

Ve203 Discrete Mathematics

Runze Cai

University of Michigan - Shanghai Jiao Tong University
Joint Institute

Spring 2023



JOINT INSTITUTE
交大密西根学院

Part I

Basic Set Theory and Applications

Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Sets

Definition

A set is an unordered collection of distinct objects, called *elements* or *members* of the set. A set is said to contain its elements. We write

- ▶ $a \in A$ if a is an element of the set A .
- ▶ $a \notin A$ if a is not an element of the set A .

Examples

- ▶ The set P of primes less than 10: $P = \{2, 3, 5, 7\}$.
- ▶ The set V of all vowels in the English alphabet: $V = \{a, e, i, o, u\}$.
- ▶ The set S of all suits $S = \{\heartsuit, \diamondsuit, \clubsuit, \spadesuit\}$.
- ▶ Different kinds of objects: $S = \{\text{USA}, \text{USB}, \text{UCSB}, 0.0, \{\text{CAT}\}\}$.
- ▶ The empty set $S = \{\} = \emptyset = \emptyset$.
- ▶ n -set, $[n] = \{1, 2, \dots, n\}$, where $n \in \mathbb{N} \setminus \{0\}$.
- ▶ $B^A = \{f \in \mathcal{P}(A \times B) \mid f : A \rightarrow B\}$.
- ▶ $SL_2(\mathbb{R}) = \left\{ \begin{pmatrix} a & b \\ c & d \end{pmatrix} \mid a, b, c, d \in \mathbb{R}, ad - bc = 1 \right\}$

Multisets

Caution

- ▶ Elements in a set are *distinct* and *unordered*.

Example

- ▶ $\{1, 0, 0, 0\} = \{0, 0, 1, 1\} = \{0, 1\} = \{1, 0\}$
- ▶ $\{\pm 1\} = \{(-1)^n \mid n \in \mathbb{N}\} = \{1, -1, 1, -1, \dots\}$

A **multiset** does allow repeated objects. (but order still does not matter)

Example

- ▶ Roots of a polynomial.
- ▶ Eigenvalues of a square matrix.
- ▶ Stock of drinks.

Set Notation

Number Systems

- ▶ \mathbb{N} , the natural numbers
- ▶ \mathbb{Z} , the integers
- ▶ \mathbb{Q} , the rational numbers
- ▶ \mathbb{R} , the real numbers
- ▶ \mathbb{C} , the complex numbers

Cardinality

The **size** of a set A is called its **cardinality**, denoted by $|A|$, $\#A$, or $\text{card } A$.

- ▶ $|A| = n \in \mathbb{N}$ if A is a finite set;
- ▶ $|A| = \infty$ otherwise. (Question: infinities?)

Set Operations

Let A, B be sets.

Inclusion

- ▶ A is a subset of B , denoted by $A \subset B$, if every element of A is an element of B .
- ▶ B is called a superset of A , denoted by $B \supset A$.

Remark

Unlike $<$ and \leq , $A \subset B$ is the same as $A \subseteq B$. Similarly for \supset and \supseteq .

Proper Subset/Superset

A is a proper subset of B if $A \subset B$ and $A \neq B$, denoted by $A \subsetneq B$ or $A \subsetneqq B$. Similarly for proper superset.

Remark

$A = B$ if and only if $A \subset B$ and $B \subset A$. (cf., $x = y$ iff $x \leq y$ and $y \leq x$.)

Set Operations

Let A, B be sets.

Union

The **union** of A and B is the set of elements in either A or B , denoted by

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

We will also denote the union of a collection of sets \mathcal{C} as

$$\bigcup \mathcal{C} = \{x \mid \exists X \in \mathcal{C} \text{ s.t. } x \in X\} = \bigcup \{X \mid X \in \mathcal{C}\} = \bigcup_{X \in \mathcal{C}} X$$

Example

- ▶ $\bigcup \{A, B\} = A \cup B$, and $\bigcup \{A\} = A$.
- ▶ $\bigcup \emptyset = \emptyset$, or in terms of a family of sets $\{S_i\}_{i \in I}$ indexed by $I = \emptyset$, we have

$$\bigcup_{i \in \emptyset} S_i = \{x \mid \exists i \in \emptyset \text{ s.t. } x \in S_i\} = \emptyset$$

Set Operations

Let A, B be sets.

Intersection

The **intersection** of A and B is the set of elements in both A and B , denoted by

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

We will also denote the intersection of a **nonempty** collection of sets \mathcal{C} as

$$\bigcap \mathcal{C} = \{x \mid \forall X \in \mathcal{C} \text{ s.t. } x \in X\} = \bigcap \{X \mid X \in \mathcal{C}\} = \bigcap_{X \in \mathcal{C}} X$$

Example

- ▶ $\bigcap \{A, B\} = A \cap B$ and $\bigcap \{A\} = A$.
- ▶ $\bigcap \emptyset = \bigcap \{\}$ is undefined.

Set Operations

Let A , B be sets.

Set Difference

The **set difference** of A and B , denoted by $A - B$, or $A \setminus B$, is the set of elements in A but not in B , that is,

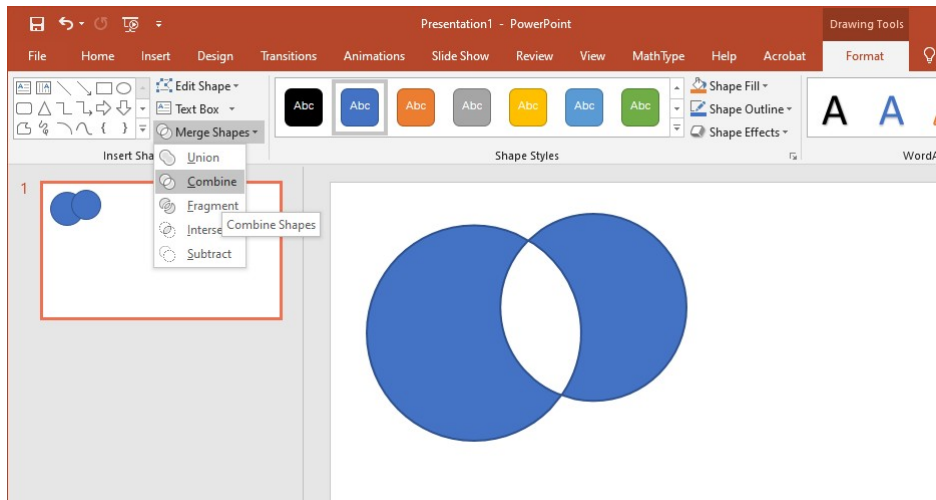
$$\begin{aligned} A - B &:= \{x \mid x \in A \text{ and } x \notin B\} \\ &= \{x \in A \mid x \notin B\} \end{aligned}$$

Symmetric Difference

The **symmetric difference** of A and B is the set of elements that are in **exclusively** one of A and B , but not the other.

$$A \triangle B = (A - B) \cup (B - A)$$

Set Operations



Set Operations

Power Set

The power set of a set A is the set of all subsets of A , denoted by $\mathcal{P}(A)$ or 2^A .

Example

- ▶ $\mathcal{P}(\{j, i\}) = \{\emptyset, \{j\}, \{i\}, \{j, i\}\}.$
- ▶ $\mathcal{P}(\{0\}) = \{\emptyset, \{0\}\}.$
- ▶ $\mathcal{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}.$
- ▶ $\mathcal{P}(\emptyset) = \{\emptyset\}.$

Cardinality of Power sets

Given a finite set A ,

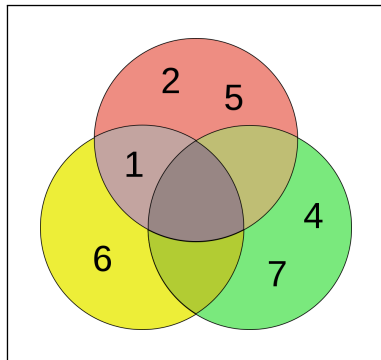
$$|2^A| = |\mathcal{P}(A)| = 2^{|A|}$$

Venn Diagram vs Euler Diagram

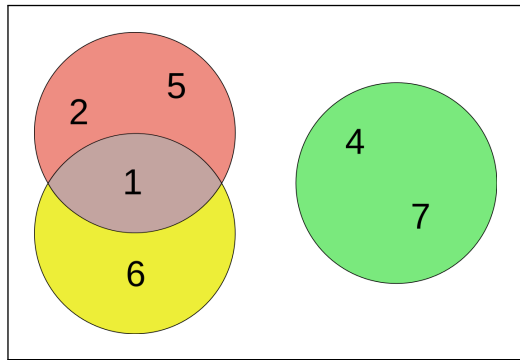
Given

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

Venn Diagram



Euler Diagram



Venn Diagram vs Euler Diagram



<https://xkcd.com/2721/>

Set Algebras

Let A , B , C be sets.

- ▶ Commutative Laws

- ▶ $A \cup B = B \cup A$

- ▶ $A \cap B = B \cap A$

- ▶ Associative Laws

- ▶ $(A \cup B) \cup C = A \cup (B \cup C)$

- ▶ $(A \cap B) \cap C = A \cap (B \cap C)$

- ▶ (Left) Distributive Laws

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

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

- ▶ De Morgan's Laws

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

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

Cartesian Product

Definition

The Cartesian product of sets A and B is the set of **ordered pairs**, such that

$$A \times B = \{(a, b) \mid a \in A, b \in B\}$$

Definition (Kuratowski)

An ordered pair (a, b) is given by

$$(a, b) := \{\{a\}, \{a, b\}\}$$

Theorem

If $a \in C$ and $b \in C$, then $(a, b) \in \mathcal{P}(\mathcal{P}(C))$.

Proof.

- ▶ $a \in C \Rightarrow \{a\} \in \mathcal{P}(C)$; $a, b \in C \Rightarrow \{a, b\} \in \mathcal{P}(C)$.
- ▶ Hence $(a, b) = \{\{a\}, \{a, b\}\} \in \mathcal{P}(\mathcal{P}(C))$. □

Cartesian Product

Theorem

$(x, y) = (a, b)$ iff $x = a$ and $y = b$.

Proof.

The \Leftarrow direction is trivial. For the other direction, let $(x, y) = (a, b)$, that is, $\{\{x\}, \{x, y\}\} = \{\{a\}, \{a, b\}\}$. So either $\{x\} = \{a\}$ or $\{x\} = \{a, b\}$.

- ▶ If $\{x\} = \{a\}$, then $x = a$.
- ▶ If $\{x\} = \{a, b\}$, then $a \in \{x\}$, so $x = a$.

So in either case, we have $x = a$. Now $\{\{a\}, \{a, y\}\} = \{\{a\}, \{a, b\}\}$, so $\{a, y\} \in \{\{a\}, \{a, b\}\}$.

- ▶ If $\{a, y\} = \{a, b\}$, then $y = a$ or $y = b$. If $y = b$, we are done. If $y = a$, then $b \in \{a, y\} = \{a\}$. So $b = a$ and also $y = b$.
- ▶ If $\{a, y\} = \{a\}$, then $y = a$. So $\{a, b\} = \{a\}$, thus $b = a$ and $y = b$ as well. □

Cartesian Product of Sets

In this manner, we can define Cartesian product of three sets as the set of *ordered triples*, e.g.,

$$A \times B \times C := \{(a, b, c) \mid a \in A, b \in B, c \in C\}.$$

where $(a, b, c) := ((a, b), c)$. More generally, given sets A_1, \dots, A_n , $n \in \mathbb{N}$, we can define the n -fold Cartesian product

$$A_1 \times \dots \times A_n := \{(a_1, \dots, a_n) \mid a_k \in A_k, k = 1, \dots, n\}$$

where the ordered *n -tuple* (a_1, \dots, a_n) is given by

$$(a_1, a_2, \dots, a_{n-1}, a_n) := ((a_1, a_2, \dots, a_{n-1}), a_n)$$

If we take the cartesian product of a set with itself, we may abbreviate it using exponents, e.g.,

$$A^2 := A \times A, \quad A^3 := A \times A \times A, \dots$$

$$\mathbb{N}^2 := \mathbb{N} \times \mathbb{N}, \quad \mathbb{N}^3 := \mathbb{N} \times \mathbb{N} \times \mathbb{N}, \dots$$

Associative Set Operations

Let A_1, A_2, \dots, A_n be sets, then

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

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

$$\blacktriangleright A_1 \times A_2 \times \dots \times A_n = \prod_{i=1}^n A_i$$

$$\blacktriangleright A_1 \triangle A_2 \triangle \dots \triangle A_n = \bigtriangleup_{i=1}^n A_i$$

Remark

Brackets are permitted everywhere but not required anywhere.

Question

How many ways to put the brackets?

Simple Graphs

k -element subsets

Let X be a finite set. For a positive integer k , let $\binom{X}{k}$ denote the set of all k -element subsets. Note that $|\binom{X}{k}| = \binom{|X|}{k}$.

Definition

A finite simple **graph** G is a pair (V, E) where V is a non-empty finite set and E is a set of 2-element subsets of V , i.e., $E \subset \binom{V}{2}$. Elements of V are called **vertices** and elements of E are called **edges**. We also call V the **vertex set** of G , denoted $V(G)$, and E the **edge set** of G , denoted $E(G)$.

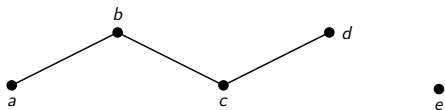
Example

Consider the following simple graph $G = (V, E)$, where

- ▶ $V(G) = \{a, b, c, d, e\}$,
- ▶ $E(G) = \{\{a, b\}, \{b, c\}, \{c, d\}\}$.

For simplicity, we also write

$$E(G) = \{ab, bc, cd\}$$



Directed Graph/Digraph

Definition

A **directed graph** (or **digraph**) is a quadruple $G = (V, E, s, t)$, where V is a set of nodes or vertices, E is a set of arcs or edges, and $s, t : E \rightarrow V$ are two functions, s being the **source functions** and t the **target function**. Given an edge $e \in E$, we call $s(e)$ the **origin** or **source** of e , and t the **endpoint** or **target** of e .

Directed Graph/Digraph

Example

We can define a digraph G such that

- ▶ $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$, $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9\}$, and
- ▶ the functions s and t are specified by the following table

$e \in E$	$s(e)$	$t(e)$
e_1	v_1	v_2
e_2	v_2	v_3
e_3	v_3	v_4
e_4	v_4	v_2
e_5	v_2	v_5
e_6	v_5	v_5
e_7	v_5	v_6
e_8	v_5	v_6
e_9	v_6	v_4

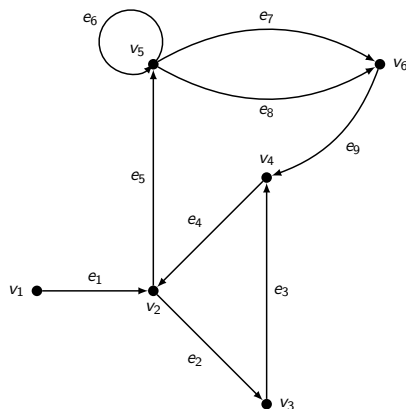


Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Propositional Logic

Definition

A **proposition** or **statement** is a declarative sentence that is either **true** or **false**, but not both.

Examples

- ▶ Washington, D.C., is the capital of the United States of America.
- ▶ Toronto is the capital of Canada.
- ▶ $1 + 1 = 2$.
- ▶ $2 + 2 = 3$.

Non-examples

- ▶ What time is it?
- ▶ Read this carefully.
- ▶ $x + 1 = 2$.
- ▶ $x + y = z$.

Notation

Propositional/Logical Variables

Denoted by p, q, r, \dots

Booleans variables: True/False

- ▶ True: $\top, 1, \top, \text{true}$
- ▶ False: $\bot, 0, \perp, \text{false}$

We sometimes denote the booleans by $\mathbb{B} = \{\top, \perp\}$.

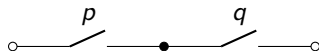
Connectives

- ▶ \neg , negation/not
- ▶ \wedge , and
- ▶ \vee , or (inclusive or)
- ▶ \rightarrow , implies
- ▶ \leftrightarrow , if and only if (iff)

Conjunction and disjunction

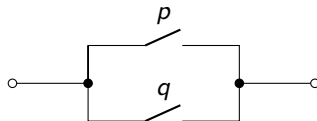
AND

p	q	$p \wedge q$
0	0	0
0	1	0
1	0	0
1	1	1



OR (inclusive or)

p	q	$p \vee q$
0	0	0
0	1	1
1	0	1
1	1	1



Remark

- ▶ In the conjunction $p \wedge q$, the proposition p and q are called **conjuncts**;
- ▶ In the disjunction $p \vee q$, the proposition p and q are called **disjuncts**.

Define Conjunction (&&) in Haskell

`(&&) :: Bool -> Bool -> Bool`

Method 1

```
True  && True   = True
True  && False  = False
False && True    = False
False && False  = False
```

Method 2

```
True && True = True
_    && _    = False
```

Method 3

```
False && _ = False
True  && b = b
```

Method 4

```
b && c | b == c    = b
      | otherwise = False
```

Negation and Exclusive OR

NOT

p	$\neg p$
0	1
1	0

XOR (exclusive or)

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

NOT in Haskell

```
not :: Bool -> Bool
not False = True
not True  = False
```

XOR in Haskell

```
xor :: Eq a => a -> a -> Bool
xor x y = x /= y
-- or use infix notation
-- x `xor` y = x /= y
-- or even simpler
-- xor = (/=)
```

Conditional Statements

Conditional statement (material implication)

- ▶ p :hypothesis/antecedent/premise
- ▶ q :thesis/conclusion/consequence

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Equivalent forms

- ▶ if p , then q
- ▶ if p , q
- ▶ q if p
- ▶ q when p
- ▶ q unless $\neg p$
- ▶ p implies q
- ▶ p only if q
- ▶ q whenever p
- ▶ p is sufficient for q
- ▶ a sufficient condition for q is p
- ▶ a necessary condition for p is q
- ▶ q is necessary for p
- ▶ q follows from p

Conditional Statements

“No underage drinking”

If somebody is drinking alcohol, then they are over 18 y/o.

- ▶ A is drinking beer
- ▶ B is drinking coda
- ▶ C is 55 y/o
- ▶ D is 15 y/o

p	q	$p \rightarrow q$
0	0	1
0	1	1
1	0	0
1	1	1

Remark

We seek to find out whether a certain promise or guarantee is kept. That is,

- ▶ either p is false
- ▶ or q is true

p	q	$\neg p$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

Converse/Inverse/Contrapositive/Negation

Given $p \rightarrow q$,

► Converse: $q \rightarrow p$

► Inverse: $\neg p \rightarrow \neg q$

► Contrapositive: $\neg q \rightarrow \neg p$

► Negation: $\neg(p \rightarrow q)$

p	q	$p \rightarrow q$	$\neg q \rightarrow \neg p$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	1	1

Example: p -value

In null-hypothesis significance testing, the p -value is the probability of obtaining test results at least as extreme as the result actually observed, **under the assumption that the null hypothesis is correct.**

Biconditional Statements

if and only if (iff)

p	q	$p \rightarrow q$	$q \rightarrow p$	$p \leftrightarrow q$
0	0	1	1	1
0	1	1	0	0
1	0	0	1	0
1	1	1	1	1

Compound proposition

Order of operations (use brackets “(...)” when in doubt)

► \neg

► \wedge

► \vee

► \rightarrow

► \leftrightarrow

Tautology and Contradiction

In the truth table,

- ▶ Tautology: All cases evaluates to 1. (e.g., $p \vee \neg p$)
- ▶ Contradiction: All cases evaluates to 0. (e.g., $p \wedge \neg p$)

Equivalence

p and q are called **equivalent** iff $p \leftrightarrow q$ is a tautology, denoted by $p \Leftrightarrow q$.

