8,9\}$ and $\overline{A \cap B} = \{1,2,3,5,6,7,8,9\}$

$\overline{A} = \{5,6,7,8,9\}$ and $\overline{B} = \{1,2,3,8,9\}$

$\overline{A} \cap \overline{B} = \{8,9\} = \overline{A \cup B}$

$\overline{A} \cup \overline{B} = \{1,2,3,5,6,7,8,9\} = \overline{A \cap B}$

Union	Name	Intersection
$A \cup B = B \cup A$	commutative	$A \cap B = B \cap A$
$(A \cup B) \cup C = A \cup (B \cup C)$	associative	$(A \cap B) \cap C = A \cap (B \cap C)$
$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$	distributive	$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
$\overline{A \cup B} = \overline{A} \cap \overline{B}$	De Morgan's laws	$\overline{A \cap B} = \overline{A} \cup \overline{B}$
$A \cup \emptyset = A$ $A \cup U = U$	Identities	$A \cap \emptyset = \emptyset$ $A \cap U = A$
$A \cup \overline{A} = U$ $\overline{U} = \emptyset$	complement	$A \cap \overline{A} = \emptyset$ $\overline{\emptyset} = U$
$\overline{\overline{A}} = A$	double complement	
$A \cup (A \cap B) = A$	absorption	$A \cap (A \cup B) = A$
$A - B = A \cap \overline{B}$	set difference	

Ex) Show that $\overline{(A \cap B) \cup \overline{B}} = B \cap \overline{A}$

$$\overline{(A \cap B) \cup \overline{B}} = \overline{(A \cap B)} \cap \overline{\overline{B}} \qquad \text{De Morgan's law}$$

$$= \overline{(A \cap B)} \cap B$$

$$= (\overline{A} \cup \overline{B}) \cap B \qquad \text{De morgan's law}$$

$$= B \cap (\overline{A} \cup \overline{B}) \qquad \text{commutative}$$

$$= (B \cap \overline{A}) \cup (B \cap \overline{B}) \qquad \text{Distributive}$$

$$= (B \cap \overline{A}) \cup \emptyset$$

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

Example from 1.206:

Let A and B be two sets. Which of the following is equivalent to $\overline{A \cup B} \cup A$?

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

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

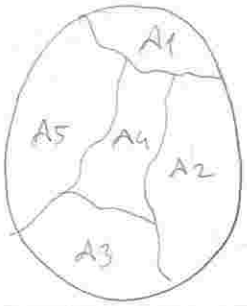
$$= (A \cup B) \cap A$$

$$= A$$

- LECTURE 1.207 -

- Dis-joint sets: Two sets A and B are disjoint if and only if $A \cap B = \emptyset$
- Partition

Def: A partition of A is a set of subsets A_i of A such that:
all the subsets A_i are dis-joints the union of all subsets A_i is equal to A



$A_1 \cap A_2 = A_2 \cap A_3 = \dots = A_4 \cap A_5 = \emptyset$

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

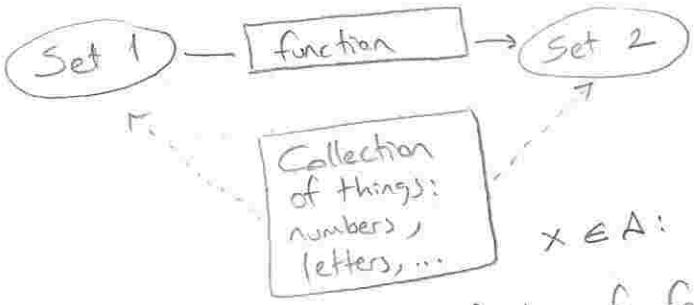
$\{A_1, A_2, A_3, A_4, A_5\}$ is a partition on A.

- LECTURE 2.102 -

Introduction:

A function is a relation between a set of inputs and a set of outputs so that each input maps to exactly one output.

The concept of a function is central to computer programming. Most of what a programmer writes consists of 'functions' that do parts of the work of the program.



$f: A \to B$

f maps A to B

$x \in A: \; x \to f(x) = y \quad (y \in B)$

Definition of a function: A function f from a set A to a set B is an assignment of exactly one element of B to each element of A

Terminology

Given a function $f: A \to B$

$x \in A \to f(x) = y \in B$

A is the set of inputs and is called the domain of f. We write $D_f = A$. B is the set containing the outputs and is called the co-domain of f. We write $co\text{-}D_f = B$.

The set of all outputs is called the range of f and it is written as R_f. y is called the image of x, whereas x is called the pre-image of y. We write $f(x)=y$.

Domain, co-domain & Range

$A=\{$on, sea, Land, Sky$\}$ — function (number of characters) — $B=\{1,2,3,4,5,6,7\}$

$D_f = A = \{$on, sea, land, sky$\}$

$Co-D_f = B = \{1,2,3,4,5,6,7\}$ /// $R_f = \{2,3,4\}$

Ex| $A=\{0,2,3,4\}$ $\quad B=\{1,2,3,4,5,6,7,8,9\}$

A → f → B

$x = 2x+1$

$D_f = A = \{0,1,2,3,4\}$, $\quad R_f = \{1,3,5,7,9\}$

Ex| Given the following function:

$f: \mathbb{Z} \to \mathbb{Z}$ with $f(x)=|x|$

Find the domain, co-domain and range of the function f.

Find the set of pre-images(1).

$D_f = \mathbb{Z}$

$Co-D_f = \mathbb{Z}$

$R_f = \mathbb{Z}^+ \cup \{0\} = \{0,1,2,3,\dots,\dots\}$

$f(1) = f(1) = 1$ hence, pre-images of $1 = \{-1,1\}$

Ex| Given the following function:

$g: \mathbb{R} \to \mathbb{R}$ with $g(x)=x^2+1$

Find the domain, co-domain and range of the function g.

Find the set of pre-images(5).

$D_g = \mathbb{R}$

$Co-D_g = \mathbb{R}$

$R_g = [1, +\infty]$

$g(-2) = g(2) = 5$ hence, pre-images of $5 = \{-2,2\}$

- LECTURE 2.104 -

Outlines
- Linear functions : $f(x)=ax+b$, a is gradient
- Quadratic functions : $f(x)=ax^2+bx+c$, $a\neq 0$ & a,b,c are numbers
- Exponential functions : $f(x)=b^x$ where $b>0$ & $b\neq 1$

$b^x b^y = b^{x+y}$	$(b^x)^y = b^{xy}$	$b^{-x} = \frac{1}{b^x}$
$\frac{b^x}{b^y} = b^{x-y}$	$(ab)^x = a^x b^x$	$\left(\frac{a}{b}\right)^x = \frac{a^x}{b^x}$

In exponential function, Domain is $=]-\infty, \infty[$ and the Range is $=]0, \infty[$

- LECTURE 2.106 -

Outlines
- Injective (one-to-one) functions
- Surjective (onto) functions

→ Injective (one-to-one) functions

Let $f: A \to B$ be a function

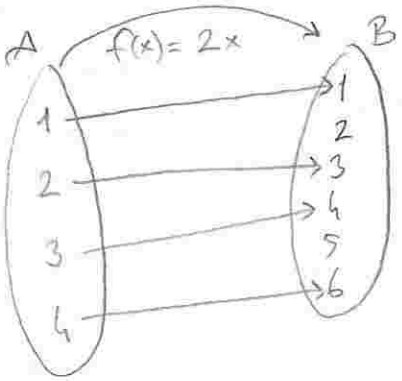
f is said to be an injective (one-to-one) function iff:

for all $a, b \in A$, if $a \neq b$ then $f(a) \neq f(b)$

≡

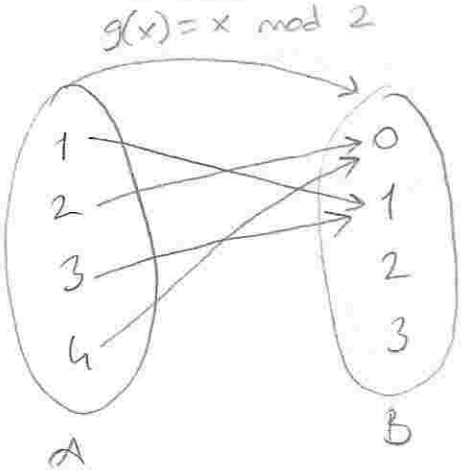
for all $a, b \in A$, if $f(a) = f(b)$ then $a = b$

Ex1



f is injective (one-to-one) as every element of A has a unique image in B.

Ex1



g is not injective (not one-to-one) as 2 or 4 in A has the same image 0 or 1 and 3 in A has the same image 1

Ex| Let's show that the function $f: \mathbb{R} \to \mathbb{R}$ with $f(x) = 2x+3$ is an injection (one-to-one) as linear function

Proof 1:

Let $a, b \in \mathbb{R}$, show that if $f(a) = f(b)$ then $a = b$

$f(a) = f(b) \Rightarrow 2a+3 = 2b+3 \Rightarrow 2a = 2b \Rightarrow a = b$

Therefore, f is injective

Proof 2:

Let $a, b \in \mathbb{R}$, show that if $a \neq b$ then $f(a) \neq f(b)$

$a \neq b \Rightarrow 2a \neq 2b \Rightarrow 2a+3 \neq 2b+3 \Rightarrow f(a) \neq f(b)$

Therefore, f is injective

Ex) Let's show that the function $f: \mathbb{R}^+ \to \mathbb{R}$ with $f(x) = x^2$ is an injection (one-to-one) as quadratic functions

Proof 1:

Let $a, b \in \mathbb{R}^+$, show that if $f(a) = f(b)$ then $a = b$

$f(a) = f(b) \Rightarrow a^2 = b^2 \Rightarrow a = b$ as $a, b \in \mathbb{R}^+$ $\Rightarrow$ f is injective

Proof 2:

Let $a, b \in \mathbb{R}^+$, show that if $a \neq b$ then $f(a) \neq f(b)$

$a \neq b \Rightarrow a^2 \neq b^2$ as $a, b \in \mathbb{R}^+$ $\Rightarrow f(a) \neq f(b) \Rightarrow$ f is injective

---

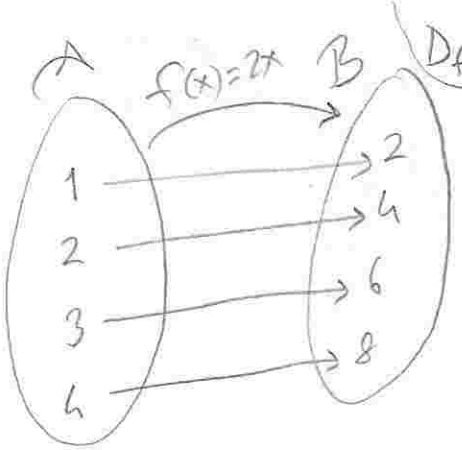
→ Surjective (onto) functions

Let $f: A \to B$ be a function

f is said to be a surjective (onto) function iff if every element of the co-domain of f, B, has at least one pre-image in the domain of f, A.

for all $y \in B$ there exists $x \in A$ such that $y = f(x)$

$\equiv$ $D_f = R_f$

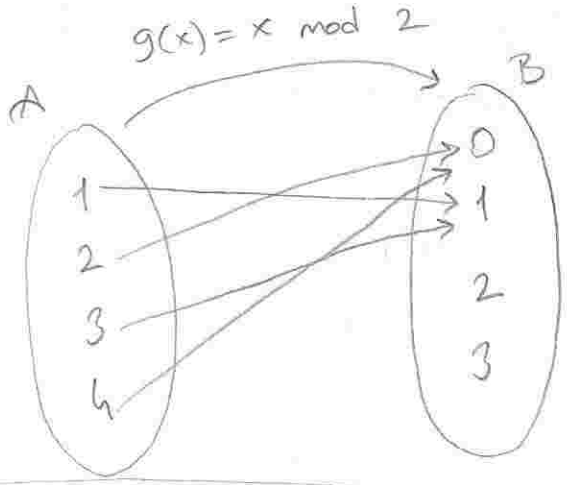


f is surjective function as every element of B has at least one pre-image

Or $D_f = R_f$

Co-$D_f = \{0, 1, 2, 3\}$

$R_f = \{2, 4, 6, 8\}$



g is not surjective as 2 & 3 have no pre-images
or $D_g \neq R_g$
$Co\text{-}D_g = \{0,1,2,3\}$
$R_g = \{0,1\}$

Ex) Let's show that the function $f: \mathbb{R} \to \mathbb{R}$ with $f(x) = 2x+3$ is a surjective (onto) function as linear function

Proof:

Let $y \in \mathbb{R}$, show that there exists $x \in \mathbb{R}$ such that $f(x) = y$

$f(x) = y \Rightarrow 2x+3 = y \Rightarrow 2x = y-3 \Rightarrow x = \frac{y-3}{2} \in \mathbb{R}$

Hence, for all y in $\in \mathbb{R}$, there exists $x = \frac{y-3}{2} \in \mathbb{R}$ such that $f(x) = y \rightarrow f$ is a surjective (onto) function

---

Ex) Let's show that the function $f: \mathbb{R} \to \mathbb{R}$ with $f(x) = x^2$ is not a surjective (onto) function as quadratic functions

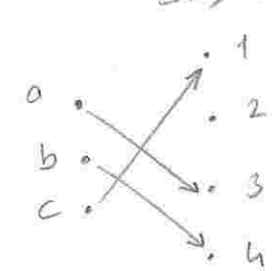
Proof:

Let $y \in \mathbb{R}$, show that there exists $x \in \mathbb{R}$ such that $f(x) = y$.

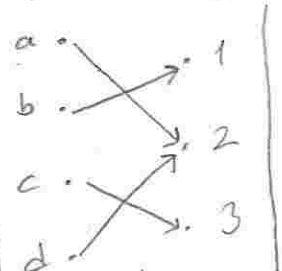
R_f (set images) $= [0, +\infty[\neq R(Co\text{-}D_f) = \mathbb{R}$

- All negative numbers have no pre-images

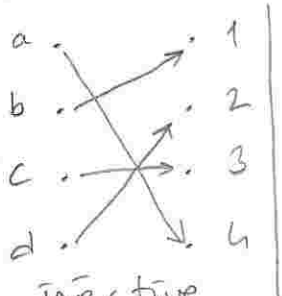
So, f is not a surjective function



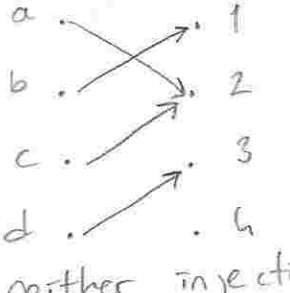
injective, not surjective



surjective, not injective



injective & surjective



neither injective nor surjective

— Practice Quiz 2.107 —

1) A function f is said to be one-to-one (injective) iff:
$f(a) = f(b)$ implies $a = b$ for all a & b in the domain of f

2) A function f is said to be onto (surjective) iff:
for all b in the co-domain of f there exists a in the domain of f such that $f(a) = b$

3) Let $f: R \to [1, +\infty)$ with $f(x) = x^2 + 1$;
f is one-to-one function as for all $y \in Co\text{-}D_f$ there exists $x \in D_f$ such that $f(x) = y$

4) Let $f: Z \to Z$ with $f(x) = 2x + 3$;
f one-to-one (injective) as there exists $a, b \in Z$ where if $a \neq b$ then $f(a) \neq f(b)$

— LECTURE 2.201 —

Outlines:
- Composition of two functions $(f \circ g)$
- Function composition is not commutative

Function composition
Given two functions f & g:

$$(f \circ g)(x) = f(g(x))$$

Ex) $f(x) = 2x$ & $g(x) = x^2$, what is $(f \circ g)(x) = ?$

$$(f \circ g)(x) = f(g(x)) = g(x^2) = 2x^2$$
$$(f \circ g)(1) = 2 = f(g(1)) = 1(1) = 2$$

- Function composition is not commutative!

$$(f \circ g) \neq (g \circ f)$$

Ex) $f(x) = 2x$ & $g(x) = x^2$

$$(f \circ g)(x) = 2(x^2) = 2x^2$$
$$(g \circ f)(x) = (2x)^2 = 4x^2$$

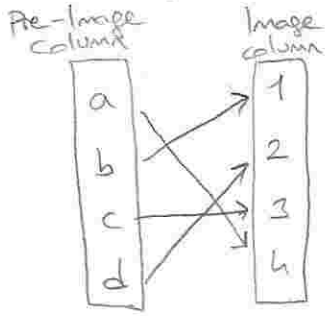
So, $2x^2 \neq 4x^2$

-LECTURE 2.203-

Outlines:
- Def. of a bijective (invertible) function
- Finding the inverse function
- Graph of the inverse function

- Bijective functions: invertible

This following function is both injective (one-to-one) & surjective (onto):



- Such a function is called a bijective or an invertible function.
- A function is said to be bijective or invertible if and only if it is both injective & surjective

Ex) Show that the function $f: \mathbb{R} \to \mathbb{R}$ with $f(x) = 2x+3$ is a bijective function.

→ Show that f is injective (one-to-one)

Proof: Let $a, b \in \mathbb{R}$, show that if $a \neq b$ then $f(a) \neq f(b)$

$a \neq b \Rightarrow 2a \neq 2b \Rightarrow 2a+3 \neq 2b+3 \Rightarrow f(a) \neq f(b) \Rightarrow f$ is injective

→ Show that f is surjective (onto)

Proof: Let $y \in \mathbb{R}$, show that there exists $x \in \mathbb{R}$ such that $f(x) = y$

$f(x) = y \Rightarrow 2x+3 = y \Rightarrow 2x = y-3 \Rightarrow x = \frac{y-3}{2} \in \mathbb{R}$

Hence, for all y in $\in \mathbb{R}$, there exists $x = \frac{y-3}{2} \in \mathbb{R}$ such that $f(x) = y$

- Inverse function

Defn: Let $f: A \to B$, if f is bijective (invertible) then the inverse function, f^{-1} exists and is defined as $f^{-1}: B \to A$

Ex) Prove that $f(x) = 2x+3$ has inverse function.

$f: \mathbb{R} \to \mathbb{R}$

$x \to f(x) = 2x+3$

$f(x) = y$

$2x+3 = y$

$2x = y-3$

$x = \frac{y-3}{2}$

$f^{-1}: \mathbb{R} \to \mathbb{R}$

$x \to f^{-1}(x) = \frac{x-3}{2}$

$(f \circ f^{-1})(x) = (f^{-1} \circ f)(x) = x$

$f: \mathbb{R} \to \mathbb{R}$ with $f(x) = 2x$

$f^{-1}: \mathbb{R} \to \mathbb{R}$ with $f^{-1}(x) = \frac{x}{2}$

$(f \circ f^{-1})(x) = f(f^{-1}(x)) = f\left(\frac{x}{2}\right) = 2\frac{x}{2} = x$

$(f^{-1} \circ f)(x) = f^{-1}(f(x)) = f^{-1}(2x) = \frac{2x}{2} = x$

→ Plotting the inverse function

The curves of f & f^{-1} are always symmetric with respect to the straight line $y = x$.

— LECTURE 2.20 —

Outlines:
- Exponential function (recap)
- Logarithmic function

- Exponential functions: The function f defined by: $f: \mathbb{R} \to \mathbb{R}^+$ and $f(x) = b^x$ where $b > 0$ and $b \neq 1$ is called an exponential function with a base b.
- Properties of the exponential function

$y = f(x) = b^x$ ($b > 0$ & $b \neq 1$) → The domain is $(-\infty, \infty)$

The range is $(0, \infty)$, It passes through the point $(0, 1)$,

If $b > 1$ it is increasing on $(-\infty, \infty)$

If $b < 1$ it is decreasing on $(-\infty, \infty)$

- Logarithmic functions

The logarithmic function with base b where $b > 0$ & $b \neq 1$ is defined as follows:

$\log_b x = y$ if and only if $x = b^y$

$\log_b x$ is the inverse function of the exponential function b^x.

→ Conversion between logarithmic & exponential forms

$y = \log_b x$ iff $x = b^y$

logarithmic form — exponential form

↳ log for x base b is equal to y

- Laws of logarithms

$\log_b m \times n = \log_b m + \log_b n$

$\log_b \frac{m}{n} = \log_b m - \log_b n$

$\log_b m^n = n \log_b m$

$\log_b 1 = 0$

$\log_b b = 1$

Natural logarithm: $\ln x$

$\ln x = \log_e x$ where $e = 2.71828$

$\ln e = \log_e e = 1$

- LECTURE 2.207 -

Outlines:

- The floor function → $n \leq x < n+1$
- The ceiling function → $n < x \leq n+1$

- Floor function: It is a function $\mathbb{R} \to \mathbb{Z}$. It takes a real number x as an input and returns the largest integer that is less than or equal to x, denoted as floor $(x) = \lfloor x \rfloor$

numbers	floor numbers	
10	10	$\lfloor x \rfloor$
1.1	1	
1.99	1	
-1.1	-2	
-1.99	-2	

- Ceiling function: It is a function $\mathbb{R} \to \mathbb{Z}$. It takes a real number x as an input and returns the smallest integer that is greater than or equal to x, denoted as ceiling $(x) = \lceil x \rceil$

numbers	Ceiling number	
10	10	$\lceil x \rceil$
1.1	2	
1.99	2	
-1.1	-1	
-1.99	-1	

Ex) Let n be an integer & x a real number. Show that:

$$\lfloor x+n \rfloor = \lfloor x \rfloor + n$$

Proof:

Let $m = \lfloor x \rfloor$

hence, $m \leq x < m+1$ (by definition)

$m+n \leq x+n < m+n+1$

this implies that $\lfloor x+n \rfloor = m+n$ (by def'n)

hence, $\lfloor x+n \rfloor = \lfloor x \rfloor + n$

— LECTURE 3.101 —

Outlines
- propositional logic
- application of propositional logic

Def'n: Propositional logic is a branch of logic interested in studying mathematical statements. It is dating back to Aristotle, was to model reasoning. It is effectively an algebra of propositions.
→ Operators: and, or, not, implies, if and only if
→ Many systems for reasoning by computers, including theorem provers, program verifiers and applications in the field of artificial intelligence, have been implemented in logic-based programming languages.
- These languages generally use "predicate logic", a more powerful form of logic that extends the capabilities of propositional logic.

— LECTURE 3.103 —

Outlines
- definition of a proposition
- propositional variables

Def'n: A proposition is a declarative sentence that is either true or false, but not both & is the most basic element of logic.

Ex)
- London is the capital of the United Kingdom → True
- $1+1=2$ → True
- $2<3$ → True

- Madrid is the capital of France → False
- $3<2$ → False
- 10 is an odd number → False

* The false ones are still propositions but false.

Ex) Examples that are not propositions

- $x + 1 = 2$
- $x + y = z$
- What time is it?
- Read this carefully
- This coffee is strong

→ To avoid writing long & repetitive propositions, we use propositional variables

→ A propositional variable is typically a letter, such as: $p, q, r, \ldots$

Ex) p: London is the capital of the United Kingdom

q: $1+1=2$

r: $2<3$

s: $1+0=1$

— LECTURE 3.105 —

Outlines

- definition of a truth table
- constructing a truth table
- definition of a truth table

Def'n of truth table: A truth table is a tabular representation of all the possible combinations of its constituent variables.

→ To construct the truth table for n propositions:

Create a table with 2^n rows & n columns

Ex) 2^3 → which are p, q, r → 8 rows

p	q	r
F	F	F
F	F	T
F	T	F
F	T	T
T	F	F
T	F	T
T	T	F
T	T	T

Begin the truth table with False value and repeat it with 2^{n-x}, means;

$p \to 2^2$, $q \to 2^1$, $r \to 2^0$

- Def'n of a truth set

Let p be a proposition on a set S. The truth set of p is the set of elements of S for which p is true.

→ We use a capital letter to refer to a truth set of a proposition, i.e. the truth set of a proposition p is noted as P.

