Discrete Mathematics: Conceptual Insights and Advanced Formulas

February 20, 2025

Contents

1 Introduction and Fundamental Concepts

Discrete mathematics forms the backbone of computer science, cryptography, and combinatorial optimization. Its language—sets, functions, relations, and counting—is essential for structuring rigorous arguments and designing algorithms. In these notes, we present definitions and theorems while discussing the ideas behind them and illustrating concepts with diagrams.

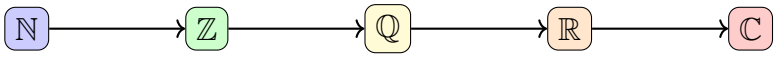
1.1 Numbers and Sets: Notation and Concepts

Understanding number systems and set theory is critical because they provide the basic vocabulary for nearly every mathematical discussion.

Definition 1.1 (Common Number Sets)**.**

- $\mathbb{N}$: the set of *natural numbers*. (Note: Some define $\mathbb{N} = \{0, 1, 2, \dots\}$, others $\{1, 2, \dots\}$.)
- $\mathbb{Z}$: the set of *integers*: $\{\dots, -2, -1, 0, 1, 2, \dots\}$.
- $\mathbb{Q}$: the set of *rational numbers*.
- $\mathbb{R}$: the set of *real numbers*.
- $\mathbb{C}$: the set of *complex numbers*.

Why It Matters: Each number set expands the types of problems you can solve. For example, $\mathbb{N}$ underpins counting, while $\mathbb{R}$ allows us to model continuous change.



Definition 1.2 (Basic Set Notation)**.** A *set* is a collection of distinct objects. We denote a set as

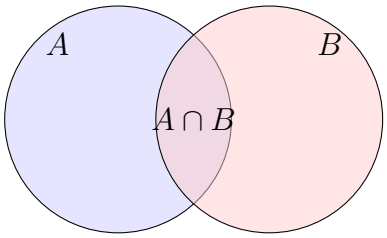
$$A = \{a, b, c, \dots\},$$

and write $a \in A$ if a is an element of A. Important notations include:

- $|A|$: the *cardinality* (number of elements in A).
- $A \subseteq B$: A is a *subset* of B.
- $A \cup B$: the *union* of A and B.
- $A \cap B$: the *intersection* of A and B.

- $A \setminus B$: the *set difference*.
- A^c or $\overline{A}$: the *complement* of A (relative to a universal set U).
- $\mathcal{P}(A)$: the *power set* of A, with $|\mathcal{P}(A)| = 2^{|A|}$ when A is finite.

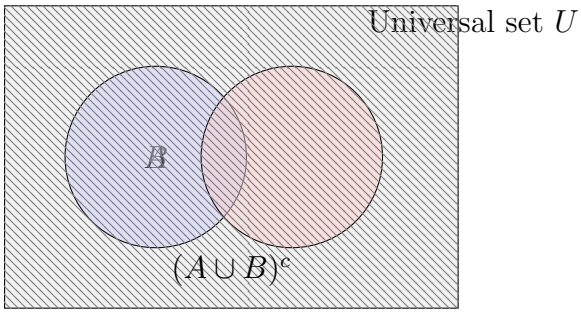
Why It Matters: Mastering set operations is essential for formal proofs, algorithm design, and understanding data structures.



Lemma 1.3 (De Morgan's Laws)**.** *For any two sets A and B (with respect to a universal set U):*

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

Conceptual Insight: *These laws show the duality between union and intersection under complementation.*



1.2 Mathematical Induction and Other Proof Techniques

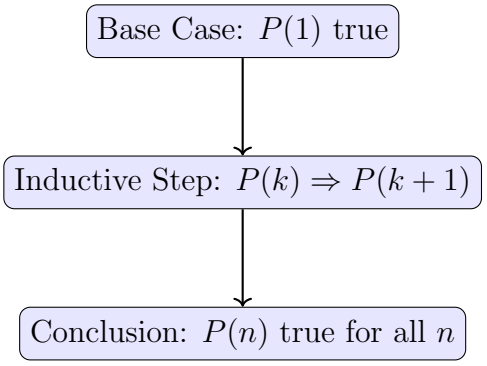
Mathematical induction is like a row of dominos: if the first falls (the base case) and every domino knocks over the next (the inductive step), then the entire row falls (the statement holds for all natural numbers).