Examples

- ▶ $p \vee \neg p \Leftrightarrow 1$
- ▶ $p \wedge \neg p \Leftrightarrow 0$
- ▶ $p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p \Leftrightarrow \neg p \vee q$

Tautological Equivalence

► Commutativity

$$p \wedge q \Leftrightarrow q \wedge p$$

$$p \vee q \Leftrightarrow q \vee p$$

► Associativity

$$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$$

$$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$$

$$(p \leftrightarrow q) \leftrightarrow r \Leftrightarrow p \leftrightarrow (q \leftrightarrow r)$$

$$(p \oplus q) \oplus r \Leftrightarrow p \oplus (q \oplus r)$$

► Distributivity

$$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$$

$$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$$

Tautological Equivalence

► Negation

$$\neg\neg p \Leftrightarrow p$$

$$p \rightarrow q \Leftrightarrow \neg q \rightarrow \neg p$$

$$\neg(p \vee q) \Leftrightarrow (\neg p) \wedge (\neg q)$$

$$\neg(p \wedge q) \Leftrightarrow (\neg p) \vee (\neg q)$$

► Identity

$$p \vee 0 \Leftrightarrow p \qquad (\max\{p, 0\} = p)$$

$$p \wedge 1 \Leftrightarrow p \qquad (\min\{p, 1\} = p)$$

► Null

$$p \wedge 0 \Leftrightarrow 0 \qquad (\min\{p, 0\} = 0)$$

$$p \vee 1 \Leftrightarrow 1 \qquad (\max\{p, 1\} = 1)$$

Tautological Equivalence

- ▶ Idempotent

$$p \wedge p \Leftrightarrow p$$

$$p \vee p \Leftrightarrow p$$

- ▶ Absorption

$$p \wedge (p \vee q) \Leftrightarrow p$$

$$p \vee (p \wedge q) \Leftrightarrow p$$

- ▶ Cases

$$(p \rightarrow q) \wedge (p \rightarrow r) \Leftrightarrow p \rightarrow (q \wedge r)$$

$$(p \rightarrow q) \vee (p \rightarrow r) \Leftrightarrow p \rightarrow (q \vee r)$$

$$(p \rightarrow r) \wedge (q \rightarrow r) \Leftrightarrow (p \vee q) \rightarrow r$$

$$(p \rightarrow r) \vee (q \rightarrow r) \Leftrightarrow (p \wedge q) \rightarrow r$$

- ▶ Added premise/exportation

$$(p \wedge q) \rightarrow r \Leftrightarrow p \rightarrow (q \rightarrow r)$$

$$\Leftrightarrow q \rightarrow (p \rightarrow r)$$

Added Premise/Exportation

p	q	r	$p \wedge q$	r	$(p \wedge q) \rightarrow r$	p	$q \rightarrow r$	$p \rightarrow (q \rightarrow r)$
0	0	0	0	0	1	0	1	1
0	0	1	0	1	1	0	1	1
0	1	0	0	0	1	0	0	1
0	1	1	0	1	1	0	1	1
1	0	0	0	0	1	1	1	1
1	0	1	0	1	1	1	1	1
1	1	0	1	0	0	1	0	0
1	1	1	1	1	1	1	1	1

Tautological Equivalence

Example

Prove the absorption rule $p \wedge (p \vee q) \Leftrightarrow p$.

Proof.

$$\begin{aligned} p \wedge (p \vee q) &\Leftrightarrow (p \vee 0) \wedge (p \vee q) \\ &\Leftrightarrow p \vee (0 \wedge q) \\ &\Leftrightarrow p \vee 0 \\ &\Leftrightarrow p \end{aligned}$$



Remark

Consider the saturation function with parameter $a, b \in \mathbb{R}$, $a \leq b$,

$$\text{SAT}_{a,b}(x) := \begin{cases} a, & x < a \\ x, & a \leq x \leq b \\ b, & x > b \end{cases}$$

Note that $\text{SAT}_{a,b}(x) = \min\{b, \max\{a, x\}\} = \max\{a, \min\{b, x\}\}$.

Tautological Equivalence

Remark (Cont.)

Then we can write

$$p \wedge (p \vee q) = \min\{p, \max\{p, q\}\} = \text{SAT}_{p,p}(q) = p$$

or

$$p \wedge (p \vee q) = \min\{p, \max\{q, p\}\} = \text{SAT}_{q,p}(p) = p$$

Remark

Note that

$$\text{SAT}_{a,b}(x) = \text{MEDIAN}(a, b, x)$$

Now we can see that the order of a, b, x in SAT does not matter.

Tautological Equivalence

Example

Show that $(p \rightarrow r) \wedge (q \rightarrow r) \Leftrightarrow (p \vee q) \rightarrow r$.

Proof.

$$\begin{aligned}(p \rightarrow r) \wedge (q \rightarrow r) &\Leftrightarrow (\neg p \vee r) \wedge (\neg q \vee r) \\ &\Leftrightarrow (\neg p \wedge \neg q) \vee r \\ &\Leftrightarrow (\neg(p \vee q)) \vee r \\ &\Leftrightarrow (p \vee q) \rightarrow r\end{aligned}$$



Example

Show that $(p \rightarrow r) \vee (q \rightarrow r) \Leftrightarrow (p \wedge q) \rightarrow r$.

Proof.

$$\begin{aligned}(p \rightarrow r) \vee (q \rightarrow r) &\Leftrightarrow (\neg p) \vee r \vee (\neg q) \vee r \\ &\Leftrightarrow (\neg p) \vee (\neg q) \vee r \\ &\Leftrightarrow (\neg(p \wedge q)) \vee r \\ &\Leftrightarrow (p \wedge q) \rightarrow r\end{aligned}$$



CNF and DNF

Conjunctive Normal Form (CNF)

A proposition is in **Conjunctive Normal Form (CNF)** if it is a conjunction of one or more clauses, where a clause is a disjunction of literals; i.e., it is a **product of sums** or an **AND of ORs**.

Disjunctive Normal Form (DNF)

A proposition is in **Disjunctive Normal Form (CNF)** if it is a Disjunction of one or more clauses, where a clause is a conjunction of literals; i.e., it is a **sum of products** or an **OR of ANDS**.

Remark

A **literal** is a Boolean variable, (i.e., an atomic proposition) or its negation.

Example

- ▶ CNF: $(\neg p \vee q \vee r) \wedge (\neg q \vee \neg r) \wedge (r)$
- ▶ DNF: $(\neg p \wedge q \wedge r) \vee (\neg q \wedge \neg r) \vee (r)$

CNF and DNF

Theorem

For any proposition φ , there is a proposition φ_{dnf} over the same Boolean variables and in DNF such that $\varphi \Leftrightarrow \varphi_{dnf}$.

Example

► $\varphi = p \vee q$

$$\varphi_{dnf} = (p) \vee (q)$$

► $\varphi = p \wedge q$

$$\varphi_{dnf} = (p \wedge q)$$

► $\varphi = p \rightarrow q$

$$\varphi_{dnf} = (\neg p) \vee (q)$$

► $\varphi = p \leftrightarrow q$

$$\varphi_{dnf} = (p \wedge q) \vee (\neg q \wedge \neg p)$$

► $\varphi = p \oplus q$

$$\varphi_{dnf} = (\neg p \wedge q) \vee (p \wedge \neg q)$$

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

$$\varphi_{dnf} = (\neg p \wedge q) \vee (p \wedge \neg q)$$

CNF and DNF

Theorem

For any proposition φ , there is a proposition φ_{cnf} over the same Boolean variables and in CNF such that $\varphi \Leftrightarrow \varphi_{cnf}$.

Example

► $\varphi = p \vee q$

$$\varphi_{cnf} = (p \vee q)$$

► $\varphi = p \wedge q$

$$\varphi_{cnf} = (p) \wedge (q)$$

► $\varphi = p \rightarrow q$

$$\varphi_{cnf} = (\neg p \vee q)$$

► $\varphi = p \leftrightarrow q$

$$\varphi_{cnf} = (\neg p \vee q) \wedge (\neg q \vee p)$$

► $\varphi = p \oplus q$

$$\varphi_{cnf} = (p \vee q) \wedge (\neg q \vee \neg p)$$

p	q	$p \oplus q$
0	0	0
0	1	1
1	0	1
1	1	0

Since

$$\neg(\varphi_{cnf}) = (\neg p \wedge \neg q) \vee (p \wedge q)$$

Then by De Morgan's law

$$\varphi_{cnf} = (p \vee q) \wedge (\neg q \vee \neg p)$$

Rules of Inference/Formal Implication/Deduction

- ▶ Detachment (Modus ponens): $(p \rightarrow q) \wedge p \Rightarrow q$
- ▶ Indirect reasoning (Modus tollens): $(p \rightarrow q) \wedge \neg q \Rightarrow \neg p$
- ▶ Disjunctive addition: $p \Rightarrow (p \vee q)$
- ▶ Conjunctive simplification: $(p \wedge q) \Rightarrow p$
- ▶ Disjunctive syllogism: $(p \vee q) \wedge \neg p \Rightarrow q$
- ▶ Hypothetical syllogism: $(p \rightarrow q) \wedge (q \rightarrow r) \Rightarrow p \rightarrow r$
- ▶ Resolution: $(p \vee q) \wedge (\neg p \vee r) \Rightarrow q \vee r$

Proof by Contrapositive

$$p \rightarrow q \Leftrightarrow (\neg q \rightarrow \neg p)$$

Proof by Contradiction

$$p \rightarrow q \Leftrightarrow (p \wedge \neg q) \rightarrow 0$$

Rules of Inference/Formal Implication/Deduction

Definition

We say that p formally implies q if $p \rightarrow q$ is a tautology, denoted by $p \Rightarrow q$.

Example

Disjunctive addition:

$$p \Rightarrow (p \vee q)$$

p	q	$p \vee q$	$p \rightarrow p \vee q$
0	0	0	1
0	1	1	1
1	0	1	1
1	1	1	1

Summary (in terms of truth tables)

- ▶ Tautology: all 1's in the truth table;
- ▶ Contradiction: all 0's in the truth table;
- ▶ Equivalence $p \Leftrightarrow q$: same value in the truth table;
- ▶ Formal Implication $p \Rightarrow q$: if $p = 1$, then $q = 1$.

Rules of Inference/Formal Implication/Deduction

Note that $p \Leftrightarrow q$ iff $p \Rightarrow q$ and $q \Rightarrow p$.

Example

Recall the the absorption rule $p \wedge (p \vee q) \Leftrightarrow p$. To establish this, we can show both $p \wedge (p \vee q) \Rightarrow p$ and $p \Rightarrow p \wedge (p \vee q)$.

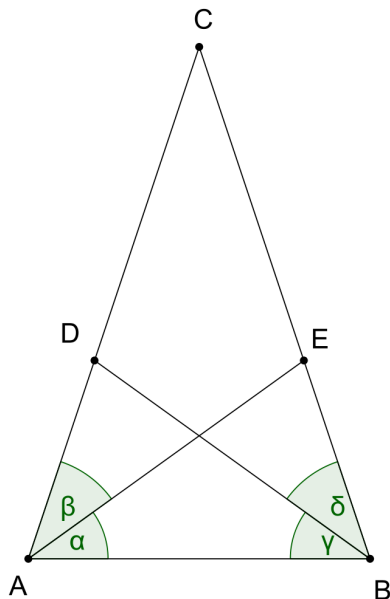
- ▶ $p \wedge (p \vee q) \Rightarrow p$. Indeed, since if $p \wedge (p \vee q) = 1$, then $p = 1$.
- ▶ $p \Rightarrow p \wedge (p \vee q)$. Indeed, since if $p = 1$, then $p \wedge (p \vee q) = 1$ regardless of q .

Direct Proofs Might Be Hard

Theorem (Steiner-Lehmus)

Every triangle with two angle bisectors of equal lengths is isosceles.

$$|AE| = |BD|, \alpha = \beta, \gamma = \delta \\ \Rightarrow \triangle ABC \text{ is isosceles.}$$



The Natural Numbers

Definition

Let $m, n \in \mathbb{N}$ be natural numbers.

- (i) We say that n is **greater than or equal to** m , writing $n \geq m$, if there exists some $k \in \mathbb{N}$ such that $n = m + k$. If we can choose $k \neq 0$, we say n is **greater than** m and write $n > m$.
- (ii) We say that m **divides** n , writing $m \mid n$, if there exists some $k \in \mathbb{N}$ such that $n = m \cdot k$.
- (iii) If $2 \mid n$, we say that n is even.
- (iv) If there exists some $k \in \mathbb{N}$ such that $n = 2k + 1$, we say that n is odd.
- (v) Suppose that $n > 1$. If there does not exist any $k \in \mathbb{N}$ with $1 < k < n$ such that $k \mid n$, we say that n is **prime**.

Remark

It can be proven that every number is either even or odd and not both (a special property of **equivalence classes**). We also assume this for the purposes of our examples.

Infinitude of Primes

Theorem

There are infinitely many prime numbers.

Proof (NOT due to Euclid).

Assume that there are only finitely many primes, say $\mathbb{P} = \{p_1, \dots, p_k\}$. consider the integer $N = p_1 p_2 \cdots p_k + 1$, observe that $p_i \nmid N$ for any $i = 1, \dots, k$, so N must be a prime, but $N \notin \mathbb{P}$, contradiction! \square

Proof (by Euclid).

Consider a finite set of primes $\{p_1, \dots, p_k\}$. Let $N = p_1 p_2 \cdots p_k + 1$, so

- ▶ either N is a prime;
- ▶ or N is not a prime, so N must admit a prime factor, which is not in $\{p_1, \dots, p_k\}$. Call this new prime p_{k+1} .

So we can always generate a new prime from a finite set of primes. \square

Remark

Euclid's proof of the infinitude of primes is **NOT** a proof by contradiction.

Proof by Contradiction

Recall a proof by contradiction admits the form

$$p \rightarrow q \Leftrightarrow (p \wedge \neg q) \rightarrow 0$$

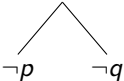

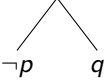
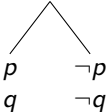
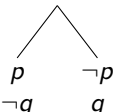
Specifically, for some proposition r ,

$$\begin{aligned} p \rightarrow q &\Leftrightarrow (p \wedge \neg q) \rightarrow 0 \\ &\Leftrightarrow (p \wedge \neg q) \rightarrow (r \wedge \neg r) \\ &\Leftrightarrow (p \wedge \neg q \rightarrow r) \wedge (p \wedge \neg q \rightarrow \neg r) \end{aligned}$$

What if $r = q$?

$$\begin{aligned} p \rightarrow q &\Leftrightarrow (p \wedge \neg q) \rightarrow (q \wedge \neg q) \\ &\Leftrightarrow (p \wedge \neg q \rightarrow q) \wedge \underbrace{(p \wedge \neg q \rightarrow \neg q)}_{=1} \\ &\Leftrightarrow p \wedge \neg q \rightarrow q \\ &\Leftrightarrow \neg q \rightarrow (p \rightarrow q) \\ &\Leftrightarrow q \vee (p \rightarrow q) \end{aligned}$$

Method of Truth Trees

$p \wedge q$ p q	$\neg(p \vee q)$ $\neg p$ $\neg q$	$\neg(p \rightarrow q)$ p $\neg q$	$\neg\neg p$ p
$\neg(p \wedge q)$ 	$p \vee q$ 	$p \rightarrow q$ 	
$p \leftrightarrow q$ 	$\neg(p \leftrightarrow q)$ 		

Method of Truth Trees

Example

Say we want to show $((p \wedge q) \rightarrow r) \Rightarrow (p \rightarrow (q \rightarrow r))$, we can use truth table, or the following truth tree method (as an application of proof by contradiction).

$$((p \wedge q) \rightarrow r) \Rightarrow (p \rightarrow (q \rightarrow r))$$

$$(p \wedge q) \rightarrow r$$

$$\neg(p \rightarrow (q \rightarrow r))$$

p

$$\neg(q \rightarrow r)$$

q

$\neg r$

$$\neg(p \wedge q)$$

r

$\neg p$

X

$\neg q$

x

x

Method of Truth Trees

Example

We can also write $((p \wedge q) \rightarrow r) \Rightarrow (p \rightarrow (q \rightarrow r))$ in the form $(p \wedge q) \rightarrow r \vdash p \rightarrow (q \rightarrow r)$.

$$\begin{array}{c}
 (p \wedge q) \rightarrow r \vdash p \rightarrow (q \rightarrow r) \\
 (p \wedge q) \rightarrow r \\
 p \\
 \neg(q \rightarrow r) \\
 q \\
 \neg r \\
 \swarrow \quad \searrow \\
 \neg(p \wedge q) \quad r \\
 \swarrow \quad \searrow \quad \text{\textbf{x}} \\
 \neg p \quad \neg q \\
 \text{\textbf{x}} \quad \text{\textbf{x}}
 \end{array}$$

Basically, $p_1, \dots, p_n \vdash q_1, \dots, q_m$ means $p_1 \wedge \dots \wedge p_n \Rightarrow q_1 \vee \dots \vee q_m$.

Method of Truth Trees

$$(\neg q \rightarrow \neg p) \wedge (\neg r \rightarrow \neg s), s \wedge (s \rightarrow \neg r), t \rightarrow p \vdash \neg t$$

$$(\neg q \rightarrow \neg p) \wedge (\neg r \rightarrow \neg s)$$

$$s \wedge (s \rightarrow \neg r)$$

$$t \rightarrow p$$

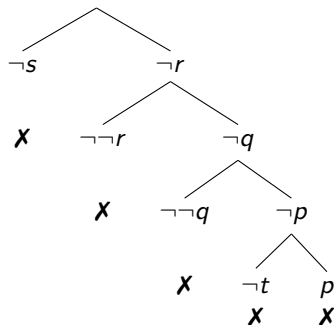
$$\neg \neg t$$

$$s$$

$$s \rightarrow \neg r$$

$$\neg q \rightarrow \neg p$$

$$\neg r \rightarrow \neg s$$



Statements

Examples

- ▶ “ $3 > 2$ ” is a *true statement*.
- ▶ “ $x^3 > 10$ ” is not a statement, because we can not decide whether it is true or not.
- ▶ “the cube of any natural number is greater than 10” is a *false statement*.

The last example can be written using a *statement variable* n :

- ▶ “For any natural number n , $n^3 > 10$ ”

The first part of the statement is a *quantifier* (“for any natural number n ”), while the second part is called a *statement form* or *predicate* (“ $n^3 > 10$ ”).

Statements

Definition

A function $P : X \rightarrow \{\top, \perp\}$ is called a *predicate* on its domain X .

Remark

A statement form or predicate becomes a statement (which can then be either true or false) when the variable takes on a specific value.

Example

If $P(x)$ stands for “ $x^3 > 10$ ”, then

- ▶ $P(10) = \top$, i.e., $10^3 > 10$ is a TRUE statement;
- ▶ $P(1) = \perp$, i.e., $1^3 > 10$ is a FALSE statement.

Recall We indicate that an *element* x is a member of a *set* X by writing $x \in X$. We may characterize the elements of a set X by some predicate P :

$$x \in X \Leftrightarrow P(x).$$

We write $X = \{x : P(x)\} = \{x \mid P(x)\}$, which is called the *set-builder notation*.

Logical Quantifiers

There are two types of quantifiers:

- ▶ the **universal quantifier**, denoted by the symbol \forall , read as “for all” and
- ▶ the **existential quantifier**, denoted by \exists , read as “there exists.”

Definition

Let M be a set and $A(x)$ be a predicate. Then we define the quantifier \forall by

$$(\forall x \in M)A(x) \quad \Leftrightarrow \quad A(x) \text{ is true for } \textbf{all } x \in M$$

We define the quantifier \exists by

$$(\exists x \in M)A(x) \quad \Leftrightarrow \quad A(x) \text{ is true for } \textbf{at least one } x \in M$$

We may also write $\forall x \in M: A(x)$, $\forall_{x \in M} A(x)$, or $\bigwedge_{x \in M} A(x)$ instead of $(\forall x \in M)A(x)$. Similarly for \exists .

Scope of Quantifiers

Example

$$\forall x(P(x, y) \rightarrow \forall y(\underbrace{\exists x(P(z, x))}_{\text{scope of } \exists x} \wedge P(x, z)))$$

► Free variables

$$\forall x(P(x, y) \rightarrow \forall y(\exists x(P(z, x)) \wedge P(x, z)))$$

► Bound variables

Remark

A proposition has no free variables.

