Binary Relations

Discrete Math, Fall 2025

Konstantin Chukharev

Set Theory

- Operations & laws
- Power sets
- Russell's paradox
- ZFC axioms
- Cartesian products
- Cardinality

Binary Relations

- Relation properties
- Equivalence relations
- Functions
- Partial orders
- Lattices
- Well-orders

Boolean Algebra

- Truth tables & laws
- Logic circuits
- Normal forms
- Karnaugh maps
- Binary Decision Diagrams (BDDs)

Formal Logic

- Syntax & semantics
- Natural deduction
- Soundness & completeness
- Categorical logic
- First-order logic

Relations

"In mathematics you don't understand things. You just get used to them."

— John von Neumann


René Descartes


Évariste Galois


Ernst Schröder


Michael Rabin


Herbert Wilf

What is a Relation?

Relations capture connections between elements of sets.

Example (Everyday examples):

- "is less than" relates *numbers*: 3 is *related to* 5 because $3 < 5$
- "is a parent of" relates *people*: Alice is *related to* Bob if Alice is Bob's parent
- "divides" relates *integers*: 2 is *related to* 6 because $2 \mid 6$

Intuition: A relation tells us which pairs of elements are connected.

Definition 1: A *binary relation* R from set A to set B is a subset of the Cartesian product:

$$R \subseteq A \times B$$

Each ordered pair $\langle a, b \rangle \in R$ means "a is related to b."

Notation for Relations

Notation: If $R \subseteq A \times B$, we write:

- $a \; R \; b$ to mean $\langle a, b \rangle \in R$ (element a is related to element b)
- $a \not R \; b$ to mean $\langle a, b \rangle \notin R$ (element a is not related to element b)

Example: Let $R = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 1, 3 \rangle\}$ be a relation on naturals.

- We can write $1 \; R \; 2$ instead of $\langle 1, 2 \rangle \in R$
- We can write $2 \not R \; 1$ instead of $\langle 2, 1 \rangle \notin R$

Key insight: Order matters! $a \; R \; b$ and $b \; R \; a$ are different statements.

Types of Relations

Definition 2: A binary relation $R \subseteq A \times B$ is:
- *Heterogeneous* if A and B are different sets
- *Homogeneous* if $A = B$ (we write $R \subseteq M^2$)

Example (Heterogeneous relation): Let Students $= \{\text{Alice}, \text{Bob}\}$ and Subjects $= \{\text{Math}, \text{Algorithms}\}$.

The relation "likes" (between students and subjects) might be:

$$R = \{\langle \text{Alice}, \text{Math} \rangle, \langle \text{Alice}, \text{Algorithms} \rangle, \langle \text{Bob}, \text{Algorithms} \rangle\}$$

We write "Alice R Math" to denote that "Alice likes Math".

Example (Homogeneous relation): The "less than" relation on $\mathbb{N}$:

$$R = \{\langle n, k \rangle \mid n, k \in \mathbb{N} \text{ and } n < k\} \subseteq \mathbb{N}^2$$

For instance: $2\ R\ 3$ (since $2 < 3$) and $3 \not R\ 2$ (since not $3 < 2$).

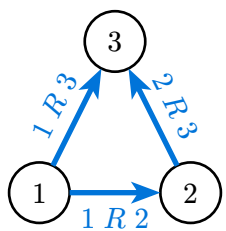
Visualizing Relations: Directed Graphs

Relations can be visualized as graphs, making their structure easier to understand.

Definition 3: A homogeneous relation $R \subseteq M^2$ is represented as a *directed graph*:

- Each element of M becomes a vertex
- Draw edge $x \to y$ whenever $x \; R \; y$ (*i.e.*, $\langle x, y \rangle \in R$)

Example: Relation $R = \{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 1, 3 \rangle\}$ on $M = \{1, 2, 3\}$:



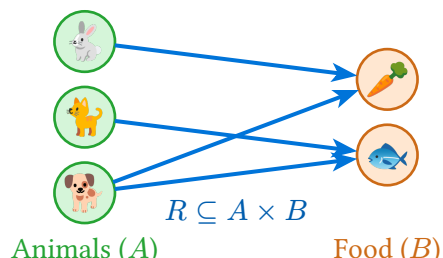
Visualizing Relations: Bipartite Graphs

Heterogeneous relations connect elements from two different sets.

Definition 4: A heterogeneous relation $R \subseteq A \times B$ can be represented as a *bipartite graph*:

- Left partition: vertices for elements of A
- Right partition: vertices for elements of B
- Draw edge $a \to b$ whenever $a \; R \; b$ (*i.e.*, $\langle a, b \rangle \in R$)

Example: Animals $A = \{$🐇, 🐈, 🐕$\}$ and food $B = \{$🥕, 🐟$\}$. Relation R: "likes to eat".



Matrix Representation

Definition 5: A relation $R \subseteq A \times B$ with $A = \{a_1, ..., a_m\}$ and $B = \{b_1, ..., b_n\}$ is represented by an $m \times n$ *boolean matrix* $[\![R]\!]$:

$$[\![R]\!][i, j] = \begin{cases} 1 \text{ if } a_i \; R \; b_j \\ 0 \text{ if } a_i \not R \; b_j \end{cases}$$

Example: Let $A = \{a, b, c\}$, $B = \{x, y\}$, and $R = \{\langle a, x\rangle, \langle b, x\rangle, \langle c, y\rangle\}$.

Matrix representation with rows = A and columns = B:

$$[\![R]\!] = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \iff \begin{cases} a \; R \; x, \;\; a \not R \; y \\ b \; R \; x, \;\; b \not R \; y \\ c \not R \; x, \;\; c \; R \; y \end{cases}$$

Why matrices? Matrix operations (multiply, transpose) match relation operations (compose, inverse).

Special Relations

Definition 6: For any set M, we have:

- *Empty relation*: $\emptyset \subseteq M^2$ — No pairs at all; nothing is related to anything.
- *Identity relation*: $I_M = \{\langle x, x\rangle \mid x \in M\}$ — Each element is related only to itself.
- *Universal relation*: $U_M = M^2$ — All possible pairs; everything is related to everything.

Example: For $M = \{a, b, c\}$:

- *Empty*: $\emptyset$ (no relations)
- *Identity*: $I_M = \{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle\}$ (3 pairs on diagonal)
- *Universal*: $U_M = \{\langle a, a\rangle, \langle a, b\rangle, \langle a, c\rangle, \langle b, a\rangle, \langle b, b\rangle, \langle b, c\rangle, \langle c, a\rangle, \langle c, b\rangle, \langle c, c\rangle\}$ (all $3^2 = 9$ pairs)

Matrix representations:

$$[\![\emptyset]\!] = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \qquad [\![I_M]\!] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad [\![U_M]\!] = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Set Operations on Relations

Since relations are sets of pairs, we can apply standard set operations.

Definition 7: For relations $R, S \subseteq A \times B$:

- *Union*: $R \cup S = \{\langle a, b\rangle \mid \langle a, b\rangle \in R \vee \langle a, b\rangle \in S\}$
- *Intersection*: $R \cap S = \{\langle a, b\rangle \mid \langle a, b\rangle \in R \wedge \langle a, b\rangle \in S\}$
- *Complement*: $\overline{R} = (A \times B) \setminus R = \{\langle a, b\rangle \mid \langle a, b\rangle \notin R\}$

Example: Let $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$ and $S = \{\langle 1, 2\rangle, \langle 3, 4\rangle\}$ on $\mathbb{N}$.

- $R \cup S = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}$ (all pairs from either relation)
- $R \cap S = \{\langle 1, 2\rangle\}$ (only common pairs)
- If we restrict to $M = \{1, 2, 3\}$, then $\overline{R} = \{\langle 1, 1\rangle, \langle 1, 3\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \langle 3, 1\rangle, \langle 3, 2\rangle, \langle 3, 3\rangle\}$

Matrix view: Union = OR, Intersection = AND, Complement = NOT (element-wise on matrices).

Inverse Relation

Definition 8: For $R \subseteq A \times B$, the *inverse* (or *converse*, or *dual*) relation is:

$$R^{-1} = \{\langle b, a\rangle \mid \langle a, b\rangle \in R\} \subseteq B \times A$$

Note: R^{-1} reverses all connections. If $a\ R\ b$, then $b\ R^{-1}\ a$.

Note: Other notations: R^T (transpose), $R^{\mathbb{C}}$ (converse), R° (reciprocal).

Example:

- If $R = \{\langle 1, x\rangle, \langle 2, y\rangle, \langle 2, z\rangle\}$, then $R^{-1} = \{\langle x, 1\rangle, \langle y, 2\rangle, \langle z, 2\rangle\}$.
- For order relations:
  - $(<)^{-1} = (>)$ (inverse of "less than" is "greater than")
  - $(\leq)^{-1} = (\geq)$ (inverse of "less or equal" is "greater or equal")
- For the "parent of" relation: $(\text{parent of})^{-1}$ = "child of"

Matrix view: $[\![R^{-1}]\!]$ is the transpose of $[\![R]\!]$.

Properties of Relations

Reflexivity

A relation is reflexive if every element is related to itself.

Definition 9: $R \subseteq M^2$ is *reflexive* if:

$$\forall x \in M.\, x\ R\ x$$

Examples:

- *Reflexive:* $\leq$, $=$, "is the same age as"
- *Not reflexive:* $<$, $\neq$, "is parent of"

Graph view: Every vertex has a self-loop.

Matrix view: All diagonal entries are 1.

Symmetry

A relation is symmetric if the relation goes both ways.

Definition 10: $R \subseteq M^2$ is *symmetric* if:

$$\forall x, y \in M. (x \ R \ y) \to (y \ R \ x)$$

Example:

- *Symmetric:* $=$, "is sibling of", "is married to"
- *Not symmetric:* $\leq$, "is parent of", "likes"

Graph view: If there's an edge $x \to y$, there's also $y \to x$.

Matrix view: Matrix equals its transpose.

Transitivity

A relation is transitive if connections chain together.

Definition 11: $R \subseteq M^2$ is *transitive* if:

$$\forall x, y, z \in M. (x \ R \ y \land y \ R \ z) \to (x \ R \ z)$$

Example:

- *Transitive:* $\leq$, $=$, "is ancestor of"
- *Not transitive:* "is parent of" (parent of parent is grandparent, not parent)

Intuition: If you can reach z from x through y, you can reach z directly from x.

Irreflexivity

A relation is irreflexive if no element is related to itself.

Definition 12: $R \subseteq M^2$ is *irreflexive* if:

$$\forall x \in M.\, x \not R\, x$$

Example:

- *Irreflexive:* $<$, $\neq$, "is parent of"
- *Not irreflexive:* $\leq$, $=$

Note: Irreflexive is *not* the same as "not reflexive"! A relation can be neither reflexive nor irreflexive.

Antisymmetry

A relation is antisymmetric if different elements can't be mutually related.

Definition 13: $R \subseteq M^2$ is *antisymmetric* if:

$$\forall x, y \in M. (x \ R \ y \land y \ R \ x) \to (x = y)$$

Note: Alternative definition:

$$\forall x, y \in M. (x \neq y) \to (x \ R \ y \to y \not R \ x)$$

Example:

- *Antisymmetric:* $\leq$, $\subseteq$, "divides" (on positive integers)
- *Not antisymmetric:* "is sibling of", "is friend of"

Intuition: At most one direction exists between distinct elements.

Asymmetry

A relation is asymmetric if it never goes both ways.

Definition 14: $R \subseteq M^2$ is *asymmetric* if:

$$\forall x, y \in M. (x \ R \ y) \to (y \not R \ x)$$

Example:

- *Asymmetric:* $<$, "is parent of"
- *Not asymmetric:* $\leq$, $=$

Note: Asymmetric = irreflexive + antisymmetric. Asymmetry is the strongest directional property.

Properties: Important Notes

Note: Properties are *not always opposites*:

- Reflexive vs irreflexive: A relation can be neither (*e.g.*, $R = \{\langle 1, 1\rangle, \langle 1, 2\rangle\}$ on $M = \{1, 2\}$)
- Symmetric vs antisymmetric: A relation can be both (*e.g.*, identity I_M)

Note: *Empty set edge case:* On $M = \emptyset$, the empty relation is vacuously[1] reflexive, irreflexive, symmetric, antisymmetric, asymmetric, and transitive!

Example (Combining properties): Common combinations:

- *Equivalence relation:* reflexive + symmetric + transitive (*e.g.*, $=$)
- *Partial order:* reflexive + antisymmetric + transitive (*e.g.*, $\leq$)
- *Strict order:* irreflexive + antisymmetric + transitive (*e.g.*, $<$)

[1] A universal statement "$\forall x \in \emptyset.\, P(x)$" is true because there are no counterexamples.

Additional Properties

Definition 15: A relation $R \subseteq M^2$ is:

- *Coreflexive*: If $R \subseteq I_M$ (only self-loops are allowed).

$$\forall x, y \in M. (x \ R \ y) \to (x = y)$$

- *Right Euclidean*: If x relates to both y and z, then y and z are related.

$$\forall x, y, z \in M. (x \ R \ y \land x \ R \ z) \to (y \ R \ z)$$

- *Left Euclidean*: If both y and z relate to x, then they relate to each other.

$$\forall x, y, z \in M. (y \ R \ x \land z \ R \ x) \to (y \ R \ z)$$

Example:

- Identity relation I_M is coreflexive (and any subset of I_M)
- Equality "=" is both left and right Euclidean
- Equivalence relations are Euclidean in both directions

Equivalence Relations

Equivalence Relations

Definition 16: A relation $R \subseteq M^2$ is an *equivalence relation* if it is reflexive, symmetric and transitive.

Example (Equality): The *identity relation* $I_M = \{\langle x, x\rangle \mid x \in M\}$ is an equivalence relation on any set M.

Note: The identity relation is just the common equality relation "$=$".

Verification:

- **Reflexive:** $x \; I_M \; x$ for all $x \in M$ (by definition of I_M) ✓
- **Symmetric:** If $x \; I_M \; y$, then $x = y$, thus $y = x$, so $y \; I_M \; x$ ✓
- **Transitive:** If $x \; I_M \; y$ and $y \; I_M \; z$, then $x = y = z$, so $x \; I_M \; z$ ✓

This is the "finest" possible equivalence relation — it distinguishes every element.

Equivalence Classes

Definition 17: Let $R \subseteq M^2$ be an equivalence relation on a set M. The *equivalence class* of an element $x \in M$ under R is the set of all elements related to x:

$$[x]_R = \{y \in M \mid x \; R \; y\}$$

Example (Equality): For the identity relation I_M on set $M = \{a, b, c\}$:

- $[a]_{I_M} = \{a\}$ (only a is equal to a)
- $[b]_{I_M} = \{b\}$ (only b is equal to b)
- $[c]_{I_M} = \{c\}$ (only c is equal to c)

Each element forms its own equivalence class under equality.

Examples of Equivalence Relations

Example (Modular arithmetic): For any positive integer n, *congruence modulo n* on $\mathbb{Z}$ is defined by:

$$a \equiv b \pmod{n} \quad \text{iff} \quad n \mid (b - a)$$

Verification:

- **Reflexive:** $a \equiv a \pmod{n}$ since $n \mid (a - a) \iff n \mid 0$ ✓
- **Symmetric:** If $a \equiv b \pmod{n}$, then $n \mid (b - a) \iff n \mid (a - b)$, thus $b \equiv a \pmod{n}$ ✓
- **Transitive:** If $a \equiv b \pmod{n}$ and $b \equiv c \pmod{n}$, then $n \mid (b - a)$ and $n \mid (c - b)$, so $n \mid ((c - b) + (b - a)) \iff n \mid (c - a)$, thus $a \equiv c \pmod{n}$ ✓

Equivalence classes (remainders): Let $n = 5$ and $M = \{0, 1, ..., 9\}$.

- $[0]_{\equiv} = \{0, 5\} = \{x \in M \mid x \equiv 0 \pmod{5}\}$
- $[1]_{\equiv} = \{1, 6\} = \{x \in M \mid x \equiv 1 \pmod{5}\}$
- $[2]_{\equiv} = \{2, 7\} = \{x \in M \mid x \equiv 2 \pmod{5}\}$
- $[3]_{\equiv} = \{3, 8\} = \{x \in M \mid x \equiv 3 \pmod{5}\}$
- $[4]_{\equiv} = \{4, 9\} = \{x \in M \mid x \equiv 4 \pmod{5}\}$

Examples of Equivalence Relations [2]

Example (Same absolute value): On $M = \{-3, -2, -1, 0, 1, 2, 3\}$, define relation R by:

$$x \; R \; y \quad \text{iff} \quad |x| = |y|$$

Verification:

- **Reflexive:** $|x| = |x|$ for all x ✓
- **Symmetric:** If $|x| = |y|$, then $|y| = |x|$ ✓
- **Transitive:** If $|x| = |y|$ and $|y| = |z|$, then $|x| = |z|$ ✓

Equivalence classes:

- $[0]_R = \{0\}$
- $[1]_R = [-1]_R = \{-1, 1\}$
- $[2]_R = [-2]_R = \{-2, 2\}$
- $[3]_R = [-3]_R = \{-3, 3\}$

Each positive number is equivalent to its negative counterpart.

Examples of Equivalence Relations [3]

Example (Same string length): On the set of all finite strings Σ^* over alphabet Σ, define:

$$s_1 \; R \; s_2 \quad \text{iff} \quad |s_1| = |s_2|$$

where $|s|$ denotes the length of string s.

Verification:

- **Reflexive:** Every string has the same length as itself ✓
- **Symmetric:** If two strings have the same length, the relation is symmetric ✓
- **Transitive:** If $|s_1| = |s_2|$ and $|s_2| = |s_3|$, then $|s_1| = |s_3|$ ✓

Equivalence classes: $[s]_R = \{t \in \Sigma^* \mid |t| = |s|\}$

For example, over $\Sigma = \{a, b\}$:

- $[\varepsilon]_R = \{\varepsilon\}$ (empty string)
- $[\mathrm{a}]_R = \{\mathrm{a}, \mathrm{b}\}$ (all strings of length 1)
- $[\mathrm{ab}]_R = \{\mathrm{aa}, \mathrm{ab}, \mathrm{ba}, \mathrm{bb}\}$ (all strings of length 2)

Examples of Equivalence Relations [4]

Example (Living in the same city): Let P be the set of people, and define relation R by:

$$p_1 \; R \; p_2 \quad \text{iff} \quad p_1 \text{ and } p_2 \text{ live in the same city}$$

Verification:

- **Reflexive:** Every person lives in the same city as themselves ✓
- **Symmetric:** If person A and B live in the same city, then B and A live in the same city ✓
- **Transitive:** If A and B live in the same city, and B and C do so, then A and C live in the same city ✓

Equivalence classes: Each equivalence class consists of all people living in the same city.

- $[\text{Alice}]_R$ = all people living in Alice's city
- This naturally partitions the population by cities

Application: This relation captures a common social grouping based on location.

Examples of Equivalence Relations [5]

Example (Similarity of triangles): Let T be the set of all triangles in the plane. Define relation $\sim$ by:

$$\triangle_1 \sim \triangle_2 \quad \text{iff} \quad \triangle_1 \text{ and } \triangle_2 \text{ are similar2}$$

Verification:

- **Reflexive:** Every triangle is similar to itself ✓
- **Symmetric:** If triangle A is similar to triangle B, then triangle B is similar to triangle A ✓
- **Transitive:** If $A \sim B$ and $B \sim C$, then $A \sim C$ (similarity is transitive) ✓

Equivalence classes: Each class consists of all triangles with the same shape (but possibly different sizes).

- All equilateral triangles form one equivalence class
- All right triangles with legs in ratio 3:4:5 form another equivalence class

Geometric significance: This relation captures the concept of "same shape, different size."

[2] Two triangles are similar if their corresponding angles are equal.

Examples of Equivalence Relations [6]

Example (Rational numbers): On the set of ordered pairs $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$, define:

$$\langle a, b \rangle \sim \langle c, d \rangle \quad \text{iff} \quad a \cdot d = b \cdot c$$

This relation is used to construct rational numbers $\mathbb{Q}$ from integer pairs.

Verification:

- **Reflexive:** $\langle a, b \rangle \sim \langle a, b \rangle$ since $a \cdot b = b \cdot a$ ✓
- **Symmetric:** If $a \cdot d = b \cdot c$, then $c \cdot b = d \cdot a$ ✓
- **Transitive:** If $a \cdot d = b \cdot c$ and $c \cdot f = d \cdot e$, then $a \cdot f = b \cdot e$ ✓

Equivalence classes: Each equivalence class represents a rational number.

- $[\langle 1, 2 \rangle]_\sim = \{\langle 1, 2 \rangle, \langle 2, 4 \rangle, \langle 3, 6 \rangle, \langle -1, -2 \rangle, \ldots\}$ represents $\frac{1}{2}$
- $[\langle 3, 4 \rangle]_\sim = \{\langle 3, 4 \rangle, \langle 6, 8 \rangle, \langle -3, -4 \rangle, \ldots\}$ represents $\frac{3}{4}$

Mathematical significance: This construction shows how equivalence relations create new mathematical objects (rationals) from simpler ones (integers).

Quotient Sets

Definition 18: The *quotient set* of M by the equivalence relation R is the set of all equivalence classes:

$$M/_R = \{[x]_R \mid x \in M\}$$

Example: Consider $M = \{0, 1, 2, 3, 4, 5\}$ with the equivalence relation "congruent modulo 3":

$$x \equiv y \pmod 3 \quad \text{iff} \quad 3 \mid (x - y)$$

Equivalence classes:

- $[0]_\equiv = \{0, 3\} = \{x \in M \mid x \equiv 0 \pmod 3\}$
- $[1]_\equiv = \{1, 4\} = \{x \in M \mid x \equiv 1 \pmod 3\}$
- $[2]_\equiv = \{2, 5\} = \{x \in M \mid x \equiv 2 \pmod 3\}$

Quotient set:

$$M/_\equiv = \{\{0, 3\}, \{1, 4\}, \{2, 5\}\}$$

Note that $[3]_\equiv = [0]_\equiv = \{0, 3\}$, so we don't get a new equivalence class, since $M/_R$ is a *set*.

Quotient Sets [2]

Example: Let $M = \{\mathrm{a}, \mathrm{ab}, \mathrm{abc}, \mathrm{x}, \mathrm{xy}, \mathrm{z}\}$ with equivalence relation R defined by:

$$s_1 \; R \; s_2 \quad \text{iff} \quad |s_1| = |s_2|$$

where $|s|$ denotes the length of string s.

Equivalence classes by length:

- $[\mathrm{a}]_R = \{\mathrm{a}, \mathrm{x}, \mathrm{z}\}$ (strings of length 1)
- $[\mathrm{ab}]_R = \{\mathrm{ab}, \mathrm{xy}\}$ (strings of length 2)
- $[\mathrm{abc}]_R = \{\mathrm{abc}\}$ (strings of length 3)

Quotient set:

$$M/_R = \{\{\mathrm{a}, \mathrm{x}, \mathrm{z}\}, \{\mathrm{ab}, \mathrm{xy}\}, \{\mathrm{abc}\}\}$$

This partitions strings by their length, creating 3 equivalence classes.

Quotient Sets [3]

Example (Construction of rational numbers): Consider the set of ordered pairs $M = \mathbb{Z} \times (\mathbb{Z} \setminus \{0\})$.

Define the equivalence relation $\sim$ by:

$$\langle a, b\rangle \sim \langle c, d\rangle \quad \text{iff} \quad a \cdot d = b \cdot c$$

Equivalence classes: Each represents a rational number in its "reduced form".

Representative	Equivalence Class	Rational Number
$\langle 1, 2\rangle$	$\{\langle 1, 2\rangle, \langle 2, 4\rangle, \langle 3, 6\rangle, \langle -1, -2\rangle, \langle -2, -4\rangle, \ldots\}$	$1/2$
$\langle 0, 1\rangle$	$\{\langle 0, 1\rangle, \langle 0, 2\rangle, \langle 0, -3\rangle, \langle 0, 7\rangle, \ldots\}$	0
$\langle 3, 4\rangle$	$\{\langle 3, 4\rangle, \langle 6, 8\rangle, \langle -3, -4\rangle, \langle 9, 12\rangle, \ldots\}$	$3/4$
$\langle -5, 3\rangle$	$\{\langle -5, 3\rangle, \langle 5, -3\rangle, \langle -10, 6\rangle, \langle 10, -6\rangle, \ldots\}$	$-5/3$

Quotient set: All equivalence classes together form the set of rational numbers:

$$\mathbb{Q} := (\mathbb{Z} \times (\mathbb{Z} \setminus \{0\}))/_{\sim} = \{[\langle a, b\rangle]_{\sim} \mid \langle a, b\rangle \in \mathbb{Z} \times (\mathbb{Z} \setminus \{0\})\}$$

Quotient Sets [4]

Interpretation: Each equivalence class $[\langle a, b\rangle]_\sim$ corresponds exactly to the rational number $\frac{a}{b}$.