Ex| Let $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$

Let p & q be two propositions concerning an integer n in S, defined as follows:

p : n is even

q : n is odd

The truth set of p written as P is:

$P = \{2, 4, 6, 8, 10\}$

The truth set of q is:

$Q = \{1, 3, 5, 7, 9\}$

-LECTURE 3.107-

Outlines

- def'n of compound statements
- negation
- conjuction
- disjunction
- exclusive-or
- precedence of logical operations

→ Def'n of compound statements: Compound propositions are statements built by combining multiple propositions using certain rules.

→ Negation (¬)

Let p be a proposition. The negation of p, denoted by ¬p, and read "not p", is the statement:

"It is not the case that p."

* The truth value of the negation of p, ¬p, is the opposite of the truth value of p

Ex|

- p: "John's program is written in Python"
- ¬p: "John's program is not written in Python."

p	¬p
F	T
T	F

→ Conjunction ($\wedge$)

Let p and q be propositions. The conjunction of p and q, denoted by $p \wedge q$, is the proposition "p and q".

- It is only true when both are true.

Ex) p: "John's program is written in Python"
q: "John's program has less than 20 lines of code."
$p \wedge q$: "John's program is written in Python and has less than 20 lines of code."

→ Disjunction ($\vee$)

Let p and q be propositions. The disjunction of p and q, denoted by $p \vee q$, is the proposition "p or q"

- It is only false when both are false.

→ Exclusive-or ($\oplus$)

Let p and q be propositions. The exclusive-or of p and q, denoted by $p \oplus q$, is the proposition "p or q" (but not both)

- The exclusive-or (xor) $p \oplus q$ is true when p is true and q is false and when p is false and q is true.

Ex) $p \oplus q$: "John's program is written in Python or has less than 20 lines of code, but not both."

→ Precedence of logical operations

To build complex compound propositions we need to use parentheses

Ex) • $(p \vee q) \wedge (\neg r)$ is different from $p \vee (q \wedge \neg r)$

p	q	$(p \wedge q)$	$(p \vee q)$	$p \oplus q$	$\neg p$	$\neg p \vee (p \wedge q)$
F	F	F	F	F	T	T
F	T	F	T	T	T	T
T	F	F	T	T	F	F
T	T	T	T	F	F	T

- LECTURE 3.202 -

Outlines
- def'n of implication
- truth table for implication
- different expression of implication
- converse, contrapositive and inverse

→ def'n of implication

Let p & q be propositions. The conditional statement, or implication $p \to q$ is the proposition: ($p \to q$: p implies q)

"if p then q"

Ex) Let p and q be the following statements:
- p: "John did well in discrete mathematics."
- q: "John will do well in the programming course."

→ The conditional statement $p \to q$ can be written as follows:

"If John did well in discrete mathematics then John will do well in the programming course."

→ Truth table for implication

p	q	$p \to q$
F	F	T
F	T	T
T	F	F
T	T	T

→ Different expressions of $p \to q$

Let p and q be the following statements:
- p: "It's sunny."
- q: "John goes to the park"

$p \to q$	
if p then q	
if p, q	
p implies q	If it is sunny then John goes to the park
p only if q	
q follows from p	
p is sufficient for q	
q unless $\neg p$	
q is necessary for p	

→ Converse, contrapositive and inverse

- Let p and q be propositions and A the conditional statement $p \to q$
- The proposition $q \to p$ is the converse of A
- The proposition $\neg q \to \neg p$ is the contrapositive of A
- The proposition $\neg p \to \neg q$ is the inverse of A

Ex| Let p and q be the following statements:

→ And A the statement $p \to q$: "If it's sunny then John goes to the park."

→ The converse of A is: "If John goes to the park then it's sunny"

→ The contrapositive of A is:
"If John doesn't go to the park then it's not sunny."

→ The inverse of A is:
"If it's not sunny, then John doesn't go to the park."

Ex) Let p and q be two propositions concerning an integer n

- p: n has one digit
- q: n is less 10

1. "If the integer n has one digit then it is less than 10"
$p \to q$
2. Contrapositive of 1.
$\neg q \to \neg p$
"If n is greater than or equal to 10 then n has more than one digit"

— LECTURE 3.2.4 —

Outlines:

- def'n of equivalence
- equivalence properties
- equivalent propositions
- proving equivalence
- proving non-equivalence
- precedence of logical operations

→ Def'n of equivalence

Let p and q be propositions. The biconditional or equivalence statement $p \leftrightarrow$ is the proposition

"$p \to q$ and $q \to p$"

→ Equivalence properties ($\leftrightarrow$)

- Biconditional statements are also called bi-implications.
- $p \leftrightarrow$ can also be read "p if and only if q".
- The biconditional statement $p \leftrightarrow q$ is true when p and q have the same truth values, and is false otherwise.

p	q	$p \to q$	$q \to p$	$(p \to q) \wedge (q \to p)$: equivalence
F	F	T	T	T
F	T	T	F	F
T	F	F	T	F
T	T	T	T	T

→ Equivalent propositions

- Let p and q be propositions. p and q are logically equivalent if they always have the same truth value

  We write $p \equiv q$
- The symbol $\equiv$ is not a logical operator, and $p \equiv q$ is not a compound proposition but, rather, is the statement that $p \leftrightarrow q$ is a always true

→ Providing equivalence

To determine equivalence, we can use truth tables

Ex) Let's compare the two propositions $p \to q$ and $\neg p \vee q$

p	q	$p \to q$	$\neg p$	$\neg p \vee q$
F	F	T	T	T
F	T	T	T	T
T	F	F	F	F
T	T	T	F	T

$\neg p \vee q$ is equivalent to $p \to q$

→ Proving non-equivalence

To determine equivalence, we can use truth tables and find at least one row where values differ.

Ex) Let's examine whether the converse or the inverse of an implication is equivalent to the original implication.

p	q	$\neg p$	$\neg q$	$p \to q$	$\neg p \to \neg q$	$q \to p$
F	F	T	T	T	T	T
F	T	T	F	T	F	F
T	F	F	T	F	T	T
T	T	F	F	T	T	T

-LECTURE 3.2.06

Outlines

- laws of propositional logic
- equivalence proof.

→ laws of propositional logic 1

Lets share some important equivalences!

	Conjunction	Disjunction
idempotent law	$p \vee p \equiv p$	$p \wedge p \equiv p$
commutative laws	$p \vee q \equiv q \vee p$	$p \wedge q \equiv q \wedge p$
associative laws	$(p \vee q) \vee r \equiv p \vee (q \vee r)$	$(p \wedge q) \wedge r \equiv$
distributive laws	$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$	$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$
identity laws	$p \vee F \equiv p$	$p \wedge T \equiv p$
domination laws	$p \vee T \equiv T$	$p \wedge F \equiv F$

→ Laws of propositional logic 2

	Conjunction	Disjunction
De Morgan's laws	$\neg(p \vee q) \equiv \neg p \wedge \neg q$	$\neg(p \wedge q) \equiv \neg p \vee \neg q$
absorption laws	$p \vee (p \wedge q) \equiv p$	$p \wedge (p \vee q) \equiv p$
negation laws	$p \vee \neg p \equiv T$	$p \wedge \neg p \equiv F$
double negation laws	$\neg\neg p \equiv p$	

→ Equivalence proof

Let's examine the equivalence between $\neg(p \wedge (\neg p \vee q))$ & $(\neg p \vee \neg q)$:

Given proposition : $\neg(p \wedge (\neg p \vee q))$

De Morgan's law : $\neg p \vee \neg(\neg p \vee q)$

De Morgan's law : $\neg p \vee ((\neg\neg p) \wedge \neg q)$

double negation law : $\neg p \vee (p \wedge \neg q)$

distributive laws : $(\neg p \vee p) \wedge (\neg p \vee \neg q)$

complement laws : $T \wedge (\neg p \vee \neg q)$

identity laws : $\neg p \vee \neg q$

— LECTURE 4.101 —

- Propositional logic is helpful for studying propositions, but has some limitations:
  - It cannot express precisely the meaning of complex statements in mathematics
  - It only studies propositions, which are statements with known truth values
- Predicate logic overcomes these limitations and can be used to build more complex reasoning.

Ex) Given the statements:
- "All men are mortal"
- "Socrates is a man"
- Naturally, it follows that: "Socrates is mortal"

- Propositional logic can't express this reasoning, but predicate logic will enable us to formalise it.

Ex) Given the statement:
- "x squared is equal to 4."
- It's not a proposition, as its truth value is a function depending on x
- This can't be expressed with propositional logic, and predicate logic will enable us to formalise it.

— LECTURE 4.103 —

Outlines
- Predicates
- Predicates for multiple variables
- Logical operations

→ Def'n of a predicate - 1
- Predicates are a more general form of proposition
- Predicates behave as functions whose values are T or F depending on their variables
- Predicates become propositions when their variables are given actual values

→ Def'n of a predicate - 2

The statement "x squared is equal to 4" has two parts:
- The variable x, which is the subject of the statement
- The predicate "squared is equal to 4", which is the property that the subject of the statement can have
- This statement can be formalised as P(x) where P is the predicate "squared is equal to 4" and x is the variable
- P is said to be the propositional function
- Once a value is assigned to the variable x, the statement P(x) becomes a proposition and has a truth value

Ex)
- Let x be an integer
- Let P be the propositional function "squared is equal to 4"
- Let evaluate P(x) for specific values of the variable x:
  - P(2) is True
  - P(3) is False

→ Predicates with multiple variables

A predicate can depend on more than one variable

Ex) Let $P(x,y)$ denote "$x^2 > y$"
- $P(-2,3) \equiv (4>3)$ is True
- $P(2,4) \equiv (4>4)$ is False

Let $Q(x,y,z)$ denote "$x+y<z$";
- $Q(2,4,5) \equiv (6<5)$ is False
- $Q(1,3,7) \equiv (4<7)$ is True
- $Q(1,3,z) \equiv (4<z)$ is not a proposition

→ Logical operations

Logical operations from propositional logic carry over to predicate logic

Ex) If $P(x)$ denotes "$x^2<16$", then:
- $P(1) \vee P(-5) \equiv (1<16) \vee (25<16) \equiv T \vee F \equiv$ True
- $P(1) \wedge P(-5) \equiv (T \wedge F) \equiv$ False
- $P(3) \wedge P(y) \equiv$ It is not a proposition. It becomes a proposition when y is assigned a value

→ A predicate can depend on more than one variable!

— LECTURE 4.105 —

Outlines
- Def'n of quantifiers
- Universal quantifier
- Existential quantifier
- Uniqueness quantifier

→ Quantifiers
- Quantification expresses the extent to which a predicate is true over a range of elements
- They express the meaning of the words all and some
- The two most important quantifiers are:
  - Universal quantifier
  - existential quantifier

Ex) • "All men are mortal"
• "Some computers are not connected to the network"

→ Universal quantifier

defn: The universal quantification of a predicate $P(x)$ is the proposition:

- "$P(x)$ is true for all values of x in the universe of discourse"
- We use the notation: $\forall x P(x)$, which is read "for all x."

• If the universe of discourse is finite, say $\{n_1, n_2, \dots n_k\}$, then the universal quantifier is simply the conjunction of the propositions over all the elements:

$$\forall x P(x) \Leftrightarrow P(n_1) \wedge P(n_2) \wedge \dots \wedge P(n_k)$$

Ex) Let P, Q denote the following propositional functions of x:

- $P(x)$: "x must take a discrete mathematics course."
- $Q(x)$: "x is a Computer Science student"
- Where, the universe of discourse for both $P(x)$ and $Q(x)$ is all university students.

Let's express the following statements:

- "Every CS student must take a discrete mathematics course."
$$\forall x Q(x) \rightarrow P(x)$$
- "Everybody must take a discrete mathematics course or be a CS student"
$$\forall x (P(x) \vee Q(x))$$
- "Everybody must take a discrete mathematics course and be a CS student"
$$\forall x (P(x) \wedge Q(x))$$

Ex) S: "For every x and every y, $x+y > 10$."

- The statement S is: $\forall x \forall y P(x,y)$
- Also written as: $\forall x,y P(x,y)$

→ Existential quantifier

def'n: The existential quantification of a predicate $P(x)$ is the proposition:

- There exists a value x in the universe of discourse such that $P(x)$ is true.
- We use the notation: $\exists x\, P(x)$, which is read "there exists x".

If the universe of discourse is finite, say $\{n_1, n_2, \dots, n_k\}$, then the existential quantifier is simply the disjunction of the propositions over all the elements:

$$\exists x\, P(x) \Leftrightarrow P(n_1) \vee P(n_2) \vee \dots \vee P(n_k)$$

Ex) Let $P(x,y)$ denote the statement "$x+y=5$"

The expression $E: \exists x \exists y\, P(x,y)$ means:

- There exists a value x and a value y in the universe of discourse such that $x+y=5$ is true"

Ex) • Let a, b, c denote fixed real numbers.

- And S be the statement: "There exists a real solution to $ax^2+bx-c=0$"
- S can be expressed as $\exists x\, P(x)$, where:
  - $P(x)$ is $ax^2+bx-c=0$ and the universe of discourse for x is the set of real numbers.

Let's evaluate the truth value of S:

- When $b^2 >= 4ac$, S is true, as $P((-b \pm \sqrt{(b^2-4ac)})/2a) = 0$
- When $b^2 < 4ac$, S is false, as there is no real number x

→ Uniqueness quantifier

def'n; The uniqueness quantification of a predicate $P(x)$ is the proposition:

- "There exists a unique value x in the universe of discourse such that $P(x)$ is true."
- We use the notation: $\exists !\, x\, P(x)$, which is read "There exists a unique x."

Ex) Let $P(x)$ denote the statement "$x^2 = 4$"

- The expression $E: \exists! x\, P(x)$ means:
  - "There exists a unique value x in the universe of discourse such that $x^2 = 4$ is true."

For instance, if the universe of discourse is positive integers, E is True (as $x=2$ is the unique solution)

- If the universe of discourse is integers, E is False (as $x=2$ and $x=-2$ are both solutions).

— LECTURE 4.107 —

Outlines

- Nested quantifiers
- Binding variables
- Logical operations
- Order of operators
- Precedence of quantifiers

→ Nested quantifiers: To express statements with multiple variables we use nested quantifiers.

$\forall x \forall y\, P(x,y)$: $P(x,y)$ is true for every pair x, y.

$\exists x \exists y\, P(x,y)$: There is a pair x, y for which $P(x,y)$ is true.

$\forall x \exists y\, P(x,y)$: For every x there is a y for which $P(x,y)$ is true

$\exists x \forall y\, P(x,y)$: There is an x for which $P(x,y)$ is true for every y

→ Binding variables

- A variable is said to be bound if it is within the scope of a quantifier.
- A variable is free if it is not bound by a quantifier or particular values

Ex) · Let P be a propositional function
· And S the statement: $\exists x P(x,y)$
· We can say that:
  · x is bound
  · y is free

→ Logical operations, which were discussed in the topic on propositional logic ($\neg \wedge \vee \rightarrow \leftrightarrow$), can also be applied to quantified statements.

Ex) · If $P(x)$ denotes "$x > 3$" and $Q(x)$ denotes "x squared is even" then:
· $\exists x(P(x) \wedge Q(x)) \equiv T$ (ex. $x=4$)
· $\forall x (P(x) \rightarrow Q(x)) \equiv F$ (ex. $x=5$)

→ Order of operators
· When nested quantifiers are of the same type, the order does not matter, otherwise matter.

Ex) $\forall x \forall y P(x,y) \equiv \forall y \forall x P(x,y)$
$\exists x \exists y P(x,y) \equiv \exists y \exists x P(x,y)$
$\forall x \exists y P(x,y)$ is different from $\exists y \forall x P(x,y)$

→ Precedence of quantifiers
· The quantifiers $\forall$ and $\exists$ have a higher precedence priority than all logical operators.

Ex) · Let $P(x)$ and $Q(x)$ denote two propositional functions.
· $\forall x P(x) \vee Q(x)$ is the disjunction of $\forall x P(x)$ and $Q(x)$ rather than $\forall x (P(x) \vee Q(x))$
· $\forall x P(x) \rightarrow Q(x)$ is the implication of $\forall x P(x)$ and $Q(x)$ rather than $\forall x (P(x) \rightarrow Q(x))$

— LECTURE 4.201 —

Outlines
- The intuition of De Morgan's laws
- De Morgan's laws
- Negating nested quantifiers

→ The intuition of De Morgan's laws

Usually, we need to consider the negation of a quantified expression.

Ex) • S: "All the university's computers are connected to the network"
- P: "There is at least one computer in the university operating on Linux."

Intuitively
- The negation of S can be verified if there is at least one computer not connected to the network
- The negation of P can be verified if all university computers are not operating on linux

. De Morgan's laws formalise these intuitions.

→ De Morgan's Laws for negating quantifiers:

$$\neg \forall x P(x) \equiv \exists x \neg P(x)$$

$$\neg \exists x P(x) \equiv \forall x \neg P(x)$$

Ex) Let S: "Every student of Computer Science has taken a course in Neural Networks"
- S can be expressed as: $\forall x P(x)$
- U = {students in CS}
- P(x): "x has taken a course in Neural Networks"

The Negation of S:
- "It is not the case that every student of CS has taken a course in Neural Networks."

$$\neg(\forall x P(x)) \equiv \exists x \neg P(x)$$

- This implies that: "There is at least one student who has not taken a course in Neural Networks"

Ex) Let R denote: "There is a student in Computer Science who didn't take a course in Machine Learning"

- R can be expressed as: $\exists x\, Q(x)$
- $U = \{\text{students in CS}\}$
- $Q(x)$: "x didn't take a course in Machine Learning"

The negation of S:

- "It is not the case that there is a student in CS who didn't take a course in ML."

$$\neg(\exists x\, Q(x)) \equiv \forall x \neg Q(x)$$

- This implies that: "Every student in CS has taken a ML"

→ Negating nested quantifiers

In the case of nested quantifiers, we apply De Morgan's Laws successively from left to right.

Ex) Let $P(x,y,z)$ denote a propositional function of variables x, y and z

$$\equiv \exists x \neg \exists y\, \forall z\, P(x,y,z)$$
$$\equiv \exists x\, \forall y \neg \forall z\, P(x,y,z)$$
$$\equiv \exists x\, \forall x\, \exists z \neg P(x,y,z)$$

$\neg \forall x\, \exists y\, \forall z\, P(x,y,z)$ is built by moving the negation to the right through all the quantifiers and replacing each $\forall$ with $\exists$, and vice versa

— LECTURE 4.2.03 —

Outlines

- Valid arguments
- Rules of inference:
  - Modus ponens, Modus tollens, Conjunction, Simplification, Addition, Hypothetical syllogism, Disjunctive syllogism, Resolution
- Building valid arguments
- Fallacies

→ Valid argument

- An argument in propositional logic is a sequence of propositions
- The final proposition is called the conclusion and the other propositions in the argument are called premises (or hypotheses)
- An argument is valid if the truth of all its premises implies the truth of the conclusion.

Ex1 Let's consider this argument:

- "If you have access the internet, you can order a book on Machine Learning"
- "You have access to the internet"

∴ Therefore: "You can order a book on Machine Learning."

This argument is valid because whenever all its premises are true, the conclusion must also be true.

Ex1 Let's consider this argument:

- "If you have access to internet, you can order a book on Machine learning."
- "You can order a book on ML."

∴ Therefore: "You have access to internet"

This argument is not valid, premises are true, conclusion is false.

→ Rules of inference

- Rules of inference can be seen as building blocks in constructing incrementally complex valid arguments.
- We can use a truth table to determine whether an argument is True or False, but this can be laborious, especially with multiple variables
- Instead, rules of inference provide a much simpler way of proving the validity of an argument.
- Every rule of inference can be proved using a tautology.

→ Modus ponens

- Tautology: $(p \wedge (p \rightarrow q)) \rightarrow q$
- The rule of inference:

$$\begin{array}{l} p \rightarrow q \\ p \\ \hline \therefore q \end{array}$$

Ex) · p: "It is snowing"

· q: "I will study Discrete Mathematics"

· "If it is snowing, I will study Discrete Mathematics"

· "It is snowing"

· Therefore: "I will study discrete Mathematics"

→ Modus tollens

- Tautology: $(\neg q \wedge (p \rightarrow q)) \rightarrow \neg p$
- The rule of inference:

$$\begin{array}{l} \neg q \\ p \rightarrow q \\ \hline \therefore \neg p \end{array}$$

Ex) · p: "It is snowing"

· q: "I will study discrete mathematics"

· "If it is snowing, I will study discrete mathematics"

· "I will not study discrete mathematics."

· Therefore: "It is not snowing"

→ Conjunction

- Tautology: $((p) \wedge (q)) \rightarrow (p \wedge q)$
- The rule of inference:

$$\begin{array}{l} p \\ q \\ \hline \therefore p \wedge q \end{array}$$

Ex)

· p: "I will study programming"

· q: "I will study discrete mathematics"

$(p \wedge q)$

∴ Therefore: "I will study programming and discrete mathematics"

→ Simplification

- Tautology: $(p \wedge q) \rightarrow p$
- The rule of inference: $\dfrac{p \wedge q}{\therefore p}$

Ex) • p: "I will study programming"

• q: "I will study discrete mathematics"

• I will study programming and discrete mathematics

• Therefore: "I will study discrete mathematics"

→ Addition

- Tautology: $p \rightarrow (p \vee q)$
- The rule of inference: $\dfrac{p}{\therefore p \vee q}$

Ex) • p: "I wish visit Paris"

• q: "I will study discrete mathematics"

• "I will visit Paris."

• Therefore: "I will visit Paris or I will study disc. math."

→ Hypothetical syllogism

- Tautology: $((p \rightarrow q) \wedge (q \rightarrow r)) \rightarrow (p \rightarrow r)$
- The rule of inference: $\dfrac{p \rightarrow q \quad q \rightarrow r}{\therefore p \rightarrow r}$

Ex) • p: "It is snowing."

• q: "I will study discrete mathematics"

• r: "I will pass the quizzes"

• "If it is snowing, I will study discrete math"

• "If I study disc. math, I will pass the quizzes"

Therefore: "If it is snowing, I will pass the quizzes"

→ Disjunctive syllogism

- Tautology: $((p \vee q) \wedge \neg p) \rightarrow q$
- The rule of inference:

$$\begin{array}{l} p \vee q \\ \neg p \\ \hline \therefore q \end{array}$$

Ex). p: "I will study art"
q: "I will study discrete mathematics"
"I will not study disc. math."
Therefore: "I will study art"

→ Resolution

- Tautology: $((p \vee q) \wedge (\neg p \vee r)) \rightarrow (q \vee r)$
- The rule of inference:

$$\begin{array}{l} p \vee q \\ \neg p \vee r \\ \hline \therefore q \vee r \end{array}$$

Ex) • p: "It is raining"
• q: "It is snowing"
• r: "It is cold"
• "It is raining or it is snowing"
• "It is not raining or it is cold"
Therefore: "It is snowing or it is cold"

→ Building valid arguments

To build a valid argument we need to follow the steps below:

- If initially written in English, transform the statement into an argument from by choosing a variable for each simple proposition.
- Start with the hypothesis of the argument
- Build a sequence of steps in which each step follows from the previous step by applying:
  - rules of inference
  - laws of logic

Ex) Let's build a valid argument from the following premises:

$\neg p$: "It is not cold tonight"

$q \to p$: "We will go to the theatre only if it is cold"

$\neg q \to r$: "If we do not go to the theatre, we will watch a movie at home"

$r \to s$: "If we watch a movie at home, we will need to make popcorn"

Propositional variables:

p: "It is cold tonight"

q: "We will go to the theatre"

r: "We will watch a movie at home"

s: "We will need to make popcorn"

→ Fallacies

- A fallacy is the use of incorrect argument when reasoning
- Formal fallacies can be expressed in propositional logic and proved to be incorrect
- Some of the widely used formal fallacies are:
  - affirming the consequent
  - a conclusion that denies premises
  - contradictory premises
  - denying the antecedent
  - existential fallacy
  - exculusive premises

Ex) Let's consider the following argument:

- If you have internet access, you can order this book.
- You can order this book

Therefore, you have Internet access

- This argument can be formalised as: if $p \to q$ and q, then p
- Where p: "You have Internet access"
  q: "You can order this book"
- The proposition $((p \to q) \wedge q) \to p$ is not a tautology, because it is false when p is false and q is true.