Other forms of quantification

It is also common to represent “there exists a unique x such that $P(x)$ is true” as $\exists!xP(x)$, meaning

$$\exists x(P(x) \wedge \forall y(P(y) \rightarrow y = x)) \quad \text{or} \quad \exists x \forall y(P(y) \leftrightarrow y = x)$$

The following are also convenient,

- ▶ $\exists_{\geq 1}$: there exists at least one
- ▶ $\exists_{\leq 1}$: there exists at most one

Sometimes we might restrict the quantifier domain, for example (more on this later)

- ▶ $\forall x > 0(x^2 > 0) \Leftrightarrow \forall x(x > 0 \rightarrow x^2 > 0)$
- ▶ $\forall y \neq 0(y^2 > 0) \Leftrightarrow \forall y(x \neq 0 \rightarrow y^2 \neq 0)$
- ▶ $\exists z > 0(z^2 = 2) \Leftrightarrow \exists z(z > 0 \wedge z^2 = 2)$

Logical Quantifiers

We may also state the domain before making the statements, as in the following example.

Examples

Let x be a real number. Then

- ▶ $\forall x: x > 0 \rightarrow x^3 > 0$ is a true statement;
- ▶ $\forall x: x > 0 \leftrightarrow x^2 > 0$ is a false statement;
- ▶ $\exists x: x > 0 \leftrightarrow x^2 > 0$ is a true statement.

Sometimes mathematicians put a quantifier at the end of a statement form; this is known as a *hanging quantifier*. Such a hanging quantifier will be interpreted as being located just before the statement form:

$$\exists y: y + x^2 > 0 \qquad \forall x$$

is equivalent to $\exists y \forall x: y + x^2 > 0$.

Contraposition and Negation of Quantifiers

We do not actually need the quantifier \exists since

$$\begin{aligned}\exists x \in M : A(x) &\Leftrightarrow A(x) \text{ is true for at least one } x \in M \\ &\Leftrightarrow \text{"}A(x) \text{ is not false for all } x \in M\text{"}^1 \\ &\Leftrightarrow \neg \forall x \in M (\neg A(x))\end{aligned}$$

The last equivalence is called *contraposition of quantifiers*. It implies that the negation of $\exists x \in M : A(x)$ is equivalent to $\forall x \in M : \neg A(x)$. For example,

$$\neg (\exists x \in \mathbb{R} : x^2 < 0) \Leftrightarrow \forall x \in \mathbb{R} : x^2 \not< 0.$$

Conversely,

$$\neg [\forall x \in M : A(x)] \Leftrightarrow \exists x \in M : \neg A(x).$$

1. cf., all is not lost.

De Morgan's Laws for Quantifiers

It is intuitive to consider the following

$$\forall x \in \{x_1, x_2, \dots, x_n\} : P(x) \Leftrightarrow P(x_1) \wedge P(x_2) \wedge \dots \wedge P(x_n)$$

$$\exists x \in \{x_1, x_2, \dots, x_n\} : P(x) \Leftrightarrow P(x_1) \vee P(x_2) \vee \dots \vee P(x_n)$$

thus

$$\neg[\forall x \in \{x_1, x_2, \dots, x_n\} : P(x)] \Leftrightarrow \neg P(x_1) \vee \neg P(x_2) \vee \dots \vee \neg P(x_n)$$

$$\neg[\exists x \in \{x_1, x_2, \dots, x_n\} : P(x)] \Leftrightarrow \neg P(x_1) \wedge \neg P(x_2) \wedge \dots \wedge \neg P(x_n)$$

To summarize,

$$\blacktriangleright \neg[\exists x \in M : A(x)] \Leftrightarrow \forall x \in M : \neg A(x)$$

$$\blacktriangleright \neg[\forall x \in M : A(x)] \Leftrightarrow \exists x \in M : \neg A(x)$$

$$\blacktriangleright \neg[\forall x \in M : \neg A(x)] \Leftrightarrow \exists x \in M : A(x)$$

$$\blacktriangleright \neg[\exists x \in M : \neg A(x)] \Leftrightarrow \forall x \in M : A(x)$$

Theorems in Predicate Logic

- ▶ $(\forall x \in M)[P(x) \vee \neg P(x)]$
- ▶ $\neg[(\forall x \in M)P(x)] \Leftrightarrow [(\exists x \in M)\neg P(x)]$
- ▶ $(\forall x \in \emptyset)P(x)$
- ▶ $\neg[(\exists x \in M)P(x)] \Leftrightarrow [(\forall x \in M)\neg P(x)]$
- ▶ $\neg(\exists x \in \emptyset)P(x)$
- ▶ $[(\forall x \in M)P(x)] \Rightarrow [(\exists x \in M)P(x)]$, if $M \neq \emptyset$
- ▶ $(\exists x \in M)[P(x) \vee Q(x)] \Leftrightarrow [(\exists x \in M)P(x)] \vee [(\exists x \in M)Q(x)]$
- ▶ $(\forall x \in M)[P(x) \wedge Q(x)] \Leftrightarrow [(\forall x \in M)P(x)] \wedge [(\forall x \in M)Q(x)]$
- ▶ $(\exists x \in M)[P(x) \wedge Q(x)] \Rightarrow [(\exists x \in M)P(x)] \wedge [(\exists x \in M)Q(x)]$
- ▶ $(\forall x \in M)[P(x) \vee Q(x)] \Leftarrow [(\forall x \in M)P(x)] \vee [(\forall x \in M)Q(x)]$
- ▶ $(\exists x \in M)[P(x) \wedge Q(x)] \Leftarrow [(\forall x \in M)P(x)] \wedge [(\exists x \in M)Q(x)]$
- ▶ $[(\forall x \in M)P(x) \rightarrow Q(x)] \wedge [(\forall x \in M)P(x)] \Rightarrow [(\forall x \in M)Q(x)]$
- ▶ $[(\forall x \in \{y \in M \mid P(y)\})Q(x)] \Leftrightarrow (\forall x \in M)[P(x) \rightarrow Q(x)]$
- ▶ $[(\exists x \in \{y \in M \mid P(y)\})Q(x)] \Leftrightarrow (\exists x \in M)[P(x) \wedge Q(x)]$
- ▶ $\varphi \wedge [(\exists x \in M)P(x)] \Leftrightarrow (\exists x \in M)[\varphi \wedge P(x)]$, if no x in φ
- ▶ $\varphi \vee [(\forall x \in M)P(x)] \Leftrightarrow (\forall x \in M)[\varphi \vee P(x)]$, if no x in φ

Theorems in Predicate Logic

$$(\forall x \in \{y \in M \mid P(y)\})Q(x) \Leftrightarrow (\forall x \in M)[P(x) \rightarrow Q(x)]$$

Take any $x \in M$, note that

$$(x \in M \wedge P(x)) \rightarrow Q(x) \Leftrightarrow x \in M \rightarrow (P(x) \rightarrow Q(x))$$

$$(\exists x \in \{y \in M \mid P(y)\})Q(x) \Leftrightarrow (\exists x \in M)[P(x) \wedge Q(x)]$$

Consider the negation,

$$\begin{aligned}(\forall x \in \{y \in M \mid P(y)\})\neg Q(x) &\Leftrightarrow (\forall x \in M)(\neg P(x) \vee \neg Q(x)) \\ &\Leftrightarrow (\forall x \in M)(P(x) \rightarrow \neg Q(x))\end{aligned}$$

which is done above.

Vacuous Truth

If the domain of the universal quantifier \forall is the empty set $M = \emptyset$, then the statement

$$(\forall x \in M)A(x) \Leftrightarrow \forall x(x \in M \rightarrow A(x))$$

is true regardless of the predicate $A(x)$. It is then said that $A(x)$ is ***vacuously true***.

Example

Let $M := \{x \in \mathbb{R} \mid x = x + 1\}$. Then the statement $\forall x \in M(x > x)$ is true.

This is similar to saying that “all **pink elephants** can fly” is a true statement, because it is impossible (as far as we know) to find a **pink elephant** that cannot fly.

Remark

This is not the same for existential quantifier, since

$$(\exists x \in M)A(x) \Leftrightarrow \exists x(x \in M \wedge A(x))$$

Q: How about $\exists x(x \in M \rightarrow A(x))$?

Nesting Quantifiers

We can also treat predicates with more than one variable. Quantifiers can be grouped into blocks as

$$\forall a \forall b \dots \forall c \quad \exists u \exists v \dots \exists w \quad \forall x \forall y \dots \forall z$$

In general, quantifiers can be swapped inside a block, but not between blocks.

Examples

In the following examples, $x, y \in \mathbb{R}$.

- ▶ $\forall x \forall y: x^2 + y^2 - 2xy \geq 0$ is equivalent to $\forall y \forall x: x^2 + y^2 - 2xy \geq 0$.
Therefore, one often writes $\forall x, y: x^2 + y^2 - 2xy \geq 0$.
- ▶ $\exists x \exists y: x + y > 0$ is equivalent to $\exists y \exists x: x + y > 0$, often abbreviated to $\exists x, y: x + y > 0$.
- ▶ $\forall x \exists y: x + y > 0$ is a true statement.
- ▶ $\exists x \forall y: x + y > 0$ is a false statement.

Nesting Quantifiers

Consider $M = \{1, 2, 3, 4, 5\}$, and the predicate

$$P : M \times M \rightarrow \{\top, \perp\}, \quad (x, y) \mapsto P(x, y).$$

$P(x, y)$	1	2	3	4	5	
1	\top	\perp	\perp	\top	\top	(i) $\forall x \exists y P(x, y) \Rightarrow \top$
2	\top	\perp	\top	\top	\top	(ii) $\exists y \forall x P(x, y) \Rightarrow \top$
3	\perp	\perp	\top	\top	\top	(iii) $\forall y \exists x P(x, y) \Rightarrow \top$
4	\perp	\top	\perp	\top	\perp	(iv) $\exists x \forall y P(x, y) \Rightarrow \perp$
5	\top	\perp	\top	\top	\top	

Note that we can get (i) from (ii) for free, since

$$\exists y \forall x P(x, y) \Rightarrow \forall x \exists y P(x, y)$$

which can be viewed as a full-blown version of

$$\exists y (P_{x_1}(y) \wedge P_{x_2}(y)) \Rightarrow \exists y P_{x_1}(y) \wedge \exists y P_{x_2}(y)$$

Nesting Quantifiers

Example: $f_n \rightarrow f$

- Pointwise convergence: $\lim_{n \rightarrow \infty} (f_n(x) - f(x)) = 0$ for all $x \in I$

$$\forall \varepsilon > 0, \forall x \in I, \exists n \in \mathbb{N}, \forall m \geq n, |f_m(x) - f(x)| < \varepsilon$$

- Uniform convergence: $\lim_{n \rightarrow \infty} \sup_{x \in I} |f_n(x) - f(x)| = 0$

$$\forall \varepsilon > 0, \exists n \in \mathbb{N}, \forall x \in I, \forall m \geq n, |f_m(x) - f(x)| < \varepsilon$$

Nesting Quantifiers

We show that $\exists y \forall x P(x, y) \rightarrow \forall x \exists y P(x, y)$ is a tautology. We can make the following transformations,

$$\begin{aligned} & \exists y \forall x P(x, y) \rightarrow \forall x \exists y P(x, y) \\ \Leftrightarrow & \exists y \forall x P(x, y) \rightarrow \forall z \exists w P(z, w) \\ \Leftrightarrow & \forall y \forall z \exists x \exists w (P(x, y) \rightarrow P(z, w)) \end{aligned}$$

which is indeed true by taking $x = z$ and $w = y$.

More Theorems

If no x in φ and $M \neq \emptyset$, then

- ▶ $\varphi \Leftrightarrow [(\forall x \in M)\varphi]$
- ▶ $\varphi \vee [(\forall x \in M)P(x)] \Leftrightarrow (\forall x \in M)[\varphi \vee P(x)]$
- ▶ $\varphi \wedge [(\exists x \in M)P(x)] \Leftrightarrow (\exists x \in M)[\varphi \wedge P(x)]$
- ▶ $\varphi \rightarrow [(\exists x \in M)P(x)] \Leftrightarrow (\exists x \in M)[\varphi \rightarrow P(x)]$
- ▶ $[(\exists x \in M)P(x)] \rightarrow \varphi \Leftrightarrow (\forall x \in M)[P(x) \rightarrow \varphi]$

Nesting Quantifiers

We can do better by considering another predicate $Q : M \times M \rightarrow \{\top, \perp\}$ given as follows, we also list P below,

$Q(x, y)$	1	2	3	4	5	$P(x, y)$	1	2	3	4	5
1	\perp	\perp	\perp	\top	\perp	1	\top	\perp	\perp	\top	\top
2	\perp	\perp	\perp	\top	\perp	2	\top	\perp	\top	\top	\top
3	\perp	\perp	\perp	\top	\perp	3	\perp	\perp	\top	\top	\top
4	\perp	\perp	\perp	\top	\perp	4	\perp	\top	\perp	\top	\perp
5	\perp	\perp	\perp	\top	\perp	5	\top	\perp	\top	\top	\top

Note that $\forall x \forall y [Q(x, y) \Rightarrow P(x, y)]$, which implies $\exists y \forall x Q(x, y) \Rightarrow \exists y \forall x P(x, y)$. Indeed, since

$$\begin{aligned} & \exists y \forall x Q(x, y) \rightarrow \exists y \forall x P(x, y) \\ \Leftrightarrow & \exists y \forall x Q(x, y) \rightarrow \exists w \forall z P(z, w) \\ \Leftrightarrow & \forall y \exists w \forall z \exists x [Q(x, y) \rightarrow P(z, w)] \end{aligned}$$

which is indeed true by taking $w = y$ and $x = z$.

Nesting Quantifiers

The good news is that Q is much easier to deal with, since we can write Q as $Q(x, y) = R(x) \wedge S(y)$, where $R(x) = 1$ for all x , and $S(y) = \delta_{4y}$. In vector notation, we have

$$v_R = \mathbf{1}^\top = [1 \quad 1 \quad 1 \quad 1 \quad 1]^\top \text{ and } v_S = [0 \quad 0 \quad 0 \quad 1 \quad 0]^\top$$

thus the matrix representation for Q is given by

$$M_Q = v_R v_S^\top = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Now note that it is straightforward to show $\exists y \forall x Q(x, y)$ or $\forall x \exists y Q(x, y)$, since

$$\exists y \forall x [R(x) \wedge S(y)] \Leftrightarrow \forall x \exists y [R(x) \wedge S(y)] \Leftrightarrow \forall x R(x) \wedge \exists y S(y)$$

Nesting Quantifiers

We can also write Q as $Q(x, y) = R'(x) \vee S(y)$, where $R'(x) = 0$ for all x . In vector notation,

$$v_{R'} = \mathbf{0}^\top = [0 \ 0 \ 0 \ 0 \ 0]^\top$$

thus (cf., disjunctive normal form)

$$M_Q = v_{R'} \mathbf{1}^\top + \mathbf{1} v_S^\top$$

Similarly, note that

$$\exists y \forall x [R'(x) \vee S(y)] \Leftrightarrow \forall x \exists y [R'(x) \vee S(y)] \Leftrightarrow \forall x R'(x) \vee \exists y S(y)$$

Remark

For more than two variables, we should really use the tensor product instead of the outer product or dyad above.

Axiomatic Set Theory Examples

- ▶ Extensionality Axiom

$$\forall A, B (\forall x (x \in A \leftrightarrow x \in B) \rightarrow A = B)$$

- ▶ Empty Set Axiom

$$\exists B \forall x (x \notin B)$$

- ▶ Pairing Axiom

$$\forall u, v \exists B \forall x (x \in B \leftrightarrow x = u \vee x = v)$$

- ▶ Union Axiom

$$\forall a, b \exists B \forall x (x \in B \leftrightarrow x \in a \vee x \in b)$$

- ▶ Powerset Axiom

$$\forall a \exists b \forall x (x \in B \leftrightarrow x \subseteq a)$$

where $x \subseteq a$ is shorthand for $\forall t (t \in x \rightarrow t \in a)$.

Set-Theoretic Proofs

Basic Facts

$$x \in A \cup B \Leftrightarrow (x \in A) \vee (x \in B)$$

$$x \notin A \cup B \Leftrightarrow (x \notin A) \wedge (x \notin B)$$

$$x \in A \cap B \Leftrightarrow (x \in A) \wedge (x \in B)$$

$$x \notin A \cap B \Leftrightarrow (x \notin A) \vee (x \notin B)$$

$$x \in A - B \Leftrightarrow (x \in A) \wedge (x \notin B)$$

$$x \notin A - B \Leftrightarrow (x \notin A) \vee (x \in B)$$

$$A \subset B \Leftrightarrow (x \in A) \rightarrow (x \in B)$$

$$A = B \Leftrightarrow (A \subset B) \wedge (B \subset A)$$

Remark

- ▶ Prove something exists: sufficient to find an example.
- ▶ Prove $P(x)$ for all $x \in A$: Take any $x \in A$ and continue. (or use induction if $A = \mathbb{N}$, more on this later.)

Starters for Proof by Contradiction

Statement	Assume (for contradiction) that ...
All P 's are Q 's.	Some P is not a Q .
No P 's are Q 's.	Some P is a Q .
Some P 's are Q 's.	All P 's are not Q 's.
Some P is not a Q .	All P 's are Q 's.
If P is true, then Q is true.	P is true, but Q is false.
P is true and Q is true.	P is false, or Q is false.
P is true or Q is true.	P is false and Q is false.

Note that all the or's above are inclusive or's.

Set-Theoretic Proofs

Theoretically, all theorems in propositional logic can be proved using truth tables.

Example

Given sets A, B, C , if $A \subset B$ and $B \cap C = \emptyset$, then $A \cap C = \emptyset$.

Proof.

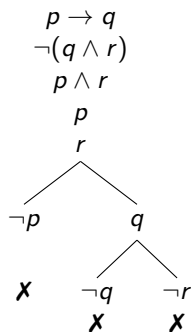
We can use truth table. Let p, q, r be the logical variables for $x \in A$, $x \in B$, $x \in C$, resp.

p	q	r	$p \rightarrow q$	$\neg(q \wedge r)$	$\neg(p \wedge r)$	$[(p \rightarrow q) \wedge \neg(q \wedge r)] \rightarrow \neg(p \wedge r)$
0	0	0	1	1	1	1
0	0	1	1	1	1	1
0	1	0	1	1	1	1
0	1	1	1	0	1	
1	0	0	0	1	1	
1	0	1	0	1	0	
1	1	0	1	1	1	1
1	1	1	1	0	0	

Set-Theoretic Proofs

Or use truth trees.

$$[(p \rightarrow q) \wedge \neg(q \wedge r)] \Rightarrow \neg(p \wedge r)$$



Truth Trees for First-Order Logic

Example

$$\exists y \forall x P(x, y) \Rightarrow \forall x \exists y P(x, y)$$

$$\begin{array}{l} \exists y \forall x P(x, y) \\ \neg \forall x \exists y P(x, y) \\ \exists x \forall y \neg P(x, y) \\ \forall y \neg P(a, y) \\ \forall x P(x, b) \\ \neg P(a, b) \\ P(a, b) \\ \mathbf{X} \end{array}$$

Strategies

- ▶ Introduce existing variables for \forall
- ▶ Introduce new variables for \exists
- ▶ Can reuse \forall -sentences

Simple Axiomatic System (Hilbert)

Axioms

- ▶ $p \rightarrow (q \rightarrow p)$
- ▶ $(p \rightarrow (q \rightarrow r)) \rightarrow ((p \rightarrow q) \rightarrow (p \rightarrow r))$
- ▶ $(\neg p \rightarrow \neg q) \rightarrow (q \rightarrow p)$

Inference Rules

Only modus ponens:

- ▶ $p, p \rightarrow q \vdash q$

Natural Deduction System

Axioms

None

Inference Rules

- ▶ Negation introduction: $(p \rightarrow q), (p \rightarrow \neg q) \vdash \neg p$
- ▶ Negation elimination: $\neg p \vdash (p \rightarrow r)$
- ▶ Double negation elimination: $\neg\neg p \vdash p$
- ▶ Conjunction introduction: $p, q \vdash (p \wedge q)$
- ▶ Conjunction elimination: $(p \wedge q) \vdash p$
- ▶ Disjunction introduction: $p \vdash (p \vee q)$ and $q \vdash (p \vee q)$
- ▶ Disjunction elimination: $p \vee q, p \rightarrow r, q \rightarrow r \vdash r$
- ▶ Modus ponens: $p, p \rightarrow q \vdash q$
- ▶ ...