- Different representations $\langle a, b\rangle$ and $\langle c, d\rangle$ belong to the same equivalence class iff they represent the same fraction: $\frac{a}{b} = \frac{c}{d}$
- The condition $a \cdot d = b \cdot c$ is the cross-multiplication test for fraction equality

Operations on the quotient set: We can define arithmetic operations on $\mathbb{Q}$ by:

$$[\langle a, b\rangle]_\sim + [\langle c, d\rangle]_\sim := [\langle ad + bc, bd\rangle]_\sim$$
$$[\langle a, b\rangle]_\sim \cdot [\langle c, d\rangle]_\sim := [\langle ac, bd\rangle]_\sim$$

These operations are *well-defined* because the result doesn't depend on the choice of representatives.

Mathematical significance:

- This construction shows that $\mathbb{Q}$ is literally *the* quotient set $\mathbb{Z} \times (\mathbb{Z} \setminus \{0\})/_\sim$
- It demonstrates how equivalence relations and quotient sets create new mathematical structures
- This is the rigorous foundation for rational number arithmetic taught in elementary school

Quotient Sets [5]

Example: Consider the set of all points in the plane $M = \mathbb{R}^2$ with the equivalence relation:

$$\langle x_1, y_1 \rangle \sim \langle x_2, y_2 \rangle \quad \text{iff} \quad x_1 + y_1 = x_2 + y_2$$

This relation groups points that lie on the same line of the form $x + y = c$ for some constant c.

Equivalence classes:

- $[\langle 0, 0 \rangle]_\sim = \{\langle x, y \rangle \mid x + y = 0\}$ (the line $x + y = 0$)
- $[\langle 1, 0 \rangle]_\sim = \{\langle x, y \rangle \mid x + y = 1\}$ (the line $x + y = 1$)
- $[\langle 0, 2 \rangle]_\sim = \{\langle x, y \rangle \mid x + y = 2\}$ (the line $x + y = 2$)

Quotient set:

$$\mathbb{R}^2/_\sim = \{L_c \mid c \in \mathbb{R}\}$$

where $L_c = \{\langle x, y \rangle \mid x + y = c\}$ is the line with equation $x + y = c$.

Each equivalence class is an entire line, and the quotient set consists of all such parallel lines.

Quotient Sets [6]

Example: Let $M = \{\text{apple}, \text{ant}, \text{banana}, \text{bee}, \text{cherry}, \text{cat}\}$ with equivalence relation:

$$w_1 \; R \; w_2 \quad \text{iff} \quad \text{first letter of } w_1 = \text{first letter of } w_2$$

Equivalence classes:

- $[\text{apple}]_R = \{\text{apple}, \text{ant}\}$ (words starting with 'a')
- $[\text{banana}]_R = \{\text{banana}, \text{bee}\}$ (words starting with 'b')
- $[\text{cherry}]_R = \{\text{cherry}, \text{cat}\}$ (words starting with 'c')

Quotient set:

$$M/_R = \{\{\text{apple}, \text{ant}\}, \{\text{banana}, \text{bee}\}, \{\text{cherry}, \text{cat}\}\}$$

This creates a dictionary-like grouping by first letter.

Quotient Sets [7]

Example (Trivial quotient sets):

Identity relation: If $R = I_M$ (identity relation), then each element forms its own equivalence class:

$$M/_{I_M} = \{\{x\} \mid x \in M\}$$

The quotient set has the same cardinality as the original set.

Universal relation: If $R = M \times M$ (universal relation), then all elements are equivalent:

$$M/_{M \times M} = \{M\}$$

The quotient set contains exactly one equivalence class — the entire set M.

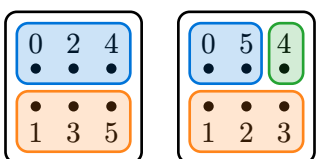
Set Partitions

Definition 19: A *partition* $\mathcal{P}$ of a set M is a family of non-empty, pairwise-disjoint subsets whose union is M:

- (Non-empty) $\forall B \in \mathcal{P}.\, (B \neq \emptyset)$
- (Disjoint) $\forall B_1, B_2 \in \mathcal{P}.\, (B_1 \neq B_2) \to (B_1 \cap B_2 = \emptyset)$
- (Cover) $\bigcup_{B \in \mathcal{P}} B = M$

Elements of $\mathcal{P}$ are *blocks* (or *cells*).

Example: For $M = \{0, 1, 2, 3, 4, 5\}$: $\{\{0, 2, 4\}, \{1, 3, 5\}\}$ and $\{\{0, 5\}, \{1, 2, 3\}, \{4\}\}$ are partitions.



Partitions and Equivalence Relations

Theorem 1 (Equivalences $\Leftrightarrow$ Partitions): Each equivalence relation R on M yields the partition $\mathcal{P}_R = \{[x]_R \mid x \in M\}$. Each partition $\mathcal{P}$ yields an equivalence $R_{\mathcal{P}}$ given by $\langle x, y\rangle \in R_{\mathcal{P}}$ iff x and y lie in the same block. These constructions invert one another.

Proof *(Sketch)*: Classes of an equivalence are non-empty, disjoint, and cover M. Conversely, "same block" relation is reflexive, symmetric, transitive. Composing the two constructions returns exactly the starting equivalence relation or partition (they are mutually inverse up to equality of sets of ordered pairs). □

Composition of Relations

Composition of Relations

Definition 20: The *composition* of two relations $R \subseteq A \times B$ and $S \subseteq B \times C$ is defined as:

$$R \mathbin{;} S = S \circ R = \{\langle a, c\rangle \mid \exists b \in B.\, (a\ R\ b) \wedge (b\ S\ c)\}$$

Example: Let $A = \{1, 2, 3\}$, $B = \{a, b, c, d\}$, $C = \{x, y\}$ with relations:

- $R = \{\langle 1, a\rangle, \langle 1, b\rangle, \langle 2, c\rangle, \langle 3, d\rangle\} \subseteq A \times B$
- $S = \{\langle a, x\rangle, \langle b, y\rangle, \langle c, x\rangle\} \subseteq B \times C$

To find $R \mathbin{;} S$, we look for pairs $\langle i, z\rangle$ where there exists w such that $\langle i, w\rangle \in R$ and $\langle w, z\rangle \in S$:

From 1: can reach a and b via R

- a connects to x via $S \Rightarrow \langle 1, x\rangle$ is in the composition
- b connects to y via $S \Rightarrow \langle 1, y\rangle$ is in the composition

From 2: can reach c via R

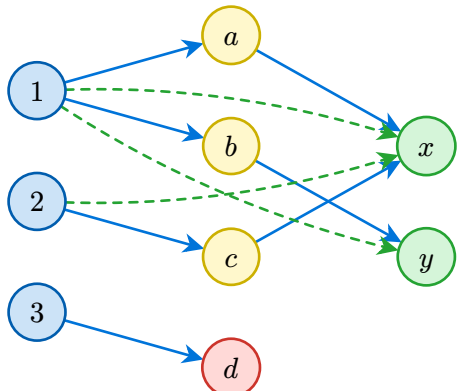
- c connects to x via $S \Rightarrow \langle 2, x\rangle$ is in the composition

From 3: can reach d via R

Composition of Relations [2]

- d has no outgoing connections in $S \Rightarrow$ no pairs from 3 in the composition

Therefore: $R \mathbin{;} S = \{\langle 1, x\rangle, \langle 1, y\rangle, \langle 2, x\rangle\}$



Edges legend:

- Solid: Relations R and S
- Dashed: Composition $R \mathbin{;} S$
- Red: Dead end (no outgoing path)

Examples of Composition

Example (Composition in a social network): Consider three sets:

- People $= \{\text{Alice}, \text{Bob}, \text{Carol}\}$
- Skills $= \{\text{Python}, \text{Design}, \text{Management}\}$
- Projects $= \{\text{WebApp}, \text{Mobile}, \text{Analytics}\}$

Define relations:

- HasSkill $= \{\langle \text{Alice}, \text{Python} \rangle, \langle \text{Alice}, \text{Design} \rangle, \langle \text{Bob}, \text{Python} \rangle, \langle \text{Carol}, \text{Management} \rangle\}$
- RequiresSkill $= \{\langle \text{Python}, \text{WebApp} \rangle, \langle \text{Python}, \text{Analytics} \rangle, \langle \text{Design}, \text{WebApp} \rangle, \langle \text{Management}, \text{Mobile} \rangle\}$

The composition HasSkill ; RequiresSkill gives us CanWorkOn:

- Alice can work on WebApp (via Python AND Design)
- Alice can work on Analytics (via Python)
- Bob can work on WebApp (via Python)
- Bob can work on Analytics (via Python)
- Carol can work on Mobile (via Management)

So: CanWorkOn $=$
$\{\langle \text{Alice}, \text{WebApp} \rangle, \langle \text{Alice}, \text{Analytics} \rangle, \langle \text{Bob}, \text{WebApp} \rangle, \langle \text{Bob}, \text{Analytics} \rangle, \langle \text{Carol}, \text{Mobile} \rangle\}$

Examples of Composition [2]

Example (Matrix composition): Relations can be composed using Boolean matrix multiplication.

Given $R \subseteq \{1, 2\} \times \{a, b\}$ and $S \subseteq \{a, b\} \times \{x, y\}$:

$$[\![R]\!] = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \quad [\![S]\!] = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

The composition $[\![R \mathbin{;} S]\!] = [\![R]\!] \odot [\![S]\!]$ using Boolean matrix multiplication:

$$[\![R \mathbin{;} S]\!] = \begin{bmatrix} (1 \wedge 1) \vee (0 \wedge 0) & (1 \wedge 1) \vee (0 \wedge 1) \\ (1 \wedge 1) \vee (1 \wedge 0) & (1 \wedge 1) \vee (1 \wedge 1) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

This means $R \mathbin{;} S = \{\langle 1, x\rangle, \langle 1, y\rangle, \langle 2, x\rangle, \langle 2, y\rangle\}$ (the universal relation).

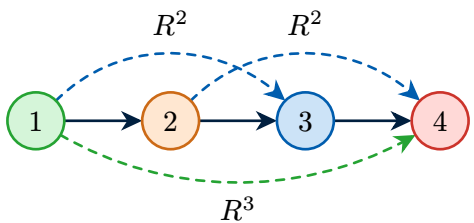
Powers of Relations

Definition 21: For a homogeneous relation $R \subseteq M^2$, we define *powers* of R:

- $R^0 = I_M$ (identity relation)
- $R^1 = R$
- $R^{n+1} = R^n \circ R$ for $n \geq 1$

Example: Let $M = \{1, 2, 3, 4\}$ and $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}$ (successor relation).

- $R^1 = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 3, 4\rangle\}$
- $R^2 = \{\langle 1, 3\rangle, \langle 2, 4\rangle\}$ (two steps)
- $R^3 = \{\langle 1, 4\rangle\}$ (three steps)
- $R^4 = \emptyset$ (no four-step paths)

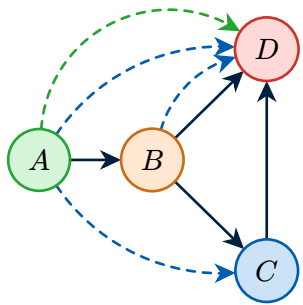


Paths in Graphs

Example (Path composition in a graph): Consider a directed graph with vertices $\{A, B, C, D\}$ and relation R representing direct edges: $R = \{\langle A, B\rangle, \langle B, C\rangle, \langle B, D\rangle, \langle C, D\rangle\}$

Powers of R represent paths of different lengths:

- $R^1 = R$ (direct connections)
- $R^2 = R \circ R$ (2-step paths):
  - $\langle A, C\rangle$: path $A \to B \to C$
  - $\langle A, D\rangle$: path $A \to B \to D$
  - $\langle B, D\rangle$: path $B \to C \to D$
  - So $R^2 = \{\langle A, C\rangle, \langle A, D\rangle, \langle B, D\rangle\}$.
- $R^3 = R^2 \circ R$ (3-step paths):
  - $\langle A, D\rangle$: path $A \to B \to C \to D$
  - No more 3-step paths!
  - So $R^3 = \{\langle A, D\rangle\}$.
- $R^4 = \emptyset$ (no 4-step paths)



Associativity of Composition

Theorem 2: Composition of relations is associative: $(R \,;\, S) \,;\, T = R \,;\, (S \,;\, T)$.

Proof: Let $R \subseteq A \times B$, $S \subseteq B \times C$, and $T \subseteq C \times D$ be three relations.

($\subseteq$): Let $\langle a, d\rangle \in (R \,;\, S) \,;\, T$.

- By definition of composition: $\exists c \in C.\, (\langle a, c\rangle \in R \,;\, S) \wedge (\langle c, d\rangle \in T)$.
- Since $\langle a, c\rangle \in R \,;\, S$, we have: $\exists b \in B.\, (\langle a, b\rangle \in R) \wedge (\langle b, c\rangle \in S)$.
- From $\langle b, c\rangle \in S$ and $\langle c, d\rangle \in T$, we have: $\langle b, d\rangle \in S \,;\, T$.
- From $\langle a, b\rangle \in R$ and $\langle b, d\rangle \in S \,;\, T$, we have: $\langle a, d\rangle \in R \,;\, (S \,;\, T)$.

($\supseteq$): Let $\langle a, d\rangle \in R \,;\, (S \,;\, T)$.

- By definition of composition: $\exists b \in B.\, (\langle a, b\rangle \in R) \wedge (\langle b, d\rangle \in S \,;\, T)$.
- Since $\langle b, d\rangle \in S \,;\, T$, we have: $\exists c \in C.\, (\langle b, c\rangle \in S) \wedge (\langle c, d\rangle \in T)$.
- From $\langle a, b\rangle \in R$ and $\langle b, c\rangle \in S$, we have: $\langle a, c\rangle \in R \,;\, S$.
- From $\langle a, c\rangle \in R \,;\, S$ and $\langle c, d\rangle \in T$, we have: $\langle a, d\rangle \in (R \,;\, S) \,;\, T$.

Therefore, $(R \,;\, S) \,;\, T = R \,;\, (S \,;\, T)$. □

Closures of Relations

What is a Closure?

Sometimes we have a relation that *almost* has a property we want, but not quite.

Example (Motivation):

- You have relation $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$ on $\{1, 2, 3\}$
- It's not reflexive (missing self-loops)
- It's not symmetric (arrows only go one way)
- It's not transitive (missing $\langle 1, 3\rangle$)

How do we "fix" R to get these properties?

Definition 22: The *closure* of relation R with respect to property P is the *smallest* relation containing R that has property P.

Think of it as: "minimally extend R to satisfy P".

Key insight: Closures add the minimum pairs needed — no more, no less.

Reflexive Closure

Definition 23: The *reflexive closure* of $R \subseteq M^2$ is:

$$r(R) = R \cup I_M = R \cup \{\langle x, x\rangle \mid x \in M\}$$

Add all missing self-loops: $\langle x, x\rangle$ for every $x \in M$ that doesn't already relate to itself.

Example: Let $M = \{1, 2, 3\}$ and $R = \{\langle 1, 2\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle\}$.

- Element 2 already has $\langle 2, 2\rangle$
- Elements 1 and 3 are missing self-loops
- Add: $\langle 1, 1\rangle$ and $\langle 3, 3\rangle$

Result: $r(R) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle, \langle 3, 3\rangle\}$

Graph view: Add self-loop to every vertex.

Matrix view: Set all diagonal entries to 1.

Symmetric Closure

Definition 24: The *symmetric closure* of $R \subseteq M^2$ is:

$$s(R) = R \cup R^{-1} = R \cup \{\langle b, a\rangle \mid \langle a, b\rangle \in R\}$$

For every pair in R, add its reverse (if not already present).

Example: Let $M = \{1, 2, 3\}$ and $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 1, 3\rangle\}$.

For each pair, add the reverse:

- $\langle 1, 2\rangle \in R \Rightarrow$ add $\langle 2, 1\rangle$
- $\langle 2, 3\rangle \in R \Rightarrow$ add $\langle 3, 2\rangle$
- $\langle 1, 3\rangle \in R \Rightarrow$ add $\langle 3, 1\rangle$

Result: $s(R) = \{\langle 1, 2\rangle, \langle 2, 1\rangle, \langle 1, 3\rangle, \langle 3, 1\rangle, \langle 2, 3\rangle, \langle 3, 2\rangle\}$

Graph view: Every directed edge becomes bidirectional.
Matrix view: Make matrix symmetric.

Transitive Closure

Definition 25: The *transitive closure* $t(R)$ of $R \subseteq M^2$ is the smallest transitive relation containing R.

Theorem 3: The transitive closure can be computed as:

$$t(R) = \bigcup_{n=1}^{\infty} R^n = R \cup R^2 \cup R^3 \cup \ldots$$

where R^n represents all n-step paths (relation composition power: $R^{n+1} = R^n \circ R$).

For finite M with $|M| = k$, we have: $t(R) = R \cup R^2 \cup \ldots \cup R^k$.

Proof: Any path longer than $|M|$ must repeat vertices, so paths up to length $|M|$ suffice. □

Intuition: Add *shortcuts* for all *multi-step paths*. If you can reach c from a through intermediate steps ($a \to b \to c$), make a directly relate to c: add $a \to c$.

Computing Transitive Closure: Example

Example: Let $M = \{1, 2, 3\}$ and $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$ (a chain: $1 \to 2 \to 3$).

Step	Description	New pairs added	Result
Step 1:	Direct connections ($R^1 = R$)	$\langle 1, 2\rangle, \langle 2, 3\rangle$	$R^1 = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$
Step 2:	Two-step paths ($R^2 = R \circ R$)	$\langle 1, 3\rangle$ from path $1 \to 2 \to 3$	$R^2 = \{\langle 1, 3\rangle\}$
Step 3:	Three-step paths ($R^3 = R^2 \circ R$)	None (no three-step paths)	$R^3 = \emptyset$
Final Result:	Transitive closure	$\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 1, 3\rangle$	$t(R) = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 1, 3\rangle\}$

Result: The "shortcut" $\langle 1, 3\rangle$ was added to connect the chain endpoints.

Combining Closures

Closures can be combined to achieve multiple properties at once.

Definition 26: Common combinations:

- *Reflexive-symmetric*: $sr(R) = s(r(R))$ — add self-loops and reverse arrows
- *Reflexive-transitive*: $tr(R) = t(r(R))$ — add self-loops and shortcuts
- *Equivalence closure*: $tsr(R) = t(s(r(R)))$ — make reflexive, symmetric, and transitive

Theorem 4 (Commutativity):

- Reflexive and symmetric closures commute: $sr(R) = rs(R)$
- Reflexive and transitive closures commute: $tr(R) = rt(R)$
- Equivalence (reflexive-symmetric-transitive) closure is independent of the order of r and s, if t is applied *last*: $tsr(R) = trs(R)$, which is equavalent to $t(rs(R)) = t(sr(R))$.

Note: Order *matters* for some combinations! Generally safe: reflexive and symmetric *commute* with most operations, but transitive closure should be computed *last* when combining all three.

Example: Reflexive-Symmetric Closure

Example: Let $M = \{1, 2, 3\}$ and $R = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$.

We want both reflexivity and symmetry. Order doesn't matter!

Method 1: Reflexive first, then symmetric

$$r(R) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle, \langle 3, 3\rangle\}$$
$$sr(R) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle, \langle 3, 2\rangle, \langle 3, 3\rangle\}$$

Method 2: Symmetric first, then reflexive

$$s(R) = \{\langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 3\rangle, \langle 3, 2\rangle\}$$
$$rs(R) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \langle 2, 3\rangle, \langle 3, 2\rangle, \langle 3, 3\rangle\}$$

Both methods give the same result: $rs(R) = sr(R)$ ✓

Example: Reflexive-Transitive Closure

The reflexive-transitive closure is particularly important and has a special notation.

Example: Let $M = \{a, b, c\}$ and $R = \{\langle a, b\rangle, \langle b, c\rangle\}$ (a chain).

Step 1: Compute transitive closure

$$t(R) = R \cup R^2 = \{\langle a, b\rangle, \langle b, c\rangle, \langle a, c\rangle\}$$

Step 2: Add reflexivity

$$rt(R) = t(R) \cup I_M = \{\langle a, a\rangle, \langle a, b\rangle, \langle a, c\rangle, \langle b, b\rangle, \langle b, c\rangle, \langle c, c\rangle\}$$

Note: The reflexive-transitive closure is often denoted R^* (Kleene star).

It represents: "reachable in zero or more steps" — fundamental in automata theory and formal languages.

Example: Equivalence Closure

The equivalence closure makes a relation reflexive, symmetric, and transitive.

Example: Let $M = \{1, 2, 3, 4\}$ and $R = \{\langle 1, 2\rangle, \langle 3, 4\rangle\}$ (two separate connections).

Step	Description	New pairs	Result
Step 1:	Reflexive closure	$\langle 1, 1\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle, \langle 4, 4\rangle$	$r(R) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle, \langle 3, 4\rangle, \langle 4, 4\rangle\}$
Step 2:	Symmetric closure	$\langle 2, 1\rangle, \langle 4, 3\rangle$	$sr(R) = \{\langle 1, 1\rangle, \langle 1, 2\rangle, \langle 2, 1\rangle, \langle 2, 2\rangle, \langle 3, 3\rangle, \langle 3, 4\rangle, \langle 4, 3\rangle, \langle 4, 4\rangle\}$
Step 3:	Transitive closure	None (no new pairs needed)	$tsr(R) = sr(R)$

Result: The equivalence closure creates two equivalence classes: $\{1, 2\}$ and $\{3, 4\}$.

Another Example: Equivalence Closure

Example: Let $M = \{a, b, c, d, e\}$ and $R = \{\langle a, b\rangle, \langle b, c\rangle, \langle d, e\rangle\}$ (chain $a \to b \to c$ and pair $d \to e$).

Step 1: Add reflexivity — all self-loops

$$r(R) = R \cup \{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle, \langle d, d\rangle, \langle e, e\rangle\}$$

Step 2: Add symmetry — reverse all arrows

$$sr(R) = r(R) \cup \{\langle b, a\rangle, \langle c, b\rangle, \langle e, d\rangle\}$$

Step 3: Add transitivity — connect all chains

- Chain $a \leftrightarrow b \leftrightarrow c$ needs shortcuts: $\langle a, c\rangle$ and $\langle c, a\rangle$
- Pair $d \leftrightarrow e$ is already bi-connected

$$tsr(R) = sr(R) \cup \{\langle a, c\rangle, \langle c, a\rangle\}$$

Result: Two equivalence classes emerge: $\{a, b, c\}$ (connected chain) and $\{d, e\}$ (connected pair).

Warshall's Algorithm

An efficient algorithm to compute transitive closure using dynamic programming.

Definition 27: *Warshall's algorithm* computes $t(R)$ in $O(n^3)$ time.

Idea: Systematically check if paths exist through each vertex as an intermediate.

```
M = relmat(R)    # Start with relation matrix
for k = 1 to n:  # Try each vertex k as intermediate
    for i = 1 to n:
        for j = 1 to n:
            # If path i→k and k→j exist, add i→j
            M[i,j] = M[i,j] OR (M[i,k] AND M[k,j])
```

Key insight: Consider each vertex in turn as a "stepping stone" to create shortcuts.

Warshall's Algorithm: Step-by-Step Example

Example: Graph: $1 \to 2 \to 3 \to 4 \to 1$ (a cycle)

Initial matrix:

$$M^{(0)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$k = 1$: Paths through vertex 1 as intermediate

$$M^{(1)} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

New: $4 \to 1 \to 2$, so add $\langle 4, 2 \rangle$

Warshall's Algorithm: Step-by-Step Example [2]

$k = 2$: Paths through vertex 2

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

New: $1 \to 2 \to 3$ and $4 \to 2 \to 3$

$k = 3$: Paths through vertex 3

$$M^{(3)} = \begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

New: paths to 4 via 3

Warshall's Algorithm: Step-by-Step Example [3]

$k = 4$: Paths through vertex 4 (closes cycle!)

$$M^{(4)} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Cycle means everything becomes reachable!

Applications of Transitive Closure

Example (Social network influence): In a "follows" network: $R = \{\langle A, B\rangle, \langle B, C\rangle, \langle C, D\rangle, \langle A, E\rangle\}$

Transitive closure $t(R)$ reveals *indirect influence*:

- A can be influenced by C (through B)
- A can be influenced by D (through B, then C)

Applications:

- Information propagation analysis
- Recommendation systems
- Influence maximization

Example (Software dependencies): Module A depends on B if it imports/uses B.

Transitive closure reveals:

- All transitive dependencies (what A ultimately needs)
- Build order (via topological sort)
- Circular dependencies (cycles in closure)

Mathematical Properties of Closures

Closures satisfy important mathematical properties.

Theorem 5: For any relation $R \subseteq M^2$:

1. *Idempotency*: Closure of closure = closure

$$r(r(R)) = r(R), \quad s(s(R)) = s(R), \quad t(t(R)) = t(R)$$

2. *Monotonicity*: Larger relation $\Rightarrow$ larger closure

$$R_1 \subseteq R_2 \rightarrow r(R_1) \subseteq r(R_2)$$

3. *Extensivity*: Closure contains original

$$R \subseteq r(R), \quad R \subseteq s(R), \quad R \subseteq t(R)$$

4. *Distributivity*: Closure of union = union of closures (for r and s)

$$r(R_1 \cup R_2) = r(R_1) \cup r(R_2)$$

Special Cases of Closures

Example (Empty relation): Let $M = \{a, b, c\}$ and $R = \emptyset$ (no pairs at all).