— LECTURE 6.2.05 —

Outlines:

- Rules of inference with quantifiers
  - universal instantiation, universal generalisation, existential instantiation, existential generalisation, universal modus ponens, univ. modus tollens
- Expressing complex statements

→ Universal instantiation (UI)

The rule of inference: $\frac{\forall x\, P(x)}{\therefore P(c)}$

Ex) • All computer science students study discrete mathematics

∴ Therefore, John, who is a computer science student, studies discrete mathematics.

→ Universal generalization (UG)

The rule of inference: $\frac{P(c) \text{ for an arbitrary element of the domain}}{\forall x\, P(x)}$

Ex) • DS = {all data science students}
• Let c be an arbitrary element in DS
• c studies machine learning

∴ Therefore, $\forall x \in DS$, x studies machine learning

→ Existential instantiation (EI)

The rule of inference: $\frac{\exists x\, P(x)}{\therefore P(c) \text{ for some element of the domain}}$

Ex) • DS = {all data science students}
• There exists a student of data science who uses Python Pandas Li

∴ There is a student, c, who is using Python Pandas Library

→ Existential generalization (EG)

The rule of inference: $\frac{P(c) \text{ for some element of the domain}}{\therefore \exists x\, P(x)}$

Ex) • DS = {all data science students}
• John, a student of data science, got an A in the ML course.

∴ Therefore, there exists someone in DS who got an A in ML.

→ Universal modus ponens

The rule of inference: $\forall x\, P(x) \rightarrow Q(x)$

$P(a)$ for some element of the domain

$\therefore Q(a)$

Ex1. DS = {all computer science students}

- Every computer science student studying data science will study machine learning
- John is studying data science

∴ Therefore, John will study machine learning

→ Universal modus tollens

The rule of inference: $\forall x\, P(x) \rightarrow Q(x)$

$\neg Q(a)$ for some element of the domain

$\therefore \neg P(a)$

Ex1. • CS = {all computer science student}

- Every computer science student studying data science will study machine learning.
- John is not studying machine learning

∴ Therefore, John is not studying data science

→ Expressing complex statements

Given a statement in natural language, we can formalise it using the following steps as appropriate:

1. Determine the universe of discourse of variables
2. Reformulate the statement by making "for all" and "there exists" explicit.
3. Reformulate the statement by introducing variables and defining predicates
4. Reformulate the statement by introducing quantifiers and logical operations

Ex) Express the statement S: "there exists a real number between any two not equal real numbers"

- The universe of discourse is: real numbers.
- Introduce variables and predicates:
  - "For all real numbers x and y, there exists z between x and y"
- Introduce quantifiers and logical operations
  - $\forall x \forall y$ if $x<y$ then $\exists z$ where $x<z<y$

Ex) Express the statement S: "every student has taken a course in Machine Learning"

The expression will depend on the choice of the universe of discourse

Case 1: U = {all students}

- Let $M(x)$ be: "x has taken a course in ML"
- S can be expressed as: $\forall x M(x)$

Case 2: U = {all people}

- Let $S(x)$ be: "x is a student" and $M(x)$ the same as in C1
- S can be expressed as $\forall x (S(x) \rightarrow M(x))$

Note: $\forall x (S(x) \wedge M(x))$ is not correct.

Ex) Express the statement S: "some student has taken a course in Machine Learning"

The expression will depend on the choice of the universe of discourse

Case 1: U = {all students}

- Let $M(x)$ be: "x has taken a course in ML"
- S can be expressed as: $\exists x M(x)$

Case 2: U = {all people}

- Let $S(x)$ be: "x is a student" and $M(x)$ the same as in case 1
- S can be expressed as $\exists x (S(x) \wedge M(x))$

Note: $\exists x (S(x) \rightarrow M(x))$ is not correct in this case.

— LECTURE 5.101 —

Outlines
- History of Boolean algebra
- Application of Boolean algebra
- Definition
- Operations of Boolean algebra

→ History of boolean algebra
- 384-322 BC: Aristotle established the foundations of logic
- 1854: George Boole published An investigation of the laws of thought"
- 1904: H.E. Huntington wrote sets of independent postulates for the algebra of logic
- 1938: Claude Shannon wrote a thesis entitled A symbolic analysis of relay switching.

→ Application of Boolean algebra
- Computer circuit analysis.
- Boolean algebra is the basic block for designing transistors, which are the basic elements in building processors
- For instance, consider the following IoT system:
  - When the system is activated, a fire sprinkler should spray water if high heat is detected.
  - This system can be designed using a Boolean algebra equa
    - w=h AND a, where:
      - h represent 'high heat is detected'
      - a represent 'the system is activated'
      - W represents 'spraying water'

→ Two-valued Boolean algebra

The most common well-known form of boolean algebra is a two-valued system, where:
- Variables take values on the set {0,1}
- the operators (+) and (.) correspond to {OR} and {AND} respectively

// It could be used to design circuits.

→ Operations of Boolean algebra

Boolean algebra is based on three fundamental operations:

AND
- logical product, intersection or conjunction
- represented as $x.y$, $x \cap y$ or $x \wedge y$

OR
- logical sum, union or disjunction
- represented as $x+y$, $x \cup y$, or $x \vee y$

NOT
- logical complement or negation
- represented as x', $\bar{x}$ or $\neg x$

When parentheses are not used, these operators have the following order of precedence

NOT > AND > OR

— LECTURE 5.103 —

Outlines
- Huntington's postulates
- basic theorems of Boolean algebra: Idempotent, associative,
- De Morgan's theorems
- Proof of Distributivity of + over .
- Principle of duality
- Ways of proving theorems

→ Huntington's postulates

It define 6 axioms that must be satisfied by any Boolean Algebra:
- closure with respect to the operators:
  - any result of logical operation belongs to the set {0,1}
- identity elements with respect to the operators:
  - $x+0 = x$, $x.1 = x$
- commutativity with respect to the operators:
  - $x+y = y+x$, $x.y = y.x$
- distributivity:
  - $x(y+z) = (x.y)+(x.z)$, $x+(y.z) = (x+y).(x+z)$

- complements exist for all the elements:
  - $x + x' = 1, \; x \cdot x' = 0$
- Distinct elements:
  - $0 \neq 1$

→ Basic theorems

Using the 6 axioms of Boolean algebra, we can establish other useful theorems for analysing and designing circuits.

- theorem 1: idempotent laws

  $x + x = x, \; x \cdot x = x$

- theorem 2: tautology and contradiction

  $x + 1 = 1, \; x \cdot 0 = 0$

- theorem 3: involution

  $(x')' = x$

- theorem 4: associative laws

  $(x+y) + z = x + (y+z), \; (x \cdot y) \cdot z = x \cdot (y \cdot z)$

- theorem 5: absorption laws

  $x + (x \cdot y) = x, \; x \cdot (x+y) = x$

- theorem 6: uniqueness of complement

  if $y + x = 1$ and $y \cdot x = 0$, then $x = y'$

- theorem 7: inversion law

  $0' = 1, \; 1' = 0$

→ De Morgan's theorems

- theorem 1: the complement of a product of variables is equal to the sum of the complements of the variables:

  $\overline{x \cdot y} = \overline{x} + \overline{y}$

- theorem 2: the complement of a sum of variables is equal to the product of the complements of the variables:

  $\overline{x + y} = \overline{x} \cdot \overline{y}$

→ Principle of duality

Starting with a Boolean relation, we can build another equivalent Boolean relation by:

- Changing each OR(+) sign to an AND(·) sign
- changing each AND(·) sign to an OR(+) sign
- changing each 0 to 1 and each 1 to 0.

Ex) Since $A + BC = (A+B)(A+C)$ (by distributive law), we can build another relation using the duality principle:

$$A(B+C) = AB + AC$$

Ex) Let's consider the Boolean equations:

- e1: $(a.1).(0+\bar{a}) = 0$
- e2: $a + \bar{a}.b = a+b$

The dual equations of e1 and e2 are obtained by interchanging + and ·, and interchanging 0 and 1, as follows:

- dual of e1: $(a+0) + (1.\bar{a}) = 1$
- dual of e2: $a.(\bar{a}+b) = a.b$

→ Ways of proving theorems

In general, there are 4 ways to prove the equivalence of Boolean relations:

- perfect induction: by showing the two expressions have identical truth tables. This is can be tedius if there are more than 3 or 4 variables
- axiomatic proof: by applying Huntington's postulates or theorems (that have already been proven) to the expressions, until identical expressions are found
- duality principle: every theorem in Boolean algebra remains valid if we interchange all AND's or OR's and interchange all 0's and 1's
- contradiction: by assuming that the hypothesis is false and then proving that the conclusion is false

Ex) Let's consider proving the absorption theorem

- The absorption theorem can be proved using perfect induction, by writing a truth table.
- It can also be proved directly as follows

$$x + (x.y) = (x.1) + (x.y) \text{ by } x.1 = x$$
$$= x.(1+y) \text{ by distributivity}$$
$$= x.(y+1) \text{ by commutativity}$$
$$= x.1 \text{ by } x+1 = x$$
$$= x \text{ by } x.1 = x$$

- From $x + (x.y) = x$, if we apply the duality principle, we can deduce: $x.(x+y) = x$

— LECTURE 5:10 5 —

Outlines:
- Def'n of boolean function
- Algebraic forms
- Standardised forms of a function
- Building a sum-of-products form
- Useful function

→ Definition
- A function defines a mapping from one or multiple Boolean input values to a Boolean output value
- For n Boolean input values, there are 2^n possible combinations

Ex) a 3 input function f can be completely defined with an 8-row truth table

→ Algebraic forms
- There is only one way to represent a Boolean function in a truth table
- In algebraic form, a function can be expressed in a variety of ways

Ex) $f(x) = x + x'.y$ and $f(x) = x + y$ are both algebraic representations of the same truth table

→ Standardised forms of a function

The two most common standardised forms are the sum-of-products form and the product-of-sums form

- The sum-of-products form:
  such as: $f(x,y,z) = xy + xz + yz$
- The product-of-sums form:
  such as: $f(x,y,z) = (x+y)(x+z)(y+z)$
- The sum-of-products form is easier to use and to simplify.

→ Build a sum-of-products form

It is easy to build a sum-of-products form using the function's truth table:

1. we focus on the values of the variables that make the function 1
2. If an input equals 1, it appears uncomplemented in the expression
3. if an input equals 0, it appears complemented in the expression
4. The function f is then expressed as the sum of products of all terms for which f=1

Ex: • Let's consider the function f represented by following truth table

• can be expressed as

$f(x,y) = x'y + xy' + xy$

x	y	f(x,y)
0	0	0
0	1	1
1	0	1
1	1	1

→ Useful functions

The 'exculusive-or' function: $x \oplus y$:

- defined as "true if either x or y is true, but not both"
- represented by the following truth table
- can be expressed as

$x \oplus y = x'y + xy'$

x	y	$x \oplus y$
0	0	0
0	1	1
1	0	1
1	1	0

The 'implies' function: $x \rightarrow y$:

- defined as "if x then y"
- represented by the following truth table
- can be expressed as:

$x \rightarrow y = x' + y$

x	y	$x \rightarrow y$
0	0	1
0	1	1
1	0	0
1	1	1

— LECTURE 5.201 —

Outlines

- Def'n of a gate
- Basic gates
- Other gates
- Multiple input gates
- Representing De Morgan's Laws

→ Def'n of a gate: A logic gate is defined as the basic element of circuits implementing a Boolean operation

. The most basic logic circuits are OR gates, AND gates and inverters, or NOT gates. All boolean functions can be written in terms of these three logic operations

- AND gate

HIGH → 1
LOW → 0

$f = x \cdot y$ or $f = xy$

- OR gate

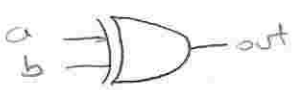
HIGH → 1
LOW → 0

$f = x + y$

- INVERTER (NOT) gate

$f = \bar{x}$

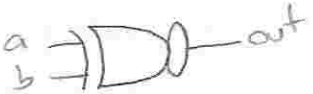
- XOR gate: true only when the values of the inputs are different



- NAND gate

- NOR gate

XNOR gate: equivalent to XOR



→ Multiple input gates: AND, OR, XOR and XNOR gates are all commutative and associative

- They can be extended to more than two inputs

XNAND and XNOR are both commutative but not associative. We must use the correct parantheses

→ Representing De Morgan's laws

Theorem 1: $\overline{x \cdot y} = \overline{x} + \overline{y}$

Theorem 2: $\overline{x + y} = \overline{x} \cdot \overline{y}$

— LECTURE 5.203 —

Outlines

- Def'n of a circuit
- Building a circuit from a function
- Writing boolean expressions from a circuit
- Building a circuit to model a problem

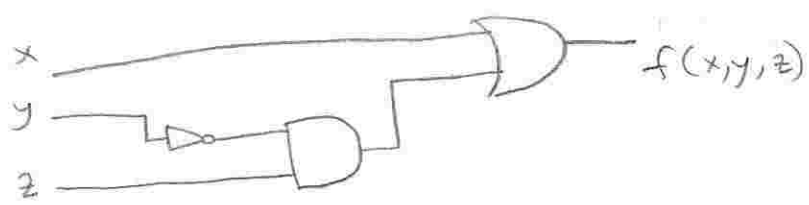
→ Def'n of a circuit

- Combinational circuits or logic networks are a combination of logic gates designed to model boolean functions
- A combinational circuit is a circuit that implements a boolean function
- The logic values assigned to the output signals is a boolean function of the current configuration of input signals.

→ Building a circuit from a function

- Given a boolean function, we can implement a logic circuit representing all the states of the function
- Intuitively, we want to minimise the number of gates used in order to minimise the cost of the circuit
- A boolean function can be implemented in different ways using circuits.

Ex) $f(x, y, z) = x + y'z$



→ Writing boolean expressions from a circuit

Given a logic network, we can work out its corresponding boolean function as follows:

1: label all gate outputs that are a function of the input variables

2: express the boolean functions for each gate in the first level

3: repeat the process until all the outputs of the circuit are written as boolean expressions.

→ Building a circuit to model a problem

Combinational circuits are useful for designing systems to solve specific problems, such as addition, multiplication, decoders and multiplexers

The step for building a circuit that solves a specific problem are:

1. labelling the inputs and outputs using variables
2. modelling the problem as a boolean expression
3. replacing each operation by the equivalent logic gate

→ Building an adder circuit

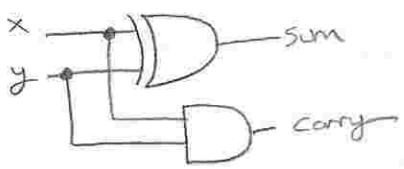
Let's consider building an adder for two one-digit binary bits x and y

From the truth table of this boolean function, we know that:

- sum = $xy' + x'y = x \oplus y$
- carry = xy

x	y	sum	carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	1	1

Which can be designed as a half adder



This half adder has limitations:

- There is no provision for carry input
- This circuit is not useful for multi-bit additions

→ Building a full adder circuit: to overcome, 3 inputs i.o. 2 inputs

$$\text{sum} = x \oplus y \oplus \text{carry in}$$

$$\text{carry out} = xy + \text{carry in} \cdot (x \oplus y)$$

LECTURE 5.205

Outlines
- Benefits of simplification
- Algebraic simplification
- Karnaugh maps

→ Benefits of simplification

Every function can be written as a sum-of-product, but this formulation is not necessarily optimal in terms of the number of gates and the depth of the circuit

This is why circuits need to be simplified

Simplification of circuits, also called minimisation or optimisation, is beneficial in circuit design, as it:
- reduces the global cost of circuits, by reducing the number of logic gates used
- might reduce the time computation cost of circuits
- allows more circuits to be fitted on the same chip

→ Algebraic simplification; is based on the use of Boolean algebra theorems to represent and simplify the behaviour of Boolean functions.

To produce a sum-of-product expression, we usually need to use one or all of the following theorems:
- De Morgan's laws and involution
- distributive laws
- commutative, idempotent and complement laws
- absorption law

Ex) Let's consider the following Boolean expression

$$E = ((xy)'z)'((x'+z)(y'+z'))'$$

Using De Morgan's laws and involution:

$$E = ((xy)'' z')((x'+z)' + (y'+z')')$$
$$= (xy + z')((x'' \cdot z') + y'' \cdot z'')$$
$$= (xy + z')(xz' + yz)$$

Using distributive laws:

$$E = xyxz' + xyyz + z'xz' + z'yz$$

Using commutative, idempotent and complement laws:

$$E = xyz' + xyz + xz' + 0$$

Using absorption law

$$E = xyz' + xyz + xz' + 0$$

→ Karnaugh maps (K-Map); is a graphical representation of Boolean functions and is different from a truth table. It can be used for expressions with 2, 3, 4 or 5 variables.

- A K-Map is shown in an array of cells and cells differing by only one variable are adjacent.
- The number of cells in a K-Map is the total number of possible input variable combinations, which equals 2^k

Ex)

		y'	y
		0	1
x'	0	0	0
x	1	1	1

x	y	f
0	0	0
0	1	0
1	0	1
1	1	1

- LECTURE 6.101 -

Outlines

- Introduction to proofs: A proof is valid argument that is used to prove the truth of a statement
- To build a proof we need to use all the blocks we have introduced previously:
  - Variables & predicates
  - Quantifiers
  - Laws of logic
  - Rules of inference

Terminology: We need to define some terms, even if choosing the appropriate term is intrinsically subjective:

- A <u>theorem</u> is a formal statement that can be shown to be true
- An <u>axiom</u> is a statement we assume to be true to serve as a premise for further arguments
- A <u>lemma</u> is proven statement used as a step to a larger result rather than as a statement of interest by itself.
- A corollary is a theorem that can be established by a short proof from a theorem

→ Formalising a theorem

- Let's consider the statement S: "There exists a real number between any two not equal real numbers."
- S can be formalised as: $\forall x, y \in R$ if $x < y$ then $\exists z \in R$ where $x < z < y$
- S is a theorem

→ Direct proof

- A direct proof is based on showing that a conditional statement: $p \rightarrow q$ is true
- We start by assuming that p is true and then use: axioms, definitions and theorems, together with rules of inference, to show that q must also be true.

Ex) Let's give a proof of the theorem:

"There exists a real number between any two not equal real numbers"

Proof:

- Let x, y be arbitrary elements in R
- Let's suppose $x < y$
- Let $z = (x+y)/2$
- $z \in R$, satisfying $x < z < y$

∴ Therefore, using the universal generalisation rule, we can conclude that: $\forall x, y \in R$ if $x < y$ then $\exists z \in R$ where $x < z < y$

→ Proof by contrapositive

- A proof by contrapositive is based on the fact that proving the conditional statement $p \rightarrow q$ is equivalent to proving its contrapositive: $\neg q \rightarrow \neg p$
- We start by assuming that $\neg q$ is true and then use: axioms, definitions, and theorems, together with rules of inference, to show that $\neg p$ must also be true.

Ex) Let's give a proof of the theorem:

"If n^2 is even then n is even."

Proof:

- Direct proof:
  - Let $n \in Z$. If n^2 is even then $\exists k \in Z, n^2 = 2k$
  - Then $\exists k \in Z, n = \pm\sqrt{2k}$. From this equation it doesn't seem intuitive to prove that n is even

- Proof by contraposition:
  - Let's suppose n is odd
  - Then $\exists k \in \mathbb{Z}$, $n = 2k+1$
  - Then $\exists k \in \mathbb{Z}$, $n^2 = (2k+1)^2 = 2(2k^2+2k)+1$
  - Then n^2 is also odd
  - We have succeeded in proving the contrapositive: if n is odd then n^2 is odd

- Proof by contradiction
  - A proof by contradiction is based on assuming that the statement we want to prove is false, and then showing that this assumption leads to a false proposition
  - We start by assuming that $\neg p$ is true and then use: axioms, definitions and theorems, together with rules of inference, to show that $\neg p$ is false. We can then conclude that it was wrong to assume that p is false, so it must be true.

Ex) Let's give a direct proof of the theorem:
"There are infinitely many prime numbers."

Proof:
- Let's suppose there are only finitely many prime numbers
- Let's list them as $p_1, p_2, p_3, \dots, p_n$ where $p_1 = 2$, $p_2 = 3$, $p_3 = 5$ and so on
- Let's consider the number $c = p_1 p_2 p_3 \dots p_n + 1$, the product of all the prime numbers, plus 1
- Then, as c is a natural number, it has at least one prime divisor
- Then $\exists k \in \{1 \dots n\}$, where $p_k \mid c$
- Then $\exists k \in \{1 \dots n\}$, $\exists d \in \mathbb{N}$ where $dp_k = c = p_1 p_2 p_3 \dots p_n + 1$
- Then $\exists k \in \{1 \dots n\}$, $\exists d \in \mathbb{N}$ where $d = p_1 p_2 \dots p_{k-1} p_{k+1} \dots p_n + \frac{1}{p_k}$
- Then $\frac{1}{p_k}$, in the expression above, is an integer, which is a contradiction

— LECTURE 6.103 —

Outlines

- Def'n of induction
- The intuition behind induction
- Structure of induction
- Some uses of induction

→ Def'n : Mathematical induction can be used to assert that a propositional function $P(n)$ is true for all positive integers n

- The rule of inference:

$$\frac{P(1) \text{ is true} \qquad \forall k\ (P(k) \rightarrow P(k+1))}{\therefore\ \forall n\ P(n)}$$

→ The intuition behind induction

- Let $P(n)$ be the propositional function verifying:
  - $P(1)$ is true
  - $\forall k\ (P(k) \rightarrow P(k+1))$
- Intuitively:
  - P is true for 1
  - Since P is true for 1, it's true for 2
  - Since P is true for 2, it's true for 3
  - And so on...
  - Since P is true for $n-1$, it's true for n ...
- In other words:
  - The base case shows that the property initially holds true
  - The inductive step shows how each iteration influences the next one

→ Structure of induction

In order to prove that a propositional function $P(n)$ is true for all, we need to verify two steps:

1. BASIS STEP: where we show that $P(1)$ is true
2. INDUCTIVE STEP: where we show that for $\forall k \in \mathbb{N}$:
   if $P(k)$ is true, called inductive hypothesis,
   then $P(k+1)$ is true

→ Some uses of induction

Mathematical induction can be used to prove $P(n)$ is true for all integers greater than a particular integer, where $P(n)$ is a propositional function. That might cover multiple cases such as:

- Proving formulas
- Proving inequalities
- Proving divisibility
- Proving properties of subsets and their cardinality

— LECTURE 6.1.06 —

Outlines

- Proving formulas
- Proving inequalities
- Proving divisibility
- Incorrect induction

→ Proving formulas

- Let's start by proving a simple formula formalised as the propositional function, $P(n): 1+2+3+\dots+n = n(n+1)/2$
- In order to prove that a propositional function $P(n)$ is true for all, we need to verify two steps:

1. BASIS STEP: where we show that $P(1)$ is true
2. INDUCTIVE STEP: where we show that for $\forall k \in \mathbb{N}$:
   if $P(k)$ is true, called inductive hypothesis,
   then $P(k+1)$ is true.

Ex) 1. BASIS STEP: The basis step, $P(1)$ reduces to $1 = 1(1+1)/2$

2. INDUCTIVE STEP:
   - Let $\forall k \in \mathbb{N}$
   - If the inductive hypothesis $P(k)$ is true:
     - we have $1+2+3+\dots+k+(k+1)$
     - then, $1+2+3+\dots+k+(k+1)$
       $= k(k+1)/2 + (k+1)$
       $= (k(k+1) + 2(k+1))/2$
       $= (k+1)((k+1)+1)/2$
     - which verifies, $P(k+1)$

→ Proving inequalities

- We may also use mathematical induction to prove an inequality that holds for all positive integers greater than a particular positive integer
- Let's consider proving the propositional function $P(n): 3^n < n!$ if n is an integer greater than or equal to 7.

