MATH1081 Revision Sheet

UNSW Mathematics Society: Isaiah Iliffe, Joanna Lin, John Kameas

We would like to preface this document by saying that this resource is first and foremost meant to be used as a reference and should NOT be used as a replacement for the course resources or lecture recordings. Said resources provided on moodle are wonderfully written and contain an abundance of fully worked solutions and in depth explanations. Studying for this course using *only* this revision sheet would not be sufficient.

In addition, although the authors have tried their best to include everything essential taught in the course, it was ultimately up to their discretion on whether or not to include results/theorems/definitions etc. Anything that is missing is most definitely a conscious choice made by the authors.

Finally, any and all errors found within this document are most certainly our own. If you have found an error, please contact us via our Facebook page, or give us an email.

Sets, Functions, and Sequences

Sets

Definition 0.1 (Set)**.** A set is a well-defined collection of distinct objects.

In MATH1081, this definition is sufficient, but the unsatisfied reader may refer to ZFC.

- An *element* of a set is any object in the set.
  - $\in$: "belongs to" or "is an element of".
  - $\notin$: "does not belong to" or "is not an element of".
- A finite set can be specified by listing its elements between braces. For example, $\{1, 2, 3\}$ is a set.
- A set can be specified with set-builder notation by specifying a property that its elements must satisfy. For example,

  $$\{x \in \mathbb{R} \mid x^3 - x > 0\}$$

  is the set of real numbers x such that $x^3 - x > 0$.
- The *cardinality* of a finite set S, denoted by $|S|$, is the number of elements in S.

Common Sets

The most commonly used sets are

- *Positive integers* $\mathbb{Z}^+ = \{1, 2, 3, \ldots\}$.
- *Natural numbers* $\mathbb{N} = \{0, 1, 2, \ldots\}$.
- *Integers* $\mathbb{Z} = \{-2, -1, 0, 1, 2, \ldots\}$.
- *Rational numbers* $\mathbb{Q} = \{\frac{p}{q} \mid p, q \in \mathbb{Z}, q \neq 0\}$.
- *Real numbers* $\mathbb{R}$ = metric completion of $\mathbb{Q}$. You don't need to know or understand this definition; just treat $\mathbb{R}$ as a space large enough so that calculus works 'intuitively'.
- *Complex numbers* $\mathbb{C}$ = algebraic completion of $\mathbb{R}$. You don't need to know or understand this definition; just treat $\mathbb{C}$ as a space large enough so that every n-degree polynomial, has n solutions.

Relations between sets

Two sets S and T are *equal*, denoted by $S = T$, if

- every element of S is also an element of T, and
- every element of T is also an element of S.

That is, when they have precisely the same elements.

The *empty set*, denoted by $\emptyset$, is a set which has no elements.

A *subset* of a set is a part of the set.

- $\subseteq$ - "is a subset of".
- $\not\subseteq$ - "is not a subset of".

A set S is a *subset* of a set T if each element of S is also an element of T.

- For any set S, we have $\emptyset \subseteq S$ and $S \subseteq S$.
- $S = T$ if and only if $S \subseteq T$ and $T \subseteq S$.

A set S is a *proper subset* of a set T if S is a subset of T and $S \neq T$.

- $\emptyset$ is a proper subset of any non-empty set.
- Any non-empty set is an improper subset of itself.

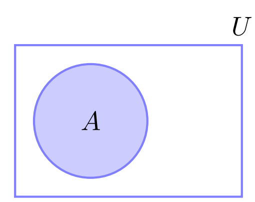
Hints for proofs:

- To prove that $S \subseteq T$, we can assume that $x \in S$ and show that $x \in T$.
- To prove that $S = T$, we can show that $S \subseteq T$ and $T \subseteq S$.

Operations on sets

It is often convenient to work inside a specified *universal set*, denoted by U, which is assumed to contain everything that is relevant.

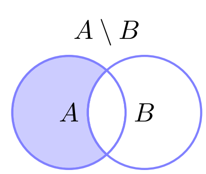
Venn diagrams are visualisations of sets as regions in the plane. For instance, here is a Venn diagram of a universal set U containing a set A:



Set operations and set algebra:

- *difference* ($-$, $\setminus$) - "but not"

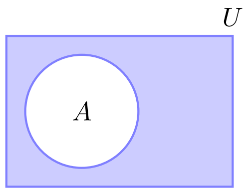
  $$A - B = A \setminus B = \{x \in U \mid x \in A \text{ and } x \notin B\}$$



- *complement* (^{c}, $\overline{}$) - "not"

$$A^c = \overline{A} = U \setminus A = \{x \in U \,|\, x \notin A\}$$

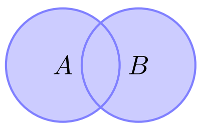
A^c



- *union* ($\cup$) - "or"

$$A \cup B = \{x \in U \,|\, x \in A \text{ or } x \in B\}$$

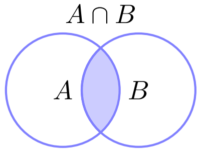
$A \cup B$



- *intersection* ($\cap$) - "and"

$$A \cap B = \{x \in U \,|\, x \in A \text{ and } x \in B\}$$

$A \cap B$



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

Generalised set operations:

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

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

Laws of set algebra

- Commutative laws:

$$A \cap B = B \cap A \quad \text{and} \quad A \cup B = B \cup A.$$

- Associative laws:

$$A \cap (B \cap C) = (A \cap B) \cap C \text{ and}$$
$$A \cup (B \cup C) = (A \cup B) \cup C.$$

- Distributive laws:

$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C) \text{ and}$$
$$A \cup (B \cap C) = (A \cup B) \cap (A \cup C).$$

- Absorption laws:

$$A \cap (A \cup B) = A \quad \text{and} \quad A \cup (A \cap B) = A.$$

- Identity laws:

$$A \cap U = U \cap A = A \text{ and } A \cup \emptyset = \emptyset \cup A = A.$$

- Idempotent laws:

$$A \cap A = A \quad \text{and} \quad A \cup A = A.$$

- Double complement law:

$$(A^c)^c = A.$$

- Difference law:

$$A - B = A \cap B^c.$$

- Domination or universal bound laws:

$$A \cap \emptyset = \emptyset \cap A = \emptyset \text{ and } A \cup U = U \cup A = U.$$

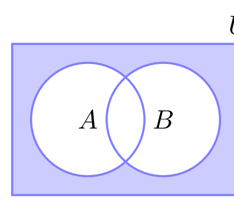
- Intersection and union with complement:

$$A \cap A^c = A^c \cap A = \emptyset \text{ and } A \cup A^c = A^c \cup A = U.$$

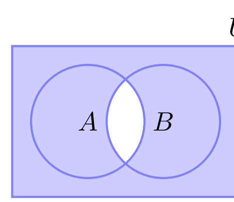
- *De Morgan's Laws*:

$$(A \cup B)^c = A^c \cap B^c \text{ and } (A \cap B)^c = A^c \cup B^c.$$

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



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



- For a set expression involving unions, intersections and complements, its *dual* is obtained by replacing $\cap$ with $\cup$, $\cup$ with $\cap$, $\emptyset$ with U, and U with $\emptyset$. The laws of set algebra mostly come in dual pairs.

Power sets

The *power set* $P(S)$ of a set S is the set of all subsets of S.

- For any set S, we have $\emptyset \in P(S)$ and $S \in P(S)$.
- The number of subsets of S is $|P(S)| = 2^{|S|}$.

Ordered collections

An *ordered pair* is a collection of two objects in a specified order. We use round brackets to denote ordered pairs; e.g., (a, b) is an ordered pair. Note that (a, b) and (b, a) are different ordered pairs (if $a \neq b$), whereas $\{a, b\}$ and $\{b, a\}$ are the same set.

An *ordered n-tuple* is a collection of n objects in a specified order; e.g., $(a_1, a_2, \ldots, a_n)$ is an ordered n-tuple. Two ordered n-tuples $(a_1, a_2, \ldots, a_n)$ and $(b_1, b_2, \ldots, b_n)$ are equal if and only if $a_i = b_i$ for all $i = 1, 2, \ldots, n$.