- $r(\emptyset) = I_M = \{\langle a, a\rangle, \langle b, b\rangle, \langle c, c\rangle\}$ (add all self-loops)
- $s(\emptyset) = \emptyset$ (no pairs to reverse)
- $t(\emptyset) = \emptyset$ (no paths to connect)

Note: Reflexive closure adds structure, but symmetric/transitive closures have no pairs to work with.

Example (Universal relation): Let $M = \{1, 2\}$ and $R = M \times M$ (all 4 pairs).

- $r(R) = R$ (already has all self-loops)
- $s(R) = R$ (already symmetric)
- $t(R) = R$ (already transitive)

Note: Universal relation already has all properties — closures don't add anything!

Non-Distributivity of Transitive Closure

Example: Let $R_1 = \{\langle 1, 2\rangle\}$ and $R_2 = \{\langle 2, 3\rangle\}$ on $M = \{1, 2, 3\}$.

Computing $t(R)$ separately:

- $t(R_1) = \{\langle 1, 2\rangle\}$ (no paths to close)
- $t(R_2) = \{\langle 2, 3\rangle\}$ (no paths to close)
- $t(R_1) \cup t(R_2) = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$

Computing $t(R)$ on union:

- $R_1 \cup R_2 = \{\langle 1, 2\rangle, \langle 2, 3\rangle\}$ (a chain!)
- $t(R_1 \cup R_2) = \{\langle 1, 2\rangle, \langle 2, 3\rangle, \langle 1, 3\rangle\}$ (adds shortcut $\langle 1, 3\rangle$)

Result: $t(R_1 \cup R_2) \supset t(R_1) \cup t(R_2)$

Key insight: Transitive closure *does not* distribute over union!

$$t(R_1 \cup R_2) \neq t(R_1) \cup t(R_2)$$

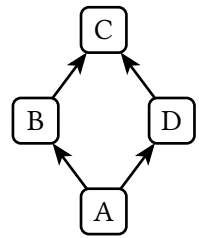
Note: Reflexive and symmetric closures *do* distribute over union:

$$r(R_1 \cup R_2) = r(R_1) \cup r(R_2) \qquad s(R_1 \cup R_2) = s(R_1) \cup s(R_2)$$

Transitive Closure in Dependency Analysis

Example: Consider a directed acyclic graph (DAG) where R represents *"depends on"* relationships:

$$R = \{\langle A, B\rangle, \langle B, C\rangle, \langle A, D\rangle, \langle D, C\rangle\}$$



Component A depends on B and D, which both depend on C.

The transitive closure reveals all *indirect* dependencies:

$$t(R) = R \cup \{\langle A, C\rangle\}$$

Component A *transitively depends* on C through two paths: $A \to B \to C$ and $A \to D \to C$.

Application: Build systems use transitive closure to determine the complete dependency tree and optimal compilation order (possibly parallel).

Applications of Relation Closures

*Example (**Graph Reachability**)*: In directed graphs, transitive closure determines reachability:

- **Social networks:** Finding mutual connections and influence propagation
- **Transportation:** Computing all possible routes between cities
- **Citation networks:** Tracking intellectual dependencies in research
- **Web crawling:** Discovering linked pages and site connectivity

*Example (**Program Analysis**)*: Closure operations enable sophisticated program optimization:

- **Data flow analysis:** Transitive closure computes variable dependencies
- **Call graph construction:** Finding all possible function invocations
- **Alias analysis:** Determining which pointers may reference the same memory
- **Taint analysis:** Tracking information flow and potential vulnerabilities

*Example (**Database Systems**)*: Relational databases extensively use closure computations:

- **Recursive queries:** Computing organizational hierarchies and bill-of-materials
- **Join optimization:** Finding efficient query execution plans
- **Integrity constraints:** Ensuring referential consistency across tables
- **Data lineage:** Tracking data provenance and transformation chains

Complexity of Closure Computations

Theorem 6: For a relation R on a set with n elements:

- **Reflexive closure:** $O(n)$ — simply add n self-loops
- **Symmetric closure:** $O(|R|)$ — reverse each of the $|R|$ pairs
- **Transitive closure:** $O(n^3)$ — Warshall's algorithm

Warning: The $O(n^3)$ transitive closure bottleneck drives many algorithmic design decisions in databases, compilers, and network analysis.

Note: Faster algorithms exist for specific graph types (*e.g.*, $O(n + m)$ for DAGs using topological sort).

Functions

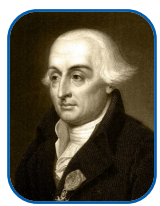
"A function is a machine which converts a certain class of inputs into a certain class of outputs."

— Norbert Wiener


Leonhard Euler


Augustin-Louis Cauchy


Karl Weierstrass


Joseph-Louis Lagrange


George Pólya


Norbert Wiener

Definition of a Function

Definition 28: A *function* f from a set A to a set B, denoted $f : A \to B$, is a special kind of relation $f \subseteq A \times B$ where every element of A is paired with *exactly one* element of B.

This "exactly one" requirement breaks down into two conditions:

1. *Functional property* (*right-unique* or *well-defined*): Each input has *at most one* output. No input can map to multiple different outputs.

$$\forall a \in A. \forall b_1, b_2 \in B. (f(a) = b_1) \land (f(a) = b_2) \to (b_1 = b_2)$$

2. *Total property* (*left-total* or *defined everywhere*): Each input has *at least one* output. Every element in the domain must map to something.

$$\forall a \in A. \exists b \in B. f(a) = b$$

Definition of a Function [2]

Definition 29: A relation that satisfies only the *functional* property (but not necessarily total) is called a *partial function*, denoted $f : A \hookrightarrow B$. It may be undefined for some inputs.

Definition 30: A relation that satisfies *both* properties is called a *total function*, denoted $f : A \to B$. It is defined for every input in its domain.

Note: In most contexts, when we say "function" we mean *total function*. Partial functions are explicitly noted when needed, especially in computability theory and programming language semantics.

Examples of Functions

Example (Partial functions):

- **Division**: $f : \mathbb{R} \times \mathbb{R} \hookrightarrow \mathbb{R}$ defined by $f(x, y) = \frac{x}{y}$ is partial because $f(x, 0)$ is undefined.
- **Square root on integers**: $g : \mathbb{Z} \hookrightarrow \mathbb{R}$ defined by $g(n) = \sqrt{n}$ is partial because negative integers have no real square root.
- **Array access**: $\text{arr} : \mathbb{N} \hookrightarrow T$ where $\text{arr}(i)$ returns the element at index i, but is undefined for out-of-bounds indices.
- **Head of list**: $\text{head} : \text{List}[T] \hookrightarrow T$ returns the first element, but is undefined for empty lists.

Example (Total functions):

- **Absolute value**: $|\cdot| : \mathbb{R} \to \mathbb{R}$ is total — defined for all real numbers.
- **Successor**: $S : \mathbb{N} \to \mathbb{N}$ defined by $S(n) = n + 1$ is total — defined for all natural numbers.
- **Constant function**: $f : A \to B$ defined by $f(x) = b_0$ for some fixed $b_0 \in B$ is always total.

Domain, Codomain, Range

Definition 31: For a function $f : A \to B$:

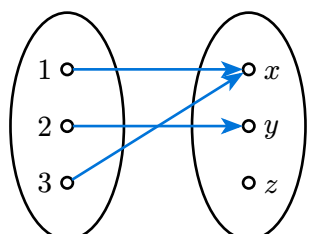
- The set A is called the *domain* of f, denoted $\mathrm{Dom}(f)$.
- The set B is called the *codomain* of f, denoted $\mathrm{Cod}(f)$.
- The *range* (or *image*) of f is the set of all values that f actually takes:

$$\mathrm{Range}(f) = \{b \in B \mid \exists a \in A.\, f(a) = b\} = \{f(a) \mid a \in A\}$$

Note: $\mathrm{Range}(f) \subseteq \mathrm{Cod}(f)$

Example: Let $A = \{1, 2, 3\}$ and $B = \{x, y, z\}$, and define $f = \{\langle 1, x\rangle, \langle 2, y\rangle, \langle 3, x\rangle\}$.

- $f : A \to B$ is a function from A to B
- $\mathrm{Dom}(f) = A = \{1, 2, 3\}$ ("from")
- $\mathrm{Cod}(f) = B = \{x, y, z\}$ ("to")
- $\mathrm{Range}(f) = \{x, y\} \subset B$ (note that z is not in the range)
- We have $f(1) = x$, $f(2) = y$, $f(3) = x$



Examples of Domain, Codomain, Range

Example: Consider the *squaring* function $g : \mathbb{Z} \to \mathbb{Z}$ defined by $g(n) = n^2$.

- $\text{Dom}(g) = \text{Cod}(g) = \mathbb{Z}$ (all integers)
- $\text{Range}(g) = \{0, 1, 4, 9, 16, ...\} = \{n^2 \mid n \in \mathbb{N}\}$ (non-negative perfect squares)

Note that the range of g is a proper subset of the codomain: $\text{Range}(g) \subset \text{Cod}(g)$, since $-1 \notin \text{Range}(g)$.

Example: The *absolute value* function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = |x|$:

- $\text{Dom}(f) = \text{Cod}(f) = \mathbb{R}$ (all real numbers)
- $\text{Range}(f) = [0, \infty) = \{y \in \mathbb{R} \mid y \geq 0\}$ (non-negative reals)

Example: The *exponential* function $\exp : \mathbb{R} \to (0, \infty)$ defined by $\exp(x) = e^x$:

- $\text{Dom}(\exp) = \mathbb{R}$ (all real numbers)
- $\text{Cod}(\exp) = (0, \infty)$ (positive real numbers)
- $\text{Range}(\exp) = (0, \infty)$ (same as codomain — this function is *surjective*!)

Note the careful choice of codomain: if we used $\exp : \mathbb{R} \to \mathbb{R}$, the range would still be $(0, \infty)$, making it not surjective.

Image and Preimage of Sets

Definition 32: Let $f : A \to B$ be a function and let $S \subseteq A$. The *image of S under f* is the set:

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

Note that $f(S) \subseteq B$. The range of f is $f(A)$.

Definition 33: Let $f : A \to B$ be a function and let $T \subseteq B$. The *preimage of T under f* (or *inverse image of T*) is the set of all elements in the domain that map into T:

$$f^{-1}(T) = \{a \in A \mid f(a) \in T\}$$

Note: The notation $f^{-1}(S)$ is used even if the inverse function f^{-1} does not exist (*i.e.*, if f is not bijective). It always refers to the set of domain elements that map into S.

Examples of Image and Preimage

Example: Let $f : \mathbb{Z} \to \mathbb{Z}$ be $f(x) = x^2$.

- Let $S = \{-2, -1, 0, 1, 2\}$.
  - Then $f(S) = \{f(-2), f(-1), f(0), f(1), f(2)\} = \{4, 1, 0, 1, 4\} = \{0, 1, 4\}$.
- Let $T_1 = \{1, 9\}$.
  - The preimage is $f^{-1}(T_1) = \{x \in \mathbb{Z} \mid x^2 \in \{1, 9\}\} = \{-3, -1, 1, 3\}$.
- Let $T_2 = \{2, 3\}$.
  - The preimage is $f^{-1}(T_2) = \{x \in \mathbb{Z} \mid x^2 \in \{2, 3\}\} = \emptyset$.

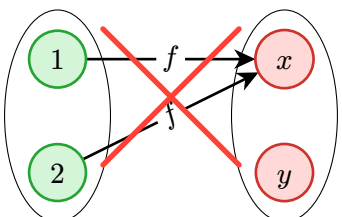
Injective Functions

Definition 34: A function $f : A \to B$ is *injective* (*left-unique* or *one-to-one*[3]) if distinct elements in the domain map to distinct elements in the codomain. Formally:

$$\forall a_1, a_2 \in A.\, (f(a_1) = f(a_2)) \to (a_1 = a_2)$$

Equivalent formulation:

$$\forall a_1, a_2 \in A.\, (a_1 \neq a_2) \to (f(a_1) \neq f(a_2))$$



[3]Do not confuse "one-to-one" with "one-to-one correspondence", which refers to a *bijection* — another concept!

Examples of Injective Functions

Key insight: An injective function never "collapses" different inputs to the same output. Each output has at most one pre-image.

Example: $f : \mathbb{N} \to \mathbb{N}$ defined by $f(n) = 2n$ is injective.

- If $f(n_1) = f(n_2)$, then $2n_1 = 2n_2$, so $n_1 = n_2$. ✓

Example: $g : \mathbb{R} \to \mathbb{R}$ defined by $g(x) = x^3$ is injective.

- The cube function is strictly increasing, so distinct inputs give distinct outputs. ✓

Example (Computer science applications):

- **Student ID assignment:** Each student gets a unique ID number.
- **Hash functions (perfect):** Different data should hash to different values (to avoid collisions).
- **Primary keys in databases:** Each record must have a unique identifier.
- **Memory addresses:** Each memory location has a unique address.

Examples of Non-Injective Functions

Example (Counter-examples (not injective functions)):

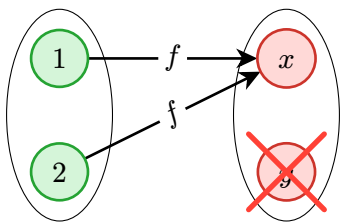
- $g : \mathbb{Z} \to \mathbb{Z}$ defined by $g(n) = n^2$ is *not* injective.
  - $g(-1) = 1$ and $g(1) = 1$, but $-1 \neq 1$. ✗
- $h : \mathbb{R} \to \mathbb{R}$ defined by $h(x) = \sin(x)$ is *not* injective.
  - $h(0) = h(\pi) = 0$, but $0 \neq \pi$. ✗
- $f : \mathbb{Z} \to \mathbb{Z}_5$ defined by $f(x) = x \bmod 5$ is *not* injective.
  - $f(2) = f(7) = 2$, but $2 \neq 7$. ✗

Surjective Functions

Definition 35: A function $f : A \to B$ is *surjective* (*right-total* or *onto*) if every element in the codomain is the image of at least one element in the domain. Formally:

$$\forall b \in B.\, \exists a \in A.\, f(a) = b$$

For surjective functions, $\text{Range}(f) = \text{Cod}(f)$, *i.e.*, there are *no "uncovered"* elements in the right side.



Examples of Surjective Functions

Key insight: A surjective function "covers" the entire codomain. Every possible output is actually achieved by some input.

Example: $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^3$ is surjective.

- For any $y \in \mathbb{R}$, let $x = \sqrt[3]{y}$. Then $f(x) = \left(\sqrt[3]{y}\right)^3 = y$. ✓

Example: $g : \mathbb{R} \to [0, \infty)$ defined by $g(x) = x^2$ is surjective.

- For any $y \geq 0$, let $x = \sqrt{y}$. Then $g(x) = \left(\sqrt{y}\right)^2 = y$. ✓

Example: $f : \mathbb{Z} \to \mathbb{Z}_5$ defined by $f(x) = x \bmod 5$ is surjective.

- Every element $\{0, 1, 2, 3, 4\}$ is achieved: $f(0) = 0, f(1) = 1, f(2) = 2, f(3) = 3, f(4) = 4$. ✓

Example (Computer science applications):

- **Color quantization:** Map 24-bit colors to 8-bit palette (should cover all palette colors).
- **Load balancing:** Distribute requests across servers (all servers should be used).
- **Hash table design:** Hash function should potentially reach every bucket.

Examples of Non-Surjective Functions

Example (Counter-examples (not surjective)):

- $g : \mathbb{N} \to \mathbb{N}$ defined by $g(n) = 2n$ is *not* surjective.
  - Odd numbers like $3, 5, 7, \ldots$ are never achieved since $2n$ is always even. ✗
- $h : \mathbb{R} \to \mathbb{R}$ defined by $h(x) = x^2$ is *not* surjective.
  - Negative numbers like $-1, -2, -3, \ldots$ are never achieved since $x^2 \geq 0$. ✗
- $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = e^x$ is *not* surjective.
  - Negative numbers and zero are never achieved since $e^x > 0$ for all x. ✗

Bijective Functions

Definition 36: A function $f : A \to B$ is *bijective* if it is both injective and surjective. A bijective function establishes a *one-to-one correspondence* between the elements of A and B.

Equivelently, f is bijective iff it has an inverse function $f^{-1} : B \to A$.

Example: $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = 2x + 1$ is bijective.

- **Injective:** If $2x_1 + 1 = 2x_2 + 1$, then $x_1 = x_2$. ✓
- **Surjective:** For any $y \in \mathbb{R}$, let $x = \frac{y-1}{2}$. Then $f(x) = y$. ✓
- **Inverse:** $f^{-1}(y) = \frac{y-1}{2}$.

Example: $g : [0, \infty) \to [0, \infty)$ defined by $g(x) = x^2$ is bijective.

- **Injective:** If $x_1^2 = x_2^2$ and $x_1, x_2 \geq 0$, then $x_1 = x_2$. ✓
- **Surjective:** For any $y \geq 0$, let $x = \sqrt{y}$. Then $g(x) = y$. ✓
- **Inverse:** $g^{-1}(y) = \sqrt{y}$.

Examples of Bijective Functions

Key insight: Bijective functions are "perfect matchings" between sets. Every element in A pairs with *exactly one* element in B, and vice versa.

Example (Computer science applications):

- **Encryption algorithms:** Must be bijective to ensure decryption is possible.
- **Base conversion:** Bijection between decimal and binary representations.
- **Perfect hash functions:** Bijective mapping from keys to table positions.
- **Coordinate transformations:** Reversible mappings between coordinate systems.

Function Composition

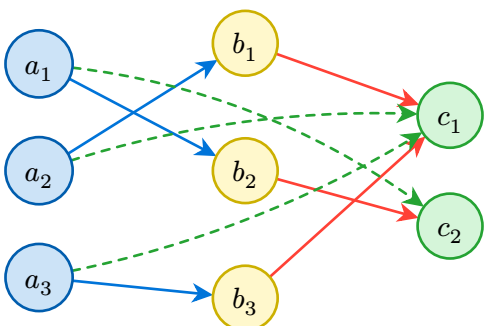
Definition 37: Let $f : A \to B$ and $g : B \to C$ be two functions. The *composition* of g and f, denoted $g \circ f$ (read as "g composed with f" or "g after f"), is a function from A to C defined by:

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

Example: Let $f : \mathbb{R} \to \mathbb{R}$ be $f(x) = x^2$ and $g : \mathbb{R} \to \mathbb{R}$ be $g(x) = x + 1$.

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

Note: $g \circ f \neq f \circ g$ (composition is not commutative!)



Examples of Function Composition

Key insight: Function composition is like a "pipeline" — the output of f becomes the input to g.

Read right-to-left: $g \circ f$ means "first apply f, then apply g", or rather, "do g after f".

Example (Computer science applications):

- **Function pipelines:** `data |> filter |> map |> reduce`
- **Compiler design:** lexer $\to$ parser $\to$ optimizer $\to$ code generator
- **Data processing:** clean $\to$ transform $\to$ aggregate $\to$ visualize

Properties of Function Composition

- **Associativity:** If $f : A \to B$, $g : B \to C$, and $h : C \to D$, then $(h \circ g) \circ f = h \circ (g \circ f)$.
- The *identity* function acts as a *neutral* element for composition:
  - $\mathrm{id}_B \circ f = f$ for any function $f : A \to B$.
  - $f \circ \mathrm{id}_A = f$ for any function $f : A \to B$.
- Composition *preserves* the properties of functions:
  - If f and g are injective, so is $g \circ f$.
  - If f and g are surjective, so is $g \circ f$.
  - If f and g are bijective, so is $g \circ f$.
- Note that in general, $g \circ f \neq f \circ g$, *i.e.*, function composition is *not commutative*.

Functional Powers

Definition 38: The *functional power* of $f : A \to A$ can defined inductively:

$$f^0 = \mathrm{id}_A$$
$$f^{n+1} = f \circ f^n = f^n \circ f$$

Note: This definition also works for functions $f : X \to Y$ with $Y \subseteq X$.

Note: To avoid the confusion with exponential powers (*e.g.*, $f^n(x)$ could be interpreted as $(f(x))^n$), we can use the notation $f^{\circ n}$ to denote the n-th functional power.

Example: Let $f : \mathbb{R} \to \mathbb{R}$ be $f(x) = 2x$.

- $f^0(x) = x$ (identity)
- $f^1(x) = 2x$
- $f^2(x) = f(f(x)) = f(2x) = 2(2x) = 4x$
- $f^3(x) = f(f^2(x)) = f(4x) = 2(4x) = 8x$
- In general: $f^n(x) = 2^n x$

Examples of Functional Powers

Example: Let $g : \mathbb{Z} \to \mathbb{Z}$ be $g(x) = x + 1$ (successor function).

- $g^{\circ 0}(x) = x$
- $g^{\circ 1}(x) = x + 1$
- $g^{\circ 2}(x) = g(g(x)) = g(x + 1) = (x + 1) + 1 = x + 2$
- $g^{\circ 3}(x) = g(g^{\circ 2}(x)) = g(x + 2) = (x + 2) + 1 = x + 3$
- In general: $g^{\circ n}(x) = x + n$

Inverse Functions

Definition 39: If $f : A \to B$ is a bijective function, then its *inverse function*, denoted $f^{-1} : B \to A$, is:

$$f^{-1}(b) = a \quad \text{iff} \quad f(a) = b$$

Note: A function has an inverse *if and only if* it is bijective.

Example: Let $f : \mathbb{R} \to \mathbb{R}$ be $f(x) = 2x + 1$. We found it's bijective. To find $f^{-1}(y)$, let $y = 2x + 1$. Solving for x, we get $x = \frac{y-1}{2}$. So, $f^{-1}(y) = \frac{y-1}{2}$.

Example: The exponential function $\exp : \mathbb{R} \to (0, \infty)$ defined by $\exp(x) = e^x$ is bijective.

- Its inverse is the natural logarithm: $\ln : (0, \infty) \to \mathbb{R}$.
- We have: $\ln(e^x) = x$ for all $x \in \mathbb{R}$ and $e^{\ln y} = y$ for all $y > 0$.

Note: If we tried $\exp : \mathbb{R} \to \mathbb{R}$, it wouldn't be surjective (negative numbers never achieved), hence no inverse! The codomain restriction is essential.

Examples of Inverse Functions

Example (Polynomial with domain restriction): Consider $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2$.

- This is **not** injective: $f(-2) = f(2) = 4$, so no inverse exists.

However, if we **restrict the domain** to $f : [0, \infty) \to [0, \infty)$, then:

- f becomes bijective (both injective and surjective on non-negative reals).
- The inverse is $f^{-1}(y) = \sqrt{y}$ for $y \geq 0$.

Key insight: Many functions need domain/codomain restrictions to become invertible. This is crucial in calculus when finding inverse functions!

Example: The sine function $\sin : \mathbb{R} \to \mathbb{R}$ is **not** injective (it's periodic), so it has no inverse.

However, with domain restriction, $\sin : [-\frac{\pi}{2}, \frac{\pi}{2}] \to [-1, 1]$ **is** bijective, with inverse $\arcsin : [-1, 1] \to [-\frac{\pi}{2}, \frac{\pi}{2}]$.

Similarly: $\cos : [0, \pi] \to [-1, 1]$ has inverse $\arccos$, and $\tan : (-\frac{\pi}{2}, \frac{\pi}{2}) \to \mathbb{R}$ has inverse $\arctan$.

Characterization of Invertible Functions

Theorem 7: A function $f : A \to B$ has an inverse function $f^{-1} : B \to A$ *if and only if* f is bijective.

Proof: We prove both directions.

($\Rightarrow$): If f has an inverse, then f is bijective.

Assume $f^{-1} : B \to A$ is an inverse of $f : A \to B$ with the following properties:

- $f^{-1}(f(x)) = x$ for all $x \in A$
- $f(f^{-1}(y)) = y$ for all $y \in B$

We are going to show that f is both injective and surjective.

1. **f is injective:** Let $x_1, x_2 \in A$ such that $f(x_1) = f(x_2)$. Applying f^{-1} to both sides, we get:

$$f^{-1}(f(x_1)) = f^{-1}(f(x_2))$$

By the property of the inverse, this simplifies to:

$$x_1 = x_2$$

Characterization of Invertible Functions [2]

Thus, f is injective.

2. **f is surjective:** Let $y \in B$. We need to find an $x \in A$ such that $f(x) = y$. Take $x = f^{-1}(y)$. Then by the property of the inverse, we have:

$$f(x) = f(f^{-1}(y)) = y$$

Hence, f is surjective.

Therefore, f is bijective. ■

($\Leftarrow$): If f is bijective, then f has an inverse f^{-1}.

Assume $f : A \to B$ is bijective. This means:

- f is injective: $(f(x_1) = f(x_2)) \to (x_1 = x_2)$
- f is surjective: for every $y \in B$, there exists $x \in A$ such that $f(x) = y$

We need to construct an inverse function $f^{-1} : B \to A$, satisfying:

$$f^{-1}(f(x)) = x \quad \text{and} \quad f(f^{-1}(y)) = y$$

Characterization of Invertible Functions [3]

Construction of f^{-1}: For each $y \in B$:

- Since f is surjective, there exists at least one $x \in A$ such that $f(x) = y$.
- Since f is injective, this x is unique.