Ex) 1. BS: The basis step, $P(7)$ reduces to $3^7 < 7!$ bec. $2187 < 5040$

2. IS:
   - Let $k \in \mathbb{N}$ and $k \geq 7$
   - If the inductive hypothesis $P(k)$ is true:
     then, $3^{k+1} = 3 * 3^k < (k+1) * k! = (k+1)!$ which verifies $P(k+1)$ is true

→ Proving divisibility

- We may also use mathematical induction to prove a divisibility that holds for all positive integers greater than a particular positive integer
- Let's consider proving the propositional function
  $P(n): \forall n \in \mathbb{N}$ $6^n + 4$ is divisible by 4

Ex) 1. BS: The basis step, $P(0)$ reduces to $6^0 + 4$ is divisible by 5

2. IS:
   - Let $k \in \mathbb{N}$
   - If the inductive hypothesis $P(k)$ is true:
     - then, $6^k + 4 = 5p$, where $p \in \mathbb{N}$
     - then, $6^{k+1} + 4 = 6 * (5p - 4) + 4$
       $= 30p - 20$
       $= 5(6p - 4)$ which is divisible by 5 and verifies $P(k+1)$ is true

→ Incorrect induction

Let's consider the statement of the following incorrect induction:

$$P(n): \forall n \in \mathbb{N} \sum_{i=0}^{n-1} 2^i = 2^n$$

Proof:
- Let $k \in \mathbb{N}$. Let's suppose the inductive hypothesis $P(k)$ is true, which means: $\sum_{i=0}^{k-1} 2^i = 2^k$
- Now let's examine $P(k+1)$
- $\sum_{i=0}^{k} 2^i = \sum_{i=0}^{k-1} 2^i + 2^k = 2^k + 2^k = 2^{k+1}$
- This means that $P(k+1)$ is also true and verifies the induction step

- Even though we have been able to prove the induction step, let's prove that the statement: $\forall n \in \mathbb{N} \quad \sum_{i=0}^{n-1} 2^i$ is FALSE
  - For example $2^0 + 2^1 = 3$ which is different from 2^2
- Our reasoning seemed correct because we haven't verified the base case and have made false assumptions
- In other words, and as we saw in propositional logic, false assumptions imply false conclusions
- To avoid this situations we need to make sure both the base case and the inductive step are verified

---

- LECTURE 6.1.08 -

Outlines
- Strong induction
- Well-ordering property
- Equivalence of the three concepts

→ Strong induction
- Sometimes, it is easier to prove statements using a different form of mathematical induction, called strong induction
- Strong induction can be formalised using the following rule of inference:

$P(1)$ is true
$\forall k \in \mathbb{N} \quad P(1), P(2) \dots P(k) \rightarrow P(k+1)$
$\therefore \forall n \in \mathbb{N}, P(n)$

Ex) Let's start by proving a simple statement, expressed as the propositional function, $P(n)$: $\forall n \in \mathbb{N}$ and $n \geq 2$, n is divisible by a prime number
- To prove it, we need to verify two steps:

1. BS: The basis step, $P(2)$ reduces to 2, which is divisible by a prime number because 2 is a prime number and divides itself.
2. IS:
   - Let $k \in \mathbb{N}$, greater than 2
   - If the inductive hypothesis is $P(k)$ is true:
     - Let's also assume $P(2) \dots P(k+1)$ is true. Then, $\forall m \in \mathbb{N}$ and $2 \leq m \leq k+1$: $\exists p$ is a prime number dividing m
   - We have two cases:
     - $k+2$ is a prime number, in which case it is trivially divisible by itself

- $k+2$ is not a prime number, in which case $\exists m$ dividing $k+2$
- as $2 \leq m \leq k+1$, $\exists p$ is a prime number dividing m. p also divides $k+2$
- which verifies $P(k+2)$ is true and proves the strong induction

→ Well-ordering property

The well-ordering property is an axiom about $\mathbb{N}$ that we assume to be true. The axioms about $\mathbb{N}$ are the following:

1. The number 1 is a positive integer
2. If $n \in \mathbb{N}$, then $n+1$, the successor of n, is also a positive integer
3. Every positive integer other than 1 is the successor of a positive integer
4. The well-ordering property: every nonempty subset of the set of positive integers has at least one element

The well-ordering property can be used as a tool in building proofs

Ex) Let's reconsider the earlier statement $P(n)$: $\forall n \in \mathbb{N}$ and $n \geq 2$, n is divisible by a prime number.

- Proof:
  - Let S be the set of positive integers greater than 1 with no prime divisor
  - Suppose S is nonempty. Let n be its smallest element
  - n cannot be prime, since n divides itself and if n were prime, it would be its own prime divisor
  - So n is composite; it must have a divisor d with $1 < d < n$.
  - Then, d must have a prime divisor (by the minimality of n), let's call it p
  - Then $p \mid d$ and $d \mid n$, so $p \mid n$, which is a contradiction
  - Therefore S is empty, which verifies $P(n)$

→ Equivalence of the three concepts

We can prove the following statements:

- mathematical induction → the well-ordering property
- the well-ordering property → strong induction
- strong induction → mathematical induction

- That is, the principal of mathematical induction, strong induction and well-ordering are all equivalent
- In other words, the validity of each of these three proof techniques implies the validity of the other two techniques

---

— LECTURE 6.201 —

Outlines
- Def'n of recursion
- Recursively defined functions
- Recursively defined sets
- Recursive algorithms

→ Def'n of recursion
- Sometimes it is difficult to define a mathematical object (e.g. a function, sequence or set) explicitly; it is easier to define the object in terms of the object itself
- This process is called recursion

→ Recursively defined functions

A recursively defined function f with domain $\mathbb{N}$ is a function defined by:
- BS: specify an initial value of the function
- Recursive-Step: give a rule for finding the value of the function at an integer from its values at smaller integers
- Such a definition is called a recursive or inductive definition
- Defining a function $f(n)$ from the set $\mathbb{N}$ to the set R is the same as a sequence $a_0, a_1 \ldots$ where $\forall i \in \mathbb{N}, a_i \in R$

Ex) Let's give a recursive definition of the sequence $\{a_n\}$, $n = 1, 2, 3 \ldots$ in the following cases:

1. $a_n = 4n$
2. $a_n = 4^n$
3. $a_n = 4$

- There may be more than one correct answer to each sequence

1. As each term in the sequence is greater than the previous term by 4, this sequence can be defined by setting $a_1 = 4$ and declaring that $\forall n \geq 1 \; a_{n+1} = 4 + a_n$

2. As each term is 4 times its predecessor, this sequence can be defined as $a_1 = 4$ and $\forall n \geq 1 \; a_{n+1} = 4a_n$

3. This sequence can be defined as $a_1 = 4$ and $\forall n \geq 1 \; a_{n+1} = a_n$

→ Recursively defined sets

Sets can also be defined recursively, by defining two steps:

- BS: where we specify some initial elements
- RS: where we provide a rule for constructing new elements from those we already have

Ex| Consider the subset S of the set of integers recursively defined by:

1. Basis Step: $4 \in S$
2. Recursive Step: if $x \in S$ and $y \in S$, then $x+y \in S$

- Later we will see how it can be proved that the set S is the set of all positive integers that are multiples of 4

→ Recursive algorithms

Def'n 1: An algorithm is a finite sequence of precise instructions for performing a computation or for solving a problem

Def'n 2: An algorithm is called recursive if it solves a problem by reducing it to an instance of the same problem with a smaller input

Ex| Let's give a recursive algorithm for computing $n!$, where n is a nonnegative integer:

- $n!$ can be recursively defined by the following two steps:
  BASIS STEP: $0! = 1$
  RECURSIVE STEP: $n! = n(n-1)!$ when n is a positive integer
- The pseudocode of this algorithms can be formalised as:

```
procedure factorial(n: nonnegative integer){
    if n=0 then return 1
    else
    return n factorial(n-1)
}
```

- LECTURE 6.204 -

Outlines
- Def'n
- Example: Hanoi Tower
- Linear Recurrences
- Arithmetic sequences
- Geometric sequences
- Divide and conquer relations

→ Def'n:
- A recurrence relation is an equation that defines a sequence based on a rule that gives the next term as a function of the previous term
- An infinite sequence is a function from the set of positive integers to the set of real numbers
- In many cases, it can be very useful to formalise the problem as a sequence before solving it

→ Hanoi Tower: !! Watch the related lecture. ($a_n = 2a_{n-1} + 1$)

→ Linear Recurrences: A linear recurrence is a relation in which each term of a sequence is a linear function of earlier terms in the sequence.
- There are two types of linear recurrence:
  - linear homogeneous recurrences:
    - formalised as $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$
    - where $c_1, c_2, \dots, c_k \in R$, and k is the degree of the relation
  - linear non-homogeneous recurrences:
    - formalised as $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + f(n)$
    - where $c_1, c_2, c_3, \dots, c_k \in R$, $f(n)$ is a function depending only on n, and k is the degree of the relation

Ex) First order recurrence

Let's consider the following case:
- a country with currently 50 million people that:
  - has a population growth rate (br - dr) of 1% per year
  - receives 50,000 immigrants per year
  - question: find this country's population in 10 years from now
- This case can be modelled as the following first-order linear recurrence:
  - where a_n is the population in n years from now
  - $\forall n \in N$, a_{n+1} is expressed as $a_{n+1} = 1.01\, a_n + 50,000$
  - $a_0 = 50,000,000$

Ex) Second order recurrence

Let's consider the following sequence:

- 0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...
- where each number is found by adding up the two numbers before it

This sequence can be modelled as the following second-order linear recurrence:

- $a_n = a_{n-1} + a_{n-2}$
- $a_0 = 0$
- $a_1 = 1$

This sequence is famously known as the fibonacci sequence

→ Arithmetic sequences

- A sequence is called arithmetic if the difference between consecutive terms is a constant c
- $\forall n$, a_{n+1} is expressed as $a_{n+1} = a_n + c$ and $a_1 = a$

Ex) . The sequence 2, 5, 8, 11, 14, ... is arithmetic with an initial term of $a_0 = 2$ and a common difference of 3
- 30, 25, 20, 15, ... is arithmetic with an initial term of $a_0 = 30$ and a common difference of -5

→ Geometric sequences

- A sequence is called geometric if the ratio between consecutive terms is a constant r
- $\forall n$, a_{n+1} is expressed as $a_{n+1} = r a_n$ and $a_0 = a$

Ex) . The sequence 3, 6, 12, 24, 48, ... is geometric with an initial term of $a_0 = 3$ and a common ratio of 2
- 125, 25, 5, 1, 1/5, 1/25, ... is geometric with an initial term of $a_0 = 125$ and a common ratio of 1/5

→ Divide and conquer recurrence ; it is an algorithm consists of 3 steps;

- Dividing a problem into smaller subproblems
- Solving (recursively) each subproblem
- And then combining solutions to find a solution to the original problem

— LECTURE 6.206 —

Outlines
- Solving linear recurrence
- Solving fibonacci recurrence
- Using strong induction to solve recurrence

→ Solving linear recurrence

- Let $a_n = c_1 a_{n-1} + c_2 a_{n+2} + \dots + c_k a_{n-k}$ be a linear homogeneous recurrence
- If a combination of the geometric sequence $a_n = r^n$ is a solution to this recurrence, it satisfies $r^n = c_1 r^{n-1} + c_2 r^{n-2} \dots c_k r^{n-k}$
- By dividing both sides by r^{n-k}, we get: $r^k = c_1 r^{k-1} + c_2 r^{k-2} \dots c_k$

This equation is called the characteristic equation,

Solving this equation is the first step towards finding a solution to linear homogeneous recurrence:

- If r is a solution of the equation with multiplicity p, then the combination $(a + \beta n + \gamma n^2 + \dots + \mu n^{p-1}) r^n$ satisfies the recurrence

Example: Solving Fibonacci

Let's consider solving the Fibonacci recurrence relation:

$f_n = f_{n-1} + f_{n-2}$, with $f_0 = 0$ and $f_1 = 1$

→ Solution
- The characteristic equation of the fibonacci recurrence rel. is:
  - $r^2 - r - 1 = 0$
- It has two distinct roots, of multiplicity 1:
  - $r_1 = (1+\sqrt{5})/2$ and $r_2 = (1-\sqrt{5})/2$
- So, $f_n = a_1 r_1^n + a_2 r_2^n$ is a solution

. To find a_1 and a_2 we need to use the initial conditions.

. From:

. $f_0 = a_1 + a_2 = 0$

. $f_1 = a_1(1+\sqrt{5})/2 + a_2(1+\sqrt{5})/2 = 1$

. we can find $a_1 = 1/\sqrt{5}$ and $a_2 = -1/\sqrt{5}$

. The solution is then formalised as:

$f_n = 1/\sqrt{5} \cdot (((1+\sqrt{5})/2)^n) - 1/\sqrt{5} \cdot ((1-\sqrt{5})/2)^n))$

Ex) Let's consider another sequence:

. $a_n = -3a_{n-1} - 3a_{n-2} - a_{n-3}$

. $a_0 = 1$, $a_1 = -2$ and $a_2 = -1$

Solution:

. The characteristic equation of this relation is:

$r^3 + 3r^2 + 3r + 1 = 0$

. It has one distinct root, whose multiplicity is 3, $r_1 = -1$

. So, $a_n = (a_0 + a_1 n + a_2 n^2) r_1^n$ is a solution.

. To find a_0, a_1 and a_2 we need to use the initial conditions.

. From:

. $a_0 = a_0 = 1$

. $a_1 = -(a_0 + a_1 + a_2) = -2$

. $a_2 = -(a_0 + 2a_1 + 4a_2) = -1$

. we can find $a_1 = 1$, $a_2 = -3$ and $a_3 = -2$

. The solution is then formalised as:

$a_n = (1 + 3n - 2n^2)(-1)^n$

→ Induction for solving recurrence

Sometimes, it is easier to verify a recurrence relation solution using strong induction

Ex) Let's try to prove the following:

. P(n): the sequence $f_n = 1/\sqrt{5}(r_1^n - r_2^n)$ verifies the fibonacci recurrence, where:

- $r_1 = (1+\sqrt{5})/2$
- $r_2 = (1-\sqrt{5})/2$ are the roots of $r^2 - r - 1 = 0$

- Let's verify $P(2)$:
  - $f_1 + f_0 = 1/\sqrt{5}(r_1 - r_2) = 1/\sqrt{5}(\sqrt{5}) = 1 = f_2$
  - because $f_2 = 1/\sqrt{5}(r_1^2 - r_2^2) = 1$
  - which verifies the initial condition
- Let $k \in \mathbb{N}$, where $P(2)P(3)\dots P(k)$ are all true
- Let's verify $P(k+1)$:
  - $f_n + f_{n-1} = \dfrac{(r_1^n - r_2^n)}{\sqrt{5}} + \dfrac{(r_1^{n-1} - r_2^{n-1})}{\sqrt{5}} =$

$= r_1^{n-1}(r_1+1)/\sqrt{5} + r_2^{n-1}(r_2+1)/\sqrt{5} = r_1^{n-1} * r_1^2 + r_2^{n-1} * r_2^2 =$

$= r_1^{n+1} + r_2^{n+1}$ which equals to f_{n+1}

We conclude that $P(k+1)$ is true and the strong induction is verified.

- LECTURE 7.101 -

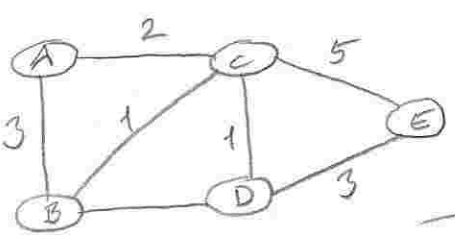
Outlines
- Definition of a graph
- Origins and applications of graph theory

→ Introduction
- Computer scientists must create abstractions of real-world problems that can be presented and manipulated by a computer.
- For instance, logic is used to design a computer circuits.
- Scheduling final exams is another example.
  - The scheduling has to take into account associations between courses, students and rooms
  - These associations (connections) between items are modelled by graphs

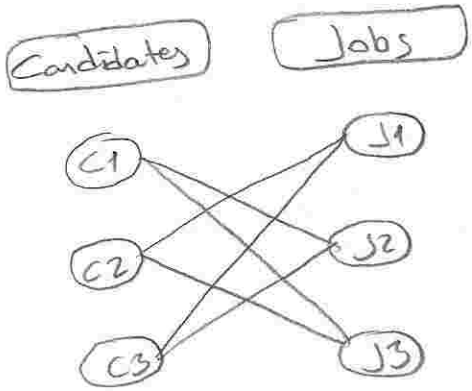
→ What is a graph?
- Graphs are discrete structures consisting of vertices (nodes) and edges connecting them
- Graph theory is an area in discrete mathematics which studies these types of discrete structures

→ Origins of graph theory
- The first problem in graph theory is the Seven Bridges of Konigsberg problem solved by Leonhard Euler in 173

→ Application of graphs
- In a variety of disciplines, problems can be solved using graph models:
  - Modelling computer networks
  - Modelling road maps
  - Solving shortest path problems between cities



- Assigning jobs to employees in an organisation



-LECTURE 7.103-

Outlines
- What is a graph?
- Definitions of graph, vertices, edges & adjacency
- Loops and parallel edges
- Directed graphs

→ Definition: Graph

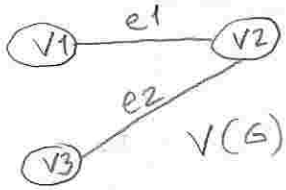
G is an ordered triple $G := (V, E)$

V is a set of nodes, or vertices

E is a set of edges, lines or connections

→ Definition: Vertex

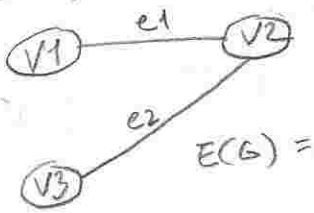
Vertex
- Basic element of a graph
- Drawn as a node or a dot
- Set of vertices of G is usually denoted by V(G) or V.



$V(G) = \{V_1, V_2, V_3\}$

→ Definition: Edges

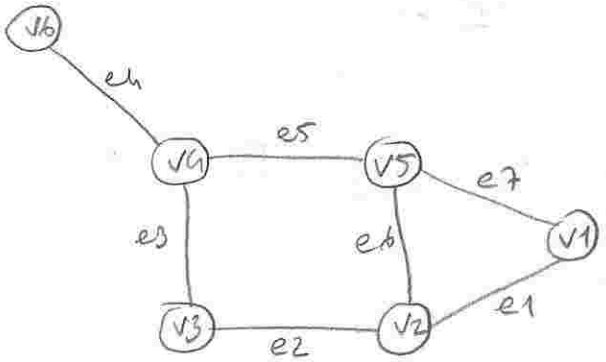
Edge
- A is a link between 2 vertices
- Drawn as a line connecting two vertices
- The set of edges in a graph G is usually denoted by E(G), or E.



$E(G) = \{e_1, e_2\} = \{\{V_1, V_2\}, \{V_2, V_3\}\}$

→ Definition: Adjacency

Adjacency
- Two vertices are said to be adjacent if they are endpoints of the same edge
- Two edges are said adjacent if they share the same vertex
- If a vertex v is an endpoint of an edge e, then we say that e and v are incident

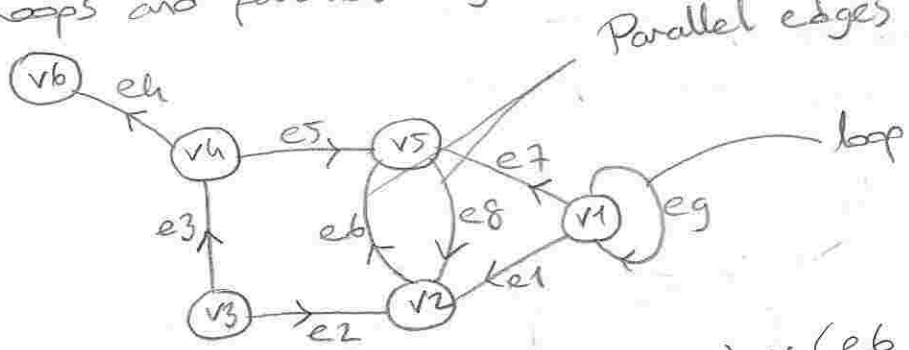


$V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

- v_1 and v_2 are endpoints of the edge e_1. We say that v_1 and v_2 are adjacent
- The edges e_1 and e_7 share the same vertex v_1. We say that e_1 and e_7 are adjacent.
- The vertex v_2 is an endpoint of the edge e_1. We say that e_1 and v_2 are incident

→ Loops and parallel edges



- v_2 and v_5 are linked with two edges (e_6 and e_8). e_6 and e_8 are called parallel edges
- v_1 is linked to itself by e_g. The edge e_g is called loop

→ Directed graphs - Digraph

- A directed graph, also called a digraph, is a graph in which the edges have a direction
- This usually indicated with an arrow on the edge

e_1 is a connection from v_1 to v_2 but not from v_2 to v_1
e_6 is a connection from v_2 to v_5 whereas e_8 is a connection from v_5 to v_2

LECTURE 7.105

Outlines
- Definition of a walk, a trail, a circuit and a path
- Definition of a Euler path
- Definition of a Hamiltonian path
- Definition of connected graphs
- Definition of strongly connected graphs
- Transitive closure of graphs

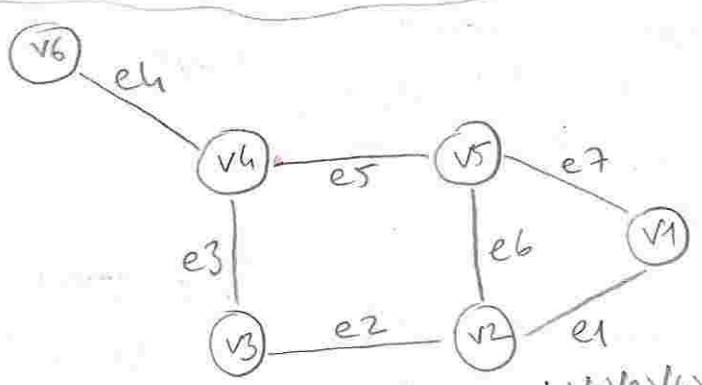
→ Definition of a Walk

A walk is a sequence of vertices and edges of a graph where vertices and edges can be repeated.

A walk of length k in a graph is a succession of k (not necessarily different) edges of the form

$uv, uw, wx, \ldots, yz$

Ex1



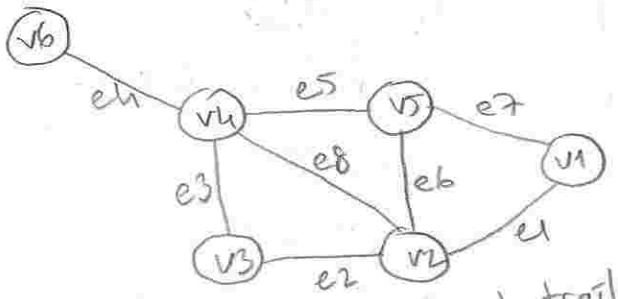
$V_1V_2, V_2V_3, V_3V_4, V_4V_6 = e_1, e_2, e_3, e_4 = V_1V_2V_3V_4V_6$

A walk of length 4 from V_1 to V_6

→ Trail: A trail is a walk in which no edge is repeated.