Theorem 1.4 (Principle of Mathematical Induction)**.** *Let $P(n)$ be a proposition about $n \in \mathbb{N}$. If:*

(i) ***Base Case:*** *$P(1)$ is true.*

(ii) ***Inductive Step:*** *For every $k \in \mathbb{N}$, $P(k)$ true implies $P(k+1)$ is true.*

Then $P(n)$ is true for all $n \in \mathbb{N}$. ***Why It Matters:*** *Induction is essential for proving infinite families of statements—a technique central to algorithm correctness and number theory.*



Theorem 1.5 (Principle of Strong Induction)**.** *Let $P(n)$ be a proposition about $n \in \mathbb{N}$. If:*

(i) ***Base Case:*** *$P(1)$ is true.*

(ii) ***Inductive Step:*** *For all $n \geq 1$, if $P(1), P(2), \ldots, P(n)$ are true, then $P(n+1)$ is true.*

Then $P(n)$ is true for all $n \in \mathbb{N}$. ***Conceptual Note:*** *Strong induction is particularly useful when $P(n+1)$ depends on several previous cases.*

Example 1.6 (Prime Factorization)**. Statement:** Every integer $n > 1$ can be written as a product of primes.

Proof (by strong induction):

- *Base Case:* $n = 2$ is prime.
- *Inductive Step:* Assume every integer $2 \leq k \leq n$ can be factored into primes. For $n+1$: if it is prime, we are done; if composite, write $n+1 = ab$ with $2 \leq a, b \leq n$. Then, by the induction hypothesis, both a and b have prime factorizations, so $n+1$ does as well.

Why It Matters: This proof underpins the Fundamental Theorem of Arithmetic.

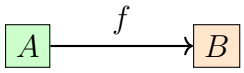
1.3 Functions

Functions formalize the idea of a rule mapping inputs to outputs, which is vital in modeling processes and computer algorithms.

Definition 1.7 (Function)**.** A *function* f from a set A to a set B, written $f : A \to B$, assigns each $a \in A$ a unique element $f(a) \in B$. Here:

- A is the *domain*.
- B is the *codomain*.
- The *image* of f is $\{f(a) \mid a \in A\}$.

Why It Matters: Functions are central to describing processes and are the building blocks of mathematical modeling.

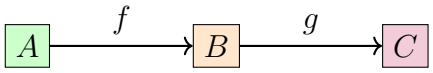


Definition 1.8 (Types of Functions)**.** A function $f : A \to B$ is:

- **Injective (one-to-one)**: $f(a_1) = f(a_2)$ implies $a_1 = a_2$.
- **Surjective (onto)**: For every $b \in B$, there exists $a \in A$ such that $f(a) = b$.
- **Bijective**: Both injective and surjective.

Conceptual Insight: Recognizing function types helps determine invertibility and structure in algorithms.

$$g \circ f : A \to C$$



Lemma 1.9 (Composition of Functions)**.** *Let $f : A \to B$ and $g : B \to C$. Then the composition $g \circ f : A \to C$ defined by*

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

is a function. Moreover:

- *If f and g are injective, then $g \circ f$ is injective.*
- *If f and g are surjective, then $g \circ f$ is surjective.*
- *If f and g are bijective, then $g \circ f$ is bijective with inverse $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.*

Why It Matters: *Composing functions is analogous to chaining processes—a key idea in building complex algorithms from simple modules.*

1.4 Relations

Relations generalize functions and describe various associations between elements, which are essential in structuring data and logical statements.

Definition 1.10 (Relation)**.** A *relation* R on a set A is a subset of the Cartesian product $A \times A$:

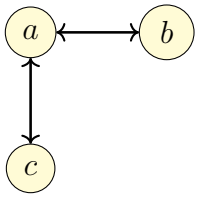
$$R \subseteq A \times A.$$

If $(a, b) \in R$, we write $a \, R \, b$.