We define a function $f^{-1} : B \to A$ by:

$$f^{-1}(y) = \text{the unique } x \in A \text{ such that } f(x) = y$$

Verification:

1. For all $x \in A$, by definition of f^{-1}:

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

2. For all $y \in B$, since $f^{-1}(y)$ is defined as the unique x with $f(x) = y$, we have:

$$f(f^{-1}(y)) = y$$

Therefore, f^{-1} is indeed the inverse of f. ■

Thus, a function has an inverse *if and only if* it is bijective. □

Some Properties of Inverse Functions

Theorem 8: If $f : A \to B$ is a bijective function with inverse $f^{-1} : B \to A$:

- f^{-1} is also bijective.
- $(f^{-1} \circ f)(a) = a$ for all $a \in A$ (*i.e.*, $f^{-1} \circ f = \mathrm{id}_A$).
- $(f \circ f^{-1})(b) = b$ for all $b \in B$ (*i.e.*, $f \circ f^{-1} = \mathrm{id}_B$).
- If $f : A \to B$ and $g : B \to C$ are both bijective, then $(g \circ f)^{-1} = f^{-1} \circ g^{-1}$.

Monotonic Functions

Definition 40: A function $f : A \to B$ is called *monotonic* if it preserves order relationships.

For real-valued functions $f : \mathbb{R} \to \mathbb{R}$:

- *Monotonic increasing* (or *isotone*) if $x \leq y \to f(x) \leq f(y)$
- *Strictly increasing* if $x < y \to f(x) < f(y)$
- *Monotonic decreasing* (or *antitone*) if $x \leq y \to f(x) \geq f(y)$
- *Strictly decreasing* if $x < y \to f(x) > f(y)$

Note: More generally, for any posets $\langle A, \leq_A \rangle$ and $\langle B, \leq_B \rangle$:

- f is *order-preserving* (monotone) if $x \leq_A y \to f(x) \leq_B f(y)$
- f is *order-reversing* (antitone) if $x \leq_A y \to f(x) \geq_B f(y)$

Monotonicity implies Injectivity

Theorem 9: If $f : \mathbb{R} \to \mathbb{R}$ is *strictly increasing* or *strictly decreasing*, then f is *injective*.

Proof: Suppose f is strictly increasing. Let $x_1, x_2 \in \mathbb{R}$ with $x_1 \neq x_2$. Then either $x_1 < x_2$ or $x_2 < x_1$.

- If $x_1 < x_2$, then $f(x_1) < f(x_2)$ by strict monotonicity, so $f(x_1) \neq f(x_2)$.
- If $x_2 < x_1$, then $f(x_2) < f(x_1)$ by strict monotonicity, so $f(x_1) \neq f(x_2)$.

In both cases, $f(x_1) \neq f(x_2)$, proving injectivity. The proof for strictly decreasing is analogous. □

Key insights:

- Monotonic functions have *predictable* behavior: they never "change direction"
- *Strictly monotonic* functions are *always injective* (one-to-one)
- Non-strict monotonic functions may have "flat" regions (also known as "*plateaus*") where different inputs map to the same output
- Fun fact: monotonic functions are *order-homomorphisms* between posets!

Examples of Monotonic Functions

Examples: *Strictly monotonic* functions (which are always injective):

- $f(x) = 2x + 1$ is strictly increasing on $\mathbb{R}$, hence injective. ✓
- $g(x) = -x$ is strictly decreasing on $\mathbb{R}$, hence injective. ✓
- $h(x) = x^3$ is strictly increasing on $\mathbb{R}$, hence injective. ✓
- $k(x) = e^x$ is strictly increasing on $\mathbb{R}$, hence injective. ✓

Examples: *Monotonic* but *NOT strictly* monotonic functions (not necessarily injective):

- $f(x) = \lfloor x \rfloor$ is monotonic increasing but NOT strictly increasing.
  - For $x \in [2, 3)$, we have $f(x) = 2$ (constant on intervals).
  - This is NOT injective: $f(2.1) = f(2.9) = 2$ but $2.1 \neq 2.9$. ✗
- $g(x) = \begin{cases} x & \text{if } x \leq 0 \\ 0 & \text{if } 0 < x < 1 \\ x - 1 & \text{if } x \geq 1 \end{cases}$ is monotonic but not injective.

Example: $f : (\mathbb{R} \setminus \{0\}) \to \mathbb{R}$ defined by $f(x) = \frac{1}{x}$ is **injective but not monotonic**.

- If $f(x_1) = f(x_2)$, then $\frac{1}{x_1} = \frac{1}{x_2}$ implies $x_1 = x_2$. ✓
- However, f is not monotonic since it decreases on $(-\infty, 0)$ and $(0, \infty)$, but $f(-1) < f(1)$. ✗

Function Properties Overview

Functions can be characterized by several key properties that determine their mathematical behavior.

Property	Definition
Functional (Right-unique)	Each input maps to *at most one* output
Total (Left-total)	Each input maps to *at least one* output (defined everywhere)
Partial	Functional but not total (may be undefined for some inputs)
Injective (Left-unique)	Different inputs $\Longrightarrow$ different outputs
Surjective (Right-total)	Every codomain element is covered
Bijective	Both injective and surjective (one-to-one correspondence)
Monotonic	Preserves or reverses order relationships
Continuous	Small input changes $\Longrightarrow$ small output changes

Characteristic Functions

Definition 41: For any set $S \subseteq A$, the *characteristic function* (or *indicator function*) $\chi_S : A \to \{0, 1\}$ is:

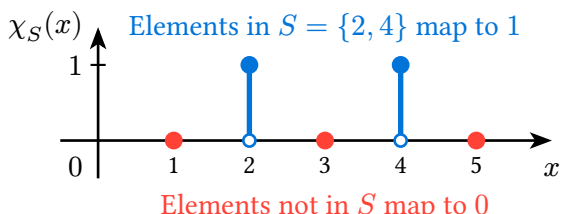
$$\chi_S(x) = \begin{cases} 1 \text{ if } x \in S \\ 0 \text{ if } x \notin S \end{cases}$$

This function "indicates" membership in set S.

Example: Let $A = \{1, 2, 3, 4, 5\}$ and $S = \{2, 4\}$. Then χ_S maps:

- $\chi_S(1) = 0$ (since $1 \notin S$)
- $\chi_S(2) = 1$ (since $2 \in S$)
- $\chi_S(3) = 0$ (since $3 \notin S$)
- $\chi_S(4) = 1$ (since $4 \in S$)
- $\chi_S(5) = 0$ (since $5 \notin S$)

So $\chi_S = \{(1, 0), (2, 1), (3, 0), (4, 1), (5, 0)\}$.



Properties of Characteristic Functions

Definition 42: For sets $A, B \subseteq U$:

- $\chi_{A \cap B} = \chi_A \cdot \chi_B$ (pointwise multiplication)
- $\chi_{A \cup B} = \chi_A + \chi_B - \chi_A \cdot \chi_B$
- $\chi_{\overline{A}} = 1 - \chi_A$ (complement)
- $\chi_{A \triangle B} = \chi_A + \chi_B - 2\chi_A \cdot \chi_B$ (symmetric difference)
- $\chi_\emptyset = 0$ and $\chi_U = 1$ (constant functions)

Example (Applications):

- **Probability theory:** Indicator random variables
- **Database queries:** Boolean conditions in WHERE clauses
- **Set operations:** Converting logical operations to arithmetic
- **Machine learning:** Feature encoding (one-hot encoding)
- **Computer graphics:** Masking and selection operations
- **Digital signal processing:** Window functions and filters

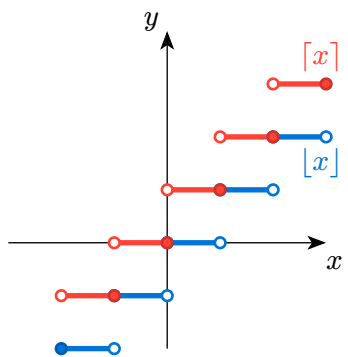
Floor and Ceiling Functions

Definition 43:

- *Floor function* $\lfloor \cdot \rfloor : \mathbb{R} \to \mathbb{Z}$ maps x to the largest integer $\leq x$: $\lfloor x \rfloor = \max\{n \in \mathbb{Z} \mid n \leq x\}$
- *Ceiling function* $\lceil \cdot \rceil : \mathbb{R} \to \mathbb{Z}$ maps x to the smallest integer $\geq x$: $\lceil x \rceil = \min\{n \in \mathbb{Z} \mid n \geq x\}$

Example:

x	$\lfloor x \rfloor$	$\lceil x \rceil$	**Note**
3.7	3	4	Positive non-integer
-2.3	-3	-2	Negative non-integer
5	5	5	Integer (floor = ceiling)
0	0	0	Zero
-1	-1	-1	Negative integer



Summary: Functions

Functions establish relationships between sets where every input has exactly one output:

- *Injective:* Different inputs $\rightarrow$ Different outputs (one-to-one)
- *Surjective:* Every output is achieved (onto)
- *Bijective:* Both injective and surjective (perfect correspondence)

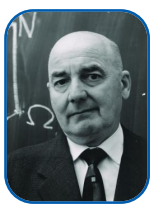
Key applications in computer science:

- **Database design:** Functions model relationships
- **Algorithm analysis:** Function properties affect complexity
- **Cryptography:** Bijective functions enable encryption/decryption
- **Type systems:** Function signatures and type hierarchies
- **Data structures:** Mappings, hashing, and uniqueness constraints

Order Theory

"Order is heaven's first law."

— Alexander Pope

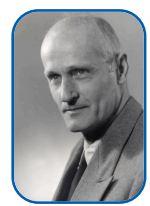

Helmut Hasse


Felix Hausdorff


Henri Poincaré


Robert Floyd


Stephen Warshall


Stephen Kleene

Orders

Definition 44: A relation $R \subseteq M^2$ is called a *preorder* if it is reflexive and transitive.

Definition 45: A *partial order* is a relation $R \subseteq M^2$ that is reflexive, antisymmetric, and transitive.

Definition 46: A *linear order* is a partial order $R \subseteq M^2$ where every pair of elements is *comparable*:

$$\forall x, y \in M. (x \ R \ y \vee y \ R \ x)$$

Note: The above condition is called *strong connectivity*. The similar property with the $x \neq y$ condition is called *semi-connectivity*. The corresponding relations are called *connex* and *semi-connex*.

Note: Hereinafter, we mainly study the "non-strict" orders (*e.g.*, $\leq$, $\subseteq$), which require the *reflexivity*. The "strict" orders (*e.g.*, $<$, $\subset$) require *irreflexivity* instead, and are defined similarly.

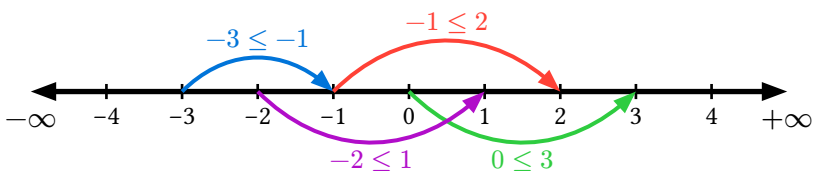
Note: Strict linear orders require *semi-connectivity* instead, since connectivity alone implies reflexivity!

Examples of Orders

Example: The relation $\leq$ on real numbers $\mathbb{R}$ is a *total order*:

- **Reflexive:** $x \leq x$ for all $x \in \mathbb{R}$ ✓
- **Antisymmetric:** If $x \leq y$ and $y \leq x$, then $x = y$ ✓
- **Transitive:** If $x \leq y$ and $y \leq z$, then $x \leq z$ ✓
- **Connected:** For any $x, y \in \mathbb{R}$, either $x \leq y$ or $y \leq x$ ✓

This is the most familiar example of an order relation. Similarly, $\mathbb{N}$, $\mathbb{Z}$, and $\mathbb{Q}$ with $\leq$ are all total orders.



Linear order $\leq$ on real numbers $\mathbb{R}$

Examples of Orders [2]

Example: The *subset relation* $\subseteq$ on the power set $\mathcal{P}(A)$ is a **partial order**.

Verification:

- **Reflexive:** $X \subseteq X$ for all $X \subseteq A$ ✓
  - Every set is a subset of itself by definition
- **Antisymmetric:** If $X \subseteq Y$ and $Y \subseteq X$, then $X = Y$ ✓
  - If every element of X is in Y, and every element of Y is in X, then X and Y have the same elements
- **Transitive:** If $X \subseteq Y$ and $Y \subseteq Z$, then $X \subseteq Z$ ✓
  - If every element of X is in Y, and every element of Y is in Z, then every element of X is in Z

For $A = \{1, 2\}$, we have $\mathcal{P}(A) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$ with:

- $\emptyset \subseteq \{1\} \subseteq \{1, 2\}$ (this is a chain)
- $\emptyset \subseteq \{2\} \subseteq \{1, 2\}$ (another chain)
- But $\{1\}$ and $\{2\}$ are *incomparable* (neither is a subset of the other)

This is *not* a total order because not all pairs are comparable.

Examples of Orders [3]

Example: Consider the *no longer than* relation $\preccurlyeq$ on binary strings $\mathbb{B}^*$:

$$x \preccurlyeq y \quad \text{iff} \quad \text{len}(x) \leq \text{len}(y)$$

Verification:

- **Reflexive:** $x \preccurlyeq x$ for all $x \in \mathbb{B}^*$ ✓
  - $\text{len}(x) \leq \text{len}(x)$ is always true
- **Transitive:** If $x \preccurlyeq y$ and $y \preccurlyeq z$, then $x \preccurlyeq z$ ✓
  - If $\text{len}(x) \leq \text{len}(y)$ and $\text{len}(y) \leq \text{len}(z)$, then $\text{len}(x) \leq \text{len}(z)$ by transitivity of $\leq$
- **Antisymmetric:** ✗
  - Counter-example: $01 \preccurlyeq 10$ and $10 \preccurlyeq 01$ (since both have length 2), but $01 \neq 10$
- **Connected:** For any $x, y \in \mathbb{B}^*$, either $x \preccurlyeq y$ or $y \preccurlyeq x$ ✓
  - Either $\text{len}(x) \leq \text{len}(y)$ or $\text{len}(y) \leq \text{len}(x)$ (or both)

This is a *preorder* (reflexive and transitive), and even connected, but *not a partial order* due to lack of antisymmetry. Different strings of the same length are all "equivalent" under this relation, but they're not actually equal.

Examples of Orders [4]

Example: *Divisibility* $\mid$ on positive integers $\mathbb{N}^+$ is a **partial order**.

Verification:

- **Reflexive:** $n \mid n$ for all $n \in \mathbb{N}^+$ (every number divides itself) ✓
- **Antisymmetric:** If $a \mid b$ and $b \mid a$, then $a = b$ ✓
  - If a divides b, then $b = ak$ for some positive integer k
  - If b divides a, then $a = b\ell$ for some positive integer ℓ
  - Substituting: $b = ak = (b\ell)k = bk\ell$, so $k\ell = 1$
  - Since $k, \ell \in \mathbb{N}^+$, we must have $k = \ell = 1$, hence $a = b$
- **Transitive:** If $a \mid b$ and $b \mid c$, then $a \mid c$ ✓
  - If $a \mid b$, then $b = ak$ for some integer k
  - If $b \mid c$, then $c = b\ell$ for some integer ℓ
  - Therefore: $c = b\ell = (ak)\ell = a(k\ell)$, so $a \mid c$
- **Connected:** ✗
  - Counter-example: 2 does not divide 3, and 3 does not divide 2

This is a *partial order* but not a total order because some pairs are incomparable.

Examples of Orders [5]

Example: *Lexicographic order* on strings A^n (like dictionary order) is a **total order**.

Verification:

- **Reflexive:** $s \preceq s$ for all strings s ✓
  - A string is lexicographically equal to itself
- **Antisymmetric:** If $s \preceq t$ and $t \preceq s$, then $s = t$ ✓
  - If s comes before or equals t AND t comes before or equals s, then $s = t$
- **Transitive:** If $s \preceq t$ and $t \preceq u$, then $s \preceq u$ ✓
  - Lexicographic comparison preserves transitivity through character-by-character comparison
- **Connected:** For any strings s, t, either $s \preceq t$ or $t \preceq s$ ✓
  - We can always compare strings lexicographically by comparing character by character

For binary strings of length 2: $00 \prec 01 \prec 10 \prec 11$

Key property: Every pair of strings is *comparable*, making this a *total order*.

Partially Ordered Sets

Definition 47: A *partially ordered set* (or *poset*) $\langle S, \leq \rangle$ is a set S equipped with a partial order $\leq$.

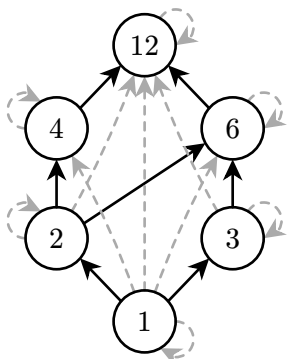
Example: Consider the poset $\langle D, | \rangle$ where $D = \{1, 2, 3, 4, 6, 12\}$ and $|$ is divisibility.

Order relation properties:

- Reflexive: $n \mid n$ for all $n \in D$
- Antisymmetric: If $a \mid b$ and $b \mid a$, then $a = b$
- Transitive: If $a \mid b$ and $b \mid c$, then $a \mid c$

Special elements:

- Least element: 1 (divides all others)
- Greatest element: 12 (divisible by all others)
- Minimal elements: just 1
- Maximal elements: just 12



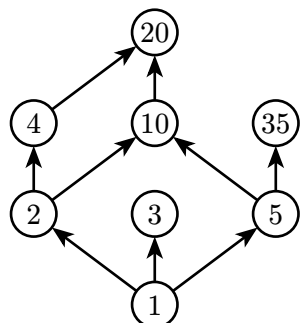
Note: This poset forms a *lattice* since every pair has a supremum (LCM) and infimum (GCD).

Hasse Diagrams

Definition 48: A *Hasse diagram* is a visual representation of a poset where:

- Each element is represented as a vertex.
- If $x < y$ and there is no z with $x < z < y$, draw an edge from x to y.
- Elements are arranged vertically by the order relation.
- Transitive connections are omitted (implied by transitivity of the partial order).

Example: For $D = \{1, 2, 3, 4, 5, 10, 20, 35\}$ with divisibility $|$.

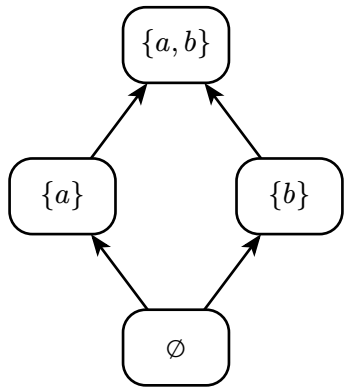


Reading the diagram:

- 1 divides everything (least element at bottom)
- Multiple *maximal* elements: 3, 20, 35 (no single greatest element)
- Some *chains* (not necessarily maximal, can skip elements):
  - Chain: $1 \mid 2 \mid 4$ (powers of 2)
  - Chain: $1 \mid 10 \mid 20$ (multiples of 10)
  - Chain: $5 \mid 35$ (multiples of 5)
- Primes 2, 3, 5 are *incomparable* to each other

Subset Poset

Example: For $\mathcal{P}(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$ with inclusion $\subseteq$:



Note: This is the classic "diamond" shape characteristic of Boolean algebras.

Note: $\langle \mathcal{P}(A), \subseteq \rangle$ is also called the *Boolean lattice*.

Covering Relation

Definition 49: In a poset $\langle S, \leq \rangle$, an element $y \in S$ *covers* $x \in S$, denoted $x \lessdot y$, if there is no other element in between them:

$$x \lessdot y \quad \text{iff} \quad (x < y) \wedge \nexists z \in S.\, (x < z < y)$$

Note: "$<$" denotes the *induced strict order*:

$$x < y \quad \text{iff} \quad (x \leq y) \wedge (x \neq y)$$

Note: Hasse diagram is just a graph of a covering relation!

Maximal and Minimal Elements

Definition 50: An element $m \in S$ is called a *maximal element* of a poset $\langle S, \leq \rangle$ if it is not less than any other element, *i.e.*, there is no even greater element.

$$\forall x \neq m. \neg(m \leq x) \iff \nexists x \neq m. (m \leq x)$$

Equivelently, $\forall x \in S. (m \leq x) \to (m = x)$

Definition 51: An element $m \in S$ is called a *minimal element* of a poset $\langle S, \leq \rangle$ if it is not greater than any other element, *i.e.*, there is no even smaller element.

$$\forall x \neq m. \neg(x \leq m) \iff \nexists x \neq m. (x \leq m)$$

Equivelently, $\forall x \in S. (x \leq m) \to (x = m)$

Note: There may be multiple maximal (or minimal) elements.

Example of Maximal and Minimal Elements

Example: Consider the divisibility poset on $S = \{2, 3, 4, 6, 8, 12\}$:

Maximal elements: 8 and 12

- 8 divides nothing else in S except itself
- 12 divides nothing else in S except itself

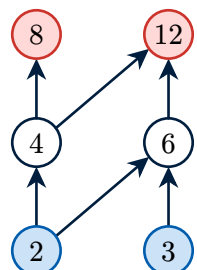
Minimal elements: 2 and 3

- Nothing in S properly divides 2 (since $1 \notin S$)
- Nothing in S properly divides 3

Note: This poset has no greatest or least element, but multiple minimal/maximal elements.

Hasse diagram:

- Maximal elements (8, 12) — they divide nothing else in S
- Minimal elements (2, 3) — nothing in S divides them
- Two separate chains: $2 \mid 4 \mid 8$ and $3 \mid 6 \mid 12$



Example of Maximal and Minimal Elements [2]

Example: Let $S = \{\text{ab}, \text{abc}, \text{abd}, \text{ac}, \text{b}, \text{bc}\}$ ordered by the *prefix* relation $\preccurlyeq$:

$$x \preccurlyeq y \text{ iff } x \text{ is a prefix of } y$$

Maximal elements: abc, abd, ac, bc

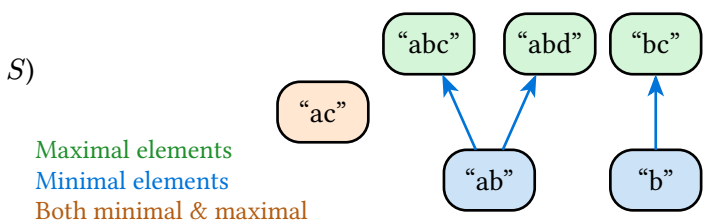
- These strings are not prefixes of any other string in S

Minimal elements: ab, ac, b

- These strings have no other string in S that is a proper prefix of them

The Hasse diagram shows three separate "trees" rooted at the minimal elements, with:

- $\text{ab} \prec \text{abc}$ and $\text{ab} \prec \text{abd}$
- $\text{b} \prec \text{bc}$
- ac stands alone (no other string extends it in S)



Greatest and Least Elements

Definition 52: The *greatest element* of a poset $\langle S, \leq \rangle$ is an element $g \in S$ that is greater than or equal to every other element in S, *i.e.*, for all $x \in S$, $x \leq g$.

Definition 53: A *least element* (bottom) b satisfies $b \leq x$ for all $x \in S$.

Note: Greatest (top) and least (bottom) elements are *unique* when they exist.

Examples:

- $\langle \mathcal{P}(A), \subseteq \rangle$: least $\emptyset$ (contained in every set), greatest A (contains every subset).
- $\langle \mathbb{N}^+, | \rangle$: least 1 (divides every positive integer), no greatest element (no integer is divisible by all others).
- $\langle \mathbb{Z}, \leq \rangle$: no least or greatest element (integers extend infinitely in both directions).
- $\langle \{1, 2, 3, 4, 5, 6\}, | \rangle$: least 1, no greatest element, maximal elements are 4, 5, 6 (prime powers and primes that don't divide anything else in the given set).

Converse Orders

Definition 54: The *dual* (or *converse*) of a poset $\langle S, \leq \rangle$ is the poset $\langle S, \geq \rangle$ where $x \geq y$ iff $y \leq x$.

Example: Consider the $\mathbb{N}^+$ ordered naturally.

- For $\leq$ order:
  - The *$\leq$-least* element is 1, since it is *$\leq$-smaller* than all others.
  - 1 is also $\leq$-minimal, since there are no other element which is $\leq$-smaller than 1.
  - There are *no $\leq$-maximal* elements, since the set is unbounded above.
  - On the Hasse diagram, 1 is at the bottom, and the diagram extends infinitely *upwards*.
- For $\geq$ order, minimal and maximal elements "flip":
  - There are *no $\geq$-minimal* elements. A $\geq$-minimal element would be an element m such that there is no other ($\geq$-smaller) element $n \neq m$ with $n \geq m$. However, for any $m \in \mathbb{N}^+$, there exists $n = m + 1$ which is $n \geq m$, so no such $\geq$-minimal element exists.
  - The *$\geq$-greatest* element is 1, since all elements are *$\geq$-smaller* than it.
  - On the Hasse diagram, 1 is at the top, and the diagram extends infinitely *downwards*.