The *Cartesian product* of two sets A and B, denoted by $A \times B$, is the set of all ordered pairs from A to B:

$$A \times B = \{(a, b) | a \in A \text{ and } b \in B\}$$

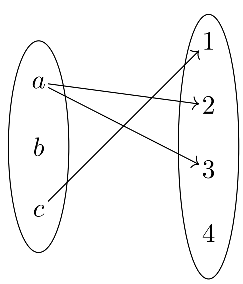
If $|A| = m$ and $|B| = n$, then we have $|A \times B| = mn$.

The Cartesian product of n sets $A_1, A_2, \ldots, A_n$ is the set of all ordered n-tuples $(a_1, a_2, \ldots, a_n)$ such that $a_i \in A_i$ for all $i = 1, 2, \ldots, n$:

$$\begin{aligned} &A_1 \times A_2 \times \cdots \times A_n \\ &= \{(a_1, a_2, \ldots, a_n) | a_i \in A_i \text{ for all } i = 1, 2, \ldots, n\} \end{aligned}$$

When X and Y are small finite sets, we can use an *arrow diagram* to represent a subset S of $X \times Y$: we list the elements of X and the elements of Y, and then we draw an arrow from x to y for each pair $(x, y) \in S$.

For example, below is the arrow diagram for S if $X = \{a, b, c\}$ and $Y = \{1, 2, 3, 4\}$ and $S = \{(a, 2), (a, 3), (c, 1)\}$.



Functions

Definition 0.2 (Function)**.** A *function* f from set X to a set Y is a subset of $X \times Y$ such that for every $x \in X$, there is exactly one $y \in Y$ for which (x, y) belongs to f.

Domain, Codomain, Range and Graph

- We write $f : X \to Y$, where X is the called the *domain* of f and Y is the *codomain* of f.
- If $(x, y) \in f$, we write $f(x) = y$ or $f : x \mapsto y$, where the latter is read as "x maps to y under f".
- The *range* of f is a subset of the codomain, given by
$$\{y \in Y \mid y = f(x) \text{ for some } x \in X\}.$$
- The graph of a function is the set of all points $(x, y) \in f$.

Ceiling and Floor

For any $x \in \mathbb{R}$,

- the *floor* function, which we denote $\lfloor x \rfloor$, outputs the largest integer less than or equal to x;
- the *ceiling* function, which we denote $\lceil x \rceil$, outputs smallest integer greater than or equal to x.

Image and Inverse Image

- The *image* of a set $A \subseteq X$ under a function $f : X \to Y$ is denoted by
$$f(A) = \{y \in Y \mid y = f(x) \text{ for some } x \in A\}.$$
- The *inverse image* or *preimage* of a set $B \subseteq Y$ under a function $f : X \to Y$ is denoted by
$$f^{-1}(B) = \{x \in X \mid f(x) \in B\}.$$

Injective, Surjective and Bijective

A function $f : X \to Y$ is

- *injective* or *one-to-one* if for every $y \in Y$, there is at most one $x \in X$ such that $f(x) = y$.

  To prove that a function f is injective, we can prove that if $f(x_1) = f(x_2)$, then $x_1 = x_2$.
- *surjective* or *onto* if for every $y \in Y$, there is at least one $x \in X$ such that $f(x) = y$. In other words, the range is the same as the codomain.
- *bijective* if it is both *injective* and *surjective.* That is, for every $y \in Y$, there is exactly one $x \in X$ such that $f(x) = y$.

Composition

For functions $f : X \to Y$ and $g : Y \to Z$, the *composite* of f and g is the function $g \circ f : X \to Z$, defined by $(g \circ f)(x) = g(f(x))$ for all $x \in X$.

- The composite function exists if and only if the range of f is a subset of the domain of g.
- Function composition is not commutative in general. That is $(g \circ f) \neq (f \circ g)$.
- Function composition (assuming it exists) is associative. That is $h \circ (g \circ f) = (h \circ g) \circ f$.

Identity and Inverse

- The *identity* function on a set X is the function $i_X : X \to X$, $i_X(x) = x$.
- A function $g : Y \to X$ is an *inverse* of $f : X \to Y$ if $g(f(x)) = x$ for all $x \in X$ and $f(g(y)) = y$ for all $y \in Y$. That is, $g \circ f = i_X$ and $f \circ g = i_Y$.
- If a function $f : X \to Y$ has an inverse, then f is invertible, and the inverse is denoted as f^{-1}.

Some theorems on inverses and invertibility:

- A function can have at most one inverse.
- A function is invertible if and only if it is bijective.
- if $f : X \to Y$ and $g : Y \to Z$ are invertible, then so is $g \circ f : X \to Z$. The inverse of $g \circ f$ is $f^{-1} \circ g^{-1}$.

Sequences

Definition 0.3 (Sequence)**.** A *sequence* $a_0, a_1, \dots, a_k, \dots$ is an ordered list of objects, where each object a_k is called a *term* and the subscript k is called an *index*. The sequence is denoted by $\{a_k\}$, or $\{a_k\}_{k=1}^{\infty}$.

Arithmetic and Geometric Progressions

Two common sequences are arithmetic and geometric progressions.

- An *arithmetic progression* is a sequence $\{b_k\}$ where $b_k = a + kd$ for all $k = 0, 1, \dots$, for some fixed numbers $a \in \mathbb{R}$ and $d \in \mathbb{R}$, written as

  $$a, a+d, a+2d, \dots, a+kd, \dots.$$

  That is, the difference across all consecutive pairs of terms is fixed, and the starting term is a.

- A *geometric progression* is a sequence $\{c_k\}$ defined by $c_k = ar^k$ for all $k = 0, 1, \dots$ for some fixed numbers $a \in \mathbb{R}$ and $r \in \mathbb{R}$, written as

  $$a, ar, ar^2, \dots, ar^k, \dots.$$

  That is, the ratio across all consecutive pairs is fixed, and the starting term is a.

Summation Notation

For $m \leq n$,

$$\sum_{k=m}^{n} a_k = a_m + a_{m+1} + a_{m+2} + \cdots + a_n$$

Summation satisfies the properties of a linear transformation. That is,

$$\sum_{k=m}^{n} (a_k + b_k) = \sum_{k=m}^{n} a_k + \sum_{k=m}^{n} b_k$$

and

$$\sum_{k=m}^{n} (\lambda a_k) = \lambda \sum_{k=m}^{n} a_k.$$

However, note that, in general

$$\sum_{k=m}^{n} (a_k b_k) \neq \left(\sum_{k=m}^{n} a_k\right)\left(\sum_{k=m}^{n} b_k\right).$$

Product Notation

For $m \leq n$,

$$\prod_{k=m}^{n} a_k = a_m \cdot a_{m+1} \cdot a_{m+2} \cdots a_n$$

We can write

$$\prod_{k=m}^{n} (a_k b_k) = \left(\prod_{k=m}^{n} a_k\right)\left(\prod_{k=m}^{n} b_k\right).$$

However, note that, in general

$$\prod_{k=m}^{n} (a_k + b_k) \neq \prod_{k=m}^{n} a_k + \prod_{k=m}^{n} b_k.$$

Integers, Modular Arithmetic and Relations

Integers

Tip: For rational numbers, a condition we also often impose on p and q is that their greatest common divisor is 1. That is, $\gcd(p, q) = 1$.

Divisibility Notation

For integers a and b, if there exists an integer m such that $b = am$, then we can write $a \mid b$, read as a divides b. Note m does not have to be unique.
If such an integer m does not exist, then we write $a \nmid b$.

Properties of Divisibility

Suppose a, b and c are integers.

- For any integer a, $a \mid 0$ is trivially true.
- If $a > 0$ and $b > 0$ and $a \mid b$, then we must have $a \leq b$.
- $0 \mid b$ is true only when $b = 0$.
- If $a \mid b$, then $a \mid bc$.
- If $a \mid b$, then $a \mid (sb + tc)$ for all integers s and t. A corollary is that if $a \mid b$, then $a \mid (b + c)$.
- If $a \mid b$ and $b \mid c$, then $a \mid c$. This property is called *transitivity*.