Definition 1.11 (Properties of Relations)**.** Let R be a relation on A. Then:

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

Why It Matters: These properties help classify relations (like orders or equivalence relations) that are central to database theory and logic.



Lemma 1.12 (Composition of Relations)**.** *If R and S are relations on A, their composition is:*

$$R \circ S = \{(a, c) \in A \times A \mid \exists\, b \in A \text{ with } (a, b) \in S \text{ and } (b, c) \in R\}.$$

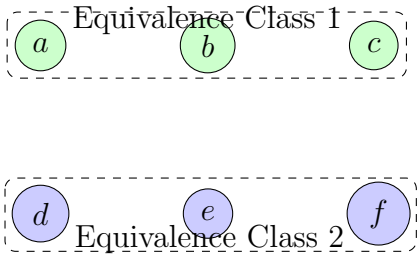
Conceptual Note: *Composing relations is analogous to linking steps in a process, useful in graph theory and state-transition analysis.*

1.5 Equivalence Relations and Partitions

Equivalence relations group elements into classes based on a shared property, simplifying analysis by reducing redundancy.

Definition 1.13 (Equivalence Relation)**.** A relation R on a set A is an *equivalence relation* if it is reflexive, symmetric, and transitive. **Why It Matters:**

Equivalence relations partition a set into disjoint equivalence classes—a key concept in modular arithmetic and classification problems.



2 Ordering and Posets

Ordering is about comparing elements. The structure of orders and partially ordered sets (posets) is central in optimization and scheduling.

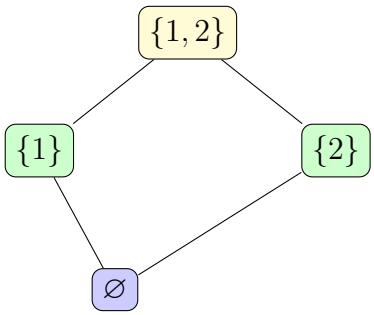
2.1 Partial Orders and Hasse Diagrams

Definition 2.1 (Partial Order)**.** A relation $\preceq$ on a set P is a *partial order* if it is:

- **Reflexive**: $a \preceq a$ for every $a \in P$.
- **Antisymmetric**: $a \preceq b$ and $b \preceq a$ imply $a = b$.
- **Transitive**: $a \preceq b$ and $b \preceq c$ imply $a \preceq c$.

Definition 2.2 (Hasse Diagram)**.** A *Hasse diagram* is a simplified drawing of a finite poset that omits edges implied by reflexivity and transitivity. If $a \prec b$ (i.e., $a \preceq b$ and $a \neq b$), then b is drawn above a. **Why It Matters:**

Hasse diagrams provide an intuitive visualization of a poset's structure.

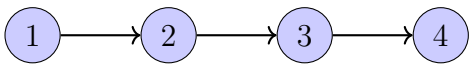


Lemma 2.3. *Every finite, nonempty poset has at least one* minimal *element and at least one* maximal *element.*

2.2 Total Orders and Chains

Definition 2.4 (Total (or Linear) Order)**.** A partial order $\preceq$ on a set P is a *total order* if for any $a, b \in P$, either $a \preceq b$ or $b \preceq a$. **Why It Matters:**

Total orders are used in sorting and scheduling algorithms, as every pair of elements can be compared.



A *chain* in a poset is a subset where every two elements are comparable, while an *antichain* is a subset where no two distinct elements are comparable.

2.3 Lattices and Boolean Algebras

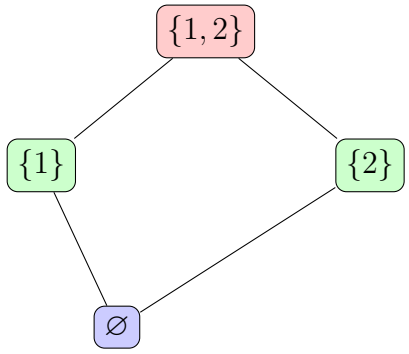
Definition 2.5 (Lattice)**.** A poset $(L, \preceq)$ is a *lattice* if every pair $a, b \in L$ has:

- A unique *least upper bound* (join, $a \vee b$).
- A unique *greatest lower bound* (meet, $a \wedge b$).