Notes on Converse Orders

Note:

- Maximal elements in $\langle S, \leq \rangle$ become minimal in $\langle S, \geq \rangle$ and vice versa.
- Greatest element in $\langle S, \leq \rangle$ becomes least in $\langle S, \geq \rangle$ and vice versa.
- Chains and antichains remain the same in both orders.
- The Hasse diagram is flipped vertically when taking the dual order.
  - For $\langle \mathbb{N}^+, \leq \rangle$:
    - The *least* element 1 is at the bottom.
    - The diagram of $\langle \mathbb{N}^+, \leq \rangle$ extends infinitely *upwards*.
  - In the dual $\langle \mathbb{N}^+, \geq \rangle$:
    - 1 becomes the *greatest* element at the top.
    - The diagram of $\langle \mathbb{N}^+, \geq \rangle$ extends infinitely *downwards*.

Chains and Antichains

Definition 55: In a partially ordered set $\langle M, \leq \rangle$:

- A *chain* is a subset $C \subseteq M$ where every two elements are comparable. Formally:

$$\forall x, y \in C. \, (x \leq y \text{ or } y \leq x)$$

- An *antichain* is a subset $A \subseteq M$ where no two distinct elements are comparable. Formally:

$$\forall x, y \in A. \, (x \neq y) \rightarrow (x \not\leq y \text{ and } y \not\leq x)$$

Note:

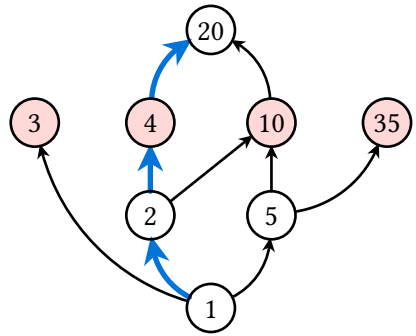
- Chains correspond to arbitrary paths or sub-sequences in the Hasse diagram.
- A *maximal chain* is a chain that cannot be extended by including any other elements from M.
- A *maximum chain* is a chain of the largest possible size in M.
- A chain is a *totally ordered subset* of the poset.
- An antichain consists of *pairwise incomparable elements*.
- Any singleton set is both a chain and an antichain.

Examples of Chains and Antichains

Example: Consider the divisibility poset $\langle D, | \rangle$ where $D = \{1, 2, 3, 4, 5, 6, 10, 20, 35\}$.

Chains: (totally ordered subsets)

- Maximal chain: $\{1, 2, 4, 20\}$ — longest path
- Not maximal: $\{5, 10\}$
- Can *skip* elements: $\{1, 5, 20\}$



Maximal elements: 3, 20, 35

Antichains: (pairwise incomparable elements)

- Maximal antichain: $\{3, 4, 10, 35\}$
- Incomparable on the same level: $\{2, 5\}$
- Incomparable from different levels: $\{3, 2, 35\}$

Dilworth's theorem: Maximum antichain size (4) = minimum chains needed to cover (4).

Examples of Chains and Antichains [2]

Example: In a Git repository, commits form a poset under the "ancestor" relation:

- **Chain:** A sequence of commits on a single branch (linear history).
- **Antichain:** Commits on different branches that have diverged (no ancestry relation).

Practical insight: Merge commits combine multiple antichains back into a single chain.

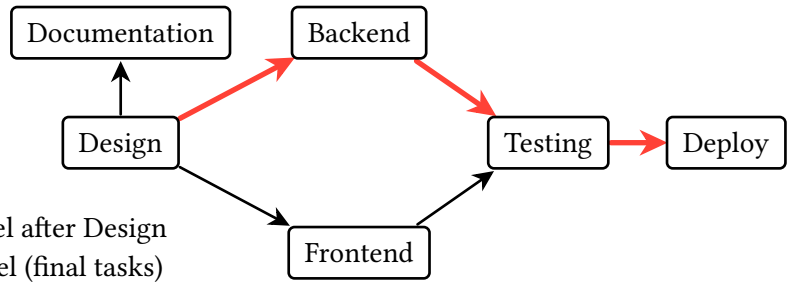
Chains and Antichains in Scheduling

Example: In project management, tasks form a scheduling poset under the "*prerequisite*" relation.

Consider web development tasks: Design, Backend, Frontend, Testing, Deploy, Documentation.

Dependencies:

- Design $\prec$ Back, Front
- Back, Front $\prec$ Test
- Test $\prec$ Deploy $\prec$ Doc



Antichain analysis:

- {Back, Front} can run in parallel after Design
- {Deploy, Doc} can run in parallel (final tasks)

Critical path: Design $\to$ Back $\to$ Test $\to$ Deploy (length 3 max chain)

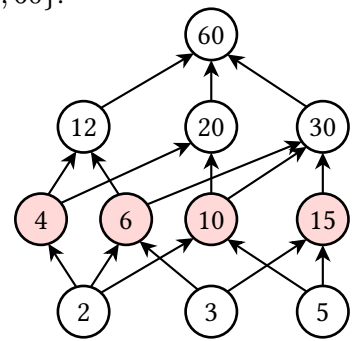
Practical insights:

- Chains = sequential dependencies (critical path)
- Antichains = tasks for parallel execution (resource allocation)
- Project duration = length of longest chain

Dilworth's Theorem

Theorem 10 (Dilworth): In any finite partially ordered set, the maximum size of an antichain equals the minimum number of chains needed to cover the entire set.

Example: Consider the divisibility poset on $P = \{2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$.



Maximum antichain: $\{4, 6, 10, 15\}$ (size 4)

- These elements are pairwise *incomparable* (none divides another).

Minimum chain decomposition: We need exactly 4 chains that *cover* P:

- Chain 1: $2 \mid 4 \mid 12 \mid 60$
- Chain 2: $3 \mid 6 \mid 30$
- Chain 3: $5 \mid 10 \mid 20$
- Chain 4: 15 (singleton chain)

Each element appears in exactly one chain, forming a *partition* of P.

Dilworth's theorem: Maximum *antichain* size (4) = Minimum number of *disjoint chains* (4). ✓

Proof of Dilworth's Theorem

Proof: Let $\langle P, \leq \rangle$ be a finite poset. Let α denote the maximum size of an antichain in P, and let β denote the minimum number of chains needed to cover P. We prove $\alpha = \beta$ by showing $\alpha \leq \beta$ and $\alpha \geq \beta$.

Easy part ($\alpha \leq \beta$): Suppose P can be partitioned into k chains $C_1, ..., C_k$. Let A be any antichain in P. Since elements in an antichain are pairwise incomparable, each chain contains at most one element of A. Therefore $|A| \leq k$. Taking the maximum over all antichains gives $\alpha \leq k$. Since this holds for any chain partition, we have $\alpha \leq \beta$. ■

Proof of Dilworth's Theorem [2]

Hard part ($\alpha \geq \beta$): Let $A \subseteq P$ be a maximal antichain of size α.

We construct a chain partition of P of size α as follows:

- Initialize $\mathcal{C} := \emptyset$.
- While $P \neq \emptyset$:
  1. Choose a maximal element $x \in P$ (no element above x in the remaining poset).
  2. Build a chain C ending at x:
     - Start with $C := \{x\}$.
     - Repeatedly add a maximal *predecessor* element $y \in P \setminus C$ such that $y <$ current bottom of C.
  3. Add C to $\mathcal{C}$ and remove all elements of C from P.

By construction:

- Each $C \in \mathcal{C}$ is a chain (elements are added only below the current bottom).
- Chains in $\mathcal{C}$ cover all elements of P.
- Each chain contains exactly one element of the maximal antichain A, so $|\mathcal{C}| = \alpha$.

Therefore P can be covered by α chains, giving $\beta \leq \alpha$. □

Summary: Orders

Orders provide structured ways to compare and rank elements:

- *Preorders:* Basic comparison (reflexive, transitive)
- *Partial orders:* Add antisymmetry for unique comparisons
- *Total orders:* Every pair of elements is comparable

Visualization: Hasse diagrams clearly show structure and hierarchy by omitting redundant transitive edges, revealing *chains* (ordered sequences) and *antichains* (incomparable elements).

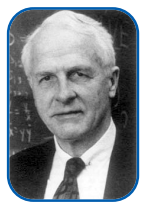
Applications: Task scheduling • Git version control • Database indexing • Type hierarchies • Boolean algebra • Concurrent systems • Build systems (dependency resolution) • Social networks • File system permissions • Web page ranking • Package managers • Distributed systems

Lattices

"Order is the shape upon which beauty depends."

— Pearl S. Buck


Alfred Tarski


Garrett Birkhoff


Dana Scott


Emmy Noether


Marshall Stone

Upper and Lower Bounds

Definition 56: In a poset $\langle S, \leq \rangle$, an element $u \in S$ is called an *upper bound* of a subset $C \subseteq S$ if it is greater than or equal to every element in C, *i.e.*, for all $x \in C$, $x \leq u$.

Definition 57: In a poset $\langle S, \leq \rangle$, an element $l \in S$ is called a *lower bound* of a subset $C \subseteq S$ if it is less than or equal to every element in C, *i.e.*, for all $x \in C$, $l \leq x$.

Note: In the simplest case, we consider a single-element subset: the upper/lower bounds of $C = \{c\}$ are just all elements greater/less than c (including c itself, since $c \leq c$). In a more general case, upper/lower bounds must be comparable to *all* elements in C.

Examples of Bounds

Example: In $\langle \mathbb{Z}, \leq \rangle$ for $C = \{-2, 3, 7\}$:

- **Lower bounds:** all integers ≤ -2, *i.e.*, $\{..., -4, -3, -2\}$
- **Upper bounds:** all integers ≥ 7, *i.e.*, $\{7, 8, 9, ...\}$

Example: In $\langle \mathbb{R}, \leq \rangle$ for interval $C = (0; 1)$:

- **Lower bounds:** every $x \leq 0$ (including $-\infty, -1, 0$)
- **Upper bounds:** every $x \geq 1$ (including $1, 2, +\infty$)

Example: In $\langle \mathcal{P}(\{1, 2, 3\}), \subseteq \rangle$ for $C = \{\{1, 2\}, \{1, 3\}\}$:

- **Lower bounds:** $\emptyset$, $\{1\}$ (subsets of both sets in C)
- **Upper bounds:** $\{1, 2, 3\}$ (supersets of both sets in C)

Example: In divisibility poset for $C = \{4, 6\}$:

- **Upper bounds:** multiples of $\text{lcm}(4, 6) = 12$, *i.e.*, $\{12, 24, 36, ...\}$
- **Lower bounds:** common divisors, *i.e.*, $\{1, 2\}$

Suprema and Infima

Definition 58: In a poset $\langle S, \leq \rangle$, the *supremum* (or *join*) of a subset $C \subseteq S$, denoted $\sup(C)$ or $\bigvee C$, is the *least upper bound* of C, *i.e.*, an upper bound $u \in S$ s.t. for any other upper bound $v \in S$, $u \leq v$.

Note: If it exists, the least upper bound is *unique*.

Definition 59: In a poset $\langle S, \leq \rangle$, the *infimum* (or *meet*) of a subset $C \subseteq S$, denoted $\inf(C)$ or $\bigwedge C$, is the *greatest lower bound* of C, *i.e.*, a lower bound $l \in S$ s.t. for any other lower bound $m \in S$, $m \leq l$.

Note: If it exists, the greatest lower bound is *unique*.

Example: $\langle \mathbb{R}, \leq \rangle$:

- For finite subsets, $\sup(C) = \max(C)$ and $\inf(C) = \min(C)$.
- For infinite subsets: $\sup((0;1)) = 1$ and $\inf((0;1)) = 0$ (even though $0, 1 \notin (0;1)$)

Examples of Suprema and Infima

Example: $\langle \mathcal{P}(A), \subseteq \rangle$:

- **Join:** $\sup(\mathcal{C}) = \bigcup_{X \in \mathcal{C}} X$ (union of all sets)
- **Meet:** $\inf(\mathcal{C}) = \bigcap_{X \in \mathcal{C}} X$ (intersection of all sets)
- $\sup\{\{1,2\},\{2,3\},\{3,4\}\} = \{1,2,3,4\}$
- $\inf\{\{1,2,3\},\{2,3,4\},\{3,4,5\}\} = \{3\}$

Example: Divisibility on $\mathbb{N}^+$:

- **Join:** $\sup\{a,b\} = \operatorname{lcm}(a,b)$ (least common multiple)
- **Meet:** $\inf\{a,b\} = \gcd(a,b)$ (greatest common divisor)
- $\sup\{6,10\} = 30$
- $\inf\{6,10\} = 2$

Lattices

Definition 60: A poset $\langle S, \leq \rangle$ where every non-empty finite subset $C \subseteq S$ has a join (supremum) is called an *upper semilattice* (or *join-semilattice*) and denoted $\langle S, \vee \rangle$.

Definition 61: A poset $\langle S, \leq \rangle$ where every non-empty finite subset $C \subseteq S$ has a meet (infimum) is called a *lower semilattice* (or *meet-semilattice*) and denoted $\langle S, \wedge \rangle$.

Definition 62: A poset $\langle S, \leq \rangle$ that is both an upper semilattice and a lower semilattice, *i.e.*, every non-empty finite subset has both a join and a meet, is called a *lattice*, denoted $(S, \vee, \wedge)$.

Example (Boolean Lattice): $\langle \mathcal{P}(X), \subseteq \rangle$ is a bounded distributive lattice $(\mathcal{P}(X), \cup, \cap)$:

- Elements are subsets of X, ordered by inclusion ($\subseteq$)
- Join is union ($\vee = \cup$), meet is intersection ($\wedge = \cap$)
- Top element is X (greatest element), bottom is $\emptyset$ (least element)

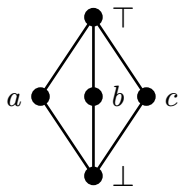
Examples: Small Lattices

Boolean Lattice B_1

Operations:

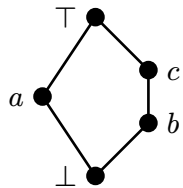
- $\top \vee \top = \top$
- $\bot \vee \bot = \bot$
- $\bot \vee \top = \top$
- $\bot \wedge \top = \bot$

Diamond Lattice M_3



- Smallest non-distributive
- But modular!

Pentagon Lattice N_5



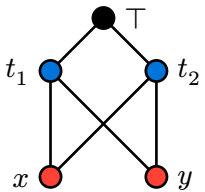
Smallest non-modular: for $b \leq c$:

$$\left.\begin{array}{l} b \vee (a \wedge c) = b \\ (b \vee a) \wedge c = c \end{array}\right\} \quad b \neq c \ \text{✗}$$

Key insight: These small lattices are fundamental building blocks. Many properties (distributivity, modularity) can be characterized by whether these appear as sublattices.

Counterexamples: Not All Posets Are Lattices!

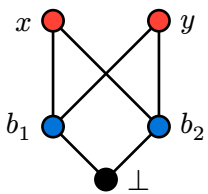
No Supremum



$\{x, y\}$ has *two* minimal upper bounds: t_1 and t_2

Not a lattice ✗

No Infimum



$\{x, y\}$ has *two* maximal lower bounds: b_1 and b_2

Not a lattice ✗

No Bounds

$\{x, y\}$ does *not* have any upper/lower bounds

Not a lattice ✗

Warning: A poset must have *unique* least upper bounds and greatest lower bounds for *every pair* of elements to be a lattice. Even one missing join or meet makes it not a lattice!

Why Lattices?

Why study lattices? Whenever you have:

- Elements that can be *compared* (ordered)
- Ways to *combine* elements (join, meet)
- Consistent behavior under combination

...you likely have a *lattice*!

This structure appears in programming languages, databases, security systems, logic circuits, and many other areas of computer science and mathematics.

Properties of Lattices

Definition 63: A lattice is *bounded* if it has a greatest element $\top$ and a least element $\bot$.

Example:

- $\langle \mathcal{P}(A), \subseteq \rangle$: bounded with $\top = A$, $\bot = \emptyset$
- $\langle \mathbb{N}^+, | \rangle$: bounded below ($\bot = 1$) but not above (no element divisible by all)
- $\langle \mathbb{Z}, \leq \rangle$: not bounded (no maximum or minimum)

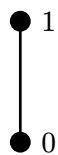
Distributive Lattices

Definition 64: A lattice is *distributive* if it satisfies the distributive laws:

$$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$$
$$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$$

Note: In a lattice, one law implies the other.

Example: B_1 is distributive



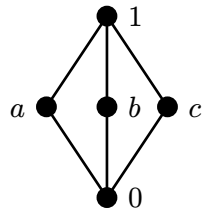
Check:

- $0 \wedge (0 \vee 1) = 0 \wedge 1 = 0$
- $(0 \wedge 0) \vee (0 \wedge 1) = 0 \vee 0 = 0$
- $0 = 0$ ✓
- Similarly for dual law

All Boolean algebras are distributive.

Non-Distributive Lattices

Example: M_3 is *not* distributive



Check:

- $a \wedge (b \vee c) = a \wedge 1 = a$
- $(a \wedge b) \vee (a \wedge c) = 0 \vee 0 = 0$
- $a \neq 0$ ✗

The diamond M_3 is the forbidden sublattice for distributivity.

A lattice is distributive if and only if it does not have a sublattice isomorphic to M_3 or N_5.

Distributive Lattices: Applications

Example: In compiler optimization, *interprocedural dataflow analysis* uses distributive lattices:

- **Elements:** Sets of dataflow *facts* (*e.g.*, "variable x is constant", "pointer p may be null")
- **Order:** $S_1 \leq S_2$ if $S_1 \subseteq S_2$ (subset of facts)
- **Join:** $S_1 \vee S_2 = S_1 \cup S_2$ (merge facts from different paths)
- **Meet:** $S_1 \wedge S_2 = S_1 \cap S_2$ (common facts)

Why distributivity matters?

- Distributivity ensures that $f(x \vee y) = f(x) \vee f(y)$ for flow functions f.
- This allows *separate analysis* of different execution paths with guaranteed precision when *merged*.
- IFDS (Interprocedural, Finite, Distributive, Subset) problems (*e.g.*, reachability, *taint analysis*) are solvable in polynomial time precisely because the lattice is distributive!

Modular Lattices

Definition 65: A lattice is *modular* if for all x, y, z with $x \leq z$:

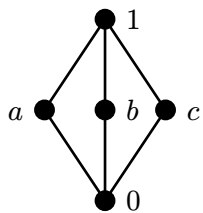
$$x \vee (y \wedge z) = (x \vee y) \wedge z$$

Note: Every distributive lattice is modular, but not vice versa.

Example: Diamond lattice M_3 is modular.

For $a \leq 1$:

- $a \vee (b \wedge 1) = a \vee b = 1$
- $(a \vee b) \wedge 1 = 1 \wedge 1 = 1$
- $1 = 1$ ✓



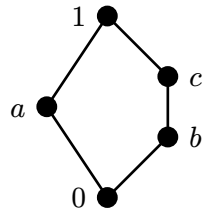
Note: M_3 is modular but NOT distributive!

Non-Modular Lattices

Example: Pentagon lattice N_5 is *not* modular.

For $b \leq c$:

- $b \vee (a \wedge c) = b \vee 0 = b$
- $(b \vee a) \wedge c = 1 \wedge c = c$
- $b \neq c$ ✗



The pentagon N_5 is the forbidden sublattice for modularity.

A lattice is modular if and only if it does not have a sublattice isomorphic to N_5.

Modular Lattices: Applications

Hierarchy: Lattice $\Rightarrow$ Modular $\Rightarrow$ Distributive

Each level adds structure and enables different applications.

Example: The subspaces of a vector space V, ordered by inclusion, form a modular lattice.

- **Elements:** subspaces $U, W, Z \subseteq V$
- **Join:** $U \vee W = U + W$ (sum / span of $U \cup W$)
- **Meet:** $U \wedge W = U \cap W$

For any two subspaces U, W of V, we have the *dimension formula*:

$$\dim(U + W) = \dim(U) + \dim(W) - \dim(U \cap W)$$

This expresses how dimensions "add up" when combining subspaces.

This is exactly the modular law in disguise!

The modular law governs how join ($+$) and meet ($\cap$) interact when one subspace is contained in another.

Modular Lattices: Applications [2]

Modular law: Whenever $U \subseteq Z$, we have: $U \vee (W \wedge Z) = (U \vee W) \wedge Z$

Proof: We show both inclusions of the equality: $U + (W \cap Z) = (U + W) \cap Z$

Step 1 ($\subseteq$):

- Suppose $x \in U + (W \cap Z)$.
- Then $x = u + w'$ for some $u \in U$, $w' \in W \cap Z$.
- Since $u \in U \subseteq Z$ and $w' \in Z$, we have $x \in Z$.
- Also $x \in U + W$, hence $x \in (U + W) \cap Z$.

Step 2 ($\supseteq$):

- Suppose $x \in (U + W) \cap Z$.
- Then $x = u + w$ for some $u \in U$, $w \in W$, and $x \in Z$.
- Because $U \subseteq Z$, we have $u \in Z$, so $w = x - u \in Z$.
- Thus $w \in W \cap Z$, and therefore $x = u + w \in U + (W \cap Z)$.

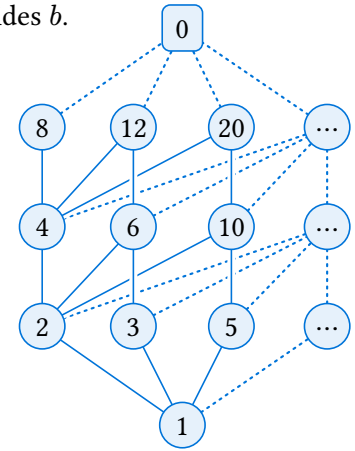
Combining both directions, we conclude: $U + (W \cap Z) = (U + W) \cap Z$. □

Interpretation: "Joining U and W, then cutting by Z" equals "cutting W by Z, then joining with U".

Example: Divisibility Lattice

Positive integers form a lattice under divisibility relation: $a \leq b$ iff a divides b.

- **Join (LCM):** $6 \vee 10 = \text{lcm}(6, 10) = 30$
- **Meet (GCD):** $6 \wedge 10 = \gcd(6, 10) = 2$
- **Bottom element:** 1 (divides everything)
- **No top element:** No positive integer is divisible by all others
  - **If we include 0:** then 0 is the top (divisible by all!)



Applications:

- Number theory and cryptography (RSA key generation)
- Computer algebra systems (polynomial GCD algorithms)
- Scheduling problems (finding common time periods)

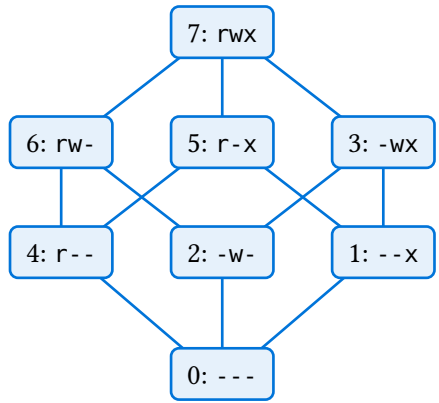
Lattices aren't just abstract algebra — they appear everywhere in computer science and mathematics.

The *join* and *meet* operations capture fundamental patterns of *combination* and *interaction*.

Example: File System Permissions

Unix file permissions form a lattice under inclusion.

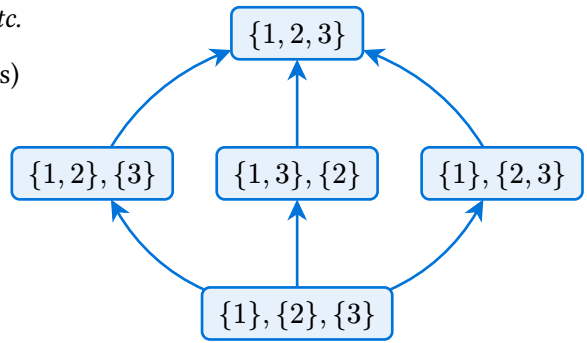
- **Elements:** Sets of permissions like $\{r, w, x\}$, $\{r, x\}$, $\{w\}$, *etc.*
- **Order:** $P_1 \leq P_2$ if $P_1 \subseteq P_2$ (fewer permissions $\Rightarrow$ more restrictive)
- **Join:** Union of permissions (less restrictive)
  - For example: $\{\text{read}\} \vee \{\text{execute}\} = \{\text{read}, \text{execute}\}$
- **Meet:** Intersection of permissions (more restrictive)
  - For example: $\{\text{read}, \text{write}\} \wedge \{\text{write}, \text{execute}\} = \{\text{write}\}$



Example: Partition Lattice

All partitions of a set S, ordered by refinement, form a lattice.