Prime and Composite Numbers

An integer $n > 1$ is *prime* if and only if its only positive factors are 1 and itself.

An integer $n > 1$ is *composite* if and only if it is not prime.

The number 1 is neither prime nor composite.

Greatest Common Divisor

For integers a and b, not both zero, their ***g**reatest **c**ommon **d**ivisor*, denoted as $\gcd(a, b)$, is the largest integer d such that $d \mid a$ and $d \mid b$.

If $\gcd(a, b) = 1$, then we call a and b *coprime* or *relatively prime*.

Least Common Multiple

For non-zero integers a and b, their ***l**east **c**ommon **m**ultiple*, denoted as $\operatorname{lcm}(a, b)$, is the smallest positive integer m such that $a \mid m$ and $b \mid m$.

The least common multiple can be determined from the greatest common divisor using the fact that

$$\operatorname{lcm}(a, b) = \frac{|ab|}{\gcd(a, b)}.$$

Fundamental Theorem of Arithmetic

The Fundamental Theorem of Arithmetic states that every positive integer has a unique prime factorisation, apart from the order of the prime factors.

Property of Prime/Composite Numbers

If n is composite, then n has a prime factor less than or equal to $\sqrt{n}$. We use the contrapositive – that if n has no prime factor less than or equal to $\sqrt{n}$, then it is prime – to determine whether a number is prime.

Preliminary Theorem

The following theorem helps us understand why Euclid's algorithm works.

Let a, b, q and r be integers such that $a = qb + r$, where a and b are non-zero. Then

$$\gcd(a, b) = \gcd(b, r).$$

Euclid's Algorithm

This algorithm is used to find the greatest common divisor of two integers. Given two integers a and b where $a \geq b$, we write $a = bq + r$, where q is the quotient and r is the remainder upon division of a by b. We recursively repeat this process on the divisor and remainder until the remainder is 0. The last divisor is the greatest common divisor of a and b.

For example, suppose $a = 16758$ and $b = 14175$. Then we use Euclid's algorithm in the following way.

$$\begin{aligned}
16758 &= 1 \times 14175 + 2583, \\
14175 &= 5 \times 2583 + 1260, \\
2583 &= 2 \times 1260 + 63, \\
1260 &= 20 \times 63 + 0.
\end{aligned}$$

We can deduce from the above theorem that $\gcd(16758, 14175) = \gcd(1260, 63) = 63$.

Bézout Property

Consider the equation $ax + by = c$, where a, b and c are integers, with a and b not both zero. Then the equation has integer solutions of x and y if and only if c is a multiple of $\gcd(a, b)$.

Extended Euclid's Algorithm

This algorithm is used to find an integer solution of x and y to the equation $ax + by = \gcd(a, b)$, where a and b are integers.

To do this algorithm, we perform Euclid's algorithm on the integers a and b, then 'undo' the algorithm by replacing the remainder of every equation with with sums of multiples of the corresponding divisor and dividend in a preceding equation.

For example, as with $a = 16758$ and $b = 14175$, from the computations in the previous section, we can write

$$\begin{aligned}63 &= 2583 - 2 \times 1260 \\ &= 2583 - 2(14175 - 5 \times 2583) \\ &= 11 \times 2583 - 2 \times 14175 \\ &= 11 \times (16758 - 1 \times 14175) - 2 \times 14175 \\ &= 11 \times 16758 - 13 \times 14175.\end{aligned}$$

We thus obtain the integer solution $x = 11$, $y = -13$. **Tip:** Perform only one substitution (namely, for the smaller remainder) at each step, then collect like terms, before performing another substitution.

Modular Arithmetic

Definition 0.4 (mod)**.** Recall that, if a is an integer and m is a positive integer, then there exist unique integers q and r, such that $a = qm+r$ and $0 \le r < m$.

We define $a \mod m$ to be this remainder r.

Congruence

Definition 0.5 (Congruence)**.** Two integers a and b are said to be *congruent modulo* m, denoted by $a \equiv b \pmod m$, if $(a \mod m) = (b \mod m)$.

Other equivalent definitions of congruence include

- $m \mid (a - b)$
- $a = b + km$ for some integer k.

Properties of congruence: If $a \equiv b \pmod m$ and $c \equiv d \pmod m$, then

- $a + c \equiv b + d \pmod m$
- $a - c \equiv b - d \pmod m$
- $ac \equiv bd \pmod m$
- $a^n \equiv b^n \pmod m$ for all $n \ge 0$.
- $la \equiv lb \pmod m$ for all integers l
- $a \equiv b \pmod n$ for all integers n satisfying $n \mid m$

Inverses

Definition 0.6 (Inverse)**.** Let m be a positive integer and $a, b \in \mathbb{Z}$ be such that $ab \equiv 1 \pmod m$. Then

- a, b are *inverses* modulo m.
- b is an *inverse* of a modulo m, and vice versa.

It can be shown that the inverse of any element, if it exists, is unique modulo m.

Solving Linear Congruences

For integers a, b and positive integer m, the aim is to find all integers x so that $ax \equiv b \pmod m$.

We are essentially finding integer solutions of x and y to the equation $ax + my = b$, except we disregard the value of y.

Existence of Solutions

Solutions of x do not necessarily exist when trying to solve linear congruence. The number of solutions of x depends on $\gcd(a, m)$.

- If $\gcd(a, m)$ is not a factor of b, then the congruence has no solution.
- Otherwise, the congruence has $\gcd(a, m)$ solutions.

Method for Solving Linear Congruence

When solving linear congruences, we often make use of the following theorems

- If $\gcd(a, m) = 1$, then $ax \equiv b \pmod m$ has the solution $x \equiv cb \pmod m$ where c is an inverse of a modulo m.
- If $c \ne 0$, then $ax \equiv b \pmod m$ and $cax \equiv cb \pmod{cm}$ have the same solutions.

Suppose we want to solve the linear congruence

$$52x \equiv 8 \pmod{60}.$$

1. Decide whether there are solutions to the equation by checking whether $\gcd(52, 60)$ divides 8. Here $\gcd(52, 60) = 4$ does divide 8, so we can proceed with finding solutions.
2. We divide the equation by $\gcd(52, 8) = 4$, leading to the congruence equation
$$13x \equiv 2 \pmod{15}. \tag{1}$$
3. Note that $\gcd(13, 15) = 1$, so we use the Extended Euclidean Algorithm to find the inverse of 13 modulo 15.
$$\begin{aligned} 15 &= 1 \times 13 + 2 & \qquad 1 &= 13 - 6 \times 2 \\ 13 &= 6 \times 2 + 1 & &= 13 - 6 \times (15 - 13) \\ 2 &= 2 \times 1 + 0 & &= 7 \times 13 - 6 \times 15 \end{aligned}$$
Notice that $7 \times 13 \equiv 1 \pmod{15}$. Thus, the inverse of 13 modulo 15 is 7 and so a solution to (1) is $x = 7 \times 2 = 14$.
4. We can write the solutions in terms of the original modulus: $x \equiv 14, 29, 44, 59 \mod 60$. There are exactly $\gcd(52, 60) = 4$ solutions.

Shortcuts

We can sometimes solve congruences using the fact that, given $\gcd(c, m) = 1$ then

$$cp \equiv cq \pmod{m} \iff p \equiv q \pmod{m}.$$

For example, given the equation $13x \equiv 1 \pmod{15}$, we can rewrite it as $-2x \equiv -14 \pmod{15}$, which means by the above fact that $x \equiv 7 \pmod{15}$.

Relations

Definition 0.7 (Relation)**.** A *relation* R from a set A to a set B is a subset of $A \times B$.

- If $(a, b) \in R$, we say that "a is related to b (by R", and we write $a \; R \; b$.
- If $(a, b) \in R$, we write $a \not R \; b$.