Conceptual Insight: Lattices structure elements in a way that mirrors logical operations, forming the foundation of Boolean algebra.

Example 2.6. The power set $\mathcal{P}(S)$ of any set S, ordered by $\subseteq$, forms a lattice where:

$$a \vee b = a \cup b \quad \text{and} \quad a \wedge b = a \cap b.$$



Lemma 2.7 (Distributive Law in Lattices)**.** *A lattice L is* distributive *if for all $a, b, c \in L$:*

$$a \wedge (b \vee c) = (a \wedge b) \vee (a \wedge c)$$

and

$$a \vee (b \wedge c) = (a \vee b) \wedge (a \vee c).$$

Why It Matters: *Distributivity simplifies computations and underpins Boolean logic.*

2.4 Dilworth's Theorem and Related Results

Dilworth's Theorem and the Erdős–Szekeres Theorem capture deep combinatorial properties, showing that large posets necessarily contain either a long chain or a large antichain.

Theorem 2.8 (Dilworth's Theorem)**.** *In any finite poset, the size of the largest antichain equals the minimum number of chains needed to cover the poset.* ***Why It Matters:*** *This theorem is a powerful tool in combinatorics with applications in scheduling and resource allocation.*

Theorem 2.9 (Erdős–Szekeres Theorem)**.** *Any sequence of $n^2 + 1$ distinct real numbers contains a monotonic (increasing or decreasing) subsequence of length $n + 1$.* ***Conceptual Insight:*** *This result shows that order must emerge in any sufficiently large dataset, an idea that resonates in Ramsey theory.*

3 Combinatorial Counting

Counting techniques are central to discrete mathematics. They allow us to enumerate possibilities, calculate probabilities, and analyze algorithms.

3.1 Counting Functions and Subsets

- The number of functions from a finite set A (with $|A| = m$) to a finite set B (with $|B| = n$) is:
$$n^m.$$
- The number of injections from A to B (when $m \leq n$) is:
$$P(n, m) = \frac{n!}{(n-m)!}.$$
- The total number of subsets of an n-element set is:
$$2^n.$$
- The number of k-element subsets is given by the binomial coefficient:
$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Why It Matters: These formulas form the core of combinatorial reasoning, vital for probability theory and algorithm analysis.

Lemma 3.1 (Binomial Sum Identity)**.** *For any non-negative integer n,*

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n.$$

3.2 Permutations and Factorials

Definition 3.2 (Factorial)**.** For $n \in \mathbb{N}$, the *factorial* $n!$ is defined as:

$$n! = n \cdot (n-1) \cdots 2 \cdot 1, \quad \text{with } 0! = 1.$$

Definition 3.3 (Permutation)**.** A *permutation* of a set of n elements is an ordered arrangement. The total number of permutations is $n!$. More generally, the number of ways to order k out of n elements is:

$$P(n,k) = \frac{n!}{(n-k)!}.$$

Lemma 3.4 (Permutations with Repetition)**.** *If there are n objects with n_1 of one type, n_2 of another, ..., n_k of the kth type (with $n_1+n_2+\cdots+n_k = n$), then the number of distinct permutations is:*

$$\frac{n!}{n_1!n_2!\cdots n_k!}.$$

3.3 Binomial Coefficients and the Binomial Theorem

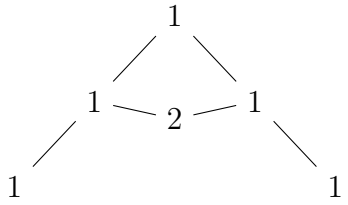
Definition 3.5 (Binomial Coefficient)**.** For non-negative integers n and k with $0 \leq k \leq n$, the binomial coefficient is:

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Lemma 3.6 (Pascal's Identity)**.** *For $0 < k < n$,*

$$\binom{n}{k} = \binom{n-1}{k-1} + \binom{n-1}{k}.$$

Conceptual Insight: *This identity is the backbone of Pascal's Triangle and reveals the recursive structure of binomial coefficients.*