- **Elements:** Partitions like $(12 \mid 3)$, $(1 \mid 23)$, $(1 \mid 2 \mid 3)$, *etc.*
- **Order:** $\pi_1 \leq \pi_2$ if π_1 is a refinement of π_2 (smaller blocks)
- **Join:** Finest common coarsening
  - For example: $(12 \mid 3) \vee (1 \mid 23) = (123)$
- **Meet:** Coarsest common refinement
  - For example: $(12 \mid 3) \wedge (1 \mid 23) = (1 \mid 2 \mid 3)$



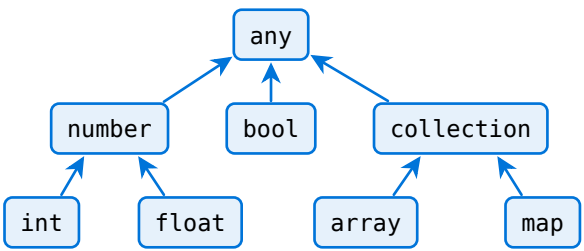
Example: Type Systems as Lattices

In programming language theory, *types* form a lattice under the *subtyping* relation $\sqsubseteq$.

The subtyping relation $\tau_1 \sqsubseteq \tau_2$ means "τ_1 is more specific than τ_2" (values of type τ_1 can be safely used where type τ_2 is expected).

Join and meet operations solve type inference problems of finding common supertypes and subtypes.

Consider a type system with numeric, boolean, and collection types:



Lattice operations:

Join ($\vee$): most specific common supertype

- `int` $\vee$ `float` = `number`
- `int` $\vee$ `bool` = `any`
- `array` $\vee$ `map` = `collection`

Meet ($\wedge$): most general common subtype

- `number` $\wedge$ `collection` = $\bot$ (bottom type)
- `int` $\wedge$ `number` = `int`

Example: Type Systems as Lattices [2]

Example (Type inference in conditionals): The code snippet below shows how join operations automatically compute types in conditional expressions:

```
let x = cond ? 42 : 3.14    // type: int ∨ float = number`
let y = cond ? x : true     // type: number ∨ bool = any`
```

The join operation gives the least upper bound type.

Example (Generic function bounds): Bounds in generic functions constrain what types can be passed:

```
<T extends Number> void process(T x) // Java
def f[T <: Number](x: T) // Scala
```

The type checker verifies that $T \sqsubseteq \text{Number}$ by checking lattice relationships.

Note: Lattice-based type systems enable *gradual typing* (mixing typed and untyped code) and sophisticated *type inference*. Examples: TypeScript, Flow, Dart, Cython, mypy.

Complemented Lattices

Definition 66: A lattice is *complemented* if for every element x there exists an element y (called the *complement* of x) such that:

$$x \vee y = \top$$
$$x \wedge y = \bot$$

Note: Complements may not be unique in general!

Boolean Algebras

Definition 67: A *Boolean algebra* is a complemented distributive lattice $(B, \vee, \wedge, \neg, \top, \bot)$:
- a set B
- with two binary operations $\vee$ (join) and $\wedge$ (meet),
- a unary operation $\neg$ (complement),
- and two distinguished elements $\top$ (top) and $\bot$ (bottom).

Note: The complement $\neg x$ is *unique* for each x in a Boolean algebras!

Example: $B_1 = \mathbf{2} = (\{0,1\}, \vee, \wedge, \neg, 1, 0)$ is the two-element Boolean algebra ($0 =$ `false`, $1 =$ `true`).

Isomorphic to (same structure up to renaming):
- Classical semantics: $(\{0,1\}, \max, \min, 1-x)$
- Algebra on sets: $(\mathcal{P}(\{\star\}), \cup, \cap, \complement)$
- GF(2) algebra: $(\{0,1\}, \oplus, \wedge, 1 \oplus x)$

x	y	$x \vee y$	$x \wedge y$	$\neg x$
0	0	0	0	1
0	1	1	0	1
1	0	1	0	0
1	1	1	1	0

Boolean Lattices as Boolean Algebras

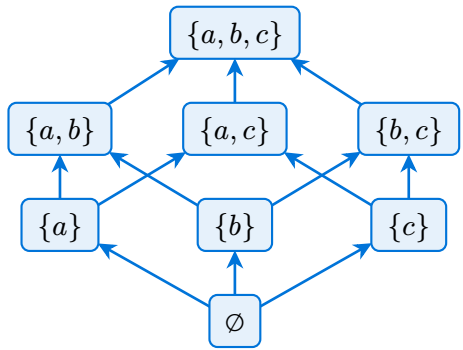
Example: Boolean lattice $B_3 = \left(\mathcal{P}(\{a, b, c\}), \cup, \cap, \overline{X}\right)$ is a Boolean algebra.

Elements: Subsets of $\{a, b, c\}$ ordered by inclusion ($\subseteq$)

Operations:

- Join: $A \vee B = A \cup B$
- Meet: $A \wedge B = A \cap B$
- Complement: $\neg A = \overline{A} = \{a, b, c\} \setminus A$
- Top: $\{a, b, c\}$
- Bottom: $\emptyset$

Check: $\{a\} \vee \neg\{a\} = \{a\} \cup \{b, c\} = \{a, b, c\} = \top$ ✓



Stone's Representation Theorem: Every Boolean algebra is isomorphic to a Boolean algebra of sets.

Boolean Algebras: Applications

Digital Circuit Design

Boolean algebras model logic gates:

- Variables: Input signals (0/1, false/true)
- Join: OR gate ($\vee$)
- Meet: AND gate ($\wedge$)
- Complement: NOT gate ($\neg$)

Applications: CPU design, memory circuits, FPGA programming

Key insight: Boolean algebras unify:

- Set theory (powerset operations)
- Logic (propositional calculus)
- Circuits (digital hardware)

Propositional Logic

Boolean algebras formalize logical reasoning:

- Elements: Truth values or propositions
- Join: Logical OR ($p \vee q$)
- Meet: Logical AND ($p \wedge q$)
- Complement: Negation ($\neg p$)

Connection to lattices:

- $\varphi \leq \psi$ means "φ implies ψ"
- Disjunction is join (weaker formula)
- Conjunction is meet (stronger formula)

Applications: SAT solvers, theorem provers, formal verification

Summary: Lattice Theory

Core definitions:

- **Lattice:** Poset where every pair has a unique join (sup) and meet (inf)
- **Bounded lattice:** Has top ($\top$) and bottom ($\bot$) elements
- **Distributive lattice:** Satisfies $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
- **Modular lattice:** Each $x \leq z$ satisfies $x \vee (y \wedge z) = (x \vee y) \wedge z$
- **Boolean algebra:** Complemented distributive lattice with $x \vee \neg x = \top$, $x \wedge \neg x = \bot$

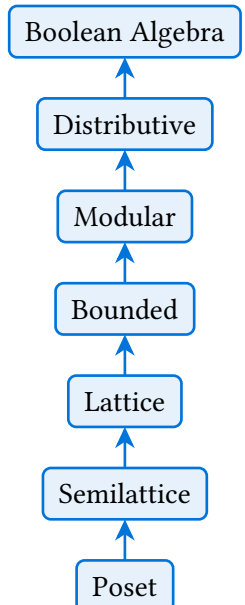
Key properties:

- Distributive lattice $\iff$ no M_3 or N_5 sublattice
- Modular lattice $\iff$ no N_5 sublattice
- **Stone's theorem:** Every Boolean algebra is isomorphic to some powerset $\mathcal{P}(S)$

Hierarchy of Lattice Structures

Key Examples of Lattices:

- $B_n = \mathcal{P}(\{x_1, ..., x_n\})$: Boolean lattice
  - Boolean algebra of sets
  - Universal model (by Stone's theorem)
- $\mathbf{2} = B_1$: Two-element Boolean algebra
  - Distributive ✓
  - Models: logic, circuits, sets
- M_3: Diamond lattice
  - Modular ✓, but NOT distributive ✗
  - Forbidden for distributivity
- N_5: Pentagon lattice
  - Lattice ✓, but NOT modular ✗
  - Forbidden for modularity



Summary: Applications of Lattices

Main applications:

- **Program analysis:** Dataflow analysis (IFDS framework), abstract interpretation
- **Type systems:** Subtyping, type inference, generic programming
- **Digital circuits:** Logic gate design, circuit optimization, BDDs
- **Abstract algebra:** Subspace lattices in linear algebra
- **Logic:** Propositional calculus, SAT solvers, theorem proving

Connection to next topics:

- **Well Orders:** Special total orders where every subset has a minimum
- **Cardinality:** Comparing sizes of infinite sets using bijections
- **Boolean Algebra:** Deep dive into axioms, laws, circuits, and applications
- **Formal Logic:** Lattices provide algebraic semantics for logical systems

Well Orders

"Every set can be well-ordered."

— Ernst Zermelo (Axiom of Choice)

Well-Ordered Sets

Definition 68: A poset $\langle M, \leq \rangle$ is *well-ordered* if every non-empty subset $S \subseteq M$ has a *least element*.

Formally: $\forall S \subseteq M.\, (S \neq \emptyset) \to (\exists m \in S.\, \forall x \in S.\, m \leq x)$

The key property: No matter how you pick elements from a well-ordered set, there's always a "first" one in any selected subset.

This is stronger than just having a minimum — well-ordering means you can't descend infinitely!

Note: A well-ordered set is automatically a *total order* (any two elements are comparable).

Examples: Well-Ordered vs Not

Example (The natural numbers): $\mathbb{N} = \{0, 1, 2, 3, \ldots\}$ with $\leq$ is *well-ordered*:

- $\{5, 17, 23, 100\}$ has least element 5
- $\{2, 4, 6, 8, \ldots\}$ has least element 2
- Even $\{n \in \mathbb{N} \mid n \geq 1000\}$ has least element 1000

Why this matters: You can always find a "starting point" in any subset!

Example (The integers): $\mathbb{Z}$ with $\leq$ is *NOT well-ordered*:

- $\{-1, -2, -3, \ldots\}$ has no minimum
- $\mathbb{Z}$ itself has no least element
- Can descend forever: $0 > -1 > -2 > -3 > \ldots$

The problem: Negative numbers allow infinite descent.

Intuition: Well-ordering prevents "falling forever." Every subset has a floor you can't go below.

Example: Ordinals

Example (Ordinal numbers): Ordinals extend natural numbers into the transfinite:

- ω = all natural numbers: $0, 1, 2, 3, \ldots$
- $\omega + 1$ = natural numbers plus one more element "at the end"
- $\omega + 2 = \omega + 1$ plus another element
- $\omega \cdot 2$ = two copies of ω in sequence
- $\omega^2, \omega^3, \ldots, \omega^\omega, \ldots$

Each ordinal is well-ordered, and ordinals themselves are well-ordered!

Key insight: Ordinals provide a canonical way to label positions in well-ordered sets.

Example: String Orderings

Two common ways to order *finite* strings over an alphabet:

Definition 69 (Shortlex order): Compare by length first, then alphabetically:

$$\varepsilon \prec \mathrm{a} \prec \mathrm{b} \prec \ldots \prec \mathrm{aa} \prec \mathrm{ab} \prec \ldots$$

Well-ordered: Every set has a shortest string, ties broken alphabetically.

Example: $\{\mathrm{cat}, \mathrm{dog}, \mathrm{ox}, \mathrm{zebra}\}$ has minimum "ox" (shortest).

Definition 70 (Dictionary order): Compare alphabetically only, ignoring length:

$$\mathrm{a} \prec \mathrm{aa} \prec \mathrm{aaa} \prec \ldots \prec \mathrm{b} \prec \mathrm{ba} \prec \ldots$$

NOT well-ordered: $\{\mathrm{b}, \mathrm{ab}, \mathrm{aab}, \mathrm{aaab}, \ldots\}$ has no minimum!

Infinite descent: $\mathrm{b} \succ \mathrm{ab} \succ \mathrm{aab} \succ \ldots$ (because $\mathrm{a} < \mathrm{b}$ alphabetically).

Well-Ordering Enables Induction

Theorem 11: Let $\langle S, \leq \rangle$ be a well-ordered set. For any property P, if

$$\forall x \in S.\, (\forall y < x.\, P(y)) \to P(x)$$

then $P(x)$ holds for all $x \in S$.

Proof: Assume the hypothesis holds but P fails for some elements.

Let $T = \{x \in S \mid \neg P(x)\}$ be the set of counterexamples.

Since S is well-ordered, T has a *least element* m. This element m is the *smallest counterexample* to P.

Then $P(k)$ holds for all $k < m$ (otherwise m wouldn't be least).

By hypothesis, this implies $P(m)$ holds — contradicting $m \in T$!

Therefore $T = \emptyset$. □

Two Forms of Induction on $\mathbb{N}$

Definition 71 (Weak Induction): To prove $\forall n \in \mathbb{N}.\, P(n)$:
- **Base:** Prove $P(0)$
- **Step:** For any n, prove $P(n) \to P(n+1)$

Definition 72 (Strong Induction): To prove $\forall n \in \mathbb{N}.\, P(n)$:
- For any n, assume $P(k)$ for *all* $k < n$, then prove $P(n)$

Note: These are *equivalent* for $\mathbb{N}$. The difference is convenience:
- Weak: uses only $P(n-1)$ to prove $P(n)$
- Strong: can use any $P(k)$ with $k < n$ to prove $P(n)$

Example:
- **Weak:** Proving $\sum_{i=0}^{n} i = \frac{n(n+1)}{2}$ (only need $P(n)$ for $P(n+1)$)
- **Strong:** Proving Fibonacci formula (need $P(n-1)$ and $P(n-2)$ for $P(n)$)

Well-Ordering Principle

Definition 73 (Well-Ordering Principle (WOP)): Every non-empty subset of $\mathbb{N}$ has a least element.

Example:

- $\{5, 10, 15, 20, \ldots\}$ has least element 5
- $\{n \in \mathbb{N} \mid n^2 > 100\}$ has least element 11
- $\{n \in \mathbb{N} \mid n \text{ is prime}\}$ has least element 2

Note: WOP is an axiom in Peano arithmetic, not provable from simpler principles.

Key: If a property fails, there's a *smallest* counterexample — this makes proof by contradiction work.

WOP = Induction

Theorem 12: Well-Ordering Principle $\iff$ Mathematical Induction

Proof *(WOP $\implies$ Induction)*: Suppose WOP holds. To prove $\forall n \in \mathbb{N}.\, P(n)$, assume P fails.

Let $S = \{n \in \mathbb{N} \mid \neg P(n)\}$. By WOP, S has a least element m.

Then $P(k)$ holds for all $k < m$ (since m is least in S).

By the induction hypothesis, $P(m)$ holds — contradicting $m \in S$! □

Note: The reverse (Induction $\implies$ WOP) can be proven similarly.

Well-Founded Relations

Well-ordering requires total comparability. Can we relax this?

Definition 74: A relation $R \subseteq M^2$ is *well-founded* if every non-empty subset has a *minimal element*.

Formally: $\forall S \subseteq M.\, (S \neq \emptyset) \to (\exists m \in S.\, \forall x \in S.\, x \not R\, m)$

Note: **Well-ordered $\neq$ Well-founded**

- Well-ordered: unique *least* element (total order)
- Well-founded: one or more *minimal* elements (may be incomparable)
- Every well-ordered set is well-founded, but not vice versa.

Key insight: Both prevent infinite descent, but well-founded allows *incomparable* elements.

Example: Well-Founded but Not Well-Ordered

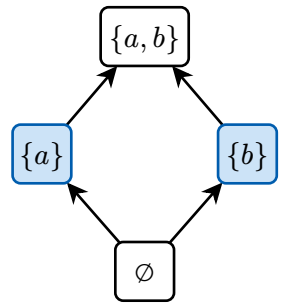
Example: Proper subset ($\subset$) relation on $M = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$

Is it well-founded? ✓

- Consider $S = \{\{a\}, \{b\}, \{a, b\}\}$
- Minimal elements: $\{a\}$ and $\{b\}$ (nothing in S is a proper subset of both)
- Note: $\{a, b\}$ is *not* minimal

Is it well-ordered? ✗

- $\{a\}$ and $\{b\}$ are *incomparable*
- No unique least element in S



Difference: Well-founded allows *multiple* incomparable minimals; well-ordered requires *unique* least.

Example: Divisibility Poset

Example: Consider $\langle \mathbb{N}^+, | \rangle$ with the divisibility relation.

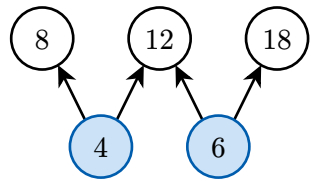
Take the subset $\{4, 6, 8, 12, 18\}$:

Minimal elements: 4 and 6

- No divisors of 4 or 6 in this set
- They're *incomparable*: $4 \nmid 6$ and $6 \nmid 4$

Conclusion:

- Well-founded: every subset has minimals ✓
- Well-ordered: no unique least ✗



Example: Set Membership $\in$

Definition 75: In ZFC set theory, the *Axiom of Regularity* (also called *Foundation*) states:

$$\forall S \neq \emptyset. \exists x \in S. (x \cap S = \emptyset)$$

Every non-empty set contains an element disjoint from itself.

Equivalently, every non-empty set has an *$\in$-minimal element*.

Examples:

- $\{\emptyset\}$ has minimal $\emptyset$
- $\{\{\emptyset\}, \{\{\emptyset\}\}\}$ has minimal $\{\emptyset\}$
- $\{\{1\}, \{2\}\}$ has two minimals: $\{1\}$ and $\{2\}$
- $\{1, 2, 3\}$ has minimal 1 (since $1 \in 2$ and $1 \in 3$)

Key properties of $\in$:

- **Well-founded?** ✓ (by Regularity)
- **Well-ordered?** ✗ (*e.g.* $\{1\} \notin \{2\}$ and $\{2\} \notin \{1\}$)
- **No** infinite descent ($\ldots \in x_2 \in x_1 \in x_0$)
- **No** self-membership ($x \in x$)
- **No** cycles ($x \in y \in x$)
- $\in$-induction!

Example: Program Termination

```
def factorial(n):
    if n == 0: return 1
    return n * factorial(n-1)
```

Claim: `factorial` terminates for all $n \in \mathbb{N}$.

Proof:

- Recursive calls form a chain: $n \to (n-1) \to (n-2) \to \ldots \to 1 \to 0$
- This is a *descending chain* in $\langle \mathbb{N}, < \rangle$
- Since $\langle \mathbb{N}, < \rangle$ is well-founded, no *infinite* descent
- Therefore, recursion terminates! □

Application: Verification tools (Dafny, Coq, Agda, Lean) use well-founded relations to prove termination of recursive functions.

Well-Founded Induction

Definition 76: Let $\langle S, \prec \rangle$ be well-founded. To prove $\forall x \in S.\, P(x)$, show:

$$\forall x \in S.\, (\forall y \prec x.\, P(y)) \to P(x)$$

Key insight: Assume $P(y)$ for all y "smaller" than x, then prove $P(x)$.

This *generalizes* strong induction — "smaller" is defined by $\prec$ instead of $<$.

Note: Also called: Noetherian induction, transfinite induction (on ordinals).

Proof of Well-Founded Induction

Proof: Assume the hypothesis but suppose P fails.

Let $S' = \{x \in S \mid \neg P(x)\}$. Since $\langle S, \prec \rangle$ is well-founded, S' has a minimal element x_0.

For any $y \prec x_0$: since x_0 is minimal, $y \notin S'$, so $P(y)$ holds.

By hypothesis: $(\forall y \prec x_0.\, P(y)) \to P(x_0)$.

So $P(x_0)$ holds — contradicting $x_0 \in S'$! □

Pattern:

- Well-foundedness gives minimal counterexample
- $\Rightarrow$ minimality makes hypothesis applicable
- $\Rightarrow$ contradiction

Example: Termination via Well-Founded Induction

```
def process(S):
    if S == ∅: return "done"
    x = S.pop()
    return process(S)
```

Claim: `process(S)` terminates for all finite S.

Proof: Use well-founded induction on $\langle \mathcal{P}_{\text{fin}}, \subset \rangle$ (finite sets ordered by proper inclusion).

Assume `process(T)` terminates for all $T \subset S$ (strictly smaller sets).

- If $S = \emptyset$: returns immediately ✓
- If $S \neq \emptyset$: recursive call uses $S \setminus \{x\} \subset S$, which terminates by hypothesis ✓

Therefore `process(S)` terminates for all finite S. □

Note: The relation $\subset$ on finite sets is well-founded but not well-ordered (sets $\{1\}$ and $\{2\}$ are incomparable).

Special Cases of Well-Founded Induction

Name	Structure	"Smaller" means
Strong induction	$\langle \mathbb{N}, > \rangle$	Numerically less
Transfinite induction	Ordinals	Earlier in order
Structural induction	Trees, lists	Proper substructure

Example (Structural induction on trees):

```
data Tree a = Empty | Node (Tree a) a (Tree a)
```

To prove $P(\text{tree})$ for all trees:

- Base: Prove $P(\text{Empty})$ (empty tree)
- Step: Assume $P(L)$ and $P(R)$, prove $P(\text{Node}(L, v, R))$

The "subtree" relation is well-founded.

Induced Strict Orders

To discuss chain conditions, we need *strict* orders derived from *partial* orders.

Definition 77 (Induced Strict Order): From any partial order $\leq$, we *induce* a strict order:

$$x < y \quad \text{iff} \quad x \leq y \text{ and } x \neq y$$

Similarly, $x > y$ means $y < x$ (the converse).

Convetion: For any poset $\langle S, \leq \rangle$, we freely use $<$ and $>$ for induced strict orders.

This lets us express strictly ascending/descending sequences: $x_1 < x_2 < x_3 < \ldots$

Example:

- In $\langle \mathbb{N}, \leq \rangle$: $3 < 5$ means $3 \leq 5$ and $3 \neq 5$
- In $\langle \mathcal{P}(M), \subseteq \rangle$: $\{a\} < \{a, b\}$ means $\{a\} \subseteq \{a, b\}$ and $\{a\} \neq \{a, b\}$

Descending Chain Condition (DCC)

Definition 78: A poset $\langle S, \leq \rangle$ satisfies DCC if there are no infinite strictly descending chains:

$$x_1 > x_2 > x_3 > x_4 > \ldots$$

Example: $\langle \mathbb{N}, \leq \rangle$ satisfies DCC:

Any descending sequence eventually stabilizes at 0.

Intuition: "Can't fall forever" — every descent hits bottom or stabilizes.

Well-Founded $\iff$ DCC

Theorem 13: R is well-founded $\iff$ R satisfies DCC

Proof *(Well-founded $\implies$ DCC)*: Suppose infinite descending chain $x_0 > x_1 > x_2 > \ldots$ exists.

Let $T = \{x_0, x_1, x_2, \ldots\}$. By well-foundedness, T has minimal x_k.

But $x_k > x_{k+1}$ contradicts minimality! □

Proof *(DCC $\implies$ Well-founded)*: Let $T \subseteq S$ be non-empty. Pick $x_0 \in T$.

If x_0 is not minimal, pick $x_1 \in T$ with $x_1 < x_0$. Repeat.

If this continues forever, we get infinite descent $x_0 > x_1 > x_2 > \ldots$, contradicting DCC. □

Ascending Chain Condition (ACC)

Definition 79: A poset satisfies ACC if no infinite strictly ascending chains exist:

$$x_1 < x_2 < x_3 < x_4 < \dots$$

Example:

- $\langle \mathbb{N}, \leq \rangle$ does NOT satisfy ACC: $0 < 1 < 2 < 3 < \dots$
- $\langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$ satisfies neither ACC nor DCC:
  - ACC fails: $\emptyset \subset \{1\} \subset \{1, 2\} \subset \dots$
  - DCC fails: $\mathbb{N} \supset \mathbb{N} \setminus \{1\} \supset \mathbb{N} \setminus \{1, 2\} \supset \dots$

Intuition: "Can't climb forever" — every ascent hits ceiling or stabilizes.

Noetherian Relations

Definition 80: R is *Noetherian* if R^{-1} is well-founded.

Equivalently: every non-empty subset has a *maximal* element.

Theorem 14: Noetherian $\iff$ ACC

Proof: Chain of equivalences:

$$\begin{aligned} R \text{ is Noetherian} &\iff R^{-1} \text{ is well-founded} \\ &\iff R^{-1} \text{ satisfies DCC} \\ &\iff R \text{ satisfies ACC} \end{aligned}$$

□

Duality: Well-founded has minimals + DCC; Noetherian has maximals + ACC.

Summary: Equivalences

Concept	Definition	Equivalent
Well-ordered	Unique least in every subset	Total + Well-founded
Well-founded	Minimal(s) in every subset	DCC
Noetherian	Maximal(s) in every subset	ACC

Key:

- Well-ordered $\to$ well-founded
- Well-founded $\iff$ DCC
- Noetherian $\iff$ ACC
- Well-founded and Noetherian are duals

Application: Noetherian Rings

Definition 81: A ring is *Noetherian* if every ascending chain of ideals stabilizes:

$$I_1 \subseteq I_2 \subseteq I_3 \subseteq \ldots$$

Example (ℤ is Noetherian): Ideals in $\mathbb{Z}$ have form $n\mathbb{Z} = \{nk \mid k \in \mathbb{Z}\}$ for $n \in \mathbb{N}$.