Duality in Propositional Logic

► \vee vs \wedge

► 0 vs 1

Basic Law	Property	Dual Law
$p \vee q \Leftrightarrow q \vee p$	Commutativity	$p \wedge q \Leftrightarrow q \wedge p$
$(p \vee q) \vee r \Leftrightarrow p \vee (q \vee r)$	Associativity	$(p \wedge q) \wedge r \Leftrightarrow p \wedge (q \wedge r)$
$p \wedge (q \vee r) \Leftrightarrow (p \wedge q) \vee (p \wedge r)$	Distributivity	$p \vee (q \wedge r) \Leftrightarrow (p \vee q) \wedge (p \vee r)$
$p \vee 0 \Leftrightarrow p$	Identity	$p \wedge 1 \Leftrightarrow p$
$p \wedge \neg p \Leftrightarrow 0$	Negation	$p \vee \neg p \Leftrightarrow 1$
$p \vee p \Leftrightarrow p$	Idempotent	$p \wedge p \Leftrightarrow p$
$p \wedge 0 \Leftrightarrow 0$	Null	$p \vee 1 \Leftrightarrow 1$
$p \wedge (p \vee q) \Leftrightarrow p$	Absorption	$p \vee (p \wedge q) \Leftrightarrow p$
$\neg(p \vee q) \Leftrightarrow \neg p \wedge \neg q$	De Morgan	$\neg(p \wedge q) \Leftrightarrow \neg p \vee \neg q$

Duality in Set Theory

► \cup vs \cap

► \emptyset vs U (U is the universe)

Basic Law	Property	Dual Law
$A \cup B = B \cup A$	Commutativity	$A \cap B = B \cap A$
$(A \cup B) \cup C = A \cup (B \cup C)$	Associativity	$(A \cap B) \cap C = A \cap (B \cap C)$
$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$	Distributivity	$A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
$A \cup \emptyset = A$	Identity	$A \cap U = A$
$A \cap A^c = \emptyset$	Negation	$A \cup A^c = U$
$A \cup A = A$	Idempotent	$A \cap A = A$
$A \cap \emptyset = \emptyset$	Null	$A \cup U = U$
$A \cap (A \cup B) = A$	Absorption	$A \cup (A \cap B) = A$
$(A \cup B)^c = A^c \cap B^c$	De Morgan	$(A \cap B)^c = A^c \cup B^c$

Russell's Paradox

Barber Paradox

The barber is the “one who shaves all those, and those only, who do not shave themselves”.

Question: does the barber shave himself?

Consider the set of all sets that do not contain themselves:

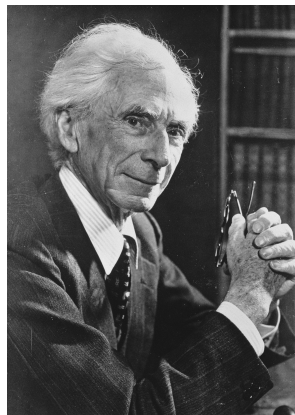
$$S = \{A \mid A \text{ is a set, and } A \notin A\}$$

If such a set exists, then

► $S \in S \rightarrow S \notin S$

► $S \notin S \rightarrow S \in S$

Contradiction!



Bertrand Russell,
Nov. 1957

Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Mathematical Induction

Typically one wants to show that some statement frame $P(n)$ is true for all $n \in \mathbb{N}$ with $n \geq n_0$ for some $n_0 \in \mathbb{N}$. Mathematical induction works by establishing two statements:

- (I) the **base case**: $P(n_0)$ is true.
- (II) the **inductive case**: $P(n+1)$ is true whenever $P(n)$ is true for $n \geq n_0$, i.e.,

$$(\forall n \in \mathbb{N}, n \geq n_0)(P(n) \rightarrow P(n+1))$$

In the inductive case, $P(n)$ is called *inductive hypothesis*, often abbreviated as *IH*.

Note that (II) does not make a statement on the situation when $P(n)$ is false; it is permitted for $P(n+1)$ to be true even if $P(n)$ is false.

The principle of mathematical induction now claims that $P(n)$ is true for all $n \geq n_0$ if (I) and (II) are true.

Introductory Example

Example

Consider the statement

$$\sum_{k=0}^n k = \frac{n(n+1)}{2} \quad \text{for all } n \in \mathbb{N}.$$

This is a typical example, in that $P(n): \sum_{k=0}^n k = \frac{n(n+1)}{2}$ is a predicate which is to be shown to hold for all natural numbers $n \in \mathbb{N}$.

We first establish that $P(0)$ is true:

$$\sum_{k=0}^0 k = 0 \quad \text{and} \quad \frac{0(0+1)}{2} = 0,$$

so $P(0): 0 = 0$ is true.

Introductory Example

We next show that $P(n) \rightarrow P(n+1)$ for all $n \in \mathbb{N} \setminus \{0\}$. This means we show that $\sum_{k=0}^{n+1} k = \frac{(n+1)(n+2)}{2}$ if $\sum_{k=0}^n k = \frac{n(n+1)}{2}$. Let n now be any n for which $P(n)$ is true. We then write

$$\sum_{k=0}^{n+1} k = \left(\sum_{k=0}^n k \right) + n + 1$$

If $P(n)$ is true for this specific n , we can replace the sum on the right by $\frac{n(n+1)}{2}$, yielding

$$\sum_{k=0}^{n+1} k = \frac{n(n+1)}{2} + n + 1 = \frac{(n+1)(n+2)}{2}$$

But this is just the statement $P(n+1)$. Therefore, if $P(n)$ is true, then $P(n+1)$ will also be true. We have shown that $P(n) \rightarrow P(n+1)$. □

Introductory Example

Example

For any $n \in \mathbb{N} \setminus \{0\}$, there exist integers a_n and b_n such that

$$\left(\frac{1 + \sqrt{5}}{2}\right)^n = \frac{a_n + b_n\sqrt{5}}{2}$$

Base case: $n = 1$. Clear by taking $a_1 = b_1 = 1$.

Inductive case: assume the IH that the result holds for some $n \geq 1$, then

$$\begin{aligned}\left(\frac{1 + \sqrt{5}}{2}\right)^{n+1} &= \frac{a_n + b_n\sqrt{5}}{2} \cdot \frac{1 + \sqrt{5}}{2} \\ &= \frac{a_n + b_n\sqrt{5} + a_n\sqrt{5} + 5b_n}{4} \\ &= \frac{(a_n + 5b_n)/2 + ((a_n + b_n)/2)\sqrt{5}}{2}\end{aligned}$$

Does not work...? (exercise: make this work.)

Application of Induction

We can use induction to prove the *efficiency* and *correctness* of a recursive algorithm.

Properties of Algorithms

- ▶ **Input.** An algorithm has input values from a specified set;
- ▶ **Output.** For given input, the algorithm produces output values from a specified set;
- ▶ **Definiteness.** The steps of the algorithm are defined precisely;
- ▶ **Correctness.** For each input, the algorithm produces the correct output values;
- ▶ **Finiteness.** For given input, the algorithm produces output after a finite number of steps;
- ▶ **Effectiveness.** Each step of the algorithm can be performed exactly;
- ▶ **Generality.** The algorithm is generally applicable, not just for certain input values.

Factorial

We can define a function recursively, via pattern matching,

```
fact :: (Eq p, Num p) => p -> p
fact 0 = 1                -- base case
fact n = n * fact (n-1)  -- recursive case
```

or use the `product` function,

```
fact :: (Num a, Enum a) => a -> a
fact n = product [1..n]
```

Remark

Recursion is inefficient (if used literally). Haskell let us write recursive functions and delegate this nuisance to the compiler/interpreter. This allows us to write clear, concise, and correct code.

Convention for Haskell Functions

Function application has the highest precedence in Haskell.

Mathematics	Haskell
$f(x)$	<code>f x</code>
$f(x, y)$	<code>f x y</code> \equiv <code>(f x) y</code>
$f(x, y, z)$	<code>f x y z</code> \equiv <code>((f x) y) z</code> \equiv <code>(f x y) z</code>
$f(x)g(x)$	<code>f x * g y</code>
$f(g(x))$	<code>f (g x)</code> \equiv <code>f \$ g x</code> \equiv <code>f . g \$ x</code>
$f(g(x, y))$	<code>f (g x y)</code> \equiv <code>f \$ g x y</code> \equiv <code>f . g x \$ y</code>
$f(x, g(y))$	<code>f x (g y)</code> \equiv <code>f x . g \$ y</code>

Correctness of fact

Proof.

Induction on n .

- ▶ **(base case):** $n = 0$. Observe that `fact(0)` returns 1 immediately, and also mathematically $0! = 1$.
- ▶ **inductive case:** ($n \geq 0$). Assume that `fact(n)` returns $n!$. We want to show that `fact($n + 1$)` returns $(n + 1)!$. Indeed, by induction hypothesis,

$$\text{fact}(n + 1) = n \cdot \text{fact}(n) = (n + 1) \cdot n! = (n + 1)!$$

Therefore the recursive algorithm `fact` is correct by induction. □

Question

Why do we have $0! = 1$?

A Few Recursively Defined Haskell Functions

- $x_1 + x_2 + \dots + x_n$

```
sum []      = 0
sum (x:xs) = x + sum xs
```

- $x_1 \times x_2 \times \dots \times x_n$

```
product []      = 1
product (x:xs) = x * product xs
```

- $x_1 \vee x_2 \vee \dots \vee x_n$

```
or []      = False
or (x:xs) = x || or xs
```

- $x_1 \wedge x_2 \wedge \dots \wedge x_n$

```
and []      = True
and (x:xs) = x && and xs
```

Remark

$(:)$:: $a \rightarrow [a] \rightarrow [a]$,
e.g.,

```
> 5 : [1,2]
[5,1,2]
> 1 : 2 : 3 : []
[1,2,3]
> 'a' : ['b','c']
['a','b','c']
```

A Few Recursively Defined Haskell Functions

Using foldr,

► $x_1 + x_2 + \cdots + x_n$

```
sum :: Num a => [a] -> a
sum xs = foldr (+) 0 xs
```

► $x_1 \times x_2 \times \cdots \times x_n$

```
product :: Num a => [a] -> a
product xs = foldr (*) 1 xs
```

► $x_1 \vee x_2 \vee \cdots \vee x_n$

```
or :: [Bool] -> Bool
or xs = foldr (||) False xs
```

► $x_1 \wedge x_2 \wedge \cdots \wedge x_n$

```
and :: [Bool] -> Bool
and xs = foldr (&&) True xs
```

Simplified further,

► $x_1 + x_2 + \cdots + x_n$

```
sum :: Num a => [a] -> a
sum = foldr (+) 0
```

► $x_1 \times x_2 \times \cdots \times x_n$

```
product :: Num a => [a] -> a
product = foldr (*) 1
```

► $x_1 \vee x_2 \vee \cdots \vee x_n$

```
or :: [Bool] -> Bool
or = foldr (||) False
```

► $x_1 \wedge x_2 \wedge \cdots \wedge x_n$

```
and :: [Bool] -> Bool
and = foldr (&&) True
```

Monoid

Definition (Monoid)

A **monoid** is a triple (M, e, \star) , where M is a set, together with an identity element $e \in M$, and a function $M \times M \rightarrow M$, such that for all $m, n, p \in M$, the following **monoid laws** hold,

- ▶ $m \star e = m$ and $e \star m = m$
- ▶ $(m \star n) \star p = m \star (n \star p)$

Remark

- ▶ Pedantically, we should phrase the monoid laws as
 - ▶ $\star(m, e) = m$ and $\star(e, m) = m$
 - ▶ $\star(\star(m, n), p) = \star(m, \star(n, p))$
- ▶ It can be shown that the identity element is unique.

Monoid

Example

- ▶ $(\mathbb{B}, \wedge, \top)$, $(\mathbb{B}, \vee, \perp)$, where $\mathbb{B} = \{\top, \perp\}$.
- ▶ $(\mathbb{N}, +, 0)$, $(\mathbb{N}, \times, 1)$.
- ▶ $(\mathcal{P}(A), \cup, \emptyset)$, $(\mathcal{P}(A), \cap, A)$, where A is any set.
- ▶ $(\mathbb{N}, \max, 0)$
- ▶ The set of all finite strings/words/lists over some fixed alphabet/set Σ forms a monoid with string concatenation as the binary function. The empty string serves as the identity element. This monoid is denoted Σ^* and is called the *free monoid* over Σ .
- ▶ The natural numbers \mathbb{N} with the function $f : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, $(a, b) \mapsto a^b$, is not a monoid.

String Concatenation

Haskell uses ++ to represent the concatenation of lists, e.g.

```
> [1,2,3] ++ [4,5]
[1,2,3,4,5]
> ['a','b'] ++ ['c','d','e']
['a','b','c','d','e']
```

Remark

- ▶ The list ['a','b','c','d','e'] is the same as the string "abcde".
- ▶ The concatenation operator can be defined recursively as follows

```
(++) :: [a] -> [a] -> [a]
[]      ++ ys = ys
(x:xs) ++ ys = x : (xs ++ ys)
```

or via foldr,

```
(++) :: [a] -> [a] -> [a]
xs ++ ys = foldr (:) ys xs
```

Insertion Sort

```
insert :: Ord a => a -> [a] -> [a]
insert x []                = [x]
insert x (y:ys) | x <= y   = x : y : ys
                  | otherwise = y : insert x ys

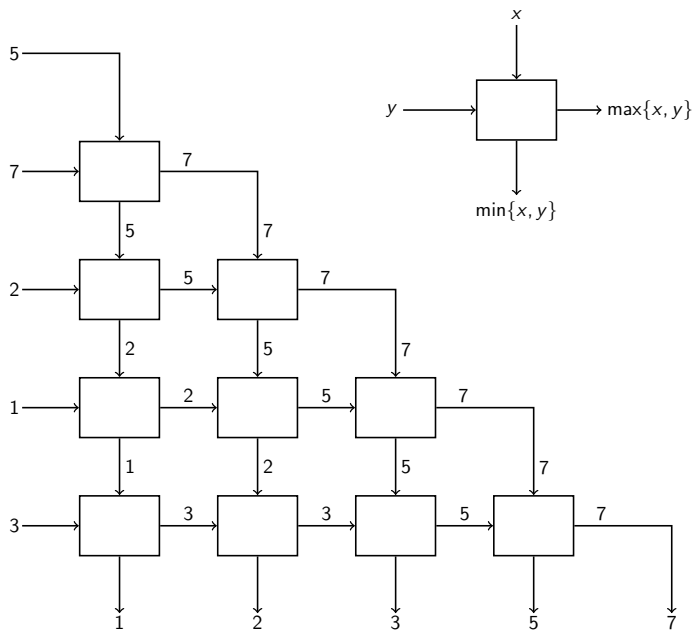
isort :: Ord a => [a] -> [a]
isort [] = []
isort (x:xs) = insert x (isort xs)
```

Selection Sort

```
select :: Ord a => [a] -> (a, [a])
select [x]                = (x, [])
select (x:xs) | x <= y    = (x, y:ys)
                  | otherwise = (y, x:ys)
                  where (y, ys) = select xs

ssort :: Ord a => [a] -> [a]
ssort [] = []
ssort xs = y: ssort ys
  where (y, ys) = select xs
```

Insertion Sort vs Selection Sort



Correctness of insertionSort and selectionSort

Proof. (Correctness of insertionSort).

- ▶ **base case** ($n = 1$): Trivial since any array of length 1 is sorted.
- ▶ **inductive case** ($n \geq 1$): Assume that $\text{insertionSort}(A[1 \dots n], n)$ is sorted. We want to show that $\text{insertionSort}(\langle A[1 \dots n+1] \rangle, n+1)$ is also sorted, which is true since $A[n+1]$ is inserted into the sorted $\text{insertionSort}(A[1 \dots n], n)$ to make this happen. □

Proof. (Correctness of selectionSort).

- ▶ **base case** ($n = 1$): Trivial since any array of length 1 is sorted.
- ▶ **inductive case** ($n \geq 1$): Assume that $\text{selectionSort}(A[1 \dots n], n)$ is sorted. We want to show that $\text{selectionSort}(\langle A[1 \dots n+1] \rangle, n+1)$ is also sorted. Since $A[\text{indexmax}] \geq A[i]$ for all $i < n+1$, then after swap, we have $A[n+1] \geq A[i]$ for all $i < n$. By induction hypothesis, $A[1 \dots n]$ is already sorted, hence $A[1 \dots n+1]$ is sorted. □

Fibonacci Numbers

A Simple Example

The Fibonacci Numbers is defined by

$$F_0 = 0,$$

$$F_1 = 1,$$

$$F_n = F_{n-1} + F_{n-2}, \quad n \geq 2$$

Show that

$$F_n = \frac{1}{\sqrt{5}} \left[\left(\frac{1 + \sqrt{5}}{2} \right)^n - \left(\frac{1 - \sqrt{5}}{2} \right)^n \right]$$

Strong (Complete) Induction

The method of induction can be strengthened. We can replace

- (I) $P(n_0)$ is true.
- (II) $P(n+1)$ is true whenever $P(n)$ is true for $n \geq n_0$.

with

- (I) $P(n_0)$ is true.
- (II') $P(n+1)$ is true whenever all the statements $P(n_0), P(n_0+1), \dots, P(n)$ are true.

Fibonacci Numbers

Proof.

We prove the formula for Fibonacci Numbers by strong induction on n . First let

$$\phi = \frac{1 + \sqrt{5}}{2}, \quad \bar{\phi} = \frac{1 - \sqrt{5}}{2}$$

then we need to show that

$$F_n = \frac{\phi^n - \bar{\phi}^n}{\phi - \bar{\phi}} = \frac{\phi^n - \bar{\phi}^n}{\sqrt{5}}$$

- ▶ **base case ($n = 0$):** $\frac{\phi^0 - \bar{\phi}^0}{\sqrt{5}} = 0 = F_0$
- ▶ **base case ($n = 1$):** $\frac{\phi^1 - \bar{\phi}^1}{\sqrt{5}} = \dots = 1 = F_1$

Fibonacci Numbers

Proof (Cont.)

- **inductive case** ($n \geq 2$): Assume that the formula holds for all $k < n$, then note that ϕ and $\bar{\phi}$ are roots of $x^2 = x + 1$,

$$\begin{aligned} F_n &= F_{n-1} + F_{n-2} \\ &= \frac{\phi^{n-1} - \bar{\phi}^{n-1}}{\sqrt{5}} + \frac{\phi^{n-2} - \bar{\phi}^{n-2}}{\sqrt{5}} \\ &= \frac{\phi^{n-2}(\phi + 1) - \bar{\phi}^{n-2}(1 + \bar{\phi})}{\sqrt{5}} \\ &= \frac{\phi^n - \bar{\phi}^n}{\sqrt{5}} \end{aligned}$$



Remark

Try the same for Lucas numbers defined by $L_0 = 2$, $L_1 = 1$, and $L_n = L_{n-1} + L_{n-2}$ for $n \geq 2$.

Prime Factorization

Theorem (Fundamental Theorem of Arithmetic (Existence Part))

Let $n \in \mathbb{N} \setminus \{0\}$, then there exist $k \geq 0$ prime numbers p_1, p_2, \dots, p_k such that $n = \prod_{i=1}^k p_i$. (also unique up to order, more on this later.)

Proof by strong induction on n .