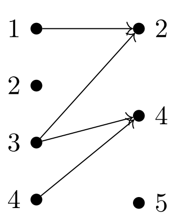
For finite sets A and B, we can represent the relation $R \subseteq A \times B$ on finite sets A and B using

- *arrow diagrams*, by listing the elements of A then listing the elements of B, then drawing an arrow from a to b for all pairs $(a, b) \in R$;
- *matrix*, by arranging the elements in some order $a_1, a_2, \ldots$ and $b_1, b_2, \ldots$, and constructing an $|A|$ by $|B|$ matrix M_R such that

$$[M_R]_{ij} = \begin{cases} 1 & \text{if } a_i \; R \; b_j; \\ 0 & \text{if } a_i \not R \; b_j. \end{cases}$$

For example, suppose $A = \{1, 2, 3, 4\}$ and $B = \{2, 4, 5\}$. We define the relation $R_1 \subseteq A \times B$ such that

$$R_1 = \{(1, 2), (3, 2), (3, 4), (4, 4)\}.$$



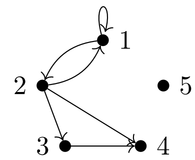
$$M_R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Figure 1: Arrow Diagram and Matrix

The matrix is based on the numerical order of A and B.

Let $C = \{1, 2, 3, 4, 5\}$. We define the relation $R_2 \subseteq C \times C$ such that

$$R_2 = \{(1, 1), (1, 2), (2, 1), (2, 3), (2, 4), (3, 4)\}$$



$$M_R = \begin{pmatrix} 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Figure 2: Arrow Diagram and Matrix

Again, the matrix is based on the numerical order of the set C.

A *function* is a relation $R \subseteq A \times B$ with the special property that for every $a \in A$, there is exactly one $b \in B$ such that $a \; R \; b$.

Reflexivity, Symmetry, Antisymmetry, Transitivity

We say that a relation R on a set A is

- *reflexive* if $a \; R \; a$ for all elements $a \in A$.
- *symmetric* if $a \; R \; b \implies b \; R \; a$ for all elements $a, b \in A$.
- *antisymmetric* if $a \; R \; b$ and $b \; R \; a \implies a = b$ for all elements $a, b \in A$.
- *transitive* if $a \; R \; b$ and $b \; R \; c \implies a \; R \; c$ for all elements $a, b, c \in A$.

Note: antisymmetric is not the opposite of symmetric. A relation can be both symmetric and antisymmetric.

	arrow diagram	matrix
reflexive	We must have a at every dot.	Diagonal entries are all 1.
symmetric	If we have a , we must have .	For $i \neq j$, $m_{i,j} = m_{j,i}$.
antisymmetric	We cannot have a .	For $i \neq j$, $m_{i,j}$ and $m_{j,i}$ cannot be both 1.
transitive	If we have a , we must have . If we have a , we must have .	For every non-zero entry in M^2, the corresponding entry in M must be 1.

Equivalence Relations and Classes

An *equivalence relation* is one that is reflexive, symmetric and transitive.

- We often write $\sim$ to denote an equivalence relation.
- $a \sim b$ is read as "a is equivalent to b".

Let $\sim$ be an equivalence relation on a set A. For any element $a \in A$, the *equivalence class* of a with respect to $\sim$, denoted by $[a]$, is the set

$$[a] = \{x \in A \,|\, x \sim a\}.$$

Theorems

Let $\sim$ be an equivalence relation on a set A. Then

- every element of A belongs to exactly one equivalence class.
- every equivalence class contains at least one element.
- for all $a, b \in A$, $a \sim b$ iff $[a] = [b]$.
- for all $a, b \in A$, $a \not\sim b$ iff $[a] \cap [b] = \emptyset$. That is, the equivalence classes are either equal or disjoint.

Equivalence Classes and Partitions

A *partition* of a set A is a collection of disjoint nonempty subsets of A whose union equals A. Let A be a set.

- The equivalence classes of an equivalence relation on A partition A.
- Any partition of A can be used to form an equivalence relation on A.

Partial Orders

A *partial order* is a reflexive, antisymmetric and transitive relation.

- We often write $\preceq$ to denote a partial order: $a \preceq b$ reads "a precedes b".
- A set A together with a partial order $\preceq$ on the set is a *partially ordered set*, or *poset*, denoted by $(A, \preceq)$.
- We say two elements $a, b \in A$ are comparable with respect to a partial order $\preceq$ iff either $a \preceq b$ or $b \preceq a$ holds.
- A partial order in which every two elements are comparable is called a *total order* or *linear order*.

Representing Partial Orders

We represent a partial order $\preceq$ on a finite set by a *Hasse diagram*:

- If $a \preceq b$ and $a \neq b$, then a line between a and b is drawn, with a positioned lower than b.
- We don't draw any lines that can be deduced by the transitive property.
- We don't draw loops to indicate the reflexive property.

For example, for the poset $(\{1, 2, 3, 4, 6\}, |)$, we draw

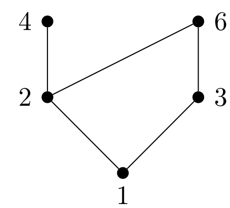


Figure 3: Hasse Diagram

Maximal, Minimal, Greatest and Least

Let $(A, \preceq)$ be a poset;. An element $x \in A$ is called

- A *maximal element* if there is no element $a \in A$ such that $x \prec a$.
- A *minimal element* if there is no element such that $a \prec x$.
- The *greatest element* if $a \preceq x$ for all $a \in A$.
- The *least element* if $x \preceq a$ for all $a \in A$.

Note that the greatest and least elements, if they exist, must be unique.

Proofs and Logic

Types of provable statements

Universal statements

- A *universal statement* has the form "$\forall x \in D$, $P(x)$ is true".
  - D is called the domain of disclosure.
  - $P(x)$ is some property/statement related to x.
- The *universal quantifier* symbol $\forall$ is read as "for all".

- The general structure for a proof of a universal statement is:
  Let $x \in D$.
  $\vdots$
  Then $P(x)$ is true.
- To disprove a universal statement, we only need to provide a single counterexample.

Existential statements

- A *universal statement* has the form "$\exists x \in D$, such that $P(x)$".
- The *existential quantifier* symbol $\exists$ is read as "there exists".
- The general structure for a proof of an existential statement is:
  Choose $x \in D$ to be ....
  $\vdots$
  Then $P(x)$ is true.
- To disprove an existential statement, we need to show the property does not hold for every element in the domain (a universal statement).

Conditional statements

- A *conditional statement* has the form "$P \Rightarrow Q$", or "If P is true, then Q is true".
- The *implication* symbol $\Rightarrow$, is read as "implies".
- The general structure for a proof of an existential statement is:
  Suppose P is true.
  $\vdots$
  Then Q is true.
- To disprove a conditional statement "If P is true, then Q is true", we must provide a counterexample where P is true but Q is false.

Converse statements

- The *converse* of a conditional statement "If P is true, then Q is true" is "If Q is true, then P is true".
- The *reverse implication* symbol $\Leftarrow$ is the opposite of the implication symbol $\Rightarrow$.
- In general, the converse of a statement is independent of the original statement. That is, proving/disproving a statement does nothing to prove/disprove its converse, or vice versa.

Biconditional statements

- A *biconditional statement* has the form "$P \Leftrightarrow Q$.
- The *double implication* symbol $\Leftrightarrow$ is read as "if and only if", written "iff" for short.
- The general structure for a proof of an existential statement is:
  Suppose P is true.
  $\vdots$
  Then Q is true.
  Now suppose Q is true.
  $\vdots$
  Then P is true.
- To disprove a conditional statement, disprove either direction.

Multiple quantifiers

- It is possible for a statement to contain more than one quantifier, such as a statement of the form "$\forall x \; \exists y \; P(x, y)$".
- The general structure for a proof using multiple quantifiers comes from adapting each part in turn. For example, the structure of a proof of the statement "$\forall x \in D \; \exists y \in D_2 \; P(x, y)$" would be:
  Let $x \in D_1$.
  Choose $y \in D_2$ to be ....
  $\vdots$
  Then $P(x, y)$ is true.
- To disprove a conditional statement, disprove either direction.