In a trail, vertices can be repeated but no edge is ever repeated

e1, e2, e3, e5, e6 is a trail



→ Circuit: A circuit is a closed trail. Circuits can have repeated vertices only

e7, e6, e8, e3, e2, e1 is a circuit

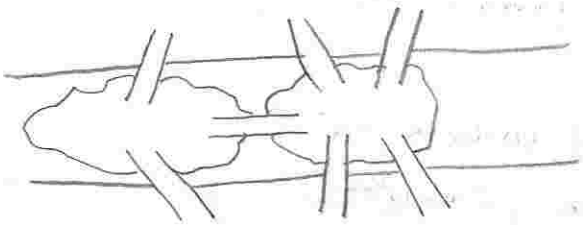
→ Definition of a path: A path is a trail in which neither vertices nor edges are repeated

→ Cycle: A cycle is a closed path, consisting of edges and vertices where a vertex is reachable from itself

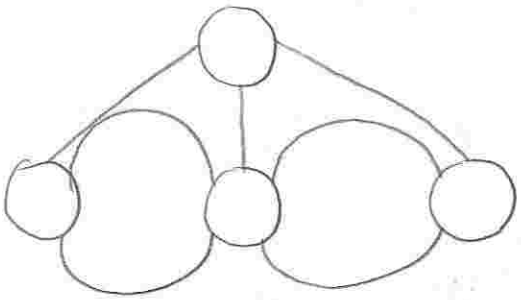
$v_1v_2, v_2v_5, v_5v_1 = e_1, e_2, e_3 = v_1v_2v_5v_1$

A walk of length 3 from v_1 to v_1 = closed path = cycle

→ The seven Bridges of Königsberg



→ Leonard Euler

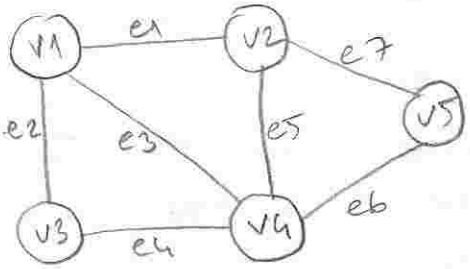


Has no walk that uses each edge exactly once (even if we allow the walk to start and finish in different places)

→ Euler path

Def'n: A Eulerian path in a graph is a path that uses each edge precisely once. If such a path exists, the graph is called traversable.

Ex)



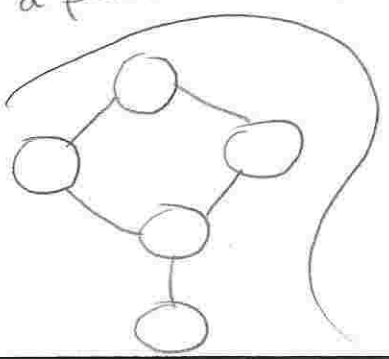
$e_2, e_4, e_6, e_7, e_5, e_3, e_1 =$

$= v_1v_3v_4v_5v_2v_4v_1v_2$

is a Euler path

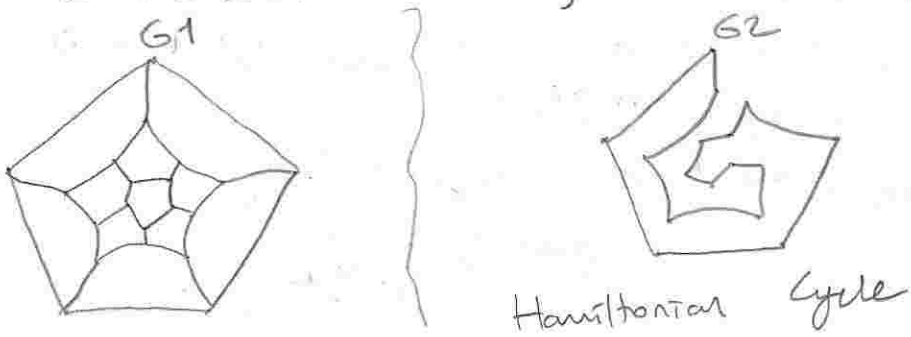
→ Hamiltonian path: It is also called as traceable path that is a path that visits each vertex exactly once

Ex)



A graph that contains a Hamiltonian path is called a traceable path

→ Hamiltonian cycle : It is a cycle that visits each vertex exactly once (except for the starting vertex, which is visited once at the start and once again at the end)

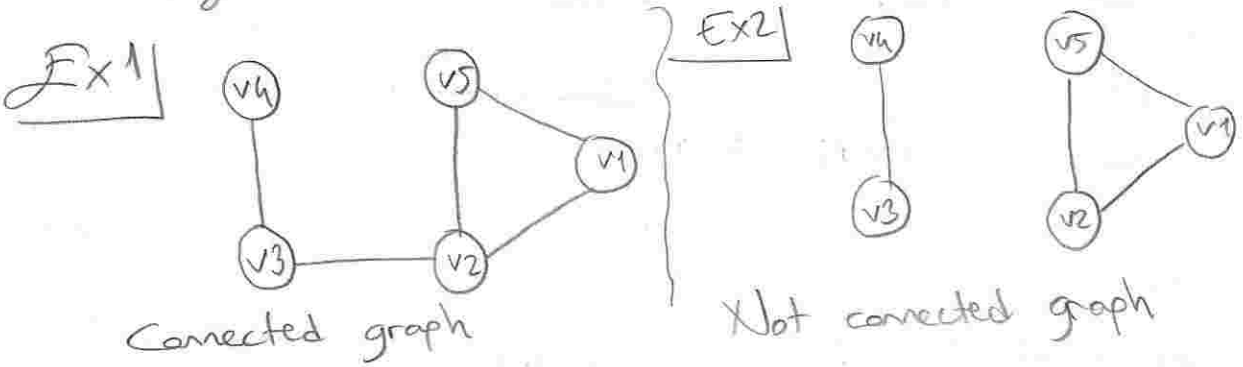


→ Connectivity

An undirected graph is connected if you can get from any node to any other by following a sequence of edges

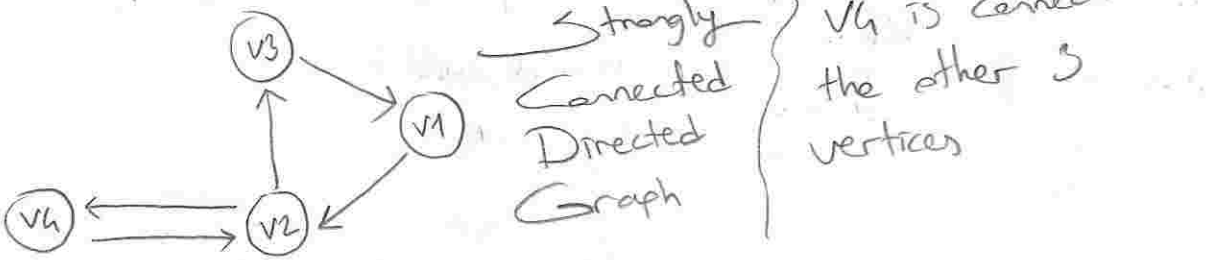
OR

any two nodes are connected by a path



→ Strong Connectivity

A directed graph is strongly connected if there is a directed path from any node to any other node.

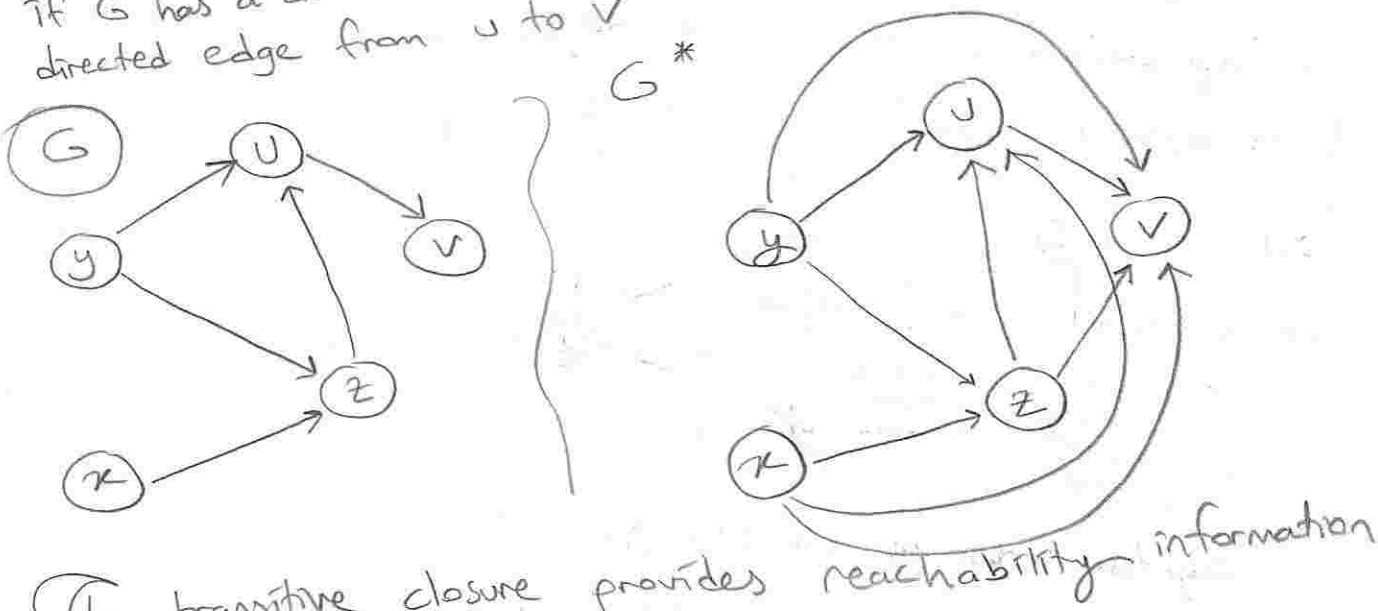


→ Transitive Closure

Given a digraph G, the transitive closure of G is the digraph G^* such that:

G^* has the same vertices as G

If G has a directed path from u to v ($u \neq v$), G^* has a directed edge from u to v



The transitive closure provides reachability information about a digraph

---

— LECTURE 7.1.07 —

Outlines

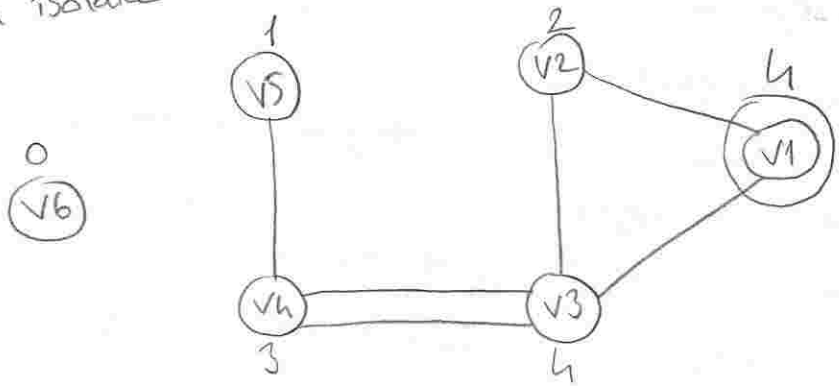
- Degree of a vertex
- Degree sequence of a graph
- Properties of the degree sequence of graph
- Example

Terminology - Undirected graphs

→ Degree of vertex (deg(v)): the number of edges incident on V

A loop contributes twice to the degree

An isolated vertex has a degree: 0



→ Terminology – Directed graphs

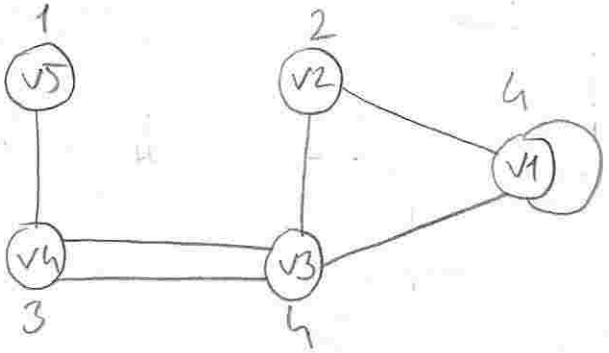
In-deg(v): number of edges for which v is the terminal vertex

Out-deg(v): number of edges for which v the is initial vertex

$$deg(v) = Out\text{-}deg(v) + IN\text{-}deg(v)$$

A loop contributes twice to the degree as it contributes 1 to both in-degree and out degree

Ex)



$deg(V_1) = in\text{-}deg(V_1) + out\text{-}deg(V_1) = 2+2 = 4$

$deg(V_2) = in\text{-}deg(V_2) + out\text{-}deg(V_2) = 1+1 = 2$

$deg(V_3) = in\text{-}deg(V_3) + out\text{-}deg(V_3) = 2+2 = 4$

$deg(V_4) = in\text{-}deg(V_4) + out\text{-}deg(V_4) = 1+2 = 3$

$deg(V_5) = in\text{-}deg(V_5) + out\text{-}deg(V_5) = 1+0 = 1$

$deg(V_6) = in\text{-}deg(V_6) + out\text{-}deg(V_6) = 0+0 = 0$

→ Degree sequence of a graph

Given an undirected graph G, a degree sequence is a monotonic nonincreasing sequence of the vertex degrees of all the vertices of G

→ Degree sequence property 1

. The sum of the degree sequence of a graph is always even

. Therefore, it is impossible to construct a graph where the sum of the degree sequence is odd

→ Degree sequence property 2

. Given a graph G, the sum of the degree sequence of G is twice the number of edges in G

. Number of edges(G) = (sum of degree sequences of G)/2

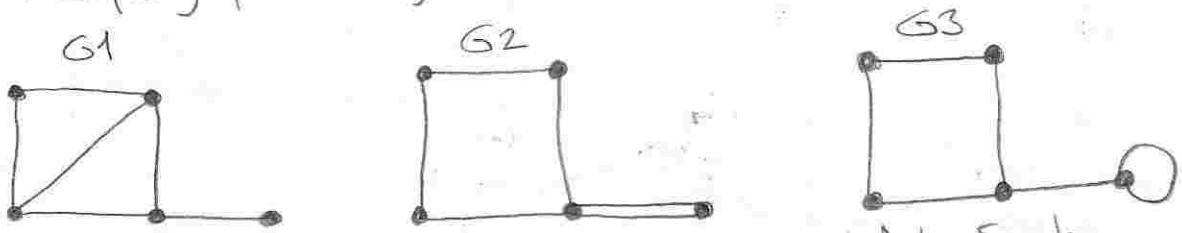
→ Sum of degree sequence of G = 2× number of edges in G

— LECTURE 7.109 —

Outlines
- Simple graphs
- Regular graphs
- Complete graphs

→ Simple graphs

A simple graph is a graph without loops and parallel edges



Simple

Not Simple (parallel edges)

Not Simple (loop)

→ More Simple graphs

→ Properties of simple graphs

Given a simple graph G with n vertices, then the degree of each vertex of G is at most equal to $n-1$

Proof:

Let v be a vertex of G such that $\deg(v) > n-1$

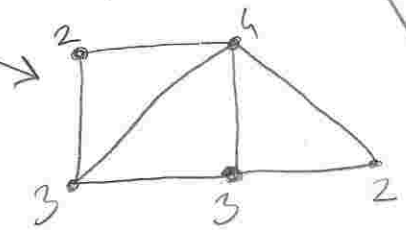
However, we have only $n-1$ other vertices for v to be connected to

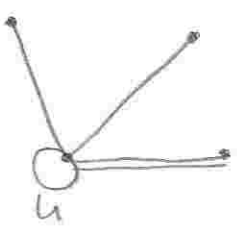
Hence, the other connections can only be a result of parallel edges of loops

Exercise: Can we draw a simple graph with the following degree sequences?

4,2,2,2 → It has to contain a loop or parallel edges
therefore, no degree sequence

4,3,3,2,2





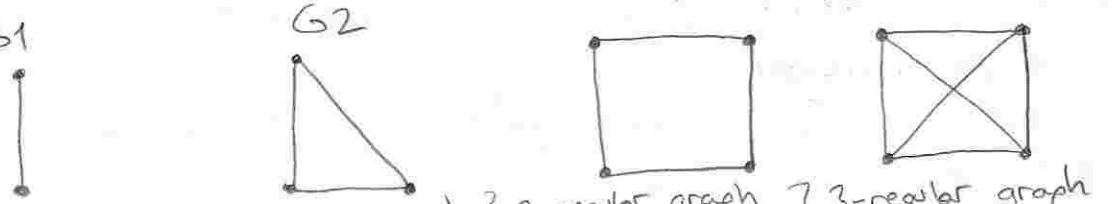
Sum degree = 4+2+2+2
= 10

→ Regular graphs

A graph is said to be regular of degree if all local degrees are the same number

A graph G where all the vertices the same degree, r, is called an r-regular graph

Ex



1-regular graph with 2 vertices
1, 1

2-regular graph with 3 vertices
2, 2, 2

2-regular graph with 4 vertices
2, 2, 2, 2

3-regular graph with 4 vertices
3, 3, 3, 3

→ Properties of regular graphs

Given an r-regular G with n vertices, then the following is true:

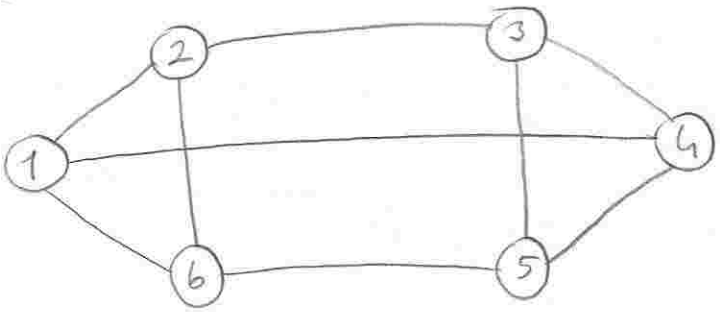
Degree sequence of $G = r, r, r, \dots, r$ (n times)

Sum of degree sequence of $G = r \times n$

Number of edges in $G = r \times n / 2$

Ex 3-regular with 6 vertices

G

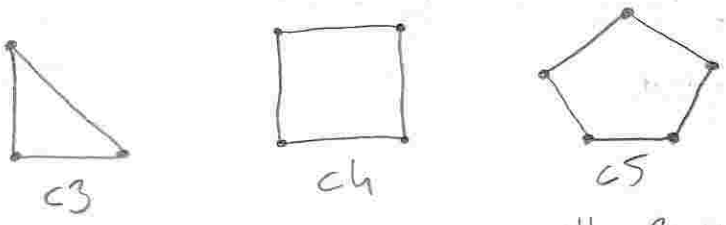


Degree Sequence = 3, 3, 3, 3, 3, 3

Sum of degree sequence = $3 \times 6 = 18$

Number of edges = $18/2 = 9$

→ Special regular graphs : cycles



C_3 C_4 C_5

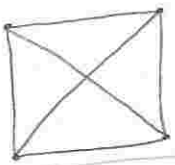
C_3 is 2-regular graph with 3 vertices
C_4 is 2-regular graph with 4 vertices
C_5 is 2-regular graph with 5 vertices

deg seq. of $C_3 = 2,2,2$
deg seq. of $C_4 = 2,2,2,2$
deg seq. of $C_5 = 2,2,2,2,2$

Exercise 1 Can we construct a 3-regular graph with 4 vertices?

Sol. 3-regular with 4 vertices

- Sum of degree sequence $= 3 \times 4 = 12$
- The sum is even, hence it is possible to construct 3-regular graph with 4 vertices



Exercise 2 Can we construct a 3-regular graph with 5 vertices?

Sol. 3-regular graph with 5 vertices

- Sum of the degree sequence $= 3 \times 5 = 15$
- The sum if odd, hence it is impossible to construct a 3-regular graph with 5 vertices

→ Complete graph: A complete graph is a simple graph where every pair of vertices are adjacent (linked with an edge)

We represent a complete graph with n vertices using the symbol K_n

→ Complete graph properties

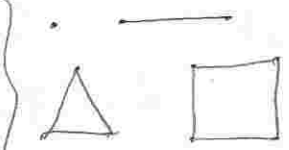
A complete graph with n vertices, k_n, has the following properties:

Every vertex has a degree $(n-1)$

Sum of degree sequence $= n(n-1)$

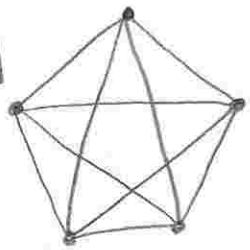
Number of edges $= n(n-1)/2$

Ex's



→ Example of a complete graph

K_5



- There are 5 vertices
- Degree of each vertex $= (5-1) = 4$
- Sum of deg. seq. $= 5(5-1) = 20$
- Number of edges $= 5(5-1)/2 = 20/2 = 10$

→ The number of edges in r-regular graph with n vertices

$$= \frac{r \times n}{2}$$

— LECTURE 7.201 —

Outlines

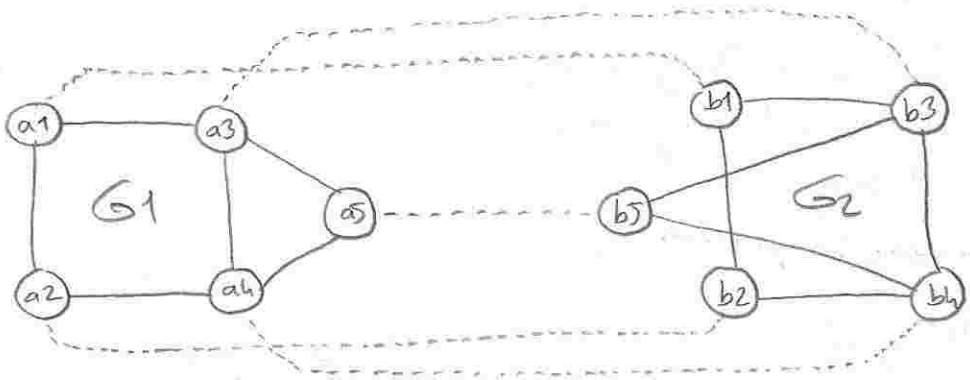
- Definition of isomorphism
- Properties of isomorphic graphs
- Examples

→ Definition of isomorphism

- Two graphs G_1 and G_2 are isomorphic if there is a bijection (invertable function): $f: G_1 \to G_2$ that preserves adjacency and non-adjacency.