- ▶ **base case** ($n = 1$): by convention, 1 is the product of zero prime numbers.
- ▶ **inductive case** ($n \geq 2$): assume the statement is true for all positive integer n' where $1 \leq n' \leq n - 1$.
 - ▶ If n is prime, which is the product of 1 prime number.
 - ▶ If n is composite, then by definition there exist positive integers a and b such that $n = a \cdot b$, with $2 \leq a, b \leq n - 1$. By inductive hypotheses, we have $a = q_1 \cdot q_2 \cdots q_\ell$ and $b = r_1 \cdot r_2 \cdots r_m$ for prime numbers q_1, \dots, q_ℓ and r_1, \dots, r_m . Therefore $n = a \cdot b = q_1 \cdot q_2 \cdots q_\ell \cdot r_1 \cdot r_2 \cdots r_m$ which is the product of $\ell + m$ prime numbers. □

Prime Factorization

```
nontrivialFactors :: Integral a => a -> [a]
nontrivialFactors n = [x | x <- [2..n-1], n `mod` x == 0]

primeFactors :: Int -> [Int]
primeFactors 1 = []
primeFactors n
  | null (nontrivialFactors n) = [n]
  | otherwise = pfactor : primeFactors (n `div` pfactor)
where
  pfactor = head (nontrivialFactors n)
```

Remark

We can check whether a number is prime as follows.

```
factors :: Integral a => a -> [a]
factors n = [x | x <- [1..n], n `mod` x == 0]

prime :: Integral a => a -> Bool
prime n = factors n == [1,n]
```

Quick Sort (Hoare, 1959)

```
qsort :: Ord a => [a] -> [a]

qsort []      = []
qsort (x:xs) = qsort smaller ++ [x] ++ qsort larger
               where
                 smaller = [a | a <- xs, a <= x]
                 larger  = [b | b <- xs, b > x]
```

Quick Sort

Proof. (Correctness of quickSort).

Let $P(n)$ denote the claim that $\text{quickSort}(A[1 \dots n])$ returns a sorted array. For simplicity, we assume that the elements in the array are distinct. We will prove $P(n)$ for all $n \geq 0$ by strong induction on n .

- ▶ **base case** ($n = 0, 1$): done b/c any array of length 0 or 1 are already sorted.
- ▶ **inductive case** ($n \geq 2$): assume $P(0), \dots, P(n-1)$ that for any array $B[1 \dots k]$ with distinct elements and $k < n$, $\text{quickSort}(B[1 \dots k])$ returns a sorted array. Let $A[1 \dots n]$ be an arbitrary array with distinct elements. Let $\text{pivot} \in \{1, \dots, n\}$ be arbitrary. We need to show that x appears before y in $\text{quicksort}(A[1 \dots n])$ iff $x < y$.

Quick Sort

Proof. (Correctness of quickSort Cont.)

Inductive case ($n \geq 2$):

- ▶ Case 1. $x = A[pivot]$. By $\text{quickSort}(A[1 \dots n])$, $y \in R$ iff $x < y$.
- ▶ Case 2. $y = A[pivot]$. Similar to Case 1.
- ▶ Case 3. $x, y < A[pivot]$. Now $x, y \in L$. Since $A[pivot] \notin L$, thus L is of at most length $n - 1$. Hence by IH x appears before y in $\text{quickSort}(L)$ iff $x < y$. Furthermore x appears before y in $\text{quickSort}(A[1 \dots n])$ iff $x < y$.
- ▶ Case 4. $x, y > A[pivot]$. Similar to Case 3.
- ▶ Case 5. $x < A[pivot] < y$. Trivial.
- ▶ Case 6. $y < A[pivot] < x$. Impossible. □

	L	$pivot$	R
1.		x	
2.		y	
3.	x, y		
4.			x, y
5.	x		y
6.	y		x

$L = A[1 \dots pivot - 1]$

$R = A[pivot + 1 \dots n]$

Recursive Definitions and the Factorial

Similar to induction, we could make *recursive definitions*. For example, we wish to define a function (to be called the *factorial*) $(\cdot)!: \mathbb{N} \rightarrow \mathbb{N}$ having the properties that

$$n! = \begin{cases} 1, & n = 0 \\ n \cdot (n-1)!, & n \in \mathbb{N} \setminus \{0\} \end{cases}$$

This is an example of a *recursive definition* and we may ask whether such a definition “makes sense”, i.e., whether such a function *exists* and is *unique*. In the present case, the function is simply

$$n! := \prod_{k=1}^n k, \quad n \in \mathbb{N} \setminus \{0\},$$

This is called a *closed formula* for $n!$.

Recursive Definitions

Recursive definitions often occur naturally in the formulation of a problem, and finding a closed formula can be extremely difficult. In some situations, a closed formula is highly desirable, while at other times, important properties are best expressed through recursive expressions.

For example, there exists a continuous extension of the factorial, given by the *Euler gamma function*, defined for $t > 0$,

$$\Gamma(t) := \int_0^{\infty} z^{t-1} e^{-z} dz, \quad t > 0.$$

It is possible to show that $\Gamma(1) = 1$ and that

$$\Gamma(t+1) = t\Gamma(t) = t\Gamma((t-1)+1) \quad t > 0.$$

we see that $\Gamma(n+1) = n!$ for all $n \in \mathbb{N}$. Furthermore, we can define functions not just based on their preceding value, but on several such values. For example, The *Fibonacci sequence*.

Recursive Definitions of Sets

In the same manner, we can define subsets of \mathbb{N} recursively. For example, consider the set $S \subset \mathbb{N}$ such that

(i) $3 \in S$,

(ii) $x, y \in S \rightarrow x + y \in S$.

We know that $3 \in S$, so $3 + 3 = 6 \in S$, $3 + 6 = 9 \in S$, $6 + 6 = 12 \in S$ and so on. Our goal is to prove that

$$S = \{n \in \mathbb{N} \mid \exists k \in \mathbb{N} \setminus \{0\}: n = 3k\}.$$

However, this requires a little preparation.

Recursively Defined Structures

David Liben-Nowell, Connecting Discrete Mathematics and Computer Science,
manuscript at www.cs.carleton.edu/faculty/dln/book/

Linked Lists

A *linked list* is either;

- ▶ $\langle \rangle$, known as the *empty list*; or
- ▶ $\langle x, L \rangle$, where x is an arbitrary element and L is a linked list.

We can declare a version of the list type parameterised by an arbitrary type

```
data List a = Nil | Cons a (List a)
```

We can calculate the length of a list

```
len :: List a -> Int
len Nil = 0
len (Cons _ xs) = 1 + len xs
```

Recursively Defined Structures

Binary trees

A **binary tree** is either:

- ▶ the empty tree, denoted by `null`; or
- ▶ a root node x , a **left subtree** T_ℓ , and a **right subtree** T_r , where x is an arbitrary value and T_ℓ and T_r are both binary trees.

In Haskell, a type for representing such trees can be declared by

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

For example,

```
t :: Tree Int
t = Node (Node (Leaf 1) 3 (Leaf 4)) 5
      (Node (Leaf 6) 7 (Leaf 9))
```

Recursively Defined Structures

Arithmetic Expressions

An *arithmetic expression* is any of the following:

- ▶ any integer n ;
- ▶ $-E$, where E is an arithmetic expression; or
- ▶ $E \odot F$, where E and F are arithmetic expressions and $\odot \in \{+, -, \cdot, /\}$ is an *operator*.

Sentences of propositional logic

A *sentence of propositional logic* (also known as a *well-formed formula*, or *wff*) over the propositional variables X is one of the following

- ▶ x , for some $x \in X$;
- ▶ $\neg P$, where P is a wff over X ; or
- ▶ $P \vee Q$, $P \wedge Q$, or $P \rightarrow Q$, where P and Q are wffs over X .

Recursively Defined Structures

Natural numbers, recursively defined

A *natural numbers* is either:

- ▶ zero, denoted by 0; or
- ▶ the successor of a natural number n , denoted by $\text{succ}(n)$ or n^+ for a natural numbers n .

The type of natural numbers can be declared recursively,

```
data Nat = Zero | Succ Nat
```

Recursion theorem on \mathbb{N}

Theorem (Recursion on \mathbb{N})

Let A be a set, $a \in A$, and $g : A \rightarrow A$. Then there **exists** a **unique** function $f : \mathbb{N} \rightarrow A$ such that

- ▶ $f(0) = a$,
- ▶ $f(n^+) = g(f(n))$ for all $n \in \mathbb{N}$.

$$\begin{array}{ccc} \mathbb{N} & \xrightarrow{n \mapsto n^+} & \mathbb{N} \\ f \downarrow & & \downarrow f \\ A & \xrightarrow{g} & A \end{array}$$

Remark

For the existence part, see Gallier (who cites Enderton) or Halmos for two VERY different proofs.

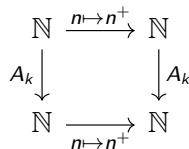
Recursion theorem on \mathbb{N}

Example

For fixed $k \in \mathbb{N}$, consider the function

$A_k : \mathbb{N} \rightarrow \mathbb{N}$ given by

- ▶ $A_k(0) := k$
- ▶ $A_k(n^+) := A_k(n)^+$



```
add :: Nat -> Nat -> Nat
add k Zero = k           -- k + 0 = 0
add k (Succ n) = Succ (add k n) -- k + (n + 1) = (k + n) + 1
```


Recursion theorem on \mathbb{N}

Example

For fixed $k \in \mathbb{N}$, consider the function $M_k : \mathbb{N} \rightarrow \mathbb{N}$ given by

- ▶ $M_k(0) := 0$
- ▶ $M_k(n^+) := A_k(M_k(n))$

$$\begin{array}{ccc} \mathbb{N} & \xrightarrow{n \mapsto n^+} & \mathbb{N} \\ M_k \downarrow & & \downarrow M_k \\ \mathbb{N} & \xrightarrow{A_k} & \mathbb{N} \end{array}$$

```
mult :: Nat -> Nat -> Nat
mult k Zero = Zero           -- k * 0 = 0
mult k (Succ n) = add k (mult k n) -- k * (n + 1) = (k * n) + k
```

Arithmetic and Order on \mathbb{N}

Definition

- ▶ $n + k := A_k(n)$
- ▶ $n \times k := M_k(n)$
- ▶ $n < m$ if $n \in m$

Natural Numbers \mathbb{N}

A natural number is either

- ▶ $0 := \{\} = \emptyset$, or
- ▶ $n + 1 = n^+ := n \cup \{n\}$, $n \in \mathbb{N}$.

Usual laws (to be verified)

- ▶ Associative law for addition: $(a + b) + c = a + (b + c)$
- ▶ Commutative law for addition: $a + b = b + a$
- ▶ Distributive law: $a(b + c) = ab + ac$
- ▶ Associative law for multiplication: $(ab)c = a(bc)$
- ▶ Commutative law for multiplication: $ab = ba$
- ▶ Order preserved by addition: $a + c < b + c$ if $a < b$
- ▶ Order preserved by multiplication: $ac < bc$ if $a < b$ and $c \neq 0$

Structural Induction

Structural induction is a useful variant of induction that allows us to prove properties for recursively defined objects.

Structural induction establishes a statement on a recursively defined set in two steps. We call those elements specifically included in the set the basis elements of the set.

1. Establish the statement for the basis elements.
2. Show that if the statement is true for each of the elements used to construct new elements in the recursive step of the definition, the statement holds for these new elements.

The justification for structural induction lies in ordinary induction, applied to the statement

$P(n)$: The claim is true for all elements of the set generated with n or fewer applications of the recursive rules for the set.

Structural induction first establishes $P(0)$ and then $P(n) \rightarrow P(n + 1)$.

Explicit Representation of Recursively Defined Sets

Example

Define $S \subset \mathbb{N}$ to be the set such that

- (i) $3 \in S$,
- (ii) $x, y \in S \rightarrow x + y \in S$.

Let

$$T = \{n \in \mathbb{N} \mid \exists k \in \mathbb{N} \setminus \{0\} \text{ such that } n = 3k\}$$

We want to show that $S = T$.

Proof.

First, we show $S \subset T$ by structural induction (on the structure of S):

base case: $3 \in S$, and $3 = 3 \cdot 1 \in T$.

inductive case: Take $x, y \in S$, assume the IH that $x, y \in T$, we want to show that $x + y \in T$. Indeed, $x, y \in T$ implies that $x = 3k$ and $y = 3k'$ for some $k, k' \in \mathbb{N} \setminus \{0\}$. Then

$$x + y = 3k + 3k' = 3(k + k')$$

so $x + y \in T$. This shows that $S \subset T$.

Explicit Representation of Recursively Defined Sets

Proof (Cont.)

Next, we show $T \subset S$, that is,

$$\forall k \in \mathbb{N} \setminus \{0\} : 3k \in S.$$

We apply (ordinary) induction on k .

base case: For $k = 1$, $3k = 3 \cdot 1 = 3 \in S$.

inductive case: Assume the IH that $3k \in S$, we want to show that $3(k+1) \in S$. Indeed, since $3 \in S$ by definition of S , thus

$$3(k+1) = 3k + 3 \in S.$$

This shows that $T \subset S$.

We finally conclude that $S = T$.



Propositional Logic Using \neg and \wedge

Example

For any logical proposition φ using the connectives $\{\neg, \wedge, \vee, \rightarrow\}$, there exists a proposition using only $\{\neg, \wedge\}$ that is logically equivalent to φ .

Proof by structural induction.

For a logical proposition φ , let $A(\varphi)$ denote the property that there exists a $\{\neg, \wedge\}$ -only proposition logically equivalent to φ . We'll prove by structural induction on φ that $A(\varphi)$ holds for any well-formed formula φ .

- ▶ **base case:** φ is a **variable**, say $\varphi = x$. No need for connectives from the set $\{\neg, \wedge\}$, $A(\varphi)$ is vacuously true.
- ▶ **inductive case I:** φ is a **negation**, say $\varphi = \neg p$. Assume the IH $A(p)$, we need to show $A(\neg p)$. By IH, there exists a $\{\neg, \wedge\}$ -only proposition $q \Leftrightarrow p$. Since $\neg q \Leftrightarrow \neg p$, and $\neg q$ contains only connectives from $\{\neg, \wedge\}$, therefore $A(\neg p)$ follows.

Propositional Logic Using \neg and \wedge

Proof by structural induction, Cont.

- **inductive case II:** φ is a conjunction, disjunction, or implication, say $\varphi = p_1 \wedge p_2$, $\varphi = p_1 \vee p_2$, $\varphi = p_1 \rightarrow p_2$. Assume IH $A(p_1)$ and $A(p_2)$, that is, there exist $\{\neg, \wedge\}$ -only propositions q_1 and q_2 such that $q_1 \Leftrightarrow p_1$ and $q_2 \Leftrightarrow p_2$. We need to show $A(p_1 \wedge p_2)$, $A(p_1 \vee p_2)$, and $A(p_1 \rightarrow p_2)$. Indeed, note that

$$p_1 \wedge p_2 \Leftrightarrow q_1 \wedge q_2$$

$$p_1 \vee p_2 \Leftrightarrow q_1 \vee q_2 \Leftrightarrow \neg(\neg q_1 \wedge \neg q_2)$$

$$p_1 \rightarrow p_2 \Leftrightarrow q_1 \rightarrow q_2 \Leftrightarrow \neg(q_1 \wedge \neg q_2)$$

Since q_1 and q_2 are $\{\neg, \wedge\}$ -only, $A(p_1 \wedge p_2)$, $A(p_1 \vee p_2)$, and $A(p_1 \rightarrow p_2)$ follow. □

Strings

A string/word is a finite sequence with entries in a finite set Σ , called alphabet, whose elements are called characters/symbols. The length of a string or word is the length of the sequence. There is a unique empty word of length 0, denoted λ or ε . A k -word over a set Σ is a sequence $w = w_1 \cdots w_k$ where $w_i \in \Sigma$ for all i .

Definition

The set Σ^* of *strings* over the alphabet Σ is defined recursively by

- ▶ $\varepsilon \in \Sigma^*$, where ε is the empty string containing no symbols.
- ▶ If $a \in \Sigma$ and $x \in \Sigma^*$, then $ax \in \Sigma^*$, where $ax := (a, x)$ is an ordered pair.

Note that $\emptyset^* = \{\varepsilon\}$.

Strings

Definition

Let Σ be a set of symbols and Σ^* the set of strings over Σ . We can define the **concatenation** of two strings, denoted by $\cdot : \Sigma^* \times \Sigma^* \rightarrow \Sigma^*$, recursively as follows.

- ▶ If $z \in \Sigma^*$, then $\varepsilon \cdot z := z$, where ε is the empty string.
- ▶ If $w, z \in \Sigma^*$ and $w = ax$, then $w \cdot z = ax \cdot z := a(x \cdot z)$.

The concatenation of the strings w_1 and w_2 is often written as the juxtaposition $w_1 w_2$ instead of $w_1 \cdot w_2$.

Recall that in Haskell we have

```
(++) :: [a] -> [a] -> [a]
[]      ++ ys = ys
(x:xs) ++ ys = x : (xs ++ ys)
```

Strings

Example

Given $abc, def \in \Sigma^*$, then

$$\begin{aligned} abc \cdot def &= a(bc \cdot def) = a(b(c \cdot def)) = a(b(c(\varepsilon \cdot def))) \\ &= a(b(c(def))) = a(b(cdef)) = a(bcdef) \\ &= abcdef \end{aligned}$$

Strings

Theorem

Given a set Σ , the triple $(\Sigma^, \cdot, \varepsilon)$ is a monoid.*

Remark

It remains to verify

- ▶ The empty string ε is also a right identity, i.e., $z \cdot \varepsilon = z$ for all $z \in \Sigma^*$.
- ▶ String concatenation is associative, i.e., $(y \cdot z) \cdot w = y \cdot (z \cdot w)$ for all $x, y, z \in \Sigma^*$.

Strings

Definition

The length of a string, $\ell : \Sigma^* \rightarrow \mathbb{N}$, $w \mapsto \ell(w)$, can be recursively defined as

- ▶ $\ell(\varepsilon) = 0$.
- ▶ $\ell(ax) = 1 + \ell(x)$ if $a \in \Sigma$ and $x \in \Sigma^*$.

We have the library function `length` given by

```
length :: [a] -> Int
length []      = 0
length (_:xs) = 1 + length xs
```

or via `foldr`

```
length :: [a] -> Int
length = foldr (\_ n -> 1+n) 0
```

Strings

Example

Given the string $ve203 \in \Sigma^*$, then

$$\begin{aligned}\ell(ve203) &= 1 + \ell(e203) = 1 + (1 + \ell(203)) = 1 + (1 + (1 + \ell(03))) \\ &= 1 + (1 + (1 + (1 + \ell(3)))) \\ &= 1 + (1 + (1 + (1 + (1 + \ell(\varepsilon))))) \\ &= 1 + (1 + (1 + (1 + (1 + 0)))) \\ &= 5\end{aligned}$$

Strings

Example

Given $x, y \in \Sigma^*$, then $\ell(x \cdot y) = \ell(x) + \ell(y)$.

Proof by structural induction.

Let $P(x)$ be the statement that $\ell(x \cdot y) = \ell(x) + \ell(y)$ whenever $y \in \Sigma^*$.

- ▶ **base case:** We show $P(\varepsilon)$ is true, i.e., $\ell(\varepsilon \cdot y) = \ell(\varepsilon) + \ell(y)$ for all $y \in \Sigma^*$. Indeed, since $\ell(\varepsilon \cdot y) = \ell(y) = 0 + \ell(y) = \ell(\varepsilon) + \ell(y)$ for any string $y \in \Sigma^*$.
- ▶ **inductive case:** Assume the IH that $P(x)$ is true, we need to show that $P(ax)$ is also true if $a \in \Sigma$, i.e., $\ell(ax \cdot y) = \ell(ax) + \ell(y)$ for all $a \in \Sigma$ and $y \in \Sigma^*$. Indeed,

$$\begin{aligned}\ell(ax \cdot y) &= \ell(a(x \cdot y)) && \text{(by definition of } \cdot \text{)} \\ &= 1 + \ell(x \cdot y) && \text{(by definition of } \ell \text{)} \\ &= 1 + \ell(x) + \ell(y) && \text{(by IH)} \\ &= \ell(ax) + \ell(y) && \text{(by definition of } \ell \text{)}\end{aligned}$$



Weak Induction as Special Case of Structural Induction

Note that natural numbers \mathbb{N} can be recursively defined, i.e., a natural number is either

- ▶ 0, or
- ▶ the successor of a natural number n , denoted by $\text{succ}(n)$ or n^+ for a natural numbers n .

Under this definition, a proof of $\forall n \in \mathbb{N} : P(n)$ by structural induction and a proof of $\forall n \in \mathbb{N} : P(n)$ are identical:

- ▶ they have precisely the same **base case**: prove $P(0)$; and
- ▶ they have precisely the same **inductive case**: prove $P(n) \rightarrow P(n^+)$, i.e., prove that $P(n) \rightarrow P(n+1)$.

Well-Ordering Principle (WOP)

Theorem (Well-Ordering Principle)

Every nonempty collection of natural numbers has a least element.

Proof by induction.

We'll prove the contrapositive, namely, that if a collection of natural numbers has no least element, then it must be empty.

Let T be such a nonempty set of natural numbers. Let $S = \mathbb{N} \setminus T$. We need to show that $n \in S$ for all $n \in \mathbb{N}$.

- ▶ **base case** ($n = 0$): If $0 \in T$, then 0 is the least element. So $0 \notin T$, i.e., $0 \in S$.
- ▶ **inductive case** ($n \geq 1$): Suppose $n \in S$. If any of the numbers less than n were in T , then one of them would be least (since there are only finitely many such numbers and every finite set has a least element²). So none of the numbers $0, 1, \dots, n$ is in T . If $n + 1 \in T$, then $n + 1$ would be least. So $n + 1 \notin T$, i.e., $n + 1 \in S$. □

2. can be proven using induction

Prime Factorization

Recall

Theorem (Fundamental Theorem of Arithmetic)

Let $n \in \mathbb{N} \setminus \{0\}$, then there exist $k \geq 0$ prime numbers p_1, p_2, \dots, p_k such that $n = \prod_{i=1}^k p_i$. (also unique up to order, more on this later.)

Proof by WOP.

Let T be the set of natural numbers greater than 1 which cannot be written as the product of primes. By WOP, T has a least element n .

Clearly n cannot be prime, so n is composite. Then we can write $n = ab$, where neither of a and b is 1. So $a < n$ and $b < n$. If both a and b had prime factorizations, then so would n . Therefore at least one of a and b does not have a prime factorization (by relabeling, we can assume it is a). But $a < n$ and $a \in T$, so n was not least in T .

This contradiction establishes that T has no least element, hence by WOP must be empty. So every natural number greater than 1 can be written as the product of primes. □

Back to The Beginning

Recall the claim that $\sum_{k=0}^n k = \frac{n(n+1)}{2}$ for all $n \in \mathbb{N}$.

Proof by WOP.

Suppose not. Then there exists some $p \in \mathbb{N}$ such that $\sum_{k=0}^p k \neq \frac{p(p+1)}{2}$.

Consider the set of all such numbers:

$$T = \left\{ k \mid 1 + \dots + k \neq \frac{k(k+1)}{2} \right\}$$

Thus $p \in T$ by assumption; in particular $T \neq \emptyset$.

By WOP, T has a least element N . $N \neq 0$ since $0 = 0(0+1)/2$. Therefore $N > 1$. But then N admits a predecessor, $n = N - 1$. Since $n < N$, then $n \notin T$ (b/c N is **least** in T). That is, $\sum_{k=0}^n k = \frac{n(n+1)}{2}$. Now consider

$$\sum_{k=0}^N k = \left(\sum_{k=0}^n k \right) + (n+1) = \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2} = \frac{N(N+1)}{2},$$

which show that $N \notin T$, contradiction! So the initial supposition was incorrect, and thus the claim is true. □

Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Motivation

Primitive Models

- ▶ equivalence relations ($=$, \cong , \equiv , \sim , etc.)
- ▶ partial orders (\leq , \subset , \preceq , $|$, etc.)

Relations

Definition

A subset $R \subset A \times B$ is called a (binary) **relation** from A to B . The **domain** and **range** of R is given by

► $\text{domain}(R) := \{x \mid \exists y(xRy)\}$

► $\text{range}(R) := \{y \mid \exists x(xRy)\}$

If $A = B$, we say that R is a **relation on A** .

Notation

► $(x, y) \in R \Leftrightarrow xRy.$

► $(x, y) \notin R \Leftrightarrow x \not R y.$

► via predefined symbols, e.g.,

► $x \preceq y$, i.e., $(x, y) \in \preceq \subset A \times B$;

► $x \sim y$, i.e., $(x, y) \in \sim \subset A \times B$.

Relations

Examples

Suppose $R \subset A \times B$.

- ▶ $R = \emptyset$, the **empty relation**, with $\text{domain}(\emptyset) = \text{range}(\emptyset) = \emptyset$.
- ▶ When $A = B$, we have the **identity relation**,

$$\text{id}_A = \{(a, a) \mid a \in A\}$$

The identity relation relates every element to itself. Note that $\text{domain}(\text{id}_A) = \text{range}(\text{id}_A) = A$.

- ▶ The relation $A \times B$ itself. This relation relates every element of A to every element of B . Note that $\text{domain}(A \times B) = A$ and $\text{range}(A \times B) = B$.

Functions

Definition

A **function** $F : A \rightarrow B$ is a triple (A, F, B) , where $F \subset A \times B$ is a relation such that

$$(\forall x \in A)(\exists! y(xFy))$$

where A, B are the **domain** and **codomain** of F .

Remark

- ▶ Given function $F : A \rightarrow B$, then $(\forall x, y \in A)(x = y \rightarrow F(x) = F(y))$.
- ▶ For a function F and $x \in \text{dom}(F)$, the unique y such that xFy is called the **value** of F at x and is denoted $F(x)$. Thus $(x, F(x)) \in F$.
- ▶ Range/Image of a function is sometimes hard to characterize explicitly.
- ▶ The set of all function from A to B is denoted B^A , i.e., $B^A = \{f : A \rightarrow B\}$.

Partial Functions

Definition

A relation $R \subset A \times B$ is said to be *functional* (or *right-unique* etc.) if

$$(\forall x \in A)(\exists_{\leq 1} y(xFy))$$

or equivalently

$$(\forall x \in \text{dom } F)(\exists! y(xFy))$$

A *partial function* is a triple (A, F, B) such that $F \subset A \times B$ is a functional relation. Note that a (total) function is a partial function which is defined for all elements in the domain.

Remark

- ▶ Partial function: not (necessarily) everywhere defined.
- ▶ (Total) function: everywhere defined.

Strings/Words/Lists

Example

Let Σ be a finite set. A string/word/list over Σ of length n is any function $u : [n] \rightarrow \Sigma$. The integer n is the length of the string u . When $n \geq 1$, we denote the string u as

$$u = u_1 u_2 \cdots u_n$$

where each $u_i \in \Sigma$.

- ▶ The empty string, or null string, is the unique function $u : [0] \rightarrow \Sigma$.
- ▶ Given two strings $u : [m] \rightarrow \Sigma$ and $v : [n] \rightarrow \Sigma$, the concatenation $u \cdot v$ (or simply uv) is the string $uv : [m+n] \rightarrow \Sigma$ given by

$$uv(i) = \begin{cases} u(i), & \text{if } 1 \leq i \leq m, \\ v(i-m), & \text{if } m+1 \leq i \leq m+n. \end{cases}$$

Multiset

Definition

Given any set S , a *multiset M over S* is any function $M : S \rightarrow \mathbb{N}$.

- ▶ A *finite multiset M over S* is any function $M : S \rightarrow \mathbb{N}$ such that $M(a) \neq 0$ only for finitely many $a \in S$.
- ▶ If $M(a) = k > 0$, we say that *a appears with multiplicity k in M* .
- ▶ The *empty multiset* is the constant function $M \equiv 0$.

Operations on Relations/Functions

For arbitrary sets/relations/functions A , F , and G ,

- ▶ The **inverse** of F is the set

$$F^{\top} = F^{-1} = \{(y, x) \mid xFy\}$$

- ▶ The **composition** of F and G is the set (beware of the order)

$$G \circ F = F \circ G = \{(x, z) \mid \exists y (xGy \wedge yFz)\}$$

- ▶ The **restriction** of F to A is the set

$$F|A = \{(x, y) \mid (xFy) \wedge (x \in A)\}$$

- ▶ The **image** of A **under** F is the set

$$F(A) = \text{im}(F|A) = \{y \mid (\exists x \in A)(xFy)\}$$

If F is a function, then $F(A) = \{F(x) \mid x \in A\}$.

Matrix Representation of a Relation

Given **finite** sets A and B , where $A = \{a_1, a_2, \dots, a_m\}$, $B = \{b_1, b_2, \dots, b_n\}$, and a relation $R \subset A \times B$, we can represent R by the $m \times n$ matrix $M_R = (m_{ij})$, where

$$m_{ij} = \begin{cases} 1, & \text{if } (a_i, b_j) \in R \\ 0, & \text{if } (a_i, b_j) \notin R \end{cases}$$

We call M_R the matrix of R . Conversely, given finite sets A and B with $|A| = m$ and $|B| = n$, an $m \times n$ 0-1 matrix determines a relation.

Example

Let $A = \{1, 2, 3\}$, $B = \{r, s\}$, then the relation $R = \{(1, r), (2, s), (3, r)\}$ can be represented by the matrix

$$M_R = \begin{matrix} & \begin{matrix} r & s \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \end{matrix} & \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \end{matrix}$$

$$\begin{array}{c|cc} + & 0 & 1 \\ \hline 0 & 0 & 1 \\ 1 & 1 & 1 \end{array}$$

$$\begin{array}{c|cc} \times & 0 & 1 \\ \hline 0 & 0 & 0 \\ 1 & 0 & 1 \end{array}$$

Composing Relations

Let A, B, C be sets, and two (binary) relations, $R \subset A \times B$, and $S \subset B \times C$. Then $M_R M_S = M_{R \circ S} = M_{S \circ R}$.

Example

Given relations $R = \{(1, 3), (2, 3)\}$ and $S = \{(3, 1)\}$ on $\{1, 2, 3\}$, then

$$M_R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad M_S = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

Take the scalar addition and multiplication as \vee and \wedge , thus

$$M_R M_S = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad M_S M_R = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

It is easy to verify that $R \circ S = S \circ R = \{(1, 1), (2, 1)\}$ and $S \circ R = R \circ S = \{(3, 3)\}$.

Composing Relations

Theorem

Given a set A , the triple $(\mathcal{P}(A \times A), \circ, \text{id}_A)$ is a monoid.

Compositions Are Important!

Definition

A **category** \mathcal{C} consists of

- ▶ a collection $\text{Ob}(\mathcal{C})$, whose elements are called **objects**;
- ▶ for any two objects $c, d \in \text{Ob}(\mathcal{C})$, a hom-set from $\text{Hom}_{\mathcal{C}}(c, d)$, whose elements are called **morphisms**, **maps**, or **arrows** from c to d ;
- ▶ for every $c \in \text{ob}(\mathcal{C})$, there is a identity morphism $\text{id}_c \in \text{Hom}_{\mathcal{C}}(c, c)$;
- ▶ for any three objects $c, d, e \in \text{Ob}(\mathcal{C})$, and morphisms $f \in \text{Hom}_{\mathcal{C}}(c, d)$ and $g \in \text{Hom}_{\mathcal{C}}(d, e)$, there is a morphism $f \circ g \in \text{Hom}_{\mathcal{C}}(c, e)$.

such that

- ▶ unitality: for any $f : c \rightarrow d$, i.e., $f \in \text{Hom}_{\mathcal{C}}(c, d)$,

$$\text{id}_c \circ f = f \circ \text{id}_d = f$$

- ▶ For any $f : c_0 \rightarrow c_1$, $g : c_1 \rightarrow c_2$, and $h : c_2 \rightarrow c_3$,

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

Injection and Surjection

Definition

Given a function $F : A \rightarrow B$, with $\text{dom } F = A$ and $\text{im}(F) \subset B$, then

- ▶ F is **injective** or **one-to-one** if $(\forall x, y \in A)(F(x) = F(y) \rightarrow x = y)$.
- ▶ F is **surjective** or **onto** if $\text{im}(F) = B$.
- ▶ F is **bijective** if it is both injective and surjective.

The above function F is also called an injection, surjection, or bijection, respectively.

Example

Given a vector space V over \mathbb{R} , and fix $v_1, \dots, v_n \in V$, define the function $X : \mathbb{R}^n \rightarrow V$, $(a_1, a_2, \dots, a_n) \mapsto a_1 v_1 + a_2 v_2 + \dots + a_n v_n$, then the set $\{v_1, v_2, \dots, v_n\} \subset V$

- ▶ **spans/generates** V iff X is surjective.
- ▶ is **linearly independent** iff X is injective.
- ▶ is a **basis** iff X is both injective and surjective.

Injection and Surjection

Theorem/Definition

Given a function $F : A \rightarrow B$, $A \neq \emptyset$, then

- ▶ There exists a function $G : B \rightarrow A$ (a “left inverse”) such that $G \circ F = \text{id}_A$
 $\Leftrightarrow F$ is **one-to-one/injective**.
- ▶ There exists a function $G : B \rightarrow A$ (a “right inverse”) such that
 $F \circ G = \text{id}_B \Leftrightarrow F$ is **onto/surjective**.

Injection and Surjection

Theorem

Given $f : A \rightarrow B$, $g : B \rightarrow C$, $A \neq \emptyset$, then

- ▶ If $g \circ f$ is injective, then f is injective.
- ▶ If $g \circ f$ is surjective, then g is surjective.

Proof.

- ▶ Given $x, y \in A$, then

$$\begin{aligned} f(x) = f(y) &\Rightarrow g(f(x)) = g(f(y)) \Rightarrow (g \circ f)(x) = (g \circ f)(y) \\ &\Rightarrow x = y \text{ (b/c } g \circ f \text{ is injective)} \end{aligned}$$

Therefore f is injective.

- ▶ Since $g \circ f$ is surjective, then $\forall z \in C$, $\exists x \in A$ such that $g \circ f(x) = g(f(x)) = z$. Let $y = f(x) \in B$ (which exists for all $x \in A$ since f is a function), then $\forall z \in C$, $\exists y \in B$ such that $z = f(y)$. Therefore g is surjective. □

Injection and Surjection

Theorem

Given $f : A \rightarrow B$, $g : B \rightarrow C$, $A \neq \emptyset$, then

- ▶ If $g \circ f$ is injective, then f is injective.
- ▶ If $g \circ f$ is surjective, then g is surjective.

Proof (Alternative).

- ▶ Since $g \circ f$ is injective, then there exists $h : C \rightarrow A$ such that $h \circ (g \circ f) = id_A$, that is, there exists $h \circ g : C \rightarrow A$ such that $(h \circ g) \circ f = id_A$. Therefore f is left invertible, hence injective.
- ▶ Since $g \circ f$ is surjective, then there exists $h : C \rightarrow A$ such that $(g \circ f) \circ h = id_C$, that is, there exists $f \circ h : C \rightarrow A$ such that $g \circ (f \circ h) = id_C$. Therefore g is right invertible, hence surjective.



Relations as Functions

Relation as multivariable boolean functions

Example

- ▶ Given a relation $R \subset A \times B$, the associated boolean function is given by

$$\begin{aligned}\phi_R : A \times B &\rightarrow \{\top, \perp\} \\ (x, y) &\mapsto \begin{cases} \top, & xRy \\ \perp, & \text{otherwise} \end{cases}\end{aligned}$$

- ▶ Given a boolean function $\phi : A \times B \rightarrow \{\top, \perp\}$, the associated relation is given by

$$R_\phi = \{(x, y) \in A \times B \mid \phi(x, y) = \top\}$$

Relations as Functions

Relation as set functions

Example

- ▶ Given a relation $R \subset A \times B$, the associated set function is given by

$$\begin{aligned}\alpha_R : A &\rightarrow \mathcal{P}(B) \\ x &\mapsto \{y \in B \mid xRy\}\end{aligned}$$

- ▶ Given a set function $\alpha : A \rightarrow \mathcal{P}(B)$, the associated relation is given by

$$R_\alpha = \{(x, y) \in A \times B \mid y \in \alpha(x)\}$$

Relations as Functions

Relation as set functions

Example

- ▶ Given a relation $R \subset A \times B$, the associated set function is given by

$$\begin{aligned}\beta_R : B &\rightarrow \mathcal{P}(A) \\ x &\mapsto \{y \in A \mid xRy\}\end{aligned}$$

- ▶ Given a set function $\beta : B \rightarrow \mathcal{P}(A)$, the associated relation is given by

$$R_\beta = \{(x, y) \in A \times B \mid y \in \beta(x)\}$$

Properties of Relations

Definition

A (binary) relation R on A , i.e., $R \subset A \times A$, is

- ▶ **reflexive** if $(\forall x \in A)(xRx)$.
- ▶ **irreflexive** if $(\forall x \in A)(xRx \rightarrow \perp)$.
- ▶ **strongly connected** or **total**³ if $(\forall x, y \in A)(xRy \vee yRx)$.
- ▶ **transitive** if $(\forall x, y, z \in A)(xRy \wedge yRz \rightarrow xRz)$.
- ▶ **symmetric** if $(\forall x, y \in A)(xRy \rightarrow yRx)$.
- ▶ **anti-symmetric** if $(\forall x, y \in A)(xRy \wedge yRx \rightarrow x = y)$.
- ▶ **asymmetric** if $(\forall x, y \in A)(xRy \wedge yRx \rightarrow \perp)$.

3. there are many other names

Properties of Relations

Theorem

Let R be a relation on A , then

- ▶ *R is reflexive iff $\text{id}_A \subset R$.*
- ▶ *R is symmetric iff $R = R^{-1}$.*
- ▶ *R is antisymmetric iff $R \cap R^{-1} \subset \text{id}_A$.*
- ▶ *R is transitive iff $R \circ R \subset R$.*

Properties of Relations

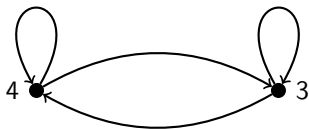
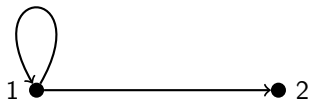
	<i>reflexive</i>	<i>transitive</i>	<i>symmetric</i>	<i>antisymmetric</i>
\leq on \mathbb{R}				
$<$ on \mathbb{R}				
\subset on 2^S				
\subsetneq on 2^S				
\equiv_n on \mathbb{Z}				
$ $ on $\mathbb{N} \setminus \{0\}$				
$ $ on \mathbb{N}				
$ $ on \mathbb{Z}				
matrix similarity				

Note that $a \equiv_n b$ if $n \mid a - b$, and we use the convention that $0 \mid 0$.

Properties of Relations

Example

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



Important Relations

	irrefl.	refl.	asym.	antisymm.	symm.	trans.	total
pre-order		✓				✓	
strict partial order	✓		✓	✓		✓	
(non-strict) partial order		✓		✓		✓	
total order		✓		✓		✓	✓
equivalence relation		✓			✓	✓	

Important Relations

Definition

A **partial order** on a set P is a relation that is

- ▶ reflexive
- ▶ antisymmetric
- ▶ transitive

Example

- ▶ On \mathbb{Z} (or \mathbb{R} etc.): $a \leq b$
- ▶ On 2^S for a given S : $A \subset B$
- ▶ On \mathbb{N} : $a \mid b$
- ▶ On partitions: refinement

Irreflexive Relations

Recall that a relation R on a set A is **irreflexive** if $aRa \rightarrow \perp$ for all $a \in A$.

Remark

- ▶ Irreflexive \neq non-reflexive
 - ▶ Irreflexive: zero self-loops.
 - ▶ Non-reflexive: missing self-loops.
- ▶ Antisymmetric \neq non-symmetric
 - ▶ Antisymmetric: no cycle of length 2.
 - ▶ Non-symmetric: exists directed edge of no return.

Terminologies on Partial Orders

Non-strict Partial Order (e.g., \leq , \subseteq)

A **reflexive**, **weak**, or **non-strict** partial order is a relation \preceq over a set A that is

- ▶ reflexive
- ▶ antisymmetric
- ▶ transitive

Strict Partial Order (e.g., $<$, \subsetneq)

An **irreflexive**, **strong**, or **strict** partial order is a relation \prec over a set A that is

- ▶ irreflexive
- ▶ asymmetric
- ▶ transitive

Remark

- ▶ For every strict partial order there is a unique corresponding non-strict partial order, and vice-versa. The two differ by the identity relation id_A .
- ▶ The term **partial order** typically refers to a non-strict partial order relation.

Important Relations

Definition

An **equivalence relation** on a set A is a relation that is

- ▶ reflexive
- ▶ symmetric
- ▶ transitive

Example

- ▶ On \mathbb{Z} (or \mathbb{R} etc.): $a = b$
- ▶ On \mathbb{Z} : $a \equiv b \pmod{12}$ (iff $12 \mid a - b$)
- ▶ On 2^S for given S : $A \equiv B$ iff $|A| = |B|$
- ▶ On square matrices: $A \cong B$ iff $A = PBP^{-1}$

Equivalence Classes

Definition

Given an equivalence relation R on A , the **equivalence class** containing x is the set

$$[x]_R := \{t \in A \mid xRt\}$$

Theorem

Given an equivalence relation R on A , then for $x, y \in A$,

$$[x]_R = [y]_R \Leftrightarrow xRy$$

Proof.

(\Rightarrow) Let $[x]_R = [y]_R$, then $y \in [y]_R$ (b/c yRy) implies $y \in [x]_R$ (b/c $[x]_R = [y]_R$), hence xRy (by definition of $[x]_R$).

Equivalence Classes

Proof.

(\Leftarrow) Assume xRy , then take any $t \in [y]_R$,

$$\begin{aligned} t \in [y]_R &\Rightarrow yRt \\ &\Rightarrow xRt \quad (xRy \text{ and transitivity}) \\ &\Rightarrow t \in [x]_R \end{aligned}$$

hence $[y]_R \subset [x]_R$. The other direction is by symmetry of R .



Partition

Definition

A **partition** Π of a set A is a set of nonempty subsets of A that is disjoint and exhaustive, i.e.,

- ▶ $(\forall a, b \in \Pi)(a \neq b \rightarrow a \cap b = \emptyset);$
- ▶ $\bigcup \Pi = A.$

An element of a partition is called a **fiber**, a **block**, or an **equivalence class**.

An element of such an equivalence class is called a **representative** of the class.

Examples

- ▶ $A = \emptyset: \Pi = \emptyset.$
- ▶ $A \neq \emptyset: \Pi = \{\{x\} \mid x \in A\},$ or $\Pi = \{A\}.$
- ▶ $A = \mathbb{N}: \Pi = \{\{\text{even numbers}\}, \{\text{odd numbers}\}\}$

Partition

Theorem

Given an equivalence relation R on A , then the set $\{[x]_R \mid x \in A\}$ of all equivalence classes is a partition of A .

Proof.

- ▶ $[x]_R \neq \emptyset \forall x$. (b/c $x \in [x]_R$)
- ▶ $\bigcup \{[x]_R \mid x \in A\} = A$. (b/c $x \in [x]_R \subset A$)
- ▶ Suppose $t \in [x]_R \cap [y]_R$, then tRx and tRy , hence xRy and $[x]_R = [y]_R$.



Quotient set

Definition

Given an equivalence relation R on A , then the **quotient set** is given by

$$A/R := \{[x]_R \mid x \in A\}$$

where A/R is read “ A modulo R ”.

Example

Let $\mathbb{N} = \{0, 1, 2, 3, \dots\}$, and define the relation \sim on \mathbb{N} by

$$m \sim n \iff 6 \mid m - n, \text{ i.e., } m - n \text{ divisible by } 6$$

then $\mathbb{N}/\sim = \{[0], [1], [2], [3], [4], [5]\}$, where explicitly

$$[0] := \{0, 6, 12, \dots\},$$

$$[1] := \{1, 7, 13, \dots\},$$

$$[2] := \{2, 8, 14, \dots\},$$

$$[3] := \{3, 9, 15, \dots\},$$

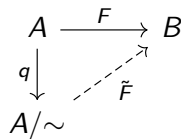
$$[4] := \{4, 10, 16, \dots\},$$

$$[5] := \{5, 11, 17, \dots\}.$$

Quotient set

Example

Let $F : A \rightarrow B$ and for elements in A define $x \sim y \Leftrightarrow F(x) = F(y)$. The relation \sim is an equivalence relation on A . There is a **unique one-to-one** function $\tilde{F} : A/\sim \rightarrow B$ such that $F = \tilde{F} \circ q$, where $q : A \rightarrow A/\sim$, $x \mapsto [x]$, is the natural quotient map. The value of \tilde{F} at a particular equivalence class is the common value at the member of that equivalence class.



- \tilde{F} is well-defined, i.e., \tilde{F} is indeed a function. Pick $[x], [y] \in A/\sim$,

$$[x] = [y] \Rightarrow x \sim y \Rightarrow F(x) = F(y) \Rightarrow \tilde{F}([x]) = \tilde{F}([y])$$

- \tilde{F} is injective. $\tilde{F}([x]) = \tilde{F}([y]) \Leftrightarrow F(x) = F(y) \Leftrightarrow x \sim y \Leftrightarrow [x] = [y]$.
- q is surjective. (b/c $x \in [x] \ \forall x \in A$)
- \tilde{F} is unique. $\tilde{F} \circ q = \tilde{G} \circ q \Rightarrow \tilde{F} = \tilde{G}$.

Note that $\tilde{F} : A/\sim \rightarrow \text{im}(F)$, $[x] \mapsto F(x)$, is bijective.

Quotient set

Definition

A **monoid homomorphism**, or **monoid morphisms**, between two monoids $(M, *, e_M)$ and (N, \cdot, e_N) is a function $f : M \rightarrow N$ such that

- ▶ $f(x * y) = f(x) \cdot f(y)$ for all $x, y \in M$, and
- ▶ $f(e_M) = e_N$.

Example

Given a monoid homomorphism $f : M \rightarrow N$, consider the **kernel** (or **kernel pair**) of f given by

$$\ker f := \{(m, m') \in M \times M \mid f(m) = f(m')\}$$

Thus $\ker f$ is an equivalence relation. Furthermore, it is a **congruence relation**. That is, consider the congruence class

$$[a] := \{x \in M \mid (x, a) \in \ker f\}$$

then $(M / \ker f, \circ, [e_M])$ is a monoid, where $[a] \circ [b] = [ab]$.

Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Integers

Definition

Let \sim be the following equivalence relation on \mathbb{N}^2 ,

$$(a, b) \sim (c, d) \iff a +_{\mathbb{N}} d = b +_{\mathbb{N}} c$$

Explicitly, note that $\sim \subset \mathbb{N}^2 \times \mathbb{N}^2$, and

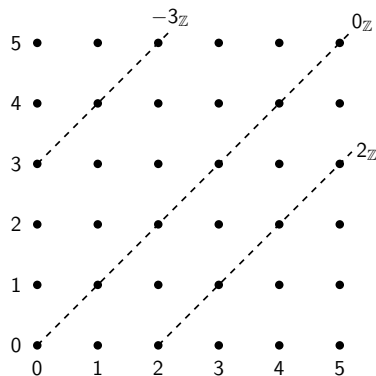
$$\sim = \{((a, b), (c, d)) \in \mathbb{N}^2 \times \mathbb{N}^2 \mid a, b, c, d \in \mathbb{N}, a +_{\mathbb{N}} d = b +_{\mathbb{N}} c\}$$

Define $\mathbb{Z} = \mathbb{N} \times \mathbb{N} / \sim$.

Integers

Example

- ▶ $0_{\mathbb{Z}} = \{(0_{\mathbb{N}}, 0_{\mathbb{N}}), (1_{\mathbb{N}}, 1_{\mathbb{N}}), \dots\} = [(0_{\mathbb{N}}, 0_{\mathbb{N}})] = [\{\{\emptyset\}\}] = [\{\{\{\}\}\}]$.
- ▶ $2_{\mathbb{Z}} = \{(2_{\mathbb{N}}, 0_{\mathbb{N}}), (3_{\mathbb{N}}, 1_{\mathbb{N}}), (4_{\mathbb{N}}, 2_{\mathbb{N}}), \dots\} = [(2_{\mathbb{N}}, 0_{\mathbb{N}})]$.
- ▶ $-3_{\mathbb{Z}} = \{(0_{\mathbb{N}}, 3_{\mathbb{N}}), (1_{\mathbb{N}}, 4_{\mathbb{N}}), (2_{\mathbb{N}}, 5_{\mathbb{N}}), \dots\} = [(0_{\mathbb{N}}, 3_{\mathbb{N}})]$.



Integers

Remark

The relation \sim is an equivalence relation on $\mathbb{N} \times \mathbb{N}$.

Proof.

Let $(a, b), (a', b'), (a'', b'') \in \mathbb{N} \times \mathbb{N}$.

- ▶ Reflexivity. It is clear that $(a, b) \sim (a, b)$, since $a + b = b + a$.
- ▶ Symmetry. Let $(a, b) \sim (a', b')$, then

$$\begin{aligned}(a, b) \sim (a', b') &\Rightarrow a + b' = a' + b \\ &\Rightarrow a' + b = a + b' \Rightarrow (a', b') \sim (a, b)\end{aligned}$$

- ▶ Transitivity. Let $(a, b) \sim (a', b')$ and $(a', b') \sim (a'', b'')$, then we have $a + b' = a' + b$ and $a' + b'' = a'' + b'$. Therefore

$$a + b' + a' + b'' = a' + b + a'' + b'$$

hence $a + b'' = a'' + b$, i.e., $(a, b) \sim (a'', b'')$.



Integers

Well-defined operations on equivalence classes

- ▶ $[(a, b)] +_{\mathbb{Z}} [(c, d)] := [(a + c, b + d)].$
- ▶ $[(a, b)] \times_{\mathbb{Z}} [(c, d)] := [(ac + bd, ad + bc)].$
- ▶ $[(a, b)] <_{\mathbb{Z}} [(c, d)]$ if $a + d <_{\mathbb{N}} b + c.$

Proof.

We want to show that the operations do not depend on the choice of representatives. choose $(a, b) \sim (a', b')$ and $(c, d) \sim (c', d')$, which indicates that $a + b' = a' + b$ and $c + d' = c' + d$, then

- ▶ We want to show $a + c + b' + d' = a' + b' + c + d$. Since
$$(a + c) + (b' + d') = (a + b') + (c + d')$$
$$= (a' + b) + (c' + d) = (a' + c') + (b + d)$$

Hence $(a + c, b + d) \sim (a' + c', b' + d')$, and

$[(a + c, b + d)] = [(a' + c', b' + d')]$, independent of the choice of representatives.

Integers

Proof (Cont.)

- We want to show $ac + bd + a'd' + b'c' = a'c' + b'd' + ad + bc$. Note that $a + b' = a' + b$ and $c + d' = c' + d$, then

$$c(a + b') = c(a' + b)$$

$$d(a' + b) = d(a + b')$$

$$a'(c + d') = a'(c' + d)$$

$$b'(c' + d) = b'(c + d')$$

which simplifies to

$$ac + b'c = a'c + bc$$

$$a'd + bd = ad + b'd$$

$$a'c + a'd' = a'c' + a'd$$

$$b'c' + b'd = b'c + b'd'$$

Add all above together and cancel the unwanted terms.

Integers

Proof (Cont.)

- We want to show that $a + d < b + c$ iff $a' + d' < b' + c'$. Recall that $a + b' = a' + b$ and $c + d' = c' + d$, then

$$\begin{aligned}a + d < b + c &\Leftrightarrow a + d + b' + d' < b + c + b' + d' \\&\Leftrightarrow a' + b + d + d' < b + b' + c' + d \\&\Leftrightarrow a' + d' < b' + c'\end{aligned}$$

Therefore the ordering $<_{\mathbb{Z}}$ on \mathbb{Z} is well-defined.



Rational Numbers

Definition

Let $\mathbb{Z}^+ = \{z \in \mathbb{Z} \mid z >_{\mathbb{Z}} 0_{\mathbb{Z}}\}$, and let \sim be the following equivalence relation on $\mathbb{Z} \times \mathbb{Z}^+$,

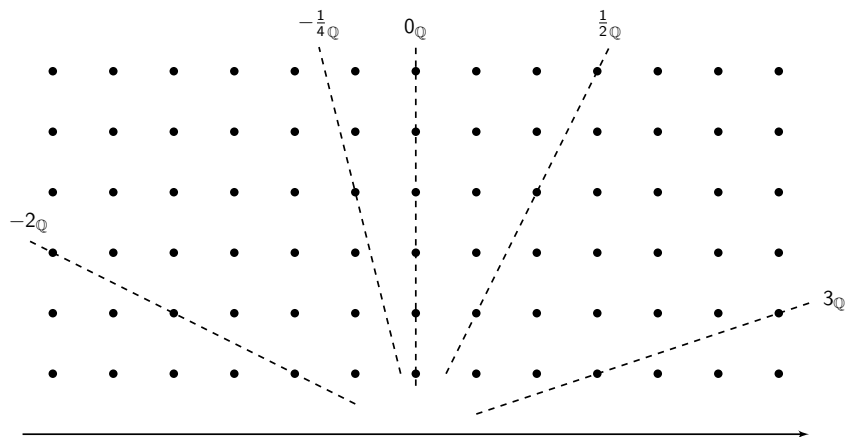
$$(a, b) \sim (c, d) \iff a \times_{\mathbb{Z}} d = b \times_{\mathbb{Z}} c$$

Explicitly, note that $\sim \subset (\mathbb{Z} \times \mathbb{Z}^+) \times (\mathbb{Z} \times \mathbb{Z}^+)$, and

$$\sim = \left\{ ((a, b), (c, d)) \in (\mathbb{Z} \times \mathbb{Z}^+)^2 \mid \begin{array}{l} a, c \in \mathbb{Z}, b, d \in \mathbb{Z}^+, \\ a \times_{\mathbb{Z}} d = b \times_{\mathbb{Z}} c \end{array} \right\}$$

Define $\mathbb{Q} := \mathbb{Z} \times \mathbb{Z}^+ / \sim$.

Rational Numbers



Rational Numbers

Remark

The relation \sim is an equivalence relation on $\mathbb{Z} \times \mathbb{Z}^+$.

Proof.

Let $(a, b), (a', b'), (a'', b'') \in \mathbb{Z} \times \mathbb{Z}^+$.

- ▶ Reflexivity. Obviously $(a, b) \sim (a, b)$ since $ab = ba$.
- ▶ Symmetry. Let $(a, b) \sim (a', b')$, then

$$(a, b) \sim (a', b') \Rightarrow ab' = a'b \Rightarrow a'b = ab' \Rightarrow (a', b') \sim (a, b)$$

- ▶ Transitivity. Let $(a, b) \sim (a', b')$ and $(a', b') \sim (a'', b'')$, then we have $ab' = a'b$ and $a'b'' = a''b'$. Thus $ab'a'b'' = a'ba''b'$. Since $b' \neq 0$, then $aa'b'' = a'ba''$.
 - ▶ If $a' \neq 0$, then $ab'' = a''b$, i.e., $(a, b) \sim (a'', b'')$.
 - ▶ If $a' = 0$, then $ab' = 0$, hence $a = 0$ ($b/c \ b' \neq 0$). Similarly $a'' = 0$. Therefore $ab'' = 0 = a''b$, i.e., $(a, b) \sim (a'', b'')$. □

Rational Numbers

Well-defined operations on equivalence classes

- ▶ $[(a, b)] +_{\mathbb{Q}} [(c, d)] := [(a \times_{\mathbb{Z}} d +_{\mathbb{Z}} b \times_{\mathbb{Z}} c, b \times_{\mathbb{Z}} d)].$
- ▶ $[(a, b)] \times_{\mathbb{Q}} [(c, d)] := [(a \times_{\mathbb{Z}} c, b \times_{\mathbb{Z}} d)].$
- ▶ $[(a, b)] <_{\mathbb{Q}} [(c, d)]$ if $a \times_{\mathbb{Z}} d <_{\mathbb{Z}} b \times_{\mathbb{Z}} c.$

Proof.

Choose $(a, b) \sim (a', b')$ and $(c, d) \sim (c', d')$, which indicates that $ab' = a'b$ and $cd' = c'd$, with $b, b', d, d' > 0$.

- ▶ We want to show that $(ad + bc, bd) \sim (a'd' + b'c', b'd')$, that is, $(ad + bc)b'd' = bd(a'd' + b'c')$, or $ab'dd' + bb'cd' = a'bdd' + bb'c'd$. Note that this is guaranteed by $ab' = a'b$ and $cd' = c'd$.
- ▶ Since $ab' = a'b$ and $cd' = c'd$, then $ab'cd' = a'bc'd$, thus $(ac, bd) \sim (a'c', b'd')$.
- ▶ $ad < bc \Leftrightarrow adb'd' < bcb'd' \Leftrightarrow a'bdd' < bb'c'd \Leftrightarrow a'd' < b'c'$ (note that $b', d' > 0$). Therefore the ordering $<_{\mathbb{Z}}$ on \mathbb{Q} is well-defined. □

Real Numbers

Definition

A **Cauchy sequence** in \mathbb{Q} is a function $s : \mathbb{N} \rightarrow \mathbb{Q}$ such that $|s_m - s_n|$ is arbitrarily small for all sufficiently large m and n ; i.e.,

$$(\forall \varepsilon \in \mathbb{Q}, \varepsilon > 0)(\exists k \in \mathbb{N})(\forall m, n > k)(|s_m - s_n| < \varepsilon)$$

Definition (Cantor)

Let C be the set of all Cauchy sequences. For $r, s \in C$, then r and s are **equivalent** ($r \sim s$) if $|r_n - s_n|$ is arbitrarily small for all sufficiently large n ; i.e.,

$$(\forall \varepsilon \in \mathbb{Q}, \varepsilon > 0)(\exists k \in \mathbb{N})(\forall n > k)(|r_n - s_n| < \varepsilon)$$

Define \mathbb{R} to be the quotient set C/\sim .

Example

- ▶ $1 = \{(1, 1, 1, \dots), (\sum_{k=1}^n 9/10^k) = (0.9, 0.99, 0.999, \dots), \dots\}$
- ▶ $e = \{(2, 2.7, 2.71, \dots), (\sum_{k=0}^n 1/k!), ((1 + 1/n)^n), \dots\}$

Real Numbers

Definition

A **Dedekind cut** is a set $x \subset \mathbb{Q}$ such that

- ▶ $x \neq \emptyset$ and $x \neq \mathbb{Q}$;
- ▶ x is closed downward, i.e., $(\forall p, q \in \mathbb{Q})(p < q \rightarrow (q \in x \rightarrow p \in x))$;
- ▶ x has no largest element.

Define \mathbb{R} to be the set of all Dedekind cuts.

Definition

Let $x \leq_{\mathbb{R}} y$ if $x \subset y$.

Theorem

Every subset $A \subset \mathbb{R}$ that is bounded above admits a least upper bound.

Proof.

Take $x = \bigcup A$. Claim: x is a Dedekind cut, and $(\forall y \in A)(y \leq x)$. □

Floor Function and Integer Division

Floor Function

Given $r \in \mathbb{R}$, consider the floor function $\lfloor r \rfloor := \max\{n \in \mathbb{Z} \mid n \leq r\}$. It is easy to verify that for all $n \in \mathbb{Z}$ and $r \in \mathbb{R}$,

$$n \leq \lfloor r \rfloor \Leftrightarrow n \leq r$$

Theorem

Given $a, b, c \in \mathbb{N} \setminus \{0\}$, $\lfloor \lfloor a/b \rfloor / c \rfloor = \lfloor a/bc \rfloor$.

Proof.

Consider $c \in \mathbb{N} \setminus \{0\}$, $r \in \mathbb{Q}$, then for any $n \in \mathbb{Z}$,

$$n \leq \lfloor \lfloor r \rfloor / c \rfloor \Leftrightarrow n \leq \lfloor r \rfloor / c \Leftrightarrow cn \leq \lfloor r \rfloor \Leftrightarrow cn \leq r \Leftrightarrow n \leq r/c \Leftrightarrow n \leq \lfloor r/c \rfloor$$

Thus $\lfloor \lfloor r \rfloor / c \rfloor = \lfloor r/c \rfloor$. Take $r = a/b$ and we are done. □

Equinumerosity

Definition

A set A is **equinumerous** to a set B (written $A \approx B$) if there is a bijection from A to B .

Theorem

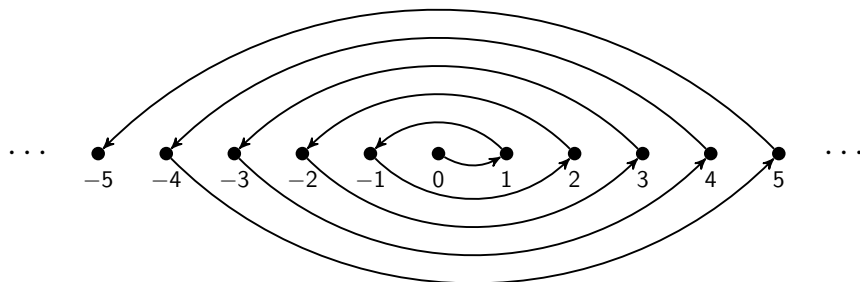
For any sets A , B , and C :

- ▶ $A \approx A$.
- ▶ $A \approx B \Rightarrow B \approx A$.
- ▶ $(A \approx B \wedge B \approx C) \Rightarrow A \approx C$.

Warning

NOT an equivalence relation since the it concerns **all** sets.

Equinumerosity $\mathbb{Z} \approx \mathbb{N}$



An integer $z \in \mathbb{Z}$ is either

- ▶ 0, or
- ▶ $-n + 1$ as then successor of $n \leq 0$, or
- ▶ $-n$ as then successor of $n > 0$.

n	0	1	2	3	4	5	6	7	9	9	10	11	...
$f(n)$	0	1	-1	2	-2	3	-3	4	-4	5	-5	6	...

Equinumerosity $\mathbb{N} \times \mathbb{N} \approx \mathbb{N}$

Cantor's Pairing Function

$$J : \mathbb{N}^2 \rightarrow \mathbb{N},$$

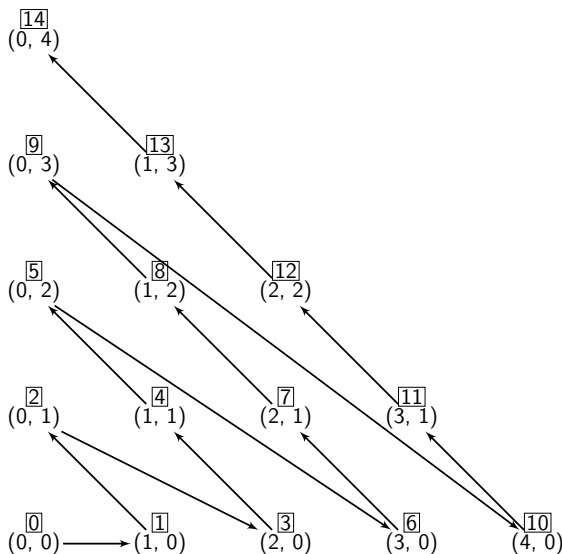
$$J(x, y) = \binom{x + y + 1}{2} + y$$

Theorem (Fueter-Pólya)

The only **quadratic** pairing functions are the Cantor polynomials (up to interchanging x and y).

Remark

It is unknown whether this the **only polynomial** pairing function.



$\mathbb{Q} \times \mathbb{Q} \approx \mathbb{Q}$?



Find me on mathstodon
@HigherGeometer

...

Terry Tao outlines a potential strategy for attacking a notorious open MathOverflow question with an incredibly simple statement and (currently) 17 deleted answers

terrytao.wordpress.com/2019/06/08/rul...

and even then, it is conditional on a conjecture in Diophantine geometry.

Polynomial bijection from $\mathbb{Q} \times \mathbb{Q}$ to \mathbb{Q} ?



Is there any polynomial $f(x, y) \in \mathbb{Q}[x, y]$ such that $f: \mathbb{Q} \times \mathbb{Q} \rightarrow \mathbb{Q}$ is a bijection?

453

nt.number-theory

open-problems

Edit tags



share cite edit close flag unprotect

edited yesterday



Community ♦

1 ● 2 ● 3

asked Apr 11 '10 at 12:03



Z.H.

2,284 ● 3 ● 12 ● 5



218

protected by [Andrés E. Caicedo](#) Dec 5 '13 at 14:23

This question is protected to prevent "thanks!", "me too!", or spam answers by new users. To answer it, you must have earned at least 10 [reputation](#) on this site (the [association bonus](#) does not count).

$$\mathbb{Q} \times \mathbb{Q} \approx \mathbb{Q}?$$

Ruling out polynomial bijections over the rationals via Bombieri-Lang?

8 June, 2019 in [math.AG](#), [math.NT](#), [question](#) | Tags: [arithmetic geometry](#), [Bombieri-Lang conjecture](#), [Diophantine geometry](#), [polynomials](#)

[**UPDATE**, Feb 1, 2021: the strategy sketched out below has been successfully implemented to rigorously obtain the desired implication in this [recent preprint of Giulio Bresciani](#).]

I recently came across [this question on MathOverflow](#) asking if there are any polynomials P of two variables with rational coefficients, such that the map $P : \mathbb{Q} \times \mathbb{Q} \rightarrow \mathbb{Q}$ is a bijection. The answer to this question is almost surely “no”, but it is remarkable how hard this problem resists any attempt at rigorous proof. (MathOverflow users with enough privileges to see deleted answers will find that there are no less than seventeen deleted attempts at a proof in response to this question!)

On the other hand, the one surviving response to the question does point out [this paper of Poonen](#) which shows that assuming a powerful conjecture in Diophantine geometry known as the [Bombieri-Lang conjecture](#) (discussed in [this previous post](#)), it is at least possible to exhibit polynomials $P : \mathbb{Q} \times \mathbb{Q} \rightarrow \mathbb{Q}$ which are injective.

$$\mathbb{Q} \times \mathbb{Q} \approx \mathbb{Q}?$$



43

There is a new manuscript on the arXiv by Giulio Bresciani, *A higher dimensional Hilbert irreducibility theorem*, arXiv:[2101.01090](https://arxiv.org/abs/2101.01090), which shows that assuming the weak Bombieri--Lang conjecture, there cannot be a polynomial bijection from $\mathbb{Q} \times \mathbb{Q} \rightarrow \mathbb{Q}$.



The author writes that:



Our strategy is essentially the one followed in a "polymath project" led by T. Tao, see [Tao19], hence this result should be credited to the polymath project as a whole.

[Tao19] <https://terrytao.wordpress.com/2019/06/08/ruling-out-polynomial-bijections-over-the-rationals-via-bombieri-lang/>

Share Cite Improve this answer Follow

edited Jan 5, 2021 at 4:03

community wiki
3 revs, 2 users 83%
Jackson Morrow

-
- 13 I should credit Daniel Loughran for pointing me to this question, see mathoverflow.net/q/373221/45660
– Giulio Bresciani Jan 5, 2021 at 6:25
-

Add a comment

Pairing Functions Recursively Defined

Consider Cantor's pairing function

$$J(x, y) = \binom{x + y + 1}{2} + y, \quad x, y \in \mathbb{N}$$

We can define it recursively as

$$J(x + 1, y) = J(x, y) + x + y + 1$$

$$J(0, y) = \binom{y + 1}{2} + y$$

$J : \mathbb{N}^2 \rightarrow \mathbb{N}$ is bijective.

A pair $(x, y) \in \mathbb{N}^2$ is either

- ▶ $(0, 0)$, or
- ▶ $(y + 1, 0)$ as the successor of $(0, y)$, or
- ▶ $(x - 1, y + 1)$ as the successor of (x, y) when $x \neq 0$.

Pairing Functions Recursively Defined

Consider another pairing function

$$P(x, y) = 2^x(2y + 1) - 1, \quad x, y \in \mathbb{N}$$

We can define it recursively as

$$P(x + 1, y) = 2P(x, y) + 1$$

$$P(0, y) = 2y$$

P is bijective

Recall fundamental theorem of arithmetic.

- ▶ Surjectivity. For all $z + 1 \in \mathbb{N}$, $z + 1 = 2^x(2y + 1)$ for some $x, y \in \mathbb{N}$.
- ▶ Injectivity. Follows from uniqueness of factorization.

Recursively Defined Functions

Recursion Theorem on \mathbb{N} (Parametric Version)

Let $a : P \rightarrow A$ and $g : \mathbb{N} \times P \times A \rightarrow A$ be functions. There exists a unique function $f : \mathbb{N} \times P \rightarrow A$ such that

- ▶ $f(0, p) = a(p)$ for all $p \in P$;
- ▶ $f(n + 1, p) = g(n, p, f(n, p))$ for all $(n, p) \in \mathbb{N} \times P$.

Recursion Theorem on \mathbb{N}

Let $a \in A$ and $g : \mathbb{N} \rightarrow A$ be functions. There exists a unique function $f : \mathbb{N} \rightarrow A$ such that

- ▶ $f(0) = a$;
- ▶ $f(n + 1) = g(n, f(n))$ for all $n \in \mathbb{N}$.

Recursively Defined Functions

Recursion Theorem on \mathbb{R} (Parametric Version)

Let $a : P \rightarrow A$ and $g : \mathbb{R} \times P \times A \rightarrow A$ be (nice) functions. There exists a unique function $f : \mathbb{R} \times P \rightarrow A$ such that

- ▶ $f^p(0) = a(p)$ for all $p \in P$;
- ▶ $\frac{df^p(t)}{dt} = g^p(t, f^p(t))$ for all $(t, p) \in \mathbb{R} \times P$.

Recursion Theorem on \mathbb{R}

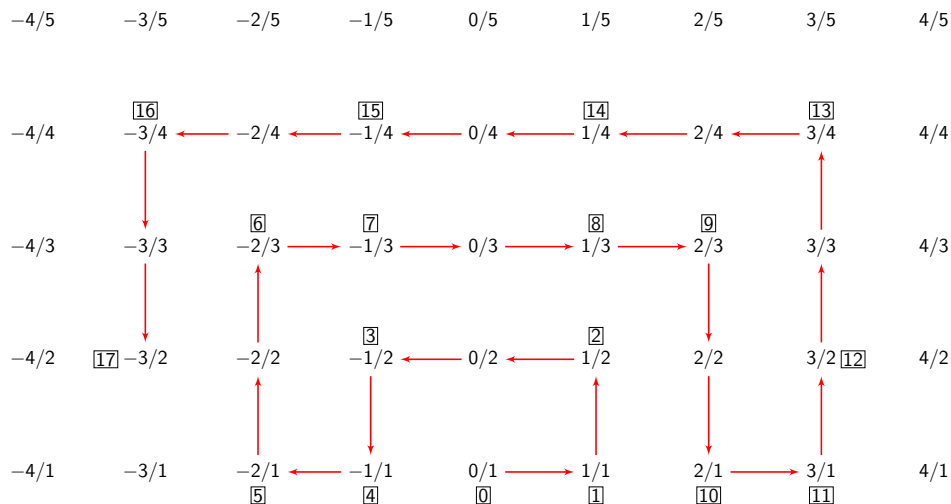
Let $a \in A$, and $g : \mathbb{R} \times A \rightarrow A$ a (nice) function. There exists a unique function $f : \mathbb{R} \rightarrow A$ such that

- ▶ $f(0) = a$;
- ▶ $\frac{df(t)}{dt} = g(t, f(t))$ for all $t \in \mathbb{R}$.

(Cristopher Moore, Recursion Theory on the Reals and Continuous-Time Computation, 1995)

Equinumerosity $\mathbb{Q} \approx \mathbb{N}$

Let $f(0) = 0/1$, $f(1) = 1/1$, $f(2) = 1/2$, $f(3) = 0/2$, etc.



Equinumerosity $\mathbb{Q} \approx \mathbb{N}$

Define $g : \mathbb{N} \rightarrow \mathbb{Q}$, such that

$$g(0) = [f(0)]$$

$$g(n+1) = [f(k)] \text{ where } k \text{ is the first such that}$$

$$\forall i \leq n, g(i) \not\sim f(k)$$

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...
$g(n)$	0	1	$\frac{1}{2}$	$-\frac{1}{2}$	-1	-2	$-\frac{2}{3}$	$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	2	3	$\frac{3}{2}$	$\frac{3}{4}$...

Cantor's Theorem

Theorem

- ▶ $\mathbb{R} \not\approx \mathbb{N}$.
- ▶ For every set A , $A \not\approx \mathcal{P}(A)$.

Proof (Sketch).

Suppose \mathbb{R} were countable, say⁴

$$x_1 = 0.\textcolor{red}{7}8790984732689\dots$$

$$x_2 = 0.2\textcolor{red}{3}456789098765\dots$$

$$x_3 = 0.98\textcolor{red}{9}65456756889\dots$$

$$x_4 = 0.237\textcolor{red}{8}9237585022\dots$$

$$x_5 = 0.1234\textcolor{red}{5}438765445\dots$$

$$\vdots$$

Consider $x_0 = 0.\textcolor{red}{84096}\dots?$



4. lots of caveats here.

Cantor's Theorem

	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	...
<i>f(a)</i>			<i>c</i>	<i>d</i>				
<i>f(b)</i>					<i>e</i>			
<i>f(c)</i>		<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>			
<i>f(d)</i>								
<i>f(e)</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>	<i>g</i>	...
<i>f(f)</i>	<i>a</i>		<i>c</i>		<i>e</i>		<i>g</i>	...
<i>f(g)</i>		<i>b</i>						<i>k, m, ...</i>
⋮								⋮

Cantor's Theorem

Proof.

Consider $f : A \rightarrow \mathcal{P}(A)$, and $B = \{x \in A \mid x \notin f(x)\} \in \mathcal{P}(A)$, e.g.,

$$f : A \longrightarrow \mathcal{P}(A)$$

$$a \mapsto \{c, d\}$$

$$b \mapsto \{e\}$$

$$c \mapsto \{b, c, d, e\}$$

$$d \mapsto \{\}$$

$$e \mapsto A$$

$$f \mapsto \{a, c, e, g, \dots\}$$

$$g \mapsto \{b, k, m, \dots\}$$

$$\vdots$$

$$B = \{a, b, d, f, g, \dots\}$$

Cantor's Theorem

Proof (Cont.)

Claim: f is not onto.

Recall $B = \{x \in A \mid x \notin f(x)\} \in \mathcal{P}(A)$. If f is onto, then $\exists z \in A$ such that $f(z) = B \in \mathcal{P}(A)$, yet

- ▶ If $z \in B$, then by definition $z \notin f(z) = B$.
- ▶ If $z \notin B$, then by definition $z \in f(z) = B$.



Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Cardinality

Cardinality

For every set A , there is a unique cardinal (or cardinal number) κ with $A \approx \kappa$. We call that κ the **cardinality** of A , denoted by $\text{card } A = \kappa$.

Example

- ▶ $\text{card } [n] = n$ for all $n \in \mathbb{N}$.
- ▶ $\text{card } \mathbb{N} = \aleph_0$ (by Cantor).
- ▶ $\text{card } \mathbb{R} = 2^{\aleph_0}$.

Continuum Hypothesis

There is no set S for which $\aleph_0 < |S| < 2^{\aleph_0}$. That is, $2^{\aleph_0} = \aleph_1$.

Caution

$\{X \mid \text{card } X = \kappa\}$ is NOT a set, except for $\kappa = 0$.

Ordering Cardinals

Definition

A set A is **dominated** by a set B (written $A \preceq B$) if there is an injection from A to B .

Examples

- ▶ $A \preceq A$.
- ▶ $A \preceq B$ if $A \subset B$. (Consider the inclusion map $\iota : A \hookrightarrow B$.)
- ▶ $A \preceq B$ iff A is equinumerous to some subset of B . (Consider a bijection between A and $f(A) \subset B$.)
- ▶ $\mathbb{N} \preceq \mathbb{Z} \preceq \mathbb{Q} \preceq \mathbb{R} \preceq \mathbb{C}$.
- ▶ $\mathbb{R} \approx (0, 1) \preceq [0, 1] \preceq \mathcal{P}(\mathbb{N}) \approx 2^{\mathbb{N}} \preceq \mathbb{R}$.

Ordering Cardinals

Definition

We write $\text{card } A \leq \text{card } B$ if $A \preceq B$.

Claim: This ordering is well-defined.

We need to verify that the definition is independent of the chosen representatives.

Suppose for sets A' and B' with $\text{card } A = \text{card } A'$ and $\text{card } B = \text{card } B'$, then $A \approx A'$ and $B \approx B'$. Now if $A \preceq B$, then there exist

- ▶ $\alpha : A' \rightarrow A$ bijective;
- ▶ $\beta : A \rightarrow B$ injective;
- ▶ $\gamma : B \rightarrow B'$ bijective.

Thus the overall composition $\gamma \circ \beta \circ \alpha : A' \rightarrow B'$ is injective, hence $A' \preceq B'$.

Ordering Cardinals

Definition

We write $\text{card } A < \text{card } B$ if $A \preceq B$ and $A \not\approx B$.

Examples

- ▶ If $A \subset B$, then $\text{card } A \leq \text{card } B$.
- ▶ For all cardinal κ , $0 \leq \kappa$.
- ▶ For all finite cardinal n , $n < \aleph_0$.
- ▶ If m and n are finite cardinals, then $m \subset n \Rightarrow m \leq n$.
- ▶ For all cardinal κ , $\kappa < 2^\kappa$. (There is no largest cardinal number.)

Countable Sets

Definition

A set A is **countable** if $A \preceq \mathbb{N}$, i.e., $\text{card } A \leq \aleph_0$. Otherwise, it is called **uncountable**.

Examples

- ▶ \mathbb{N} , \mathbb{Z} , and \mathbb{Q} are countable; \mathbb{R} is uncountable.
- ▶ A subset of a countable set is countable.
- ▶ The Cartesian product of two countable sets is countable.
- ▶ A countable union of countable sets is countable.
- ▶ If X is countable and $f : X \rightarrow Y$ is onto, then Y is countable.
- ▶ For all infinite set A , $\mathcal{P}(A)$ is uncountable.

Cantor-Schröder-Bernstein Theorem

Q: Does the ordering on the cardinals induce a “partial ordering”?

For sets A , B , and C ,

- ▶ reflexivity: $\text{card } A \leq \text{card } A$, i.e., $A \preceq A$.
- ▶ transitivity: $(\text{card } A \leq \text{card } B) \wedge (\text{card } B \leq \text{card } C) \Rightarrow \text{card } A \leq \text{card } C$, i.e., $(A \preceq B) \wedge (B \preceq C) \Rightarrow A \preceq C$.
- ▶ antisymmetry: $(\text{card } A \leq \text{card } B) \wedge (\text{card } B \leq \text{card } A) \Rightarrow ? \text{card } A = \text{card } B$,
i.e.,
 $(A \preceq B) \wedge (B \preceq A) \Rightarrow ? A \approx B$.

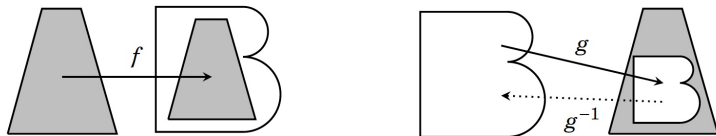
A: Yes.

Theorem (Cantor-Schröder-Bernstein)

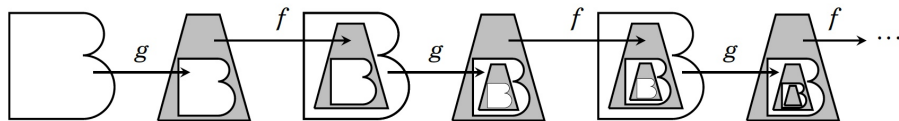
$(\text{card } A \leq \text{card } B) \wedge (\text{card } B \leq \text{card } A) \Rightarrow \text{card } A = \text