Negation

- The *negation* of a statement is a statement that is true precisely when the original statement is false.
- The *negation* symbol $\sim$, or sometimes $\neg$, is read as "not".
- For example, $\sim (x = y)$ means the same thing as $x \neq y$.
- The negation of a universal statement:
  $$\sim \Big(\forall x \in D \text{ s.t } P(x)\Big) \equiv \exists x \in D \text{ s.t} \sim P(x).$$
- The negation of an existential statement:
  $$\sim \Big(\exists x \in D \text{ s.t } P(x)\Big) \equiv \forall x \in D \text{ s.t} \sim P(x).$$

- The negation of a conditional statement:

$$\sim \Big(P \implies Q\Big) \equiv P \text{ and} \sim Q$$

- The negation of a biconditional statement:

$$\sim \Big(P \Leftrightarrow Q\Big) \equiv \sim \big(P \Rightarrow Q\big) \text{ or } \sim \big(Q \Rightarrow P\big).$$

- When negating a statement with multiple quantifiers, negate each quantifying component in turn. More precisely, apply the facts that

$$\sim (\exists x\ P(x)) \text{ is equivalent to } \forall x(\sim P(x))$$

  and

$$\sim (\forall x\ P(x)) \text{ is equivalent to } \exists x(\sim P(x)).$$

Proof techniques

Contraposition

- The *contrapositive* of a conditional statement "If P is true, then Q is true" is "If Q is false, then P is false".
- Symbolically, the contrapositive of $P \Rightarrow Q$ is $\sim P \Rightarrow \sim Q$.
- The contrapositive of a statement is equivalent to the original statement. This means that to prove any conditional statement, we can instead prove its contrapositive.

Contradiction

- To prove a statement by *contradiction*, we assume the required result is false, and eventually derive a fact that is obviously false, which affirms that the original result was in fact true.
- The general structure for a proof by contradiction of a simple statement "P is true" is:

  Suppose P is false.
  $\vdots$
  This is a contradiction.
  Thus P is true.

Mathematical Induction

- The technique of *mathematical induction* can be used to prove results for all elements of a countable set, such as the natural numbers.
- Proving the statement "For all integers $n > n_0, P(n)$ is true," is broken down into two parts:
  - *Base case*: Prove that $P(n_0)$ is true.
  - *Inductive step*: Prove that $P(k) \Rightarrow P(k+1)$ for all $k \in \mathbb{N}$.
- *Strong induction* instead proves in the inductive step that for all $k \in \mathbb{N}$, if $P(n_0)$, $P(n_0+1)$, $\ldots$, and $P(k)$, then $P(k+1)$,

Symbolic Logic Terminology

Propositions

Definition 0.8 (Proposition)**.** A *proposition* is a statement that is unambiguously true or false.

More complicated expressions, called *compound propositions* or *propositional forms*, can be constructed from propositions using logical operators. If p and q are propositions, then

- $(\sim q) \to (\sim p)$ is the contrapositive of $p \to q$;
- $q \to p$ is the converse of $p \to q$.

Truth Tables

Truth tables allow us to prove or disprove the logical equivalence of statements by listing every possible combination of truth values T or F that can be assigned to the propositions of each statement, and checking whether the statements always yield the same truth value.

If the statements always yield the same truth value, then they are *logically equivalent*, denoted by $\iff$ or $\equiv$. An alternative way of showing that the statements are **not** equivalent is to assign specific truth values to each proposition in the statement in such a way that one of the statements is true and the other is false.

Logical operators

Negation

The truth value of $\sim p$ is the opposite of the truth value of p.

p	$\sim p$
T	F
F	T

And

The *conjunction* $p \wedge q$ is true when both p and q are true, and false otherwise.

p	q	$p \wedge q$
T	T	T
T	F	F
F	T	F
F	F	F

Or

The *disjunction* $p \vee q$ is false when both p and q are false, and true otherwise.

p	q	$p \vee q$
T	T	T
T	F	T
F	T	T
F	F	F

Exclusive Or

The statement $p \oplus q$ is true when either p or q is true, but not both.

p	q	$p \oplus q$
T	T	F
T	F	T
F	T	T
F	F	F

Only If

The *conditional* proposition $p \to q$ is always true except when p is true and q is false.

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

It can be shown that

- $p \to q \iff (\sim p) \vee q$;
- $p \to q \iff (\sim q) \to (\sim p)$.

If and Only If

The *biconditional* proposition $p \leftrightarrow q$ is true when p and q are both true or both false.

p	q	$p \leftrightarrow p$
T	T	T
T	F	F
F	T	F
F	F	T

It can be shown that $p \leftrightarrow q \iff (p \to q) \wedge (q \to p)$.

Tautology, Contradiction, Contingency

- A *tautology* **T** is a propositional form that is always true. Note that all tautologies are logically equivalent. Conversely, if a statement is logically equivalent to a tautology, then the statement itself is a tautology.
- A *contradiction* **F** is a propositional form that is always false. Note that all contradictions are logically equivalent. Conversely, if a statement is logically equivalent to a contradiction, then the statement itself is a contradiction.
- A *contingency* is a form that is neither a tautology nor a contradiction, because its truth value depends on an unknown proposition.

Laws of logical equivalence

Here are some of the most useful logical equivalences.

- Commutative laws:
$$p \wedge q \iff q \wedge p \quad \text{and} \quad p \vee q \iff q \vee p.$$
- Associative laws:
$$\begin{aligned} (p \wedge q) \wedge r &\iff p \wedge (q \wedge r) \text{ and} \\ (p \vee q) \vee r &\iff p \vee (q \vee r). \end{aligned}$$
- Distributive laws:
$$\begin{aligned} p \wedge (q \vee r) &\iff (p \wedge q) \vee (p \wedge r) \text{ and} \\ p \vee (q \wedge r) &\iff (p \vee q) \wedge (p \vee r). \end{aligned}$$
- Absorption laws:
$$p \wedge (p \vee q) \iff p \quad \text{and} \quad p \vee (p \wedge q) \iff p$$
- Identity laws:
$$p \wedge \mathbf{T} \iff p \quad \text{and} \quad p \vee \mathbf{F} \iff p.$$
- Laws of negation:
$$p \vee (\sim p) \iff \mathbf{T} \quad \text{and} \quad p \wedge (\sim p) \iff \mathbf{F}.$$
- Double negation law:
$$\sim (\sim p) \iff p.$$
- Idempotent laws:
$$p \wedge p \iff p \quad \text{and} \quad p \vee p \iff p$$

- Domination laws:

$$p \vee \mathbf{T} \iff \mathbf{T} \quad \text{and} \quad p \wedge \mathbf{F} \iff \mathbf{F}$$

- De Morgan's laws:

$$\begin{aligned} \sim (p \wedge q) &\iff (\sim p) \vee (\sim q) \text{ and} \\ \sim (p \vee q) &\iff (\sim p) \wedge (\sim q). \end{aligned}$$

It is noteworthy that many of these laws have a corresponding law in set algebra.

Equivalence and Logical Implication

Two propositional forms P and Q are *logically equivalent* if and only if $P \leftrightarrow Q$ is a tautology. Symbolically, the $\leftrightarrow$ turns into a $\iff$.

We say that P *logically implies* Q if, whenever P is true, Q is also true. This happens if and only if $P \to Q$ is a tautology. Symbolically, the $\to$ turns into a $\implies$.

It can be shown that $P \iff Q$ if and only if $P \implies Q$ and $Q \implies P$.

Arguments

Suppose $P_1, \ldots P_n, Q$ are propositions. An *argument*, can be formed by writing

$$\begin{array}{l} P_1 \\ \vdots \\ \underline{P_n} \\ \therefore Q. \end{array}$$

$P_1, \ldots, P_n$ are *hypotheses*, and Q is the *conclusion*. The argument is valid when, in the case where all hypotheses are true, the conclusion is also true. This is equivalent to saying that the argument is valid if and only if $P_1 \wedge \cdots \wedge P_n \implies Q$.

Rules of Inference

The following are examples of valid arguments, known as rules of inference.

- modus ponens

$$\begin{array}{l} p \to q \\ \underline{p \qquad} \\ \therefore q. \end{array}$$

- modus tollens

$$\begin{array}{l} p \to q \\ \underline{\sim q \quad} \\ \therefore \sim p. \end{array}$$

- hypothetical syllogism

$$\begin{array}{l} p \to q \\ \underline{q \to r} \\ \therefore p \to r. \end{array}$$

We can verify the validity of an argument using rules of inference, truth tables or laws of logical equivalence.