Theorem 3.7 (Binomial Theorem)**.** *For any real numbers x and y and non-negative integer n,*

$$(x+y)^n = \sum_{k=0}^{n} \binom{n}{k} x^k y^{n-k}.$$

Why It Matters: *The binomial theorem is a powerful tool for expanding expressions and has applications in algebra, probability, and numerical methods.*

Example 3.8. For $n = 3$:

$$(x+y)^3 = y^3 + 3xy^2 + 3x^2y + x^3.$$

Definition 3.9 (Multinomial Coefficients)**.** For non-negative integers $n_1, n_2, \ldots, n_k$ satisfying $n_1 + n_2 + \cdots + n_k = n$, the multinomial coefficient is:

$$\binom{n}{n_1, n_2, \ldots, n_k} = \frac{n!}{n_1! n_2! \cdots n_k!}.$$

Theorem 3.10 (Multinomial Theorem)**.** *For any real numbers $x_1, x_2, \ldots, x_k$ and non-negative integer n,*

$$(x_1 + x_2 + \cdots + x_k)^n = \sum_{n_1+n_2+\cdots+n_k=n} \binom{n}{n_1, n_2, \ldots, n_k} \prod_{i=1}^{k} x_i^{n_i}.$$

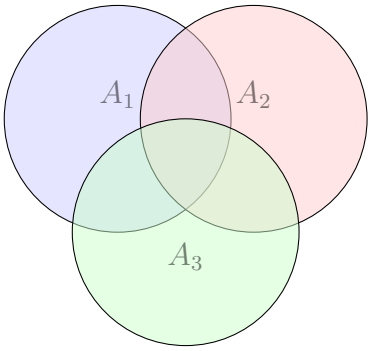
3.4 Inclusion-Exclusion Principle

The inclusion-exclusion principle corrects for over-counting in overlapping sets, a crucial concept in combinatorics.

Theorem 3.11 (Inclusion-Exclusion Principle)**.** *Let $A_1, A_2, \ldots, A_n$ be finite sets. Then:*

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

Why It Matters: *This principle is essential for accurately counting elements in overlapping sets.*



Example 3.12. For two sets A and B:

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

3.5 Derangements and the Hat-Check Problem

Derangements are permutations where no element remains in its original position. They provide insight into probability and combinatorial reasoning.

Definition 3.13 (Derangement)**.** A *derangement* is a permutation σ of $\{1, 2, \ldots, n\}$ with $\sigma(i) \neq i$ for all i. **Why It Matters:** Counting derangements uses inclusion-exclusion and highlights unexpected order in chaos.

Let D_n denote the number of derangements of n objects:

$$D_n = n! \sum_{k=0}^{n} \frac{(-1)^k}{k!}.$$

An alternative recurrence is:

$$D_n = (n-1)(D_{n-1} + D_{n-2}), \quad \text{with } D_0 = 1 \text{ and } D_1 = 0.$$

Example 3.14. For $n = 3$:

$$D_3 = 3! \left(1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!}\right) = 6 \left(1 - 1 + \frac{1}{2} - \frac{1}{6}\right) = 6 \left(\frac{1}{3}\right) = 2.$$

Another useful expression is:

$$D_n = \left\lfloor \frac{n!}{e} + \frac{1}{2} \right\rfloor,$$

where e is the base of the natural logarithm.

Conclusion

In these notes we have explored several core topics in discrete mathematics:

- **Numbers and Sets:** Fundamental structures that underpin all of mathematics.
- **Proof Techniques:** Methods like induction that enable us to rigorously prove infinite families of statements.
- **Functions and Relations:** The language for describing mappings and associations, essential in modeling processes.
- **Ordering:** Concepts such as partial orders, total orders, and lattices that bring structure and hierarchy.
- **Combinatorial Counting:** Techniques to enumerate possibilities, critical in probability and algorithm analysis.

These diagrams serve as visual aids to help cement your understanding of the concepts. Feel free to expand on them or adjust as your understanding grows.