For chain $n_1\mathbb{Z} \subseteq n_2\mathbb{Z} \subseteq \ldots$:

- Inclusion means $n_2 \mid n_1$, so $n_1 \geq n_2 \geq \ldots$ in $\mathbb{N}$
- By DCC, this sequence stabilizes
- Therefore the ideal chain stabilizes

Chain Conditions: Key Examples

Poset	DCC?	ACC?
$\langle \mathbb{N}, \leq \rangle$	✓	✗
$\langle \mathbb{Z}^-, \leq \rangle$ (negative integers)	✗	✓
$\langle \mathcal{P}(\{1,2,3\}), \subseteq \rangle$ (finite)	✓	✓
$\langle \mathcal{P}(\mathbb{N}), \subseteq \rangle$ (infinite)	✗	✗

Key observation: DCC and ACC are *independent* — a structure can satisfy one, both, or neither!

Applications: From Theory to Practice

These abstract concepts — well-orders, well-founded relations, and chain conditions — are powerful tools for proving *termination*, reasoning about *infinity*, and establishing *finiteness* properties.

The unifying principle: All capture variants of *"no infinite regress"* — essential for proving processes terminate and structures are well-behaved.

Examples (Computer science):

- **Model checking:** Well-founded relations guarantee finite state-space exploration
- **Database systems:** Datalog uses well-founded semantics for recursive queries
- **Compiler optimization:** Prove termination of optimization passes via well-founded metrics
- **Graph algorithms:** DFS/BFS termination follows from well-founded visit ordering

Applications: From Theory to Practice [2]

Examples (Mathematics & logic):

- **Transfinite induction:** Well-ordering enables proofs about infinite ordinals
- **Set theory:** Von Neumann ordinals constructed as well-ordered sets
- **Rewriting systems:** Confluence and termination via Noetherian orderings (Knuth-Bendix)
- **Combinatorics:** Dickson's lemma uses well-quasi-orders for finiteness results

Examples (Software engineering):

- **Scheduling algorithms:** Priority queues maintain well-founded task ordering
- **Garbage collection:** Reachability analysis uses well-founded heap traversal
- **Distributed systems:** Lamport clocks establish well-founded event causality
- **Version control:** Commit DAGs form well-founded partial orders

Summary: Well-Founded Relations and Chain Conditions

Concept	Definition	Equivalent to
Well-ordered	Every subset has *unique least* element	Total + well-founded
Well-founded	Every subset has *minimal* elements	DCC
Noetherian	Every subset has *maximal* elements	ACC
DCC (Artinian)	No infinite descending chains	Well-founded
ACC (Noetherian)	No infinite ascending chains	Noetherian

Key relationships:

- Well-ordered $\rightarrow$ well-founded (least $\rightarrow$ minimal)
- Well-ordered $\rightarrow$ total order (least elements force comparability)
- Well-founded $\iff$ DCC (equivalent characterizations)
- Noetherian $\iff$ ACC (equivalent characterizations)

Cardinality & Infinity

"God made the integers, all else is the work of man."

— Leopold Kronecker

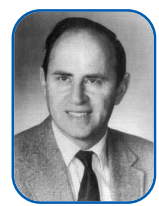
Giuseppe Peano

Leopold Kronecker

David Hilbert

Kurt Gödel

John von Neumann



Paul Cohen

Cardinality of Sets

Definition 82: The *cardinality* of a set X, denoted $|X|$, is a measure of its "size".

- For *finite* sets, cardinality equals the number of elements: $|\{a, b, c\}| = 3$.
- For *infinite* sets, cardinality describes the "type" of infinity, distinguishing between *countable* (like $\mathbb{N}$) and *uncountable* (like $\mathbb{R}$) infinities.

Key insight: Not all infinities are equal!
Some infinite sets are "larger" than others in a precise mathematical sense.

Example:

- $|\mathbb{N}| = \aleph_0$ (*"aleph-null"* — the smallest infinite cardinal, denoting *countable* infinity)
- $|\mathbb{Z}| = \aleph_0$ (surprisingly, same size as $\mathbb{N}$!)
- $|\mathbb{Q}| = \aleph_0$ (rationals are also countably infinite)
- $|\mathbb{R}| = 2^{\aleph_0} = \mathfrak{c}$ (*"continuum"* — *uncountably* infinite)
- $|\mathcal{P}(\mathbb{N})| = 2^{\aleph_0} > \aleph_0$ (power set is always larger than the original set)

Cardinal Numbers

Note: $|X|$ is not just a number, but a *cardinal number*.

- Cardinal numbers extend natural numbers to describe sizes of infinite sets
- The finite cardinals are: $0, 1, 2, 3, \ldots$
- The first infinite cardinal is $\aleph_0 = |\mathbb{N}|$
- Arithmetic on infinite cardinals behaves differently: $\aleph_0 + 1 = \aleph_0$ and $\aleph_0 \cdot 2 = \aleph_0$

Cantor's revolutionary insight (1874): We can compare sizes of infinite sets using bijections, just as we compare finite sets by counting!

Equinumerosity

Definition 83: Two sets A and B have the same *cardinality* and are called *equinumerous*, denoted $|A| = |B|$ or $A \approx B$, iff there exists a *bijection* (one-to-one correspondence) between A and B.

Intuition: If you can pair up every element of A with exactly one element of B, with nothing left over on either side, then A and B have the same cardinality.

Theorem 15: Equinumerosity is an equivalence relation.

Proof: Let A, B, C be sets.

- **Reflexivity:** The identity map $\mathrm{id}_A : A \to A$, where $\mathrm{id}_A(x) = x$, is a bijection, so $A \approx A$.
- **Symmetry:** Suppose $A \approx B$, then there is a bijection $f : A \to B$. Since it is a bijection, its inverse f^{-1} exists and is also a bijection. Hence, $f^{-1} : B \to A$ is a bijection, so $B \approx A$.
- **Transitivity:** Suppose that $A \approx B$ and $B \approx C$, *i.e.*, there are bijections $f : A \to B$ and $g : B \to C$. Then the composition $g \circ f : A \to C$ is also a bijection. So $A \approx C$.

Therefore, equinumerosity is an equivalence relation. □

Hilbert's Grand Hotel

Imagine a hotel with *infinitely many* rooms, numbered $1, 2, 3, \ldots$, and *all rooms are occupied*.

A new guest arrives. Can we accommodate them? **YES!**

The solution: Move each guest in room n to room $n + 1$.

- Guest in room 1 moves to room 2
- Guest in room 2 moves to room 3
- And so on...

Now room 1 is *vacant* for the new guest!

Formally: Define the *shift map* $f : \mathbb{N} \to \mathbb{N}$ by $f(n) = n + 1$

- **Injective:** Different rooms map (shift) to different rooms
- **Not surjective:** Room 1 has no pre-image

The hotel can accommodate one more guest even though it is "full"!

Dedekind-Infinite Sets

Definition 84: A set X is *Dedekind-infinite* if some proper subset $Y \subset X$ is equinumerous to it, *i.e.*, there is a *bijection* between X and one of its *proper* subsets.

Equivalently, X is Dedekind-infinite if there exists an *injective* but *not surjective* function $f : X \to X$.

A set that is *not* Dedekind-infinite is called *Dedekind-finite*.

Note: Intuitively, an "infinite" set can be put in one-to-one correspondence with a part of itself, which is impossible for finite sets.

Example: The set of natural numbers $\mathbb{N}$ is Dedekind-infinite:

- Let $Y = \{2, 4, 6, 8, \ldots\} = \mathbb{N}_{\text{even}} \subset \mathbb{N}$ (proper subset)
- Define $f : \mathbb{N} \to Y$ by $f(n) = 2n$ (bijection)
- Since $\mathbb{N} \approx \mathbb{N}_{\text{even}}$ (equinumerous, bijection exists), the set $\mathbb{N}$ is Dedekind-infinite

Examples of Dedekind-Infinite Sets

Example: **Integers** $\mathbb{Z}$ are Dedekind-infinite:

- Define $f : \mathbb{Z} \to \mathbb{Z}_{\text{even}}$ by $f(n) = 2n$
- This is a bijection: $\mathbb{Z} \approx \mathbb{Z}_{\text{even}} \subset \mathbb{Z}$

Example: **Rationals** $\mathbb{Q}$ are Dedekind-infinite:

- Define $f : \mathbb{Q} \to \mathbb{Q} \setminus [0; 1)$ by $f(x) = \begin{cases} x & \text{if } x<0 \\ x+1 & \text{if } x \geq 0 \end{cases}$
- This "skips" the interval $[0; 1)$, giving a bijection onto $\mathbb{Q} \setminus [0, 1) \subset \mathbb{Q}$

Example: **Real numbers** $(0; 1)$ are Dedekind-infinite:

- Define $f : (0; 1) \to \left(0; \frac{1}{2}\right)$ by $f(x) = \frac{x}{2}$
- This is a bijection: $(0; 1) \approx \left(0; \frac{1}{2}\right) \subset (0; 1)$

Key insight: Being Dedekind-infinite is equivalent to being infinite (assuming the Axiom of Choice).

This gives us a purely set-theoretic definition of infinity without counting!

Hilbert's Hotel: Infinitely Many New Guests

The Greater Challenge: Now a *bus* with infinitely many passengers arrives (numbered $1, 2, 3, \ldots$). The hotel is still completely full. Can we accommodate all of them? **YES!**

The solution:

Ask each current guest in room n to move to room $2n$:

$$f_{\text{current}} : \mathbb{N} \to \mathbb{N}_{\text{even}}, \quad f_{\text{current}}(n) = 2n$$

This creates infinitely many vacant *odd-numbered* rooms: $\{1, 3, 5, 7, 9, \ldots\}$

Then, assign bus passenger k to room $2k - 1$:

$$f_{\text{new}} : \mathbb{N} \to \mathbb{N}_{\text{odd}}, \quad f_{\text{new}}(k) = 2k - 1$$

Hilbert's Hotel: Infinitely Many New Guests [2]

Original	Current Guest	⟶	New Room	New Guest
Room 1	Guest #1		Room 2	
Room 2	Guest #2		Room 4	
Room 3	Guest #3		Room 6	
⋮	⋮		⋮	
			Room 1	Bus passenger #1
			Room 3	Bus passenger #2
			Room 5	Bus passenger #3
			⋮	⋮

This demonstrates that $\mathbb{N}$ (all guests) is equinumerous to $\mathbb{N}_{\text{even}}$ (current guests) *and* $\mathbb{N}_{\text{odd}}$ (new guests):

$$|\mathbb{N}| = |\mathbb{N}_{\text{even}}| = |\mathbb{N}_{\text{odd}}| = \aleph_0$$

Moreover, we showed that $\mathbb{N} \approx \mathbb{N}_{\text{even}} \sqcup \mathbb{N}_{\text{odd}}$, illustrating that $\aleph_0 + \aleph_0 = \aleph_0$.

Countable Sets

Definition 85: A set is *countable* if it is either finite, or has the same cardinality as $\mathbb{N}$ (*i.e.*, there exists a bijection with $\mathbb{N}$).

An infinite countable set has cardinality $\aleph_0$.

Key insight: A countable set is one whose elements can be "listed" in a sequence $a_0, a_1, a_2, \ldots$, pairing each with a natural number.

Example: **Integers** $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ are countable: $|\mathbb{Z}| = \aleph_0$

Bijection $f : \mathbb{N} \to \mathbb{Z}$:

$$f(n) = \begin{cases} \frac{n}{2} & \text{if } n \text{ is even} \\ -\frac{n+1}{2} & \text{if } n \text{ is odd} \end{cases} \qquad \begin{bmatrix} \mathbb{N}: & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & \ldots \\ \mathbb{Z}: & 0 & -1 & 1 & -2 & 2 & -3 & 3 & -4 & \ldots \end{bmatrix}$$

This systematically pairs each natural number with exactly one integer.

Properties of Countable Sets

Theorem 16:

1. Any subset of a countable set is countable
2. The union of countably many countable sets is countable
3. The Cartesian product of two countable sets is countable
4. The set of finite sequences over a countable alphabet is countable

Examples of Countable Sets

Example: **Finite binary strings** $\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$ over $\Sigma = \{0, 1\}$:

- List by length, then lexicographically: ε (length 0), then $0, 1$ (length 1), then $00, 01, 10, 11$ (length 2), *etc.*
- Since each length class is finite and there are countably many lengths, Σ^* is countable

Example: **Polynomials** with integer coefficients $\mathbb{Z}[x]$:

- Group by degree and coefficient sum, enumerate systematically
- Countable because expressible as finite sequences over $\mathbb{Z}$

Example: **Algebraic numbers** (roots of polynomials with integer coefficients):

- Each polynomial has finitely many roots
- "Countably many polynomials" $\times$ "finitely many roots each" $=$ countable

Zig-Zag Enumeration: $\mathbb{N}^2$ is Countable

Theorem 17: $\mathbb{N} \times \mathbb{N}$ is countable.

Proof: We enumerate pairs by diagonals of constant sum $s = n + k$ for $s = 0, 1, 2, \ldots$

$n \downarrow$ $k \rightarrow$	**0**	**1**	**2**	**3**	**...**
0	$\langle 0,0 \rangle$ 0	$\langle 0,1 \rangle$ 2	$\langle 0,2 \rangle$ 5	$\langle 0,3 \rangle$ 9	...
1	$\langle 1,0 \rangle$ 1	$\langle 1,1 \rangle$ 4	$\langle 1,2 \rangle$ 8	$\langle 1,3 \rangle$ 13	...
2	$\langle 2,0 \rangle$ 3	$\langle 2,1 \rangle$ 7	$\langle 2,2 \rangle$ 12	$\langle 2,3 \rangle$ 18	...
3	$\langle 3,0 \rangle$ 6	$\langle 3,1 \rangle$ 11	$\langle 3,2 \rangle$ 17	$\langle 3,3 \rangle$ 24	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$

Zig-Zag Enumeration: $\mathbb{N}^2$ is Countable [2]

The *Cantor pairing function* gives an explicit bijection $f : \mathbb{N}^2 \to \mathbb{N}$:

$$f(n,k) = \underbrace{\frac{(n+k)(n+k+1)}{2}}_{\text{pairs before diagonal } n+k} + \underbrace{k}_{\text{position within diagonal}}$$

Therefore $\mathbb{N}^2 \approx \mathbb{N}$, so $\left|\mathbb{N}^2\right| = \aleph_0$. □

Why this matters: "2-dimensional infinity" is the same size as "1-dimensional infinity"!

More generally, $\mathbb{N}^n$ is countable for any finite n.

Rationals are Countable

Theorem 18: $\mathbb{Q}$ is countable.

Proof: We construct an injection from $\mathbb{Q}$ into $\mathbb{N} \times \mathbb{N}$, which is countable.

Step 1: Map each positive rational $\frac{p}{q}$ (as a reduced fraction) to $(p, q) \in \mathbb{N} \times \mathbb{N}$.

- This is injective: different reduced fractions have different (p, q) pairs
- Since $\mathbb{N} \times \mathbb{N}$ is countable, $\mathbb{Q}^+ \preceq \mathbb{N} \times \mathbb{N}$ (injection) implies $\mathbb{Q}^+$ is countable

Step 2: Decompose $\mathbb{Q} = \mathbb{Q}^+ \cup \{0\} \cup \mathbb{Q}^-$ (union of three disjoint sets).

- $\mathbb{Q}^+$ is countable (Step 1), $\{0\}$ is finite, $\mathbb{Q}^- \approx \mathbb{Q}^+$ via $f(x) = -x$

Step 3: Union of three countable sets is countable, so $\mathbb{Q}$ is countable. □

Surprising fact: There are "as many" rationals as integers, even though rationals are *dense* (between any two, there's another) while integers are *discrete*!

Uncountable Sets: Cantor's Diagonal Argument

Definition 86: A set is *uncountable* if it is infinite but not countable (no bijection with $\mathbb{N}$ exists).

Cantor's strategy: Given any proposed "list" of all elements, construct a *new* element that differs from each item in the list, proving the list is incomplete.

Theorem 19: The set $\mathbb{B}^\omega$ of all infinite binary sequences is uncountable.

Why $\mathbb{B}^\omega$? It's simpler than $\mathbb{R}$ but has the same cardinality: $|\mathbb{B}^\omega| = |\mathbb{R}| = 2^{\aleph_0} = \mathfrak{c}$. Binary sequences can represent real numbers via binary expansions!

Proof: Suppose for contradiction that $\mathbb{B}^\omega$ is countable. Then we can enumerate its elements as $x_1, x_2, x_3, \ldots$, where $x_i = (b_{i1}, b_{i2}, b_{i3}, \ldots)$ is an infinite bit sequence.

Uncountable Sets: Cantor's Diagonal Argument [2]

We can represent this enumeration as an infinite table, where each row corresponds to a (supposedly countable) sequence x_i and each column corresponds to a bit position (natural numbers):

Seq	Bit 1	Bit 2	Bit 3	Bit 4	...
x_1	$\mathbf{b_{11}}$	b_{12}	b_{13}	b_{14}	...
x_2	b_{21}	$\mathbf{b_{22}}$	b_{23}	b_{24}	...
x_3	b_{31}	b_{32}	$\mathbf{b_{33}}$	b_{34}	...
x_4	b_{41}	b_{42}	b_{43}	$\mathbf{b_{44}}$	...
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$
$\boldsymbol{\Delta}$	$\boldsymbol{\overline{b}_{11}}$	$\boldsymbol{\overline{b}_{22}}$	$\boldsymbol{\overline{b}_{33}}$	$\boldsymbol{\overline{b}_{44}}$	...

Construct $\Delta = \left(\overline{b}_{11}, \overline{b}_{22}, \overline{b}_{33}, \ldots\right)$ by flipping each diagonal bit:

$$\overline{b}_{ii} = \begin{cases} 1 \text{ if } b_{ii} = 0 \\ 0 \text{ if } b_{ii} = 1 \end{cases}$$

Uncountable Sets: Cantor's Diagonal Argument [3]

Key observation: $\Delta \neq x_i$ for any i, because they differ at position i: $\overline{b}_{ii} \neq b_{ii}$.

But $\Delta \in \mathbb{B}^\omega$, since it is an infinite binary sequence, so it *should* appear in our enumeration. Contradiction!

Therefore, $\mathbb{B}^\omega$ is uncountable. □

Connection to zig-zag:

- **Zig-zag** (for $\mathbb{N}^2$): Each diagonal is *finite*, we can list all pairs
- **Diagonal argument** (for $\mathbb{B}^\omega$): Sequences are *infinite*, impossible to list completely!

More Uncountable Sets

Example: **Real numbers** $\mathbb{R}$ are uncountable:

- Map each $x \in (0; 1)$ to its binary expansion sequence in $\mathbb{B}^\omega$
- Since $\mathbb{B}^\omega$ is uncountable and $(0, 1) \approx \mathbb{R}$, we have $|\mathbb{R}| = 2^{\aleph_0} = \mathfrak{c}$

Example: **Power set** $\mathcal{P}(\mathbb{N})$ is uncountable:

- Map each $S \subseteq \mathbb{N}$ to its characteristic sequence $(\chi_S(0), \chi_S(1), \chi_S(2), ...) \in \mathbb{B}^\omega$
- This is a bijection: $\mathcal{P}(\mathbb{N}) \approx \mathbb{B}^\omega$
- Therefore $|\mathcal{P}(\mathbb{N})| = 2^{\aleph_0}$ (matches Cantor's theorem!)

More Uncountable Sets [2]

Example: **Irrational numbers** $\mathbb{I} = \mathbb{R} \setminus \mathbb{Q}$ are uncountable:

- If $\mathbb{I}$ were countable, then $\mathbb{R} = \mathbb{Q} \cup \mathbb{I}$ would be a union of two countable sets, hence countable
- But $\mathbb{R}$ is uncountable, contradiction!
- Most real numbers are irrational (in a measure-theoretic sense)

Example (Computer science connections):

- **Undecidable problems:** There are uncountably many *languages* over any alphabet, but only countably many *algorithms* (finite strings). Most languages have no decision algorithm!
- **Real computation:** Real numbers are uncountable, but computable reals are countable
- **Cryptography:** Security often relies on the vastness of uncountable key spaces

Comparing Cardinalities

Definition 87: The cardinality of A is *less than or equal to* that of B, denoted $|A| \leq |B|$ or $A \preceq B$, if there exists an *injection* $f : A \to B$.

Definition 88: The cardinality of A is *strictly less than* that of B, denoted $|A| < |B|$ or $A \prec B$, if $A \preceq B$ and $A \not\approx B$ (injection exists, but no bijection).

Key insight: Injections let us compare sizes without needing full bijections!

Example:

- $|\{1, 2\}| < |\{a, b, c\}|$ because $f(1) = a, f(2) = b$ is injective, but no bijection exists
- $|\mathbb{N}| \leq |\mathbb{Z}|$ and $|\mathbb{Z}| \leq |\mathbb{N}|$ (both have bijections, so $|\mathbb{N}| = |\mathbb{Z}|$)
- $|\mathbb{N}| < |\mathcal{P}(\mathbb{N})|$ because $f(n) = \{n\}$ is injective, but Cantor's theorem says no bijection exists
- $|\mathbb{N}| < |\mathbb{R}|$ because $f : \mathbb{N} \to \mathbb{R}$ via $f(n) = n$ is injective, but diagonal argument shows no bijection

Cantor's Theorem

Theorem 20 (Cantor): $A \prec \mathcal{P}(A)$ for any set A.

Proof: The map $f(x) = \{x\}$ is an injection from A to $\mathcal{P}(A)$, since if $x \neq y$, then $\{x\} \neq \{y\}$, and so $f(x) \neq f(y)$. So we have shown that $A \preceq \mathcal{P}(A)$.

To show that $A \not\approx \mathcal{P}(A)$, suppose for reductio that there is a bijection $g : A \to \mathcal{P}(A)$.

- Consider $D = \{x \in A \mid x \notin g(x)\}$. Note that $D \subseteq A$, so $D \in \mathcal{P}(A)$.
- Since g is surjective, there exists $y \in A$ such that $g(y) = D$.
  - If $y \in D$, then by definition of D, we have $y \notin g(y)$, *i.e.*, $y \notin D$. Contradiction!
  - If $y \notin D$, then $y \notin g(y)$, so by definition of D, we have $y \in D$. Contradiction!
- Therefore, no bijection $A \to \mathcal{P}(A)$ can exist, so $A \not\approx \mathcal{P}(A)$. □

Profound implication: There is no "largest" infinity! We can always construct a bigger one using the power set operation: $\aleph_0 < 2^{\aleph_0} < 2^{2^{\aleph_0}} < \ldots$

Schröder–Bernstein Theorem

Theorem 21 (Schröder–Bernstein): If $A \preceq B$ and $B \preceq A$, then $A \approx B$.

Equivalently: if injections $f : A \to B$ and $g : B \to A$ both exist, then a bijection $h : A \to B$ exists.

What this means: If each set "fits inside" the other, they have the same size!
This powerful result lets us prove equality of cardinalities without constructing explicit bijections.

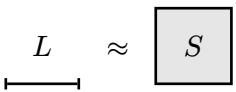
Example: $(0; 1) \approx [0; 1]$ (open interval equals closed interval):

- Injection $f : (0; 1) \to [0; 1]$ by $f(x) = x$ (identity)
- Injection $g : [0; 1] \to (0; 1)$ by $g(x) = \frac{x}{2} + \frac{1}{4}$ (shrink and shift to $\left(\frac{1}{4}; \frac{3}{4}\right)$)
- By Schröder–Bernstein, there exists a bijection between them!

Note: The proof is non-trivial and constructive. We'll see it in a dedicated section later.

Unit Line vs. Unit Square

Theorem 22: The unit line $L = [0;1]$ and unit square $S = [0;1]^2$ are equinumerous.



Surprise! A 1D object has the same cardinality as a 2D object!

Proof[4]: **Step 1:** Injection $f : L \to S$ by $f(x) = \langle x, x \rangle$ gives $L \preceq S$.

- If $f(a) = f(b)$, then $\langle a, a \rangle = \langle b, b \rangle$, so $a = b$

Step 2: Injection $g : S \to L$ by *interleaving* decimal digits gives $S \preceq L$:

$$\left.\begin{aligned} x &= 0.x_1x_2x_3\ldots \\ y &= 0.y_1y_2y_3\ldots \end{aligned}\right\} \quad g(x,y) = 0.x_1y_1x_2y_2x_3y_3\ldots$$

- If $g(a,b) = g(c,d)$, then all digits match, so $\langle a, b \rangle = \langle c, d \rangle$

Step 3: By Schröder–Bernstein (Theorem 21), we conclude $L \approx S$. □

[4] See https://math.stackexchange.com/a/183383 for more detailed analysis.