If uv is in $E(G_1)$ then $f(u)f(v)$ is in $E(G_2)$

→ Isomorphic graphs

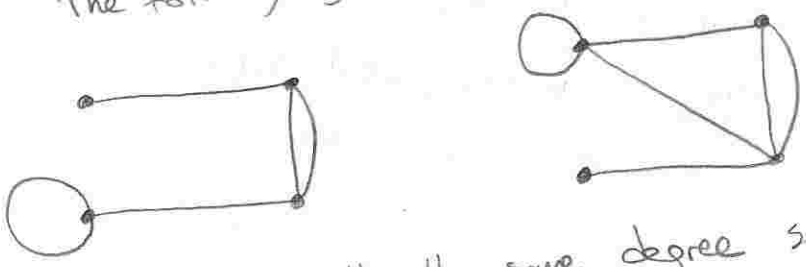


$f : G_1 \to G_2$

G_1	a_1	a_2	a_3	a_4	a_5
$f(G_1) = G_2$	b_1	b_2	b_3	b_4	b_5

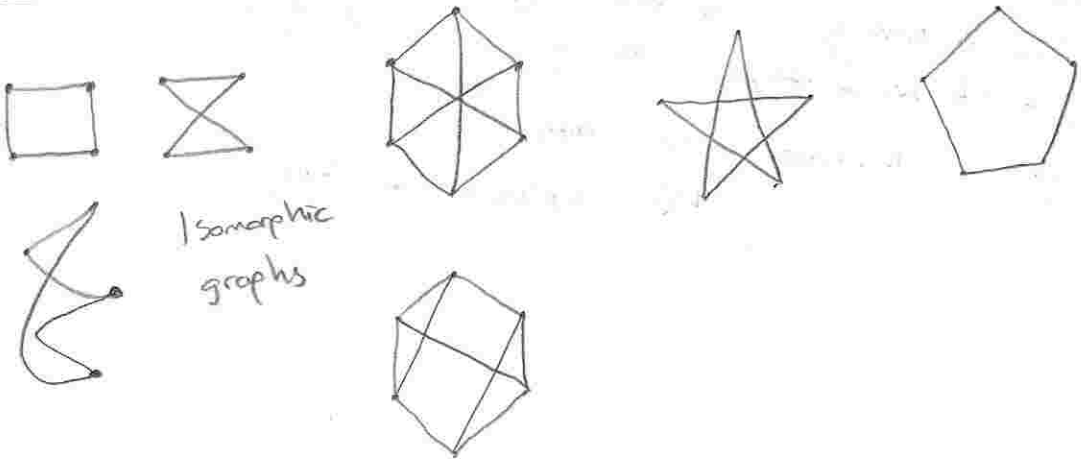
→ Properties of isomorphic graphs

- Two graphs with different degree sequences can't be isomorphic
  The following graphs <u>are not</u> isomorphic



- Two graphs with the same degree sequence aren't necessarily isomorphic
  The following graphs have the same degree sequence but <u>are not</u> isomorphic

Ex

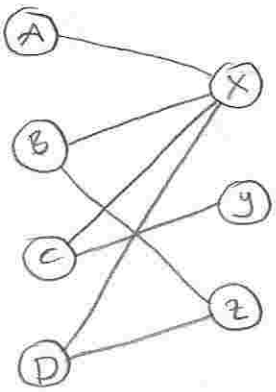


— LECTURE 7.203 —

Outlines
- Definition of bipartite graphs
- Examples of bipartite graphs
- Maximum matching
- Hopcroft-Karp algorithm

→ Bipartite graphs
- A graph $G(V, E)$ is called a bi-partite graph:
- If the set of vertices V can be partitioned in two non-empty disjoint sets V_1 and V_2 in such a way that each edge e in G has one endpoint in V_1 and another endpoint in V_2

Ex | G



$V_1 = \{A, B, C, D\}$

$V_2 = \{x, y, z\}$

- The graph is 2-colourable
- No odd-length cycles

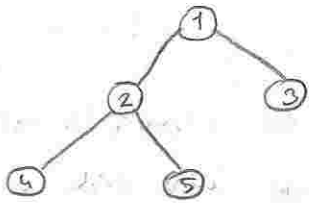
→ Matching
- A matching is a set of pairwise non-adjacent edges, none of which are loops; that is, no two edges share a common endpoint
- A vertex is matched (or saturated) if it is an endpoint of one of the edges in the matching. Otherwise the vertex is unmatched

→ Maximum matching
- A maximum matching is a matching of maximum size such that if any edge is added, it is no longer a matching

→ The Hopcroft-Karp algorithm 1
- An algorithm for solving the maximum matching problem in a bipartite graph

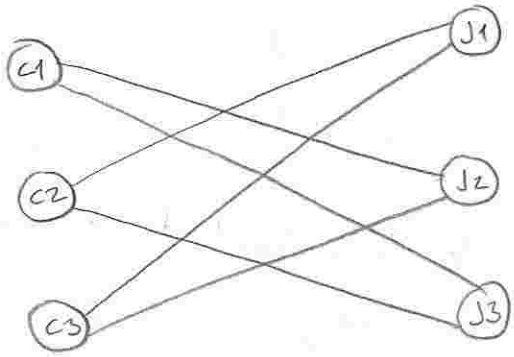
Useful concepts

- Augmenting path
- Breadth First search



→ The Hopcroft-Karp algorithm 2

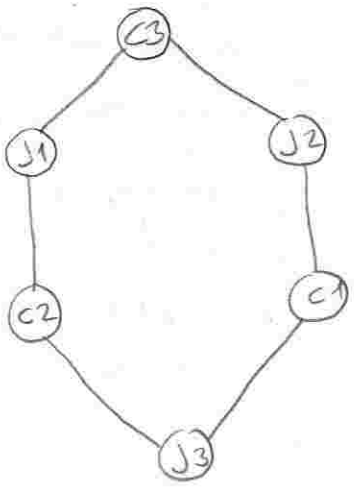
C: candidates J: Job positions



- Three candidates are applying for three Job positions
- The graph represents the match between the candidates' skills and the Job positions

Subcode:

1) Initialize $M = \{\}$
2) While there exists an Augmenting Path P
   1) Use BFS to build layers that terminate at free vertic
   2) Start at the free vertices in C, use DFS
3) Return M



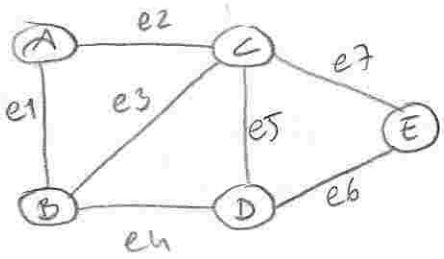
— LECTURE 7.205 —

Outlines

- Adjacency list of a graph
- Adjacency matrix of a graph
- Properties of the adjacency matrix

→ Graph representation recap

So far we have seen that a graph can be represented by a set of vertices and a set of edges

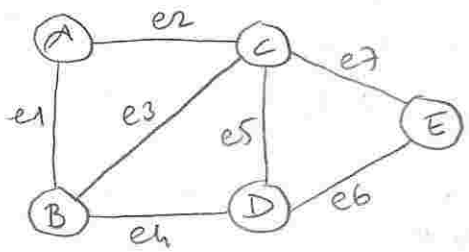


$V = \{a, b, c, d, e\}$ and

$E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\}$

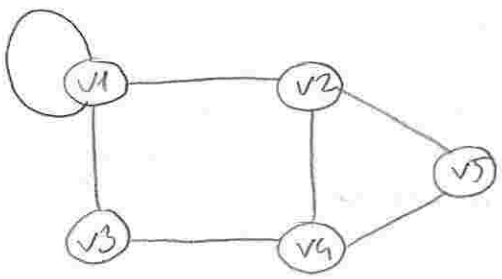
→ Adjacency list.

The adjacency list of a graph G is a list of all the vertices in G and their corresponding individual adjacent vertices



a = b, c
b = a, c, d
c = a, b, d, e
d = b, c, e
e = c, d

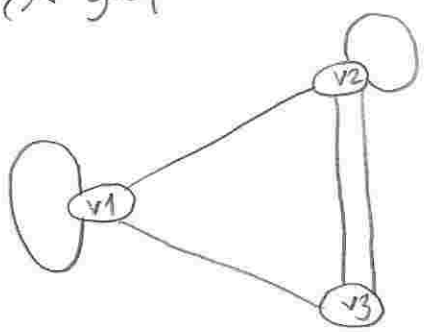
Ex) Given an undirected graph G defined by its corresponding adjacency list. Draw the graph G.



$v_1 = v1, v2, v3$
$v2 = v1, v4, v5$
$v3 = v1, v4$
$v4 = v2, v3, v5$
$v5 = v2, v4$

→ Adjacency matrix

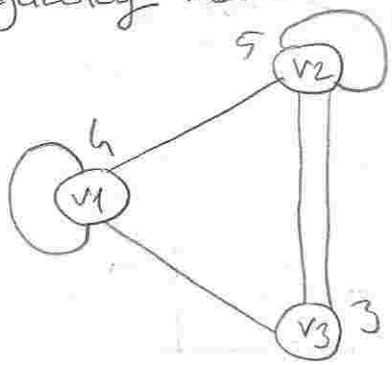
A graph can also be represented by its adjacency matrix.



$$M(G) = \begin{array}{c} \\ v1 \\ v2 \\ v3 \end{array} \begin{array}{c} \begin{array}{ccc} v1 & v2 & v3 \end{array} \\ \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 2 \\ 1 & 2 & 0 \end{bmatrix} \end{array}$$

→ Observation

- The adjacency matrix of an undirected graph is symmetric
- The number of edges in an undirected graph is equal to half the sum of all the elements (m_{ij}) of its corresponding adjacency matrix



$$M(G) = \begin{array}{c} \\ V1 \\ V2 \\ V3 \end{array} \begin{array}{c} \begin{array}{ccc} V1 & V2 & V3 \end{array} \\ \begin{bmatrix} 2 & 1 & 1 \\ 1 & 2 & 2 \\ 1 & 2 & 0 \end{bmatrix} \end{array}$$

$\sum m_{ij} = 1+1+1+1+2+2+2+2 = 5+4+3 = 12$

Number of edges in $G = (\sum m_{ij})/2 = 12/2 = 6$

Sum $(m_{ij}) = 1+1+1+1+2 = 6$ (# of edges)

→ Adjacency matrix squared (M^2)

$$M^2 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 2 \\ 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 2 \\ 1 & 0 & 0 \end{bmatrix}$$

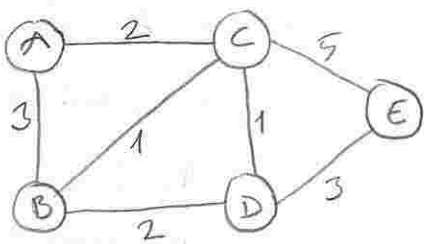
— LECTURE 7.207 —

Outlines

- Weighted graphs
- Dijkstra's algorithm
- Example
- Algorithm's pseudocode

→ Weighted graphs

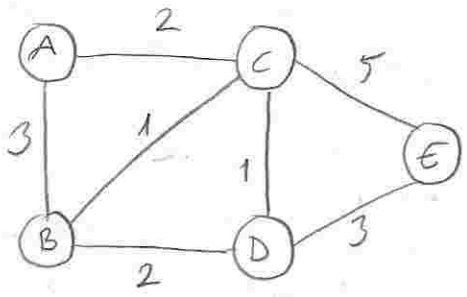
A weighted graph is a graph in which each edge is assigned a numerical weight



→ Dijkstra's algorithm

An algorithm was designed by Edsger W. Dijkstra in 1956, in order to find the shortest path between nodes in a weighted graph

Ex!



Vertex	Shortest distance from A	Previous vertex
A	0	
B	3	A
C	2	A
D	3	C
E	6	D

→ Pseudocode

Let G be a graph and s a source vertex. The following pseudocode calculates the shortest distance and the previous vertex from s to every other node in the graph

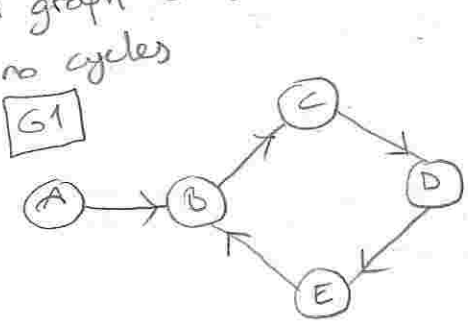
```
Unvisited = {}
for each vertex v in G:
    shortest_distance[v] ← infinity
    previous_vertex[v] ← undefined
    add v to unvisited
    shortest_distance[s] ← 0
while unvisited is not empty:
    u ← vertex in unvisited with min
shortest_distance[u]
    remove u from unvisited
    for each neighbour v of u:
        alt ← shortest_distance[u] +
length(u,v)
        if alt < shortest_distance[v]
            shortest_distance[v]
← alt
            previous_vertex[v]
u
return shortest_distance[], previous_vertex[]
```

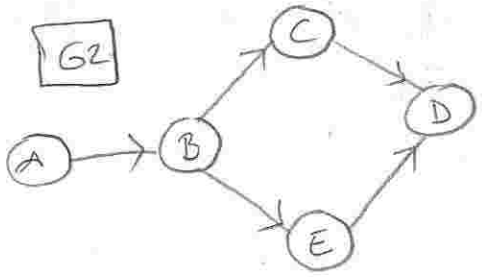
- LECTURE 8.103 -

Outlines
- Acyclic graphs
- Definition of a tree
- Definition of a forest
- Theorems on trees
- Definition of rooted trees

→ Acyclic graphs

A graph G is called an acyclic graph if and only if G has no cycles

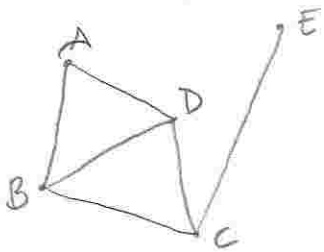




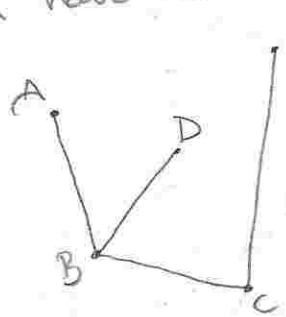
G1 is not an acyclic graph and G2 is.

→ Definition of a tree

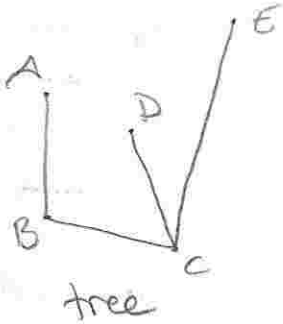
A tree is a connected acyclic undirected graph.
Hence, a tree can have neither loops nor multiple edges (parallel edges)



Not tree



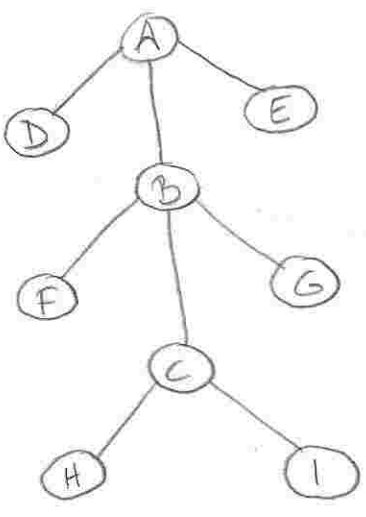
tree

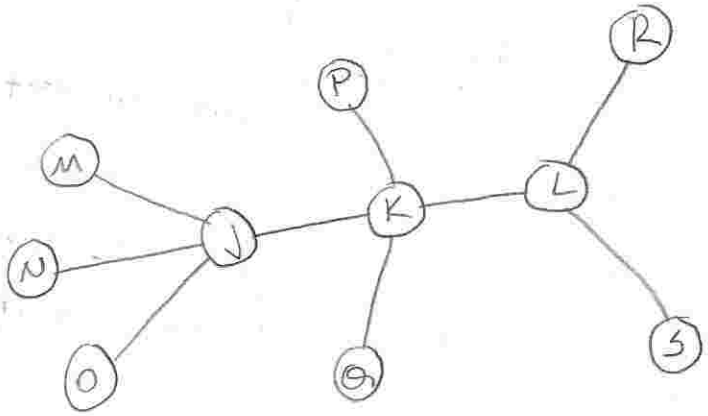


tree

→ Definition of a forest

A disconnected graph containing no cycles is called a forest





→ Theorem 1

- An undirected graph is a tree if and only if there is unique simple path between any two of its vertices

→ Theorem 2

- A tree with n vertices has $n-1$ edges

 → 3 vertices & 2 edges

→ Rooted trees

A rooted tree is when one vertex has been designated as the root and every edge is directed away from the root

---

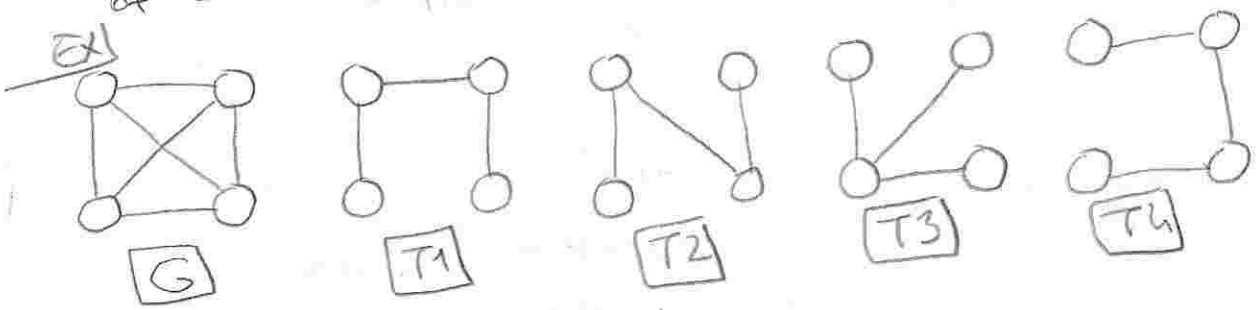
- LECTURE 8.1.05 -

Outlines

- Spanning trees of a graph
- Constructing a spanning tree
- Non-isomorphic spanning trees of a graph

→ Def'n of a spanning trees

A spanning tree of a graph G is a connected sub graph of G which contains all vertices of G, but with no cycles



→ Constructing a spanning tree

- To get a spanning tree of a graph G
  1) Keep all vertices of G
  2) Break all the cycles but keep the tree connected

→ Non-isomorphic spanning trees

- Two spanning trees are said isomorphic if there is a bijection preserving adjacency between the two trees

-LECTURE 8.107-

Outlines
- Example of use case
- Weight of spanning tree
- Minimum spanning trees
- Kruskal's algorithm
- Prim's algorithms

→ Example of use case

Suppose we want to supply a set of houses with:
- electric power
- water pipes
- sewage lines
- telephone lines

- To keep cost down, you could connect these houses with a spanning tree (power lines, for example).
- However, the houses are not all equal distances apart.
- To reduce costs even further, you could connect the houses with a minimum-cost spanning tree

→ Spanning trees cost

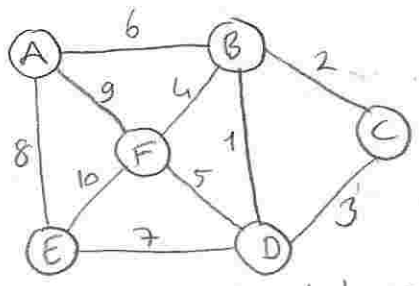
- Suppose you have a connected undirected graph with a weight (or cost) associated with each edge.
- The cost of a spanning tree would be the sum of the costs of its edges

→ The weight of spanning trees

$w_1 = 8+9+6+7+3 = 33$

$w_2 = 6+9+5+2+7 = 29$

$w_3 = 8+6+4+1+2 = 21$

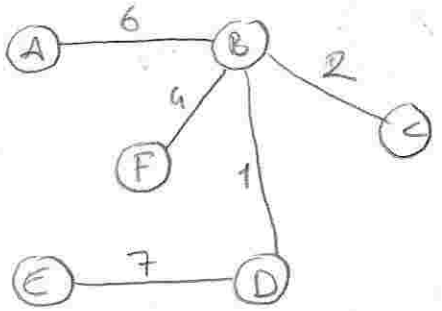


A connected, undirected graph

→ Minimum-cost spanning trees

- A minimum-cost spanning tree is a spanning tree that has the lowest weight (lowest cost)

A minimum-cost spanning tree



→ Finding spanning trees

There are two basics algorithms for finding minimum-cost spanning trees, and both are greedy algorithms:

Kruskal's algorithm

Prim's algorithm

→ Kruskal's algorithm

- Start with the cheapest edges in the spanning tree
- Repeatedly add the cheapest edge that does not create a cycle

→ Prim's algorithm

- Start with any one node in the spanning tree
- Repeatedly add the cheapest edge, and the node it leads to, for which the node is not already in the spanning tree

---

— LECTURE 8.201 —

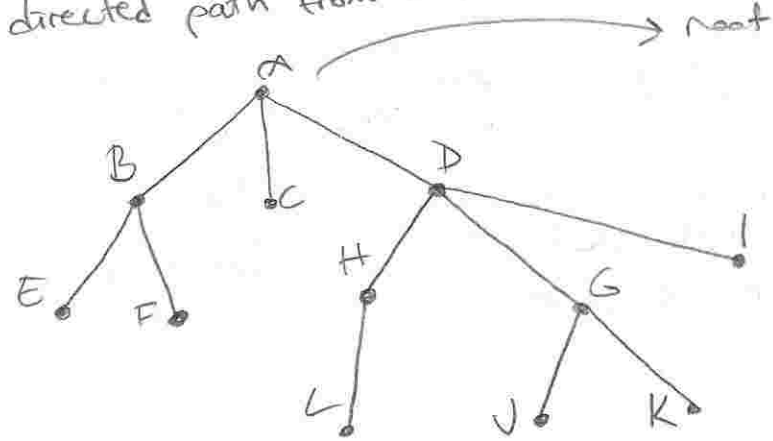
Outlines

- Def'n of rooted trees
- Terminology of rooted trees
- Depth and height in a tree
- Special trees
- Regular rooted trees and properties
- Isomorphic rooted trees and properties

→ Definition of rooted trees

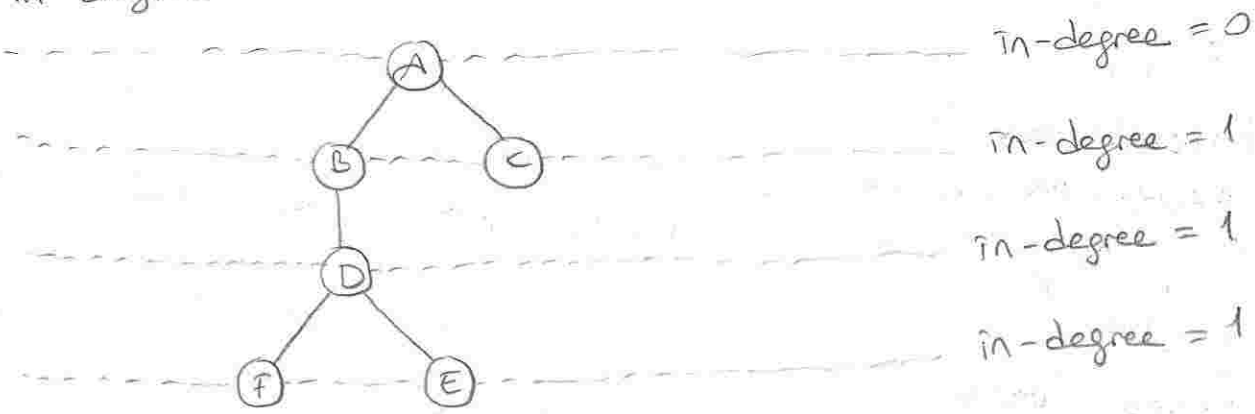
A rooted tree is a directed tree having one distinguished vertex r, called a root, such that for every vertex v there is a directed path from r to v

T



→ Theorem

A directed tree is represented as a rooted tree if and only if one vertex has in-degree 0 whereas all other vertices have in-degree 1.