Combinatorics

Counting sets

Addition law

Given finite sets $A_1, A_2, \ldots, A_n$ that are disjoint ($A_i \cap A_j = \emptyset$ for all $i \neq j$), we have

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|.$$

Likewise, if n mutually exclusive events can occur in $k_1, k_2, \ldots, k_n$ different ways, then the number of ways any one of the events can occur is $k_1 + k_2 + \cdots + k_n$.

Multiplication law

Given finite sets $A_1, A_2, \ldots, A_n$, we have

$$|A_1 \times A_2 \times \cdots \times A_n| = |A_1| \times |A_2| \times \cdots \times |A_n|.$$

Likewise, if n independent events can occur in $k_1, k_2, \ldots, k_n$ different ways, then the number of ways every event can occur is $k_1 k_2 \cdots k_n$.

Complement law

Given a finite universal set $\mathcal{U}$ and some set $A \subseteq \mathcal{U}$, we have

$$|A| = |\mathcal{U}| - |A^c|.$$

Likewise, if an event can have m different possible outcomes, and a particular event E can occur in n different ways, then the number of ways E cannot occur is $m - n$.

Inclusion-exclusion Principle

Given finite sets $A_1, A_2, A_3, \ldots$ we have

$$|A_1 \cup A_2| = |A_1| + |A_2| - |A_1 \cap A_2|,$$

$$\begin{aligned}|A_1 \cup A_2 \cup A_3| = &|A_1| + |A_2| + |A_3| \\ &- |A_1 \cap A_2| - |A_1 \cap A_3| - |A_2 \cap A_3| \\ &+ |A_1 \cap A_2 \cap A_3|,\end{aligned}$$

and, in general,

$$\begin{aligned}\left|\bigcup_{i=1}^{n} A_i\right| = &\sum_{i=1}^{n} |A_i| - \sum_{i<j}^{n} |A_i \cap A_j| + \sum_{i<j<k}^{n} |A_i \cap A_j \cap A_k| \\ &+ \cdots + (-1)^n |A_1 \cap A_2 \cap \cdots \cap A_n|.\end{aligned}$$

Arrangements of objects

Arrangements of distinct objects

Given n distinct objects, they can be arranged in $n!$ different ways.

The *factorial* of a positive integer n is given by

$$n! = 1 \times 2 \times 3 \times \ldots \times n, \text{ while } 0! = 1.$$

Arrangements of non-distinct objects

Given n objects of m distinct types, where there are k_1 of one type, k_2 of another type, and so on (with $n = k_1 + k_2 + \ldots + k_m$), they can be arranged in

$$\binom{n}{k_1, k_2, \ldots, k_m} := \frac{n!}{k_1! k_2! \ldots k_m!}$$

different ways.

This is called a multinomial coefficient, as it is also the coefficient of $x_1^{k_1} x_2^{k_2} \ldots x_m^{k_m}$ in the expanded multinomial expression for $(x_1 + x_2 + \cdots + x_m)^n$.

Selections of objects

Ordered selections

A *permutation* of objects is a selection of objects whose order of selection matters.

We denote the number of ways to choose a permutation of size k from n distinct types of object (*without repetition*) by $P(n,k)$ or ${}^nP_k$.

It is given by $P(n,k) = \dfrac{n!}{(n-k)!}$.

The number of ways to choose a permutation of size k from n distinct types of object *with repetition allowed* is n^k.

Unordered selections

A *combination* of objects is a selection of objects whose order of selection does not matter.

We denote the number of ways to choose a combination of size k from n distinct types of object (*without repetition*) by $C(n,k)$ or ${}^nC_k$ or $\dbinom{n}{k}$.

It is given by $\dbinom{n}{k} = \dfrac{n!}{k!(n-k)!}$.

This is called a binomial coefficient, as it is also the coefficient of $x^k y^{n-k}$ in the expanded binomial expression for $(x+y)^n$.

The number of ways to choose a combination of size k from n distinct types of object *with repetition allowed* is $\dbinom{n+k-1}{k}$. This result comes from the “stars and bars" approach.

Likewise, the number of solutions to $x_1 + x_2 + \cdots + x_k = n$, where $x_i \in \mathbb{N}$ for all i, is $\dbinom{n+k-1}{k}$.

Pigeonhole principle

Pigeonhole principle: Given n objects to distribute amongst k boxes, if $k < n$ then at least one box contains more than 1 object.

Generalised pigeonhole principle: Given n objects to distribute amongst k boxes, at least one box contains at least $\left\lceil \dfrac{n}{k} \right\rceil$ objects.

As a corollary, the minimum number of objects required to be distributed amongst k boxes such that at least one box contains at least m objects is $n = (m-1)k + 1$.

Recurrence relations

Terminology

- A *recurrence relation* is any relation which describes a sequence $\{a_k\}$ by defining successive terms in relation to previous terms.

- A sequence defined by a recurrence relation is not explicitly described unless it also includes *initial conditions*, which usually give the first value(s) of the sequence.

- A *closed form solution* for a sequence $\{a_n\}$ is an equation that gives an as a function of n only.

- A *linear* recurrence relation for a sequence $\{a_n\}$ recursively defines a_n in relation to a linear combination of previous terms.

- A linear recurrence relation of order k (or kth-order linear recurrence) is one in which a_n is defined recursively as far back as k earlier terms in the sequence $\{a_n\}$.

Homogeneous linear recurrence relations

A *homogeneous* linear recurrence relation is one in which all terms are constant multiples of terms from the sequence $\{a_n\}$.

To solve a homogeneous recurrence relation

$$a_n + c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} = 0$$

for given constants c_i:

- Find the roots $r_1, r_2, \ldots, r_k$ of the *characteristic equation* $r^k + c_1 r^{k-1} + c_2 r^{k-2} + \cdots + c_k = 0$.
- If there are no repeated roots, write $a_n = A_1 r_1^n + A_2 r_2^n + \cdots + A_k r_k^n$ for arbitrary constants A_i.
- If any root is repeated, multiply each repeated instance by n (possibly multiple times) to preserve independence of terms. For example, if $r_1 = r_2 = r_3$, write $a_n = A_1 r_1^n + A_2 n r_2^n + A_3 n^2 r_1^3 + \cdots + A_k r_k^n$ for arbitrary constants A_i.
- Substitute the initial conditions into the general solution to find the values of each A_i.

Inhomogeneous linear recurrence relations

A *inhomogeneous* linear recurrence relation is one in which all terms are constant multiples of terms from the sequence $\{a_n\}$, except for one term which is a function of n.

To solve a homogeneous recurrence relation

$$a_n + c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k} = f(n)$$

for given constants c_i:

- Find the homogeneous solution h_n by solving $h_n + c_1 h_{n-1} + c_2 h_{n-2} + \cdots + c_k h_{n-k} = 0$ as above, but do not yet find the values of each A_i.
- Guess the particular solution p_n by setting p_n to be a generalised form of $f(n)$. For example, guess a general polynomial of the same degree as $f(n)$, or a general exponential with the same base as $f(n)$.
- If any term in the guess for p_n appears as a term in the homogeneous solution h_n, multiply it by n (possibly multiple times) to preserve independence of terms.
- Write the general solution $a_n = h_n + p_n$.
- Substitute the initial conditions into the general solution to find the values of each A_i.

Graph Theory

The Basics

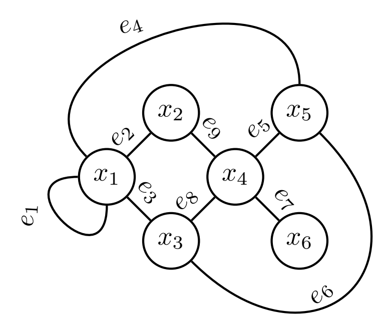
Definition

Formally, a finite graph G consists of:

- a set V whose elements are called the *vertices* of G;
- a set E whose elements are called the *edges* of G;
- a function that assigns to each edge $e \in E$ an unordered pair of vertices, called the *endpoints* of e.

Intuitively, a graph G is simply a collection of dots ('vertices') and lines ('edges') connecting them.

Here is an example:



So using our terminology above:

- the set V is $\{x_1, x_2, x_3, x_4, x_5\}$;
- the set E is $\{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$.

We can also summarise the endpoints of this graph in the following table:

Edge	Endpoints
e_1	$\{x_1\}$
e_2	$\{x_1, x_2\}$
e_3	$\{x_1, x_3\}$
e_4	$\{x_1, x_5\}$
e_5	$\{x_4, x_5\}$
e_6	$\{x_3, x_5\}$
e_7	$\{x_4, x_6\}$
e_8	$\{x_3, x_4\}$
e_9	$\{x_2, x_4\}$

Essentially, the endpoints of an edge are the vertices connected by that particular edge. For example, e_1 is only connected to x_1 and hence, the endpoint of e_1 is $\{x_1\}$.

Terminology

- If the edge $e \in E$ has endpoints $v, w \in V$, then we say that:
  - The edge e *connects* the vertices v and w.
  - The edge e is *incident* to the vertices v and w.
  - The vertices v and w are the *endpoints* of the edge e.
  - The vertices v and w are adjacent.
  - The vertices v and w are neighbours.
- Two edges with the same endpoints are *multiple* or *parallel*.
- A *loop* is an edge that connects a vertex to itself.
- The *degree* of a vertex v, denoted by $\deg(v)$, is the number of edges incident with v, counting the loops *twice*.
- An *isolated vertex* is one with degree 0, and a pendant vertex is one with degree 1.
- A *simple graph* is a graph with no loops or parallel edges.

The Handshaking Theorem

The total degree of a graph is twice the number of edges. That is,

$$2|E| = \sum_{v \in V} \deg(v).$$

Consequently, this means that:

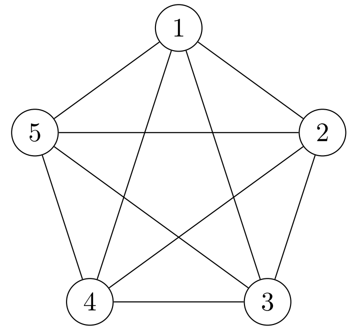
- the total degree (the sum of all vertex degrees) of a graph must be even;
- the number of vertices with odd degree must be even.

Special Kinds of Graphs

1. The complete graph, denoted as K_n, is a simple graph with:
   - n vertices;
   - exactly one edge between each pair of distinct vertices.

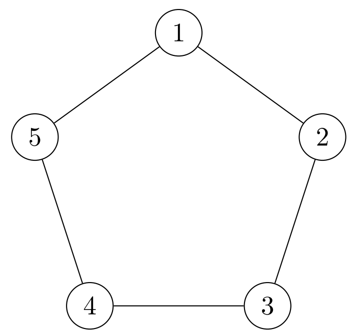
   Hence, the number of edges in K_n is $\binom{n}{2}$.

   For example, K_5 looks like:



2. The cyclic graph C_n ($n \geq 3$) consists of
   - n vertices $v_1, v_2, \dots, v_n$;
   - n edges $\{v_1, v_2\}, \{v_2, v_3\}, \dots, \{v_n, v_1\}$.

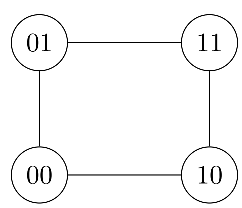
   For example, C_5 looks like:



3. The n-cube Q_n is a simple graph with:
   - vertices for each bit string $a_1 a_2 \cdots a_n$ of length n, where $a_n \in \{0, 1\}$;
   - vertices are adjacent if and only if they differ by one bit.

   The number of edges in Q_n is $n2^{n-1}$ and the number of vertices is 2^n.

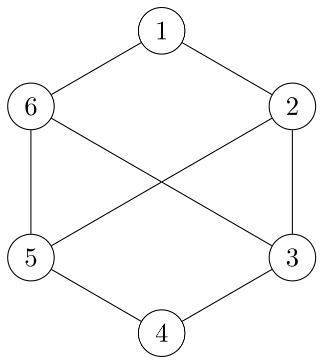
   For example, Q_2 looks like:



4. A bipartite graph is one such that:
   - the vertex set can always be partitioned into two subsets V_1, V_2;

- no vertex is adjacent to any vertex in the same subset.

For example, the following graph is bipartite:

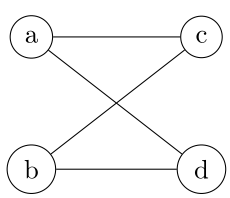


Since the complete vertex set V, can be partitioned into the two sets $V_1 = \{1, 3, 5\}$ and $V_2 = \{2, 4, 6\}$.

5. The complete bipartite graph $K_{m,n}$ is the simple bipartite graph with vertex set $V_1 \cup V_2$ with
   - V_1 containing m vertices and V_2 containing n vertices;
   - edges between **every** vertex in V_1 and **every** vertex in V_2.

   $K_{m,n}$ has $m + n$ vertices and mn edges.

   For example, $K_{2,2}$ is:



Subgraphs

Let G_1 and G_2 be two graphs with vertex sets V_1 and V_2 and edge set E_1 and E_2. Then, G_1 is a subgraph of G_2, and we write $G_1 \subseteq G_2$, if and only if

- $V_1 \subseteq V_2$;
- $E_1 \subseteq E_2$;
- each edge in G_1 has the same endpoints as in G_2.

Complementary Graphs

Let G be a simple graph. The *complementary graph* $\overline{G}$ of G is a simple graph with

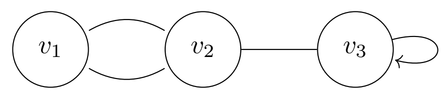
- the same vertex set as G;
- an edge joining two vertices if and only if they are **not** adjacent in G.

The Adjacency Matrix

Let G be a graph with an ordered listing of vertices $v_1, v_2, \ldots, v_n$. The adjacency matrix of G is the $n \times n$ matrix $A = [a_{ij}]$ with a_{ij} = the number of edges containing v_i and v_j.

- The entries a_{ij} depend on the order in which the vertices have been numbered.
- Changing the vertex order corresponds to permuting the rows and columns.
- The adjacency matrix A is symmetric, i.e, $A = A^T$.

For example, for the following graph,



the adjacency matrix is,

$$A = \begin{pmatrix} 0 & 2 & 0 \\ 2 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}.$$

Paths and Circuits

Definitions

- A *walk* in a graph G is an alternating sequence of vertices v_i and edges e_i in G, written as

  $$P = v_0 e_1 v_1 e_2 v_2 \ldots v_{n-1} e_n v_n,$$

  where v_{i-1} and v_i are the endpoints of edge e_i for all i.
- The *length* of the walk is the number of edges involved (n above).
- A *closed* walk is one that starts and ends in the same vertex.
- In a simple graph, a walk can be specified by stating the vertices alone.
- A *path* is a walk with **no** repeated edges.
- A *circuit* is a closed walk with **no** repeated edges.
- A path P is *simple* if and only if all the vertices visited by P are distinct. That is, there are no repeated vertices.
- A circuit C is *simple* if and only if all the vertices visited by C are distinct, except of course the first and last vertices (which are the same).

Theorems Related to Paths & Circuits

- Let a and b be vertices in a graph. If there is a walk from a to b, then there is a simple path from a to b.

- If A is the adjacency matrix for G with ordered vertices $v_1, \dots, v_n$, then the number of walks of length k from v_i to v_j in G is given by the entry in the ith row and jth column of A^k.

Connectivity

- Vertices a, b of a graph G are *connected* in G if and only if there is a walk from a to b.

- A graph G is connected if and only if every pair of distinct vertices is connected in G.

- Let G be a graph with vertex set V. The relation $\sim$ on V defined by

  $v_i \sim v_j$ if and only if v_i is connected to v_j in G,

  is an equivalence relation.

- The equivalence classes of this relation are the connected components of G. Two vertices are in the same connected component if and only if they are connected in G.

Let G be a graph with vertices $v_1, \dots, v_n$ and adjacency matrix A. Let

$$C = I + A + A^2 + \cdots + A^{n-1}.$$

Then G is connected if and only if C has no zero entries.

Euler and Hamiltonian Paths and Circuits

Suppose that G is a graph.

- An *Euler path* in G is a path that includes every edge of G exactly once.

- A *Hamiltonian path* in G is a simple path that includes every vertex of G exactly once.

- An *Eulerian circuit* in G is a circuit that includes every edge of G exactly once.

- A *Hamiltonian circuit* in G is a simple circuit which includes every vertex of G exactly once (counting the starting vertex once).

Necessary and Sufficient Condition for Existence of Euler Circuit

Let G be a connected graph. An Euler circuit exists if and only if all vertices of G have even degree i.e. there are no vertices of odd degree.

Necessary and Sufficient Condition for Existence of Euler Path

Let G be a connected graph. An Euler path which is not a circuit exists if and only if G has exactly two vertices of odd degree.

Weak Conditions for Existence of Hamiltonian Path/Circuit

In short, there is no simple necessary and sufficient criteria which are known that determine whether a graph has a Hamilton circuit or path.

However, there do exist some "weak" conditions:

- A graph with a vertex of degree 1 cannot have a Hamilton circuit.

- If a graph G has a Hamilton circuit, then the circuit must include all edges incident with vertices of degree 2.

- A Hamilton path or circuit uses at most 2 edges incident with any one vertex.

- Let G be a connected and simple graph with $n \geq 3$ vertices, such that each vertex has degree at least $n/2$. Then G has a Hamilton circuit.

Isomorphic Graphs

Definition

Let G and G' be graphs with vertices V and V' respectively, and edges E and E' respectively. Then G is *isomorphic* to G' (written as $G \cong G'$), if and only if there are two bijections $f : V \to V'$ and $g : E \to E'$, such that e is incident with v in G if and only if $g(e)$ is incident with $f(v)$ in G'.

- Two graphs are isomorphic if and only if they are the 'same' except for edge and vertex labelings.

- In this case, $\deg(v) = \deg(f(v))$.

Two simple graphs G and G' are isomorphic if and only if there is a bijection $f : V \to V'$ such that for all $v_1, v_2 \in V$, v_1 and v_2 are adjacent in G if and only if $f(v_1)$ and $f(v_2)$ are adjacent in G'.

Invariants

A property of a graph G is an *invariant* if and only if G' also has this property whenever $G' \cong G$.

Some graph invariants are

- the number of vertices;
- the number of edges;
- the total sum of all the vertex degrees;
- the number of vertices of a given degree;
- bipartiteness, number of connected components, connectedness;
- having a vertex of some degree n adjacent to a vertex of some degree m;
- the number of circuits of a given length;
- the existence of an Euler circuit;
- the existence of a Hamilton circuit.

The easiest way to show that G and G' are **not** isomorphic ($G \ncong G'$) is to find an invariant property that holds for G but not for G'.

Planar Graphs

Definition

A graph G is *planar* if and only if it can be drawn in the plane so that no edge crosses another. Such a drawing is called a *planar map* or *planar representation* of G.

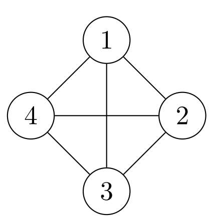


Figure 4: *Not* a planar map

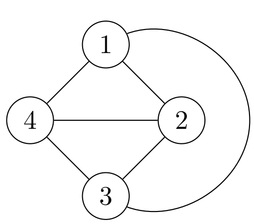


Figure 5: Planar representation of above graph

Properties

- A planar map divides the plane into a finite number of regions. Exactly one of these regions is unbounded.
- A planar graph can have different planar representations (or maps), but the number of regions is the same for all planar representations.
- **Euler's Formula:** If G is a connected planar graph with e edges and v vertices, and if r is the number of regions in a planar representation of G, then
$$v - e + r = 2.$$
- The *degree* of a region R in a planar representation is the number of edges (counting repetitions) traversed in going round the boundary of R.
- We also have that,
$$2|E| = \text{ the sum of region degrees.}$$
- The sum of region degrees equals the sum of vertex degrees.

Dual Graphs

The dual of a planar graph G is a planar graph G^* given as:

- for each region R_i of G, there is an associated vertex v_i^* in G^*;
- for each edge e in G that is surrounded by one region R_i, there is an associated loop in G^* at vertex v_i^*.
- for each edge e of G that separates two regions R_1 and R_2, there is an edge e^* in G^* that connects vertices v_1^* and v_2^* corresponding to R_1 and R_2 respectively.

Connected Planar Graphs

If G is a simple connected planar graph with at least 3 vertices, then every region degree is at least 3.

- To have a region of degree 1, G must have a loop.
- To have a region of degree 2, G must have parallel edges.

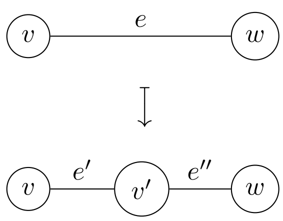
If G is a connected planar simple graph with e edges and $v \geq 3$, then

- $e \leq 3v - 6$;
- $e \leq 2v - 4$ if G has no circuits of length 3.

Elementary subdivisions

Suppose that G has an edge e with endpoints v and w. Let G' be the graph obtained from G by replacing e by a path $ve'v'e''w$.

Such an operation is called an *elementary subdivision.*



Homeomorphic Graphs

We say that two graphs are *homeomorphic* if and only if each can be obtained from a common graph by elementary subdivisions. If G is planar then so is any graph homeomorphic to G.

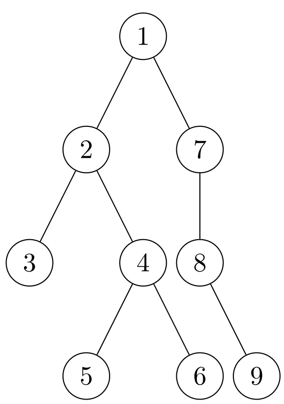
- **Kuratowski's Theorem**: A graph is planar if and only if it does not contain a subgraph homeomorphic to $K_{3,3}$ or K_5.

Trees

Definition

A *tree* is a connected graph with no circuits. A tree has no loops or multiple edges, so it is simple.

Here is an example:



Any tree T is planar.

Spanning Trees

A spanning tree in a graph G is a subgraph that is a tree and contains every vertex of G.

- Every connected graph contains a spanning tree.
- A connected graph with n vertices is a tree if and only if it has exactly $n-1$ edges.

Weighted Trees

- A weighted graph is a graph whose edges have been given numbers called weights. The weight of an edge e is denoted by $w(e)$.
- The weight of a subgraph in a weighted graph G is the sum of the weights of the edges in the subgraph.
- A minimal spanning tree in a weighted graph G is a spanning tree whose weight is less than or equal to the weight of any other spanning tree.

Kruskal's Algorithm for Minimal Spanning Tree

1. Start with the tree $T := \emptyset$.
2. Sort the edges of G into increasing order of weight (breaking ties arbitrarily).
3. Going down the list, add an edge to T if and only if it does not form a circuit with edges already in T.
4. Repeat step 3 until T has $n-1$ edges.

Then T is a minimal spanning tree for G.

Dijkstra's Algorithm

Given a connected weighted graph G and a particular vertex v_0, we want to find a shortest path from v_0 to v for each vertex v in G.

1. Start with the subgraph T consisting of v_0 only.
2. For all vertices $v \in T$, we need to also record $\mathrm{spl}(v)$, the shortest path length from v_0 to v. Initially, we just set $\mathrm{spl}(v_0) := 0$.
3. Consider all edges e with one endpoint u in T and the other endpoint v not in T.
4. Of these edges, choose an edge e which minimises $w(e) + \mathrm{spl}(u)$, where $w(e)$ is the weight of e.
5. Add this edge e, and its corresponding vertex v, to T, and set $\mathrm{spl}(v) := w(e) + \mathrm{spl}(u)$.
6. Repeat steps 3 to 5 until T contains every vertex in G.

Then, T is a minimal v_0-path spanning tree for G. That is, T is a tree, and for all vertices v, a shortest path in G between v_0 and v is a subgraph of T.