Summary: Cardinality & Infinity

Cardinality measures set "size," with surprising distinctions among infinities:

- *Finite sets:* Cardinality = element count
- *Countable infinity* ($\aleph_0$): Sets like $\mathbb{N}, \mathbb{Z}, \mathbb{Q}$ that can be listed (bijection with $\mathbb{N}$)
- *Uncountable infinity*: Sets like $\mathbb{R}$ and $\mathcal{P}(\mathbb{N})$ too large to enumerate (*e.g.*, $|\mathbb{R}| = 2^{\aleph_0} = \mathfrak{c}$)

Key techniques:

- *Equinumerosity:* Sets have equal cardinality iff a bijection exists between them
- *Injections:* Prove $|A| \leq |B|$ by constructing a one-to-one map
- *Diagonal argument:* Prove uncountability by showing no enumeration can be complete
- *Schröder–Bernstein:* Two injections (both ways) yield a bijection

Summary: Cardinality & Infinity [2]

Mind-bending facts:

- $|\mathbb{N}| = |\mathbb{Z}| = |\mathbb{Q}| = \aleph_0$ (integers and rationals are "same size" as naturals!)
- $|\mathbb{N} \times \mathbb{N}| = |\mathbb{N}|$ (2D grid has same size as 1D line)
- $[0; 1] \approx [0; 1]^2$ (unit line equals unit square!)
- $|\mathbb{N}| < |\mathcal{P}(\mathbb{N})| < |\mathcal{P}(\mathcal{P}(\mathbb{N}))| < \ldots$ (infinitely many infinities!)
- Most real numbers are *not* algebraic (computable reals are countable, but all reals are uncountable)

Enumerations and Countability

Enumerable Sets

Definition 89: A set X is *enumerable* if there exists a surjection $e : \mathbb{N} \to X$.

The function e is called an *enumeration* of X.

Intuition: An enumerable set can be "listed" as $e(0), e(1), e(2), \ldots$ (possibly with repetitions).

Note: Elements may appear multiple times since we only require a *surjection*, not a bijection.

Example: **Even numbers:** $e(n) = 2n$ gives $0, 2, 4, 6, 8, \ldots$

Example: **Perfect squares:** $e(n) = n^2$ gives $0, 1, 4, 9, 16, 25, \ldots$

Example: **Prime numbers:** $e(0) = 2, e(1) = 3, e(2) = 5, e(3) = 7, \ldots$ (via Sieve of Eratosthenes)

Three Equivalent Characterizations

Theorem 23: For any set X, the following are equivalent:

1. X is *countable*: X is finite or has a bijection with $\mathbb{N}$
2. X is *enumerable*: there exists a surjection $e : \mathbb{N} \to X$
3. X is *embeddable in* $\mathbb{N}$: $X = \emptyset$ or there exists an injection $f : X \to \mathbb{N}$

Practical guide:

- Use *bijection* when you can construct an explicit 1-1 correspondence
- Use *surjection* (enumeration) when you can algorithmically list elements
- Use *injection* when X embeds naturally into $\mathbb{N}$ (often easiest!)

Proof of Equivalence

Proof: We prove $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (1)$.

$(1) \Rightarrow (2)$: Suppose X is countable.

- If X is finite: $X = \{x_0, ..., x_{n-1}\}$, define $e(k) = \begin{cases} x_k \text{ if } k<n \\ x_0 \text{ if } k \geq n \end{cases}$ (surjection)
- If $X \approx \mathbb{N}$: any bijection $g : \mathbb{N} \to X$ is also a surjection

$(2) \Rightarrow (3)$: Suppose $e : \mathbb{N} \to X$ is a surjection.

- If $X = \emptyset$, done
- Otherwise, for each $x \in X$, define $f(x) = \min\{n \in \mathbb{N} \mid e(n) = x\}$ (first occurrence)
- This is injective: if $f(x) = f(y) = n$, then $e(n) = x$ and $e(n) = y$, so $x = y$

$(3) \Rightarrow (1)$: Suppose $X = \emptyset$ or $f : X \to \mathbb{N}$ is injective.

- If $X = \emptyset$, then X is finite (countable)
- Otherwise, $X \approx f(X) \subseteq \mathbb{N}$
- Since any subset of $\mathbb{N}$ is countable, X is countable

□

Infinite Subsets of $\mathbb{N}$

Theorem 24: Any infinite subset $A \subseteq \mathbb{N}$ is equinumerous to $\mathbb{N}$.

Proof: Construct a bijection $f : \mathbb{N} \to A$ recursively:

- $f(0) = \min A$ ($\mathbb{N}$ is well-ordered $\Rightarrow$ any $A \subseteq \mathbb{N}$ has a least element)
- $f(n+1) = \min(A \setminus \{f(0), f(1), ..., f(n)\})$

Since A is infinite, $A \setminus \{f(0), ..., f(n)\}$ is always non-empty and has a minimum.

Injective: By construction, all $f(i)$ are distinct.

Surjective: Every $a \in A$ eventually becomes the minimum of a remaining set.

Therefore f is a bijection, so $|A| = \aleph_0$. □

Key insight: There's only "one size" of countably infinite set: $\aleph_0$.

Enumeration Examples

Example (Finite strings over $\Sigma = \{a, b\}$): Enumerate Σ^* by length, then lexicographically:

$$\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \ldots$$

- Length 0: ε (1 string)
- Length 1: a, b (2 strings)
- Length 2: aa, ab, ba, bb (4 strings)
- Length n: 2^n strings

Since $\bigcup_{n=0}^{\infty} 2^n$ is a countable union of finite sets, Σ^* is countable.

Example (Rational numbers): List positive fractions $\frac{p}{q}$ by increasing $p + q$, skip non-reduced:

$$\frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \frac{3}{1}, \frac{1}{4}, \frac{2}{3}, \frac{3}{2}, \frac{4}{1}, \cdots$$

Include 0 and negatives by interleaving:

$$0, \frac{1}{1}, -\frac{1}{1}, \frac{1}{2}, -\frac{1}{2}, \frac{2}{1}, -\frac{2}{1}, \cdots$$

Schröder–Bernstein

Proof of Schröder–Bernstein Theorem

Theorem 25: If injections $f : A \to B$ and $g : B \to A$ exist, then a bijection $h : A \to B$ exists.

Proof: Let $f : A \to B$ and $g : B \to A$ be injections.

We use g^{-1} to denote the inverse of g restricted to its image: for any $a \in g(B)$, we write $g^{-1}(a)$ for the unique $b \in B$ satisfying $g(b) = a$.

Step 1: Construct auxiliary sets

Define inductively the sets $A_n \subseteq A$ and $B_n \subseteq B$ for $n \geq 0$ by:

$$\begin{aligned} A_0 &:= A \setminus g(B) \\ B_n &:= f(A_n) \quad \text{for } n \geq 0 \\ A_{n+1} &:= g(B_n) \quad \text{for } n \geq 0 \end{aligned}$$

Define $A_\infty \subseteq A$ as the union of all A_n: $A_\infty := \bigcup_{n \geq 0} A_n$.

Proof of Schröder–Bernstein Theorem [2]

Step 2: Define the candidate bijection

Define $h : A \to B$ by:

$$h(a) = \begin{cases} f(a) & \text{if } a \in A_\infty \\ g^{-1}(a) & \text{if } a \notin A_\infty \end{cases}$$

Step 3: Verify h is well-defined

For all $a \in A$, we need $h(a)$ to be defined:

- $\boldsymbol{a \in A_\infty}$**:** Then $h(a) = f(a)$ is well-defined (since $f : A \to B$).
- $\boldsymbol{a \notin A_\infty}$**:** Then $a \notin A_0$ (as $A_0 \subseteq A_\infty$), so $a \in g(B)$. Thus $g^{-1}(a)$ exists and is unique.

Proof of Schröder–Bernstein Theorem [3]

Step 4: Prove the ranges are disjoint

We show that $f(A_\infty)$ and $g^{-1}(A \setminus A_\infty)$ are disjoint.

First, observe that:

$$f(A_\infty) = f\left(\bigcup_{n\geq 0} A_n\right) = \bigcup_{n\geq 0} f(A_n) = \bigcup_{n\geq 0} B_n$$

Now take $a \notin A_\infty$ and let $b = g^{-1}(a)$. We claim $b \notin \bigcup_{n\geq 0} B_n$:

- Suppose for contradiction that $b \in B_n$ for some n
- Then $a = g(b) \in g(B_n) = A_{n+1} \subseteq A_\infty$
- This contradicts $a \notin A_\infty$

Hence $g^{-1}(A \setminus A_\infty) \subseteq B \setminus \bigcup_{n\geq 0} B_n = B \setminus f(A_\infty)$.

Therefore, the ranges $f(A_\infty)$ and $g^{-1}(A \setminus A_\infty)$ are disjoint.

Proof of Schröder–Bernstein Theorem [4]

Step 5: Prove h is injective

Take $a, a' \in A$ with $h(a) = h(a')$. Consider the cases:

- **Both in A_∞:** Then $f(a) = f(a')$, so $a = a'$ (since f is injective).
- **Both outside A_∞:** Then $g^{-1}(a) = g^{-1}(a')$, so $a = a'$ (since g is injective).
- **Mixed case** (say $a \in A_\infty$, $a' \notin A_\infty$): Then $h(a) = f(a) \in f(A_\infty)$ but $h(a') = g^{-1}(a') \in B \setminus f(A_\infty)$. By Step 4, these sets are disjoint, contradicting $h(a) = h(a')$.

Therefore h is injective.

Proof of Schröder–Bernstein Theorem [5]

Step 6: Prove h is surjective

Let $b \in B$. Consider the cases:

- **$b \in f(A_\infty)$:** Then $b = f(a)$ for some $a \in A_\infty$, so $h(a) = f(a) = b$.
- **$b \notin f(A_\infty)$:** Let $a = g(b) \in A$. We claim $a \notin A_\infty$:
  - Suppose for contradiction that $a \in A_\infty$
  - Then $a \in A_n$ for some $n \geq 1$ (as $a = g(b) \in g(B)$, but $A_0 = A \setminus g(B)$)
  - So $a \in A_n = g(B_{n-1})$, meaning $a = g(b')$ for some $b' \in B_{n-1}$
  - Since g is injective and $a = g(b) = g(b')$, we have $b = b' \in B_{n-1} \subseteq f(A_\infty)$
  - This contradicts $b \notin f(A_\infty)$

  Therefore $a \notin A_\infty$, and $h(a) = g^{-1}(a) = g^{-1}(g(b)) = b$.

Therefore h is surjective.

Conclusion: Since h is both injective and surjective, h is a bijection, so $A \approx B$. □

Large Cardinal Numbers

"The essence of mathematics is its freedom."

— Georg Cantor

Beyond $\aleph_0$: Hierarchies of Infinity

We've seen that $\aleph_0 = |\mathbb{N}|$ is the smallest infinite cardinality. But are there *larger* infinities beyond $\aleph_0$?

Cantor showed that there are *infinitely many* distinct sizes of infinity, forming an endless hierarchy of ever-growing infinities.

$$|\mathbb{N}| < |\mathbb{R}| < |\mathcal{P}(\mathbb{R})| < |\mathcal{P}(\mathcal{P}(\mathbb{R}))| < \dots$$

This discovery fundamentally changed our understanding of the infinite.

Two key hierarchies help us organize these infinities:

- The **ב (beth) numbers** $\beth_0, \beth_1, \beth_2, \dots$ — *constructive hierarchy* built by repeatedly taking powersets
- The **א (aleph) numbers** $\aleph_0, \aleph_1, \aleph_2, \dots$ — *ordinal hierarchy* indexing all infinite cardinals by their order

Beth Numbers: The Powerset Hierarchy

Definition 90: The *beth numbers* (from Hebrew letter ב) form a *constructive* hierarchy defined recursively using the powerset operation:

- $\beth_0 = \aleph_0 = |\mathbb{N}|$ (countable infinity — the starting point)
- $\beth_{n+1} = 2^{\beth_n} = |\mathcal{P}(\text{set of size } \beth_n)|$ (apply powerset operation)
- $\beth_\lambda = \sup_{\alpha<\lambda} \beth_\alpha$ (for limit ordinal λ)

Key insight: Each beth number is the cardinality of the *powerset* of the previous one.

This gives us a *concrete, algorithmic* hierarchy: we know exactly how to construct each level!

Note: The beth hierarchy is a natural generalization of Cantor's diagonal argument: each step $\beth_n \to \beth_{n+1}$ applies the result that $S \prec \mathcal{P}(S)$ for any set S.

Examples of Beth Numbers

Example: The first few beth numbers:

Beth	Value	Interpretation
$\beth_0$	$\aleph_0 = \lvert\mathbb{N}\rvert$	Countable infinity
$\beth_1$	$2^{\aleph_0} = \lvert\mathcal{P}(\mathbb{N})\rvert = \lvert\mathbb{R}\rvert = \mathfrak{c}$	The continuum (real numbers)
$\beth_2$	$2^{\mathfrak{c}} = \lvert\mathcal{P}(\mathbb{R})\rvert$	All functions $\mathbb{R} \to \mathbb{R}$, all subsets of $\mathbb{R}$
$\beth_3$	$2^{\beth_2} = \lvert\mathcal{P}(\mathcal{P}(\mathbb{R}))\rvert$	All relations on $\mathbb{R}$

By Cantor's theorem, we know $\beth_0 < \beth_1 < \beth_2 < \beth_3 < \ldots$ is a *strictly increasing* sequence.

Each powerset operation produces a provably larger infinity!

Aleph Numbers: Indexing All Infinite Cardinals

Definition 91: The *aleph numbers* (from Hebrew letter ℵ) enumerate **all** infinite cardinal numbers in their *natural order*:

- $\aleph_0 = |\mathbb{N}|$ — the smallest infinite cardinal (countable infinity)
- $\aleph_1$ — the *next* infinite cardinal after $\aleph_0$ (the smallest uncountable cardinal)
- $\aleph_2$ — the next infinite cardinal after $\aleph_1$
- In general: $\aleph_{\alpha+1}$ is the *smallest cardinal strictly larger* than $\aleph_\alpha$

Unlike beth numbers (defined by powerset), aleph numbers are defined *abstractly* by their *order*.

We don't know how to "construct" $\aleph_1$ from $\aleph_0$ — we only know it is the "next" cardinal!

Notes on Aleph Hierarchy

Note: Each aleph is actually an *initial ordinal*: the smallest ordinal of that cardinality.

- For example, $\aleph_0 = \omega$, the first infinite ordinal.
- $\aleph_1 = \omega_1$, the first uncountable ordinal.
- "Intermediate" ordinals like $\omega + 1, \omega \times 2, \omega^2, \omega^\omega, \ldots$ all have cardinality $\aleph_0$.

This connects cardinality to ordinal number theory.

Note: The alephs are "ordinal-indexed": $\aleph_0, \aleph_1, \ldots, \aleph_\omega, \aleph_{\omega+1}, \ldots$ It extends through all ordinal numbers! For a limit ordinal λ:

$$\aleph_\lambda = \bigcup_{\alpha<\lambda} \aleph_\alpha = \sup_{\alpha<\lambda} \aleph_\alpha$$

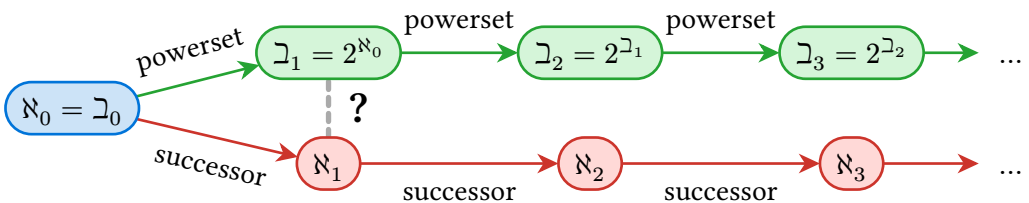
Aleph vs Beth: Two Fundamentally Different Hierarchies

Beth hierarchy (constructive):

- Start: $\beth_0 = \aleph_0 = |\mathbb{N}|$
- Rule: $\beth_{n+1} = 2^{\beth_n}$ (powerset)
- *Algorithmic:* built by iteration
- We know *exactly* what each is

Aleph hierarchy (ordinal):

- Start: $\aleph_0 = \beth_0 = |\mathbb{N}|$
- Rule: $\aleph_{n+1}$ = "next cardinal"
- *Axiomatic:* built by well-ordering
- We know the *order*, not values



How these two *diverging* hierarchies relate is one of deepest "unsolved" questions in mathematics.

Central question: Is $\aleph_1 = \beth_1 = \mathfrak{c}$? This is the *Continuum Hypothesis*.

The Continuum Hypothesis (CH): Cantor's Great Question

Definition 92: The *Continuum Hypothesis* (CH) states that there is no infinite cardinal strictly between $\aleph_0$ and the cardinality of the continuum:

$$\aleph_1 = \beth_1 = 2^{\aleph_0} = |\mathbb{R}| = \mathfrak{c}$$

In other words: *every* infinite subset of $\mathbb{R}$ is either countable ($\aleph_0$) or has the same size as $\mathbb{R}$ ($\mathfrak{c}$).

Georg Cantor formulated CH in 1878 and believed it to be true. He spent decades trying to prove it.

"The question whether there exists a transfinite number between $\aleph_0$ and $2^{\aleph_0}$ has tormented me." [5]

Despite geniously discovering the transfinite numbers, Cantor could neither prove nor disprove CH. This struggle contributed to his mental health difficulties in later life.

[5] Not a real quote, but captures his sentiment!

What CH Implies?

Examples (if CH is true):

- The real numbers $\mathbb{R}$ have cardinality $\aleph_1$ (first uncountable cardinal)
- Every uncountable subset of $\mathbb{R}$ is equinumerous to $\mathbb{R}$
- There are "no infinities between" countable and continuum
- The hierarchies align at the first step: $\aleph_1 = \beth_1$
- All "naturally occurring" uncountable sets in analysis have the same size

Examples (if CH is false):

- The real numbers $\mathbb{R}$ have cardinality strictly larger than $\aleph_1$ (*e.g.*, $\mathfrak{c} = \aleph_2$ or higher)
- There exist uncountable subsets of $\mathbb{R}$ with cardinalities between $\aleph_0$ and $|\mathbb{R}|$
- The hierarchies diverge: $\aleph_1 < \beth_1$
- A richer structure of infinite cardinalities exists
- Some "naturally occurring" sets in analysis may have different sizes

Are there hidden infinities between countable and continuum, or just these two?

The Generalized Continuum Hypothesis (GCH)

Definition 93: The *Generalized Continuum Hypothesis* (GCH) extends CH to all infinite cardinals:

For every infinite cardinal κ:

$$2^{\kappa} = \kappa^{+}$$

where κ^{+} denotes the *immediate successor cardinal* after κ.

This means: $\aleph_n = \beth_n$ for all ordinals n (finite and transfinite).

What GCH claims: The two hierarchies *completely coincide* at every level!

There is only *one* natural hierarchy of infinite cardinals, and the powerset operation always produces the very next cardinal in the sequence.

What GCH Implies?

Examples (if GCH is true):

- Every aleph is a beth: $\aleph_n = \beth_n$ for all ordinals n
- Powerset always gives the next cardinal: $2^{\aleph_\alpha} = \aleph_{\alpha+1}$
- No "gaps" in the cardinal hierarchy — maximally simple structure
- All questions about cardinal arithmetic have definite answers
- The universe of sets is "neat" and predictable

Examples (if GCH is false):

- The hierarchies diverge at some level: $\aleph_\alpha < \beth_\alpha$ for some α
- Powerset can "jump" multiple levels: $2^{\aleph_\alpha} > \aleph_{\alpha+1}$
- Cardinal arithmetic has complex, unpredictable behavior
- The universe of sets is "wild" with hidden structure
- Different models can have vastly different cardinal behaviors

Is the cardinal hierarchy maximally simple or is reality more complex?..

Hilbert's First Problem

In 1900, David Hilbert presented **23 problems** that shaped 20th-century mathematics.

Problem 1 was to resolve the Continuum Hypothesis — showing its fundamental importance.

Early attempts to settle CH:

- **Cantor** (1878–1918): Believed CH was true, could not prove it
- **Hilbert** (1900): Made it the first of his famous problems
- **Zermelo** (1908): Developed axiomatic set theory (ZF), but CH remained open
- **Fraenkel** (1922): Extended to ZFC (with Axiom of Choice), CH still unresolved
- **Gödel** (1940): Proved CH is *consistent* with ZFC (cannot be disproved)
- **Cohen** (1963): Proved ¬CH is also consistent — **breakthrough!**

Cohen's Independence Result (1963)

Theorem 26 (Cohen, 1963): The Continuum Hypothesis is *independent* of ZFC (Zermelo–Fraenkel set theory with the Axiom of Choice):

- ZFC **cannot prove** CH
- ZFC **cannot disprove** CH
- Both "ZFC + CH" and "ZFC + ¬CH" are consistent (assuming ZFC is consistent)

What Cohen proved using *forcing*: The value of $2^{\aleph_0}$ is *not determined* by the axioms of ZFC!

We can construct models where:

- $\aleph_1 = \beth_1 = 2^{\aleph_0}$ (CH holds)
- $\aleph_2 = \beth_1 = 2^{\aleph_0}$ (one intermediate cardinal)
- $\aleph_\omega = \beth_1 = 2^{\aleph_0}$ (countably many intermediate cardinals)
- $\aleph_{\omega_1} = \beth_1 = 2^{\aleph_0}$ (uncountably many intermediate cardinals)
- Even: $\aleph_\alpha = \beth_1 = 2^{\aleph_0}$ for arbitrarily large α!

The Freedom of Infinity

Different models of ZFC can have wildly different cardinal structures!

Universe where CH holds:

- $\aleph_1 = \beth_1 = 2^{\aleph_0} = |\mathbb{R}|$ — hierarchies align
- Every subset of $\mathbb{R}$ is either countable ($\aleph_0$) or has the same cardinality $\mathfrak{c}$
- No "intermediate" infinities

Universe where CH fails:

- $\aleph_1 < 2^{\aleph_0} = |\mathbb{R}|$ — hierarchies diverge
- "Intermediate" infinities exist (between $\mathbb{N}$ and $\mathbb{R}$)
- $\mathbb{R}$ can be arbitrarily large
- Much richer structure

Before 1963: We thought CH must be either true or false.

After Cohen: CH is *independent* — both answers are mathematically valid!

Key insight: There is no single fixed mathematical reality.

Different axiom systems can give different but equally consistent answers.

What Is Mathematical Truth?

If CH can be neither proved nor disproved, what does it even *mean* to ask "Is CH true?"

Two philosophical perspectives:

Platonist view:

- Mathematical objects exist independently
- CH *is* either true or false in reality
- We just haven't found the "right" axioms yet
- *Approach:* Search for natural axioms beyond ZFC (like large cardinal axioms)

Formalist view:

- Mathematics is just symbol manipulation
- "Truth" depends on which axioms you choose
- CH is true in some models, false in others
- *Approach:* Study all possible models ("multiverse") and their properties

Modern Perspectives on CH

Most mathematicians today take a pragmatic approach:

Instead of asking "Is CH absolutely true?", modern set theory asks more productive questions:

- In which models of ZFC does CH hold?
- What interesting mathematics follows from CH? From ¬CH?
- Do certain "natural" axioms beyond ZFC settle CH?
- Which axiom systems are most useful for specific areas of mathematics?

Note: This shift mirrors similar debates in physics: rather than asking if *string theory* is "The True Theory," physicists ask what predictions it makes and whether it's useful for understanding nature.

Summary: Large Cardinal Numbers

Two infinite hierarchies with different constructions:

- **Beth numbers:** $\beth_0 = \aleph_0$, $\beth_{n+1} = 2^{\beth_n}$ — built step-by-step using powersets
- **Aleph numbers:** $\aleph_0, \aleph_1, \aleph_2, \ldots$ — enumerate all infinite cardinals in order

Both start at $\aleph_0 = \beth_0$, but their relationship afterward is *not fixed by ZFC axioms*.

The Continuum Hypothesis: Does $\aleph_1 = 2^{\aleph_0}$?

- **Gödel (1940):** Showed ZFC + CH is consistent
- **Cohen (1963):** Showed ZFC + ¬CH is also consistent

Conclusion: CH is *independent* of ZFC. The "size" of $\mathbb{R}$ depends on which axioms you choose.

What we learned: Mathematics does not have a unique "reality" — different axiom systems can give different answers to the same question, yet remain equally consistent.

Outline

TODO

- Applications of lattices in:
  - Formal concept analysis
  - Domain theory in computer science
  - Algebraic topology
  - Cryptography (lattice-based cryptography)
- Advanced topics in set theory:
  - Cardinal arithmetic
  - Ordinal numbers
  - Forcing and independence results
  - Large cardinals
- Connections to Boolean algebra (next lecture)
- Applications in formal logic and proof theory

Looking Ahead: Boolean Algebra

The next lecture will explore *Boolean algebra*, which provides the mathematical foundation for:

- Digital circuit design and computer hardware
- Propositional logic and automated reasoning
- Database query optimization
- Formal verification of software and hardware systems

Key topics will include:

- Boolean functions and their representations
- Normal forms (CNF, DNF)
- Minimization techniques (Karnaugh maps, Quine-McCluskey)
- Functional completeness and Post's theorem
- The satisfiability problem (SAT) and its computational complexity

Preview: Formal Logic

Following Boolean algebra, we will study *formal logic*, covering:

- Propositional and predicate logic
- Natural deduction and proof systems
- Completeness and soundness theorems
- Applications to program verification and AI reasoning

This progression from sets → relations → functions → Boolean algebra → logic provides a solid foundation for advanced topics in discrete mathematics and computer science.

Binary relations are the bridge between sets and functions — they model how objects *connect*, *organize*, and *interact* in mathematical structures and real-world systems.