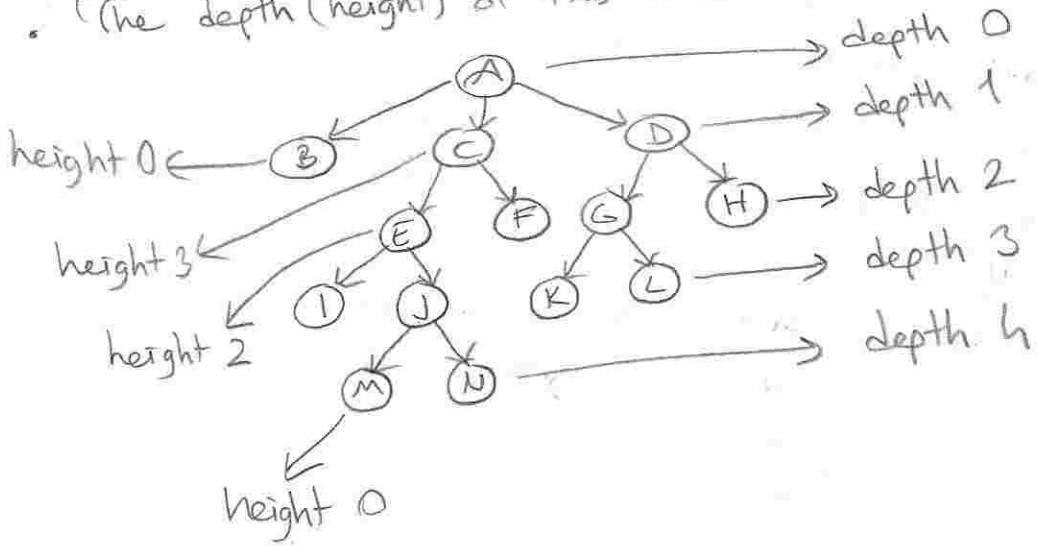


→ Terminology of rooted trees

- A is the root of the tree
- B is called the parent of D
- E and F are the children of D
- B and A are ancestors of E and F (E & F are siblings)
- B and D are called internal nodes
- C, E & F are called external nodes

→ Depth and height in a tree

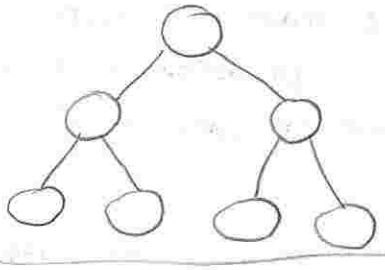
- The depth or path length of a node in a tree is the number of edges from the root to that node
- The height of a node in a tree is the longest path from that node to a leaf
- The depth or the height of a tree is the maximum path length across all its nodes
- The depth (height) of this tree is 4



→ Special trees

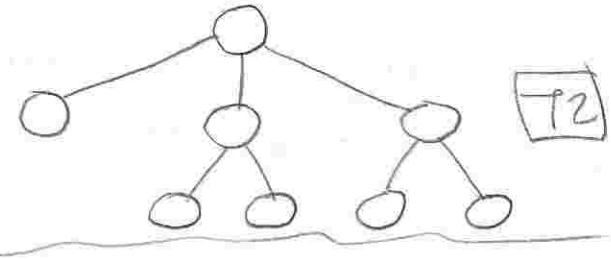
• Binary Trees

A binary tree is a rooted tree in which every vertex has 2 or fewer children



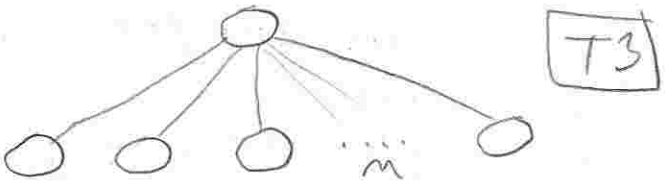
• Ternary Trees

A ternary tree is a rooted tree in which every vertex has 3 or fewer children



• n-ary Trees

A n-ary tree is a rooted tree in which every vertex has n or fewer children



→ Regular rooted trees

An n-ary tree is regular if every one of its internal nodes has exactly n children

→ Properties

An n-ary tree has at most n^h vertices at level h

max no. of nodes per level → $h, 2^h, 3^h, n^h$

level, binary tree, ternary tree

→ Isomorphic trees

Two trees T_1 and T_2 are isomorphic if there is a bijection

$f: V(T_1) \to V(T_2)$

which preserves adjacency and non-adjacency

That is, if uv is in $E(T_1)$ and $f(u)f(v)$ is in $E(T_2)$

Notation:

$T_1 \cong T_2$ means that T_1 and T_2 are isomorphic

→ Properties

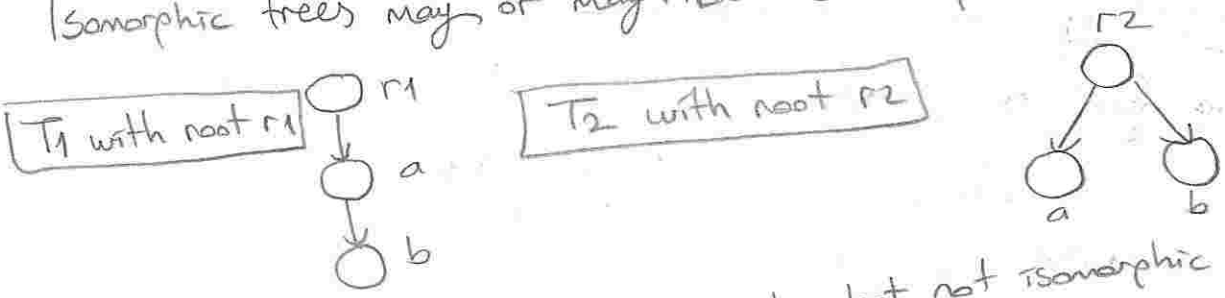
- Two trees with different degree sequences are not isomorphic
- Two trees with the same degree sequence are not necessarily isomorphic

→ Isomorphic rooted trees

Two isomorphic trees are isomorphic as rooted trees if and only if there is a bijection that maps the root of one tree to the root of the other.

→ Properties

Isomorphic trees may or may not be isomorphic as rooted trees



T_1 and T_2 are isomorphic as graphs but not isomorphic as rooted trees.

– LECTURE 8.203 –

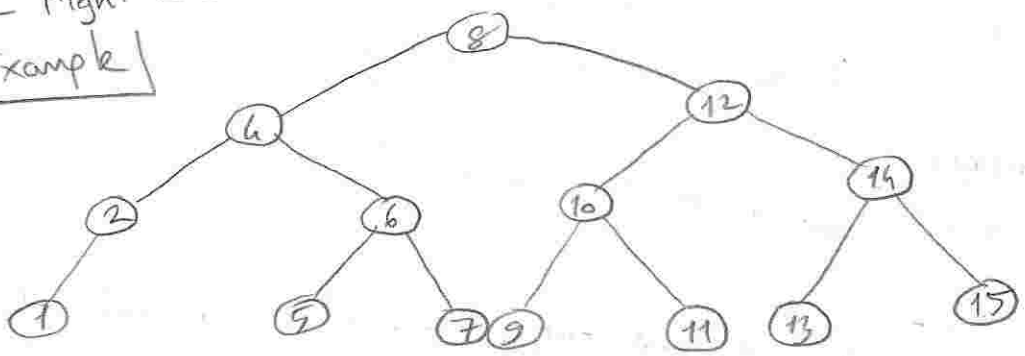
Outlines

- Definition of binary search tree
- Example of a binary search tree
- Application of a binary search tree
- Binary search algorithm
- Example of a binary search algorithm

→ Definition

• A binary search tree is a binary tree in which the vertices are labelled with items so that a label of a vertex is greater than the labels of all vertices in the left subtree of this vertex and is less than the labels of all vertices in the right subtree of this vertex

→ Example



→ Applications

• The use applies in the case where we want to store a modifiable collection in a computer's memory and be able to search, insert or remove elements from the collection in an efficient way.

Ex) Build a binary search tree to store 15 records and find the height of this trees

min ↖ 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15 ↗ max

• root = $\left\lfloor \frac{1+15}{2} \right\rfloor = 8$

• 1, 2, 3, 4, 5, 6, 7

left-hand side = $\left\lfloor \frac{1+7}{2} \right\rfloor = 4$

9, 10, 11, 12, 13, 14, 15

right-hand side = $\left\lfloor \frac{9+15}{2} \right\rfloor = 12$

→ Height of the tree

Method 1 $2^{h-1} < 1+N \leq 2^h \equiv h-1 < \log_2(1+N) \leq h$

Method 2 $\equiv h = \lceil \log_2(N+1) \rceil$

For example: if $N=15$ then $h=4$

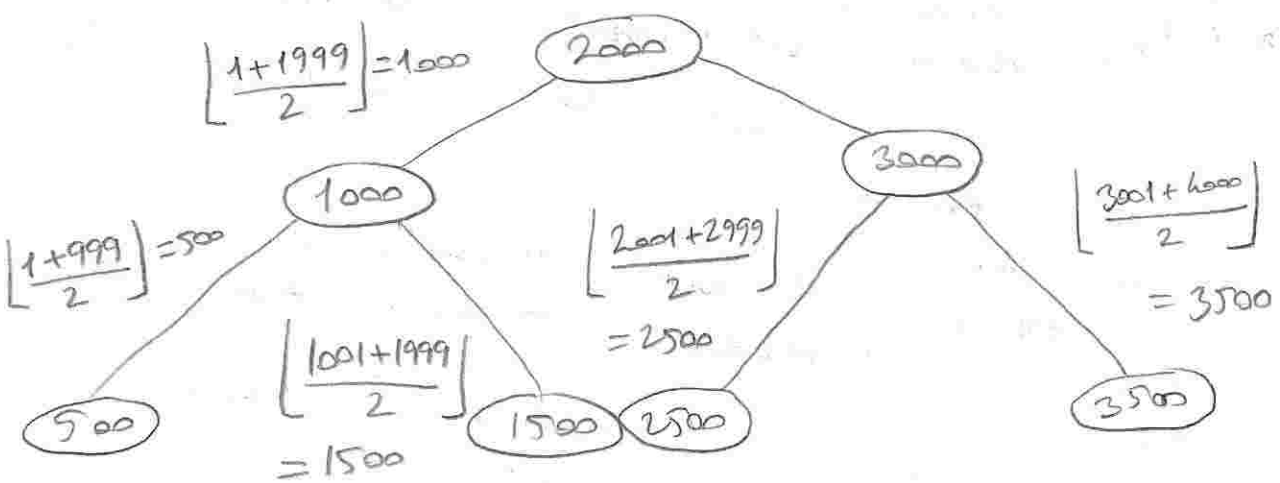
$2^{4-1} < 1+15 \leq 2^4$

$h = \lceil \log_2(15+1) \rceil = \lceil \log_2(16) \rceil = 4$

→ Exercise

• Find the first 3 level of binary search tree to store 4000 records. Find the height of this tree.

$1, 2, 3, 4, \ldots, 4000 \rightarrow root = \left\lfloor \frac{1+4000}{2} \right\rfloor = 2000$



→ Height of the tree

Method 1 → $2^{h-1} < 1+N \leq 2^h \equiv 2^{12-1} < 1+4000 \leq 2^{12} \equiv h = 12$

Method 2 → $h = \lceil \log_2(N+1) \rceil \equiv h = \lceil \log_2(4000+1) \rceil =$

$= \lceil \log_2(4001) \rceil = 12$

→ Binary search algorithm

- The algorithm starts by comparing the searched element to the middle term of the list
- The list is then split into two smaller sub-lists of the same size, or where one of these smaller lists has one fewer term than the other
- The search continues by restricting the search to the appropriate sub-list based on the comparison of the searched element and term in the middle

---

– LECTURE 9.101 –

Outlines

- Relation in general
- Relation in mathematics

→ Relation in general

Relationships between elements of sets occur in many context. We deal with relationships on daily basis:

- a relationship between a person and a relative
- a relationship between an employee and his/her salary
- a relationship between a business and its telephone number
- a relationship between a computer language and a valid statement in this language will often arise in CS

→ Relations in Mathematics

In mathematics we study relationships such as:

- a relation between a positive integer and one that it devides
- a relation between a real number and one that is larger than it
- a relation between a real number x and the value $f(x)$ where f is a function, and so on.

Mathematically:

- we can define a relationship between elements of two sets
- we can also define a relationship between the two elements of a set.

In this topic we will discuss these type of relations as well as their properties in more detail.

- LECTURE 9.103 -

Outlines
- What is a relation?
- Cartesian product
- Definition of a relation
- Relations on a set

→ What is a relation?
- A relation can be defined between elements of a set A and elements of another set B
- A relation can be defined between elements of the same set
- We always use the letter R to refer to a relation
- Let A and B be sets
- Let R be a relation linking elements of A to elements of B
- Let $x \in A$ and $y \in B$
  - We say that x is related to y with respect to the relation R and we write $x R y$
- A relation is a link between two elements of a set
  - For example:
    - A person x is a SON OF a person y
    - SON OF is a relation that links x to y
- We usually use the letter R to refer to a relation:
  - In this case R = 'SON OF'
  - If x is a SON OF y we write $x R y$
  - If y is Not a SON OF x we write $y \not R x$

Ex)
- Let A be the students in the CS program
  - A = {Sofia, Samir, Sarah}
- Let B be the courses the department offers
  - B = {Mathematics, Java, Databases, Art}
- Let R be a relation linking students in the set A to classes they are enrolled in: A student is related to a course if the student is enrolled on this course

Ex)
- Sofia is enrolled in Mathematics & Java,
- Samir is enrolled in Java & Databases,
- Sarah is enrolled in Mathematics and Art,
- Sofia is not enrolled in Art

Notations

- Sofia R Mathematics
- Sofia R Java
- ⋮
- Sofia R Art

→ Cartesian product

- Let A and B be two sets
- The cartesian product $A \times B$ is defined by a set of pairs (x, y) such that $x \in A$ and $y \in B$

$$A \times B = \{(x, y) : x \in A \text{ and } y \in B\}$$

For example:

- $A = \{a_1, a_2\}$ and $B = \{b_1, b_2, b_3\}$
- $A \times B = \{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_2, b_1), (a_2, b_2), (a_2, b_3)\}$

→ Definition of a relation

Let A and B be two sets.

A binary relation from A to B is a subset of a Cartesian product $A \times B$

- $R \subseteq A \times B$ means R is a set of ordered pairs of the form (x, y) where $x \in A$ and $y \in B$
- $(x, y) \in R$ means xRy (x is related to y)

For example:

- $A = \{a, b, c\}$ and $B = \{1, 2, 3\}$
- The following is a relation defined from A to B:

$$R = \{(a, 1), (b, 2), (c, 3)\}$$

- This means that: $aR1$, $bR2$ and $cR3$

→ Relations on a set

- When $A = B$
- A relation R on the set A is a relation from A to A
  - $R \subseteq A \times A$
- We will generally be studying relations of this type

Ex)

- $A = \{1, 2, 3, 4\}$
- Let R be relation on the set A:
  - $x, y \in A$, xRy if and only if $x < y$
- We have: $1R2, 1R3, 1R4, 2R3, 2R4, 3R4$
- $R = \{(1,2), (1,3), (1,4), (2,3), (2,4), (3,4)\}$

- LECTURE 9.105 -

Outlines
- Representing relations via matrices
- Examples
- Combining relations using matrices
- Representing relations via diagraphs
- Examples

→ Relations using matrices

- Given a relation R from a set A to set B
- List the elements of sets A and B in a particular order
- Let $n_a = |A|$ and $n_b = |B|$
- The matrix of R is a $n_a \times n_b$ matrix

$$m_r = [m_{ij}]_{n_a \times n_b}$$

$$m_{ij} = \begin{cases} 1, \text{ if } (a_i, b_j) \in R \\ 0, \text{ if } (a_i, b_j) \notin R \end{cases}$$

→ Example 1

- Let $A = \{\text{Sofia, Samir, Sarah}\}$
- Let $B = \{\text{CS100, CS101, CS102, CS103}\}$
- Consider the relation of who is enrolled in which class
- $R = \{(a,b) \mid \text{person } a \text{ is enrolled in course } b\}$

	CS100	CS101	CS102
Sofia	X	X	
Samir		X	X
Sarah	X		X

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

→ Example 2

- Let $A = \{1,2,3,4,5\}$
- Consider the relation: $< (x,y) \in R$ iff $x < y$

$$M_{\leq} = \begin{bmatrix} 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- Every element is not related to itself as $x \not< x$

→ Example 3

- Let $A = \{1,2,3,4,5\}$
- Consider a relation: $\leq$ $(x,y) \in R$ iff $x \leq y$

$$M_{\leq} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

→ Combining relations

Union:

- The union of two relations is a new set that contains all of the pairs of elements that are in at least one of the two relations
- The union is written as $R \cup S$ or "R or S"
- $R \cup S = \{(a,b) : (a,b) \in R \text{ or } (a,b) \in S\}$

Intersection:

- The intersection of two relations is a new set that contains all of the pairs that are in both sets
- The intersection is written as $R \cap S$ or "R and S".
- $R \cap S = \{(a,b) : (a,b) \in R \text{ and } (a,b) \in S\}$

→ Combining relations: via Boolean operators

Let $M_R = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ $M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{bmatrix}$

Join $M_{R \cup S} = M_R \vee M_S = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$

Meet $M_{R \cap S} = M_R \wedge M_S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

Ex) Jason R 6.042 → infix

R (Jason, 6.042) → prefix

(Jason, 6.042) $\in$ R

→ Representing relations using directed graphs

Definition:

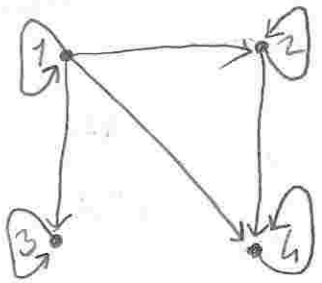
- When a relation is defined on a set, it can be represented by a digraph

Building the digraph:

- First, the elements of A are written down,
- When $(a,b) \in R$ arrows are drawn from a to b

→ Example 1

- $A = \{1,2,3,4\}$
- Let R be relation on A defined as follows:
  - $R = \{(x,y) \mid x \text{ divides } y\}$
- R can be represented by the following digraph



→ Example 2

- Let $A = \{1,2,3,4,5\}$
- Consider a relation: $\leq$ $(x,y) \in R$ iff $x \leq y$

-LECTURE 9.107-

Outlines

- Reflexivity
- Symmetry
- Anti-symmetry

→ Definition of reflexivity

A relation R in a set S is said to be reflexive iff:

$$xRx, \forall x \in S$$
$$(x,x) \in R, \forall x \in S$$

→ Example 1

- Let R be a relation of elements in $\mathbb{Z}$:
  $R = \{(a,b) \in \mathbb{Z}^2 \mid a \leq b\}$
- For all x elements of $\mathbb{Z}$, we have $x \leq x$, hence xRx
- This implies that R is reflexive

→ Example 2

- Let R be a relation of elements in Z:

$$R = \{(a,b) \in \mathbb{Z}^2 \mid a < b\}$$

- For all x elements of Z, we have $x \not< x$ hence $x \not R x$
- For example $1 \in \mathbb{Z}$, however, 1 is not strictly less than 1
- Hence, 1 is not related to 1
- This implies that R is not reflexive

→ Digraph of reflexive relation

The digraph of a reflexive relation on elements in a set S contains a loop on every element of S.

Ex) Let $S = \{1,2,3,4\}$ and let R be a relation of elements in S; $R = \{(a,b) \in S^2 \mid a \leq b\}$

→ Matrix of reflexive relation

Let M_r be the matrix of a reflexive relation, in which all the values of the diagonal of M_r are equal to 1

→ Definition of symmetry

A relation R on a set S is said to be symmetric iff

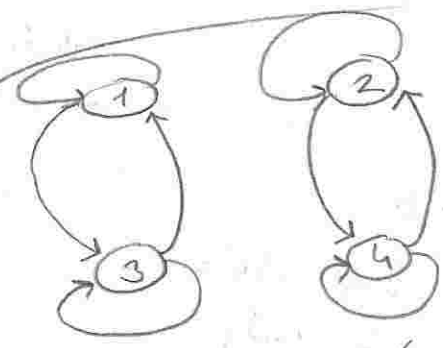
$$\forall a,b \in S, \text{ if } a\,R\,b \text{ then } b\,R\,a$$

Ex) Let R be a relation of elements in Z:

$$R = \{(a,b) \in \mathbb{Z}^2 \mid a \bmod 2 = b \bmod 2\}$$

Proof: let $a, b \in \mathbb{Z}$ with $a\,R\,b$:

- $a \bmod 2 = b \bmod 2$
- $b \bmod 2 = a \bmod 2$
- $b\,R\,a$
- R is symmetric



→ Digraph of a symmetric relation

Ex) Let $S = \{1,2,3,4\}$ and R be relation of elements in S

$$R = \{(a,b) \in S^2 \mid a \bmod 2 = b \bmod 2\}$$

→ Matrix of symmetric relation

The adjacency matrix, M_r, of a symmetric relation is symmetric

Ex) Let $S = \{1,2,3,4\}$ and let R be relation of elements in S; $R = \{(a,b) \in S^2 \mid a \bmod 2 = b \bmod 2\}$

$$M_R = \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \end{matrix}$$

Ex) $R = \emptyset$ ✓, $R = A \times A$ ✓, $R = \{(3,1),(1,3),(2,3)\}$ ✗

→ Definition of anti-symmetric

A relation R on a set S is said to be anti-symmetric iff:

$\forall a, b \in S$, if aRb and bRa then $a = b$

Ex) Let R be a relation on elements in $\mathbb{Z}$:

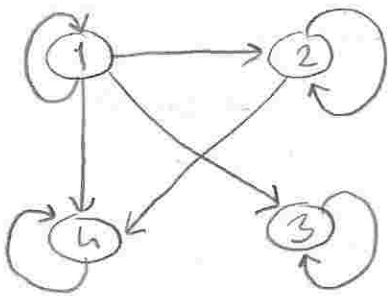
$R = \{(a,b) \in \mathbb{Z}^2 \mid a \leq b\}$

- Let $a, b \in \mathbb{Z}$, aRb and bRa
  - $a \leq b$ and $b \leq a$
  - $a = b$
- R is anti-symmetric

→ Digraph of an anti-symmetric relation

The digraph of an anti-symmetric relation on elements in a set S contains no parallel edges between any two difference vertices

Ex) Let $S = \{1,2,3,4\}$ and R be relation on elements in S; $R = \{(a,b) \in S^2 \mid a \text{ divides } b\}$



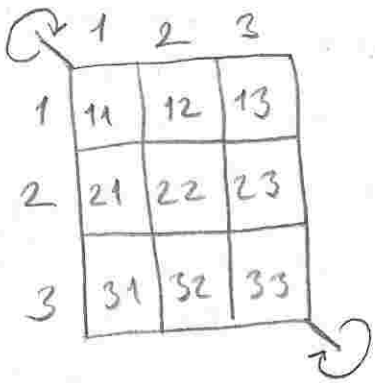
→ Matrix of an anti-symmetric relation

Let $M_R = [M_{ij}]$ be the matrix of an anti-symmetric relation

If $i \neq j$ and $M_{ij} \neq 0$ then $M_{ij} = 0$

Ex) Let $S = \{1,2,3,4\}$ and let R be relation of elements in S; $R = \{(a,b) \in S^2 \mid a \text{ divides } b\}$

$$M_R = \begin{matrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$



– LECTURE 9.109 –

Outline
- Transitive relations
- Digraph of transitive relations

→ Definition of transitivity

A relation R on set S is called transitive iff:
$\forall a, b, c \in S$, if (aRb and bRc) then aRc

Ex) Are the following relations transitive?

$R = \{(x,y) \in \mathbb{N}^2 \mid x \leq y\}$

- Yes, it is transitive as $\forall x, y, z \in \mathbb{N}$ if $x \leq y$ and $y \leq z$ then $x \leq z$

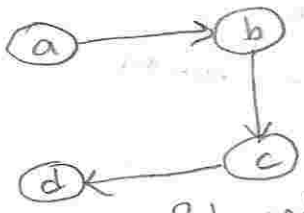
$R = \{(2,3), (3,2), (2,2)\}$

- No, it is not transitive because $3R2$ and $2R3$ but $3R3$

$R = \{(a,b) \mid a$ is an ancestor of $b\}$

- Yes, it is transitive because if a is an ancestor of b and b is an ancestor of c, then a is an ancestor of c

Ex) Let $S = \{a, b, c, d\}$ and let R be a relation on S represented by the following digraph

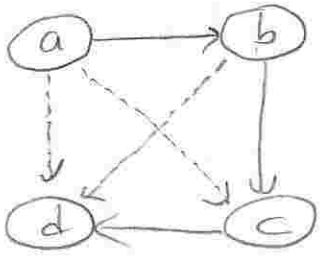


Not transitive as: aRb and bRc, however $a \not R c$
or
bRc and cRd, however $b \not R d$

→ Transitive closure of a relation

What is the minimum number of edges that need to be added to make this relation transitive?

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



$R_{enhanced} = \{(a,b), (b,c), (a,c), (c,d), (b,d), (a,d)\}$

(transitive closure of R)

- LECTURE 9.20 -

Outline
- Definition of equivalence relations
- Examples of equivalence relations
- Definition of equivalence classes & examples

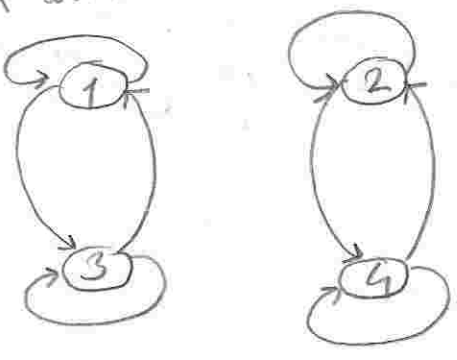
→ Definition of equivalence relation
- Let R be a relation of elements on a set S. R is an equivalence relation iff:
  R is reflexive, symmetric and transitive

Ex) - Let R be relation of element in $\mathbb{Z}$:
$R = \{(a,b) \in \mathbb{Z}^2 \mid a \bmod 2 = b \bmod 2\}$
- We have already proved that this relation is:
  - reflexive as aRa, $\forall a \in \mathbb{Z}$
  - symmetric as if aRb then bRa, $\forall a,b \in \mathbb{Z}$
  - transitive as if aRb and bRc then aRc, $\forall a,b,c \in \mathbb{Z}$
- R is an equivalence relation

Ex) - Let R be a relation of elements in $\mathbb{Z}$:
$R = \{(a,b) \in \mathbb{Z}^2 \mid a \leq b\}$
- We have already proved that this relation is:
  - reflexive as aRa for all a in $\mathbb{Z}$
  - transitive as if aRb and bRc then aRc, $\forall a,b,c \in \mathbb{Z}$
  - not symmetric as $2 \leq 3$ but $3 \not\leq 2$, $\forall a,b \in \mathbb{Z}$
- R is not an equivalence relation

→ Let R be an equivalence relation on a set S. Then, the equivalence class of $a \in S$ is:
the subset of S containing all the elements related to a through 'R'
$[a] = \{x : x \in S \text{ and } xRa\}$

Ex) - Let $S = \{1,2,3,4\}$ and R be a relation on elements in S:
$R = \{(a,b) \in S^2 \mid a \bmod 2 = b \bmod 2\}$
- R is an equivalence relation with 2 equivalence classes:
  - $[1] = [3] = \{1,3\}$
  - $[2] = [4] = \{2,4\}$



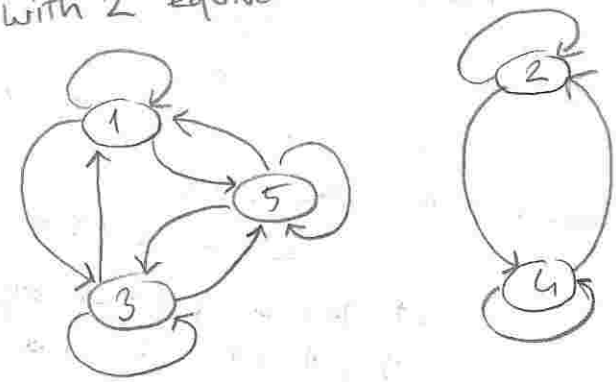
Ex) • Let $Z = \{1,2,3,4,5\}$ and R be a relation of elements in Z: $R = \{(a,b) \in Z^2 \mid a-b \text{ is an even number}\}$

$R = \{(1,1),(2,2),(3,3),(4,4),(5,5),(2,4),(4,2),(1,3),(3,1),$
$(1,5),(5,1),(3,5),(5,3)\}$

• R is an equivalence relation with 2 equivalence classes:

$[1] = [3] = [5] = \{1,3,5\}$

$[2] = [4] = \{2,4\}$



— LECTURE 9.203 —

Outline
- Definition of a partial order
- Definition of a total order

→ Definition of a partial order

Let R be a relation on elements in a set S.

R is a partial order iff:

R is reflexive, anti-symmetric and transitive

Ex) • Let R be a relation of elements in Z:

$R = \{(a,b) \in Z^2 \mid a \leq b\}$

• It can easily be proved that R is:
- reflexive as $a \leq a$, $\forall a \in Z$
- transitive as if $a \leq b$ and $b \leq c$ then $a \leq c$, $\forall a,b \in Z$
- anti-symmetric as if $a \leq b$ and $b \leq a$ then $a = b$, $\forall a,b \in Z$

• R is a partial order

Ex) • Let R be a relation of elements in Z^+:

$R = \{(a,b) \in Z^+ \mid a \text{ divides } b\}$

• It can easily be proved that R is:
- reflexive as a divides a, $\forall a \in Z^+$
- transitive as if a divides b and b divides c then a divides c, $\forall a,b,c \in Z^+$
- anti-symmetric as if a divides b and b divides a then $a = b$, $\forall a,b \in Z^+$

• R is a partial order

→ Definition of equivalence class

- Let R be an equivalence relation on a set S. Then, the equivalence class of $a \in S$ is:
- The subset of S containing all the elements related to a through 'R'.

$$[a] = \{x : x \in S \text{ and } x R a\}$$

Ex
- Let R be a relation of elements in $\mathbb{Z}$:

$$R = \{(a,b) \in \mathbb{Z}^2 \mid a \leq b\}$$

- It has been previously shown that R is a partial order
- Also, $\forall a, b \in \mathbb{Z}$, $a \leq b$ or $b \leq a$ is true
- R is a total order

Ex
- Let R be a relation on elements in $\mathbb{Z}^+$:

$$R = \{(a,b) \in \mathbb{Z}^+ \mid a \text{ divides } b\}$$

- It has been proved that R is a partial order
- $\mathbb{Z}^+$ contains elements that are incomparable, such as 5 and 7
- R is not totaly ordered

- LECTURE 10.103 -

Outline
- Product rule
- Addition rule
- Combining the sum and product rules
- Substraction rule
- Division rule

→ Product rule

To determine the number of different possible outcomes in a complex process, we can break the problem into a sequence of two independent tasks:
- if there are n ways of doing the first task
- for each of these ways of doing the first task, there are m ways of doing the second task
- then there are $n \cdot m$ different ways of doing the whole process

Ex) Let's consider a restaurant offering a combination meal where a person can order one from each of the following categories: 2 salads, 3 main dishes, 4 side dishes and 3 dessert. How many different combination meals are possible?

Sol) The problem can be broken down into 4 independent events:
- selecting a salad, selecting a main dish, selecting a side dish and selecting a dessert.

For each event, the number of available options is:
- 2 for the first event
- 3 for the second event
- 4 for the third event
- 3 for the fourth event

Thus, there are $2.3.4.3 = 72$ possible combination meals

→ Product rule in terms of sets
- Let A be the set of ways to do the first task and B the set of ways to do second task. If A and B are disjoint, then:
- The number ways to do both task 1 and 2 can be represented as $|A \times B| = |A| \cdot |B|$
- In other words, the number of elements Cartesian product of these sets is the product of the number of elements in each set.

→ Addition rule
- Suppose a task 1 can be done n ways and a task 2 can be done in m ways
- Assume that both tasks are independent, that is, performing task 1 doesn't mean performing task 2 and vice versa
- In this case, the number of ways of executing task 1 or task 2 is equal to $n+m$

Ex| • The computing department must choose either a student or a member of academic staff as a representative for a university committee
• How many ways of choosing this representative are there if there are 10 academic staff and 77 mathematics students and no one is both a member or academic staff and a student.

Sol| By the addition rule, there are 10+77 ways of choosing this representative

→ The sum rule in terms of sets

Let A be the set of ways to do task 1 and B the set of ways to do task 2, where A and B are disjoint sets
• The sum rule can be phrased in terms of sets
• $|A \cup B| = |A| + |B|$ as long as A & B are disjoint sets

→ Combining the sum and product rules

Combining the sum and product rules allows us to solve more complex problems.

Ex| • Suppose a label in a programming language can be either a single letter followed by two digits. What is the number of possible labels?

Sol| • The number of labels with one letter only is 26
• Using the product rule the number of labels with a letter followed by 2 digits is $26 \times 10 \times 10$
• Using the sum rule the total number of label is
$26 + 26.10.10 = 2626$

→ Substraction rule
• Suppose a task can be done either in one of n_1 ways or in one of n_2 ways
• Then the total number of ways to do the task is $n_1 + n_2$ minus the number of ways common to the two different ways

• This is also known as the principle of inclusion-exclusion

$$|A \cup B| = |A| + |B| - |A \cap B|$$

Ex) How many binary bit strings of length eight either start with a 1 bit or end with two bits 00

Sol) • Number of bit strings of length eight that start with a 1 bit: $2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 \cdot 2 = 2^7 = 128$

• Number of bit strings of length eight that end with the two bits 00: $2^6 = 64$

• Number of bit strings of length eight that start with a 1 bit and end with bits 00 is $2^5 = 32$

• Using the subtraction rule:

• The number of bit strings either starting with a 1 or ending with 00 is $128 + 64 - 32 = 160$

→ Division rule

• Suppose a task can be done using a procedure that can be carried out in n ways, and for every way w, exactly d of the n ways correspond to w. Then this task can done in n/d ways.

• In terms of sets: If the finite set A is the union of n pairwise disjoint subsets each with d elements, then $n = |A| / d$

• In terms of functions: if f is a function from A to B, where both are finite sets, and for every value $y \in B$ there are exactly d values $x \in A$ such that $f(x) = y$, then $|B| = |A| / d$

Ex) In how many ways can we seat 4 people around a table, where two seating arrangements are considered the same when each person has the same left and right neighbour?

Sol) Let's first number the seats around the table from 1 to 4 proceeding clockwise:

• There are four ways to select the person for seat 1, three for seat 2, two for seat 3, and one for seat 4

• Thus there are $4 \cdot 3 \cdot 2 \cdot 1 = 24$ ways to order the four people

- Since two seating arrangements are the same when each person has the same left and right neighbour, for every choice for seat 1, we get the same seating
- Therefore, by the division rule, there are $24/4 = 6$ different seating arrangements.

— LECTURE 10.10.05 —

Outline
- Pigeonhole principle
- The generalised pigeonhole principle

→ Pigeonhole principle

If k is a positive integer and $k+1$ objects are placed into k boxes, then at least one box contains two or more objects

Proof by contrapositive:
- Let's suppose none of the k boxes has more than one object
- Then the total number of objects would be at most k
- Which contradicts the statement that we have $k+1$ objects

Ex) If a flock of 10 pigeons roosts in a set of 9 pigeonholes, one of the pigeonholes must have more than 1 pigeon

Exercise) Prove that a function f from a set with $k+1$ elements to a set with k elements is not one-to-one

Sol:) We can prove this using the pigeonhole principle as follows:
- Create a boxy for each element y in the co-domain of f
- Put all of the elements x from the domain in the box for y such that $f(x) = y$
- Because there are $k+1$ elements and only k boxes, at least one box has two or more elements
- Hence, f can't be one-to-one

→ The generalised pigeonhole principle

If N objects are placed into k boxes, then there is at least one box containing at least $\lceil N/k \rceil$ objects, where $\lceil x \rceil$ is called the ceiling function, which represents the round-up value of x.

Let's prove it by contrapositive:

- Suppose that none of the boxes contains more than $\lceil N/k \rceil - 1$ objects
- Then the total number of objects is at most

$$k\left(\left\lceil \frac{N}{k} \right\rceil - 1\right) < k\left(\left(\frac{N}{k} + 1\right) - 1\right) = N$$

- This is a contradiction because there is a total of N objects

Ex) How many cards must be selected from a standard deck of 52 cards to guarantee that at least four cards of the same suit are chosen?

Sol) - We assume four boxes, one for each suit
- Using the generalised pigeonhole principle, at least one box contains at least $\lceil \frac{N}{4} \rceil$ cards, where N is the number of cards selected
- At least four cards of one suit are selected if $\lceil \frac{N}{4} \rceil \geq 4$
- The smallest integer N such that $\lceil \frac{N}{4} \rceil \geq 4$ is equal to 13

– LECTURE 10.107 –

Outline
- Definition of a permutation
- Number of permutations
- Definition of a combination
- Number of combinations

→ Definition of a permutation
- A permutation of a set of distinct objects is an ordered arrangement of these objects
- An ordered arrangement of r elements of a set is called an r-permutation
- The number of r-permutations of a set with n elements is denoted by $P(n,r)$

Ex) Let $S = \{1,2,3\}$

- The ordered arrangement 3,1,2 is a 3-permutation of S
- The ordered arrangement 3,2 is a 2-permutation of S
- The 2-permutations of $S = \{1,2,3\}$ are 1,2 ; 1,3 ; 2,1 ; 2,3 ; 3,1 ; and 3,2
- Hence, $P(3,2) = 6$

→ Number of permutations

If n is a positive integer and r is an integer with $r \leq n$, then there are $P(n,r) = n(n-1)(n-2)\cdots(n-(r-1))$ r-permutations of a set with n distinct elements.

$$P(n,r) = \frac{n!}{(n-r)!}$$

Proof:

- By the product rule
  - there are n different ways for choosing the 1st element
  - $n-1$ ways for choosing the 2nd element
  - $n-2$ ways for choosing the 3rd element, and so on
  - there are $(n-(r-1))$ ways to choose the last element
  - hence, $P(n,r) = n(n-1)(n-2)\cdots(n-(r-1))$
  - $P(n,0) = 1$, since there is only one way to order zero

Ex) How many possible ways are there of selecting a first-prize winner, a second-prize winner and a third-prize winner from 50 different people?

$$P(50,3) = 50 \cdot 49 \cdot 48 = 117.600$$

→ Definition of combinations

- An r-combination of elements of a set is an unordered selection of r elements from the set
- An r-combination is a subset of the set with r elements
- The number of r-combinations of a set with n distinct elements is denoted by $C(n,r) = \binom{n}{r}$
- The notation used is also called a binomial coefficient

→ Number of combinations

- The number of r-combinations of a set with n distinct elements can be formulated as:

$$C(n,r) = \frac{n!}{(n-r)!\,r!} = \frac{P(n,r)}{r!}$$

- $C(n,r)$ can be referred to as n choose r
- It follows that $C(n,r) = C(n,n-r)$

Ex | How many ways are there of selecting six players from a 20-member tennis team to make a trip to an international competition?

$$C(20,6) = \frac{20!}{6!\cdot 14!} = \frac{20\cdot 19\cdot 18\cdot 17\cdot 16\cdot 15}{6\cdot 5\cdot 4\cdot 3\cdot 2} = 38760$$

— LECTURE 10.201 —

Outline

- Binomial theorem
- Application of the binomial theorem
- Pascal's identity
- Pascal's triangle

→ Binomial expression

An expression consisting of two terms, connected by a + or − sign is called a binomial expression

Ex | $x+a$; $2x-y$; x^2-y^2; $2x-3y$, ...

→ Binomial theorem

$(x+1)^1 = x+y$

$(x+y)^2 = x^2+2xy+y^2$

$(x+y)^3 = x^3+3x^2y+3xy^2+y^3$

$\vdots$

$(x+y)^{30}$

Let x and y be variables, and n a non-negative integer

The expansion of $(x+y)^n$ can be formalised as:

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}$$

Ex) What is the coefficient of x^8y^7 in the expansion of $(3x-y)^{15}$?

Sol)
- We can view the expression as $(3x+(-y))^{15}$
- By the binomial theorem:

$$(3x+(-y))^{15} = \sum_{k=0}^{15} \binom{15}{k}(3x)^k(-y)^{n-k}$$

- Consequently, the coeffient of x^8y^7 in the expansion is obtained when $k=8$:

$$\binom{15}{8}(3)^8(-1)^7 = -3^8\frac{15!}{8!\,7!}$$

→ Application of the binomial theorem

Let's prove the identity $2^n = \sum_{k=0}^{n}\binom{n}{k}$

Using binomial theorem:
- With $x=1$ and $y=1$, from the binomial theorem we see that the identity is verified

Using sets:
- Consider the subsets of a set with n elements
- There are subsets with zero elements, with one element, with two elements and so on... with n elements
- Therefore the total number of subsets is $\sum_{k=0}^{n}\binom{n}{k}$
- Also, since we know that a set with n elements has 2^n subsets, we can conclude that:

$$2^n = \sum_{k=0}^{n}\binom{n}{k}$$

→ Pascal's identity

If n & k are integers with $n \geq k \geq 1$, then:

$$\binom{n}{k}+\binom{n}{k-1}=\binom{n+1}{k}$$

Proof:
- Let T be a set where $|T| = n+1$, $a \in T$, and $S = T-\{a\}$
- There are $\binom{n+1}{k}$ subsets of T containing k elements. Each of these subsets either:
  - contains a with $k-1$ other elements, or
  - contains k elements of S and not a

- There are:
  - $\binom{n}{k-1}$ subsets of k elements that contain a
  - $\binom{n}{k}$ subsets of k elements of T that don't contain a
- Hence $\binom{n+1}{k} = \binom{n}{k-1} + \binom{n}{k}$

→ Pascal's triangle

Pascal's triangle is a number triangle with numbers arranged in staggered rows such that $a_{n,r}$ is the binomial coefficient $\binom{n}{r}$:

Ex) Using the binomial theorem, which one of the following is the correct expansion of $(2x-y)^7$?

$$\sum_{k=0}^{7} \binom{7}{k} (2x)^{7-k} (-y)^k$$

— LECTURE 10.2.4 —

Outline
- Permutations with repetition
- Permutations without repetition
- Combination with repetition
- Combination without repetition
- Choice of formulas

→ Permutations with repetition

The number of r-permutations of a set of n objects with repetition allowed is n^r.

Proof:
- Since we have n choices each time, there are n possibilities for the 1st choice, n possibilities for the 2nd choice, ..., and n possibilities when choosing the last number
- By the product rule, multiplying each time:

$$\underbrace{n \times n \times n \times n \times \cdots \times n}_{r \text{ times}} = n^r$$

Ex) How many strings of length r can be formed if we are using only uppercase letters in the English alphabet?

Sol) The number of such a strings is 26^r, which is the number of r-permutations with repetition of a set with 26 elements

→ Permutations without repetition

In the case of permutations without repetition, we reduce the number of available choices each time by 1. The number of r-permutations of a set with n objects without repetition is:

$$\underbrace{n \times (n-1) \times (n-2) \times \dots \times (n-r+1)}_{r \text{ times}}$$

$$P(n,r) = P_n^r = n(n-1)(n-2)\dots(n-r+1) = \frac{n!}{(n-r)!}$$

Ex] During a running competition how many different ways can the first and the second place be awarded if 10 runners are taking part in the race?

Sol] $$P(10,2) = P_{10}^2 = \frac{10!}{(10-2)!} = \frac{10!}{8!} = 90$$

→ Combination with repetition

The number of ways in which k objects can be selected from n categories of objects, with repetition permitted, can be calculated as: $\binom{k+n-1}{k} = \frac{(k+n-1)!}{k!\,(n-1)!}$

- It is also the total number of ways to put k identical balls into n distinct boxes
- It is also the total number of functions from a set of k identical elements to a set of n distinct elements

Ex] Let's find all multisets of size 3 from the set $\{1,2,3,4\}$

Sol] • Using bar and crosses, think of the values 1, 2, 3, 4 as four categories
• We will denote each multiset of size 3 by placing three crosses in the various categories
• For instance, the multiset $\{1,1,3\}$ is represented by $\times\times||\times|$
• This counting problem can be modelled as distributing the 3 crosses among the 3+4-1 positions, the remaining positions being occupied by bars
• Thus the number of multisets of size 3 is:

$$C(6,3) = \frac{6!}{3!\,3!} = 20$$

→ Combination without repetition

- The number of ways in which r objects can be selected from n categories of objects with repetition not permitted can be calculated as:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

- This counting problem is the same as the number of ways of putting k identical balls into n distinct boxes, where each box receives at most one ball
- It is also the number of one-to-one functions from a set of k identical elements into a set of n distinct elements
- It is also the number of k-element subsets of an n-element set

→ Choice of formulas

- We have discussed four different ways of selecting k objects from a set with n elements:
  - the order in which the choices are made may or may not matter
  - repetition may or may not be allowed
- The following table summarises the formula in each case

	Order matters	Order does not matter
Repetition is not permitted	$\frac{n!}{(n-k)!}$	$\frac{n!}{k!(n-k)!}$
Repetition is permitted	n^k	$\frac{(k+n-1)!}{k!(n-1)!}$

Ex) John is the chair of a committee. In how many ways can a committee of 3 be chosen from 10 people, given that John must be one of the people selected?

Sol) • Since John is already chosen, we need to choose another 2 out of 9 people
• In choosing a committee, the order doesn't matter, so we need to apply the combinations without repetition formula:

$$C(9,2) = \frac{9!}{2!(9-2)!} = 36 \text{ ways}$$

- LECTURE 10.206 -

Outline
- Distributing objects into boxes
- Distinguishable objects into distinguishable boxes with or without exclusion
- Indistinguishable objects into distinguishable boxes with or without exclusion

→ Distributing objects into boxes

Counting problems can be phrased in terms of distributing k objects into n boxes under various conditions:
- The objects can be either distinguishable or indistinguishable
- The boxes can be either distinguishable or indistinguishable
- The distribution can be done either with exclusion or without exclusion

→ Distinguishable objects and distinguishable boxes with exclusion

In this case, we want to distribute k balls, numbered from 1 to k, into n boxes, numbered from 1 to n, in such a way that no box receives more than one ball

This is equivalent to making an ordered selection of k boxes from n boxes, where the balls do the selecting for us:
- the ball labelled 1 chooses the first box
- the ball labelled 2 chooses the second box ... and so on

- Theorem:

Distributing k distinguishable balls into n distinguishable boxes, with exclusion, is equivalent to forming a permutation of size k from a set of size n

Therefore, the number of ways of placing k distinguishable balls into n distinguishable boxes is as follows:

$$P(n,k) = n(n-1)(n-2)\dots(n-k+1) = \frac{n!}{(n-k)!}$$

→ Distinguishable objects and distinguishable boxes without exclusion

In this case, we want to distribute k balls, numbered from 1 to k, into n boxes, numbered from 1 through n, without restrictions on the number of balls in each box

This is equivalent to making an ordered selection of k boxes from n, with repetition, where the balls do the selecting for us:

- the ball labelled 1 chooses the first box

- Theorem:

Distributing k distinguishable balls into n distinguishable boxes, without exclusion, is equivalent to forming a permutation of size k from a set of size n, with repetition.

Therefore, there are:

$$n^k \text{ different ways}$$

→ Indistinguishable objects and distinguishable boxes with exclusion

In this case, we want to distribute k balls, into n boxes, numbered from 1 through n, in such a way that no box receives more than 1 ball

- Theorem:

Distributing k indistinguishable balls into n distinguishable boxes, with exclusion, is equivalent to forming a combination of size k from a set of size n.

Therefore, there are $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ different ways

→ Indistinguishable objects and distinguishable boxes without exclusion

In this case, we want to distribute k balls, into n boxes, numbered from 1 through n, without restrictions on the number of balls in each box

- Theorem:

Distributing k indistinguishable balls into n distinguishable boxes, without exclusion, is equivalent to forming a combination of size k from a set of size n, with repetition.

Therefore, there are

$$\binom{n+k-1}{k} = \frac{(n+k-1)!}{k!(n-1)!} \text{ different ways}$$

Ex) How many ways are there of placing 8 indistinguishable balls into 6 distinguishable boxes?

$$\binom{8+6-1}{8} = \binom{13}{8} = \frac{13!}{8!5!} = 1287$$

reflexive

$\forall x \; xRx$ ✗

Symmetric

$\forall x \forall y \; xRy \rightarrow yRx$

Transitive

$\forall x \forall y \forall z$

$xRy \wedge yRz \rightarrow xRz$

$p \rightarrow q = \bar{p} \vee q$

T T F T

$p \leftrightarrow q = p \rightarrow q \wedge q \rightarrow p$

T F F T

$(A - B) = (A \cap \bar{B})$

$|S|$ = # of elements = 5

$|P(S)| = 2^{|A|} = 32$

$\wedge$ = conjunction

$\vee$ = disjunction

$\Rightarrow$ = implication

$\Leftarrow$ = reduction

$\Leftrightarrow$ = equivalence

---

k ordering not important $= \frac{n!}{k!(n-k)!}$

---

in simple graph, the total number of degree sequence should be even

---

Hamiltonian path

vertex in G exactly once

$p \oplus q = (p \cup q) - (p \cap q)$

F T T F

Conditional = $p \rightarrow q$

Converse = $q \rightarrow p$

Inverse = $\neg p \rightarrow \neg q$

Contr. post. = $\neg q \rightarrow \neg p$

AND = $\cdot$

OR = $+$

---

f is injective if for all $a, b \in R$, if $a \neq b$, then $f(a) \neq f(b)$

---

f is surjective for all $y \in R$, there exists $x \in R$, such that $f(x) = y$

$\log_a b^c = c \log_a b$

Simple graph: A simple graph has no loops, no parallel edges and its undirected

---

max vertex for each edge should be $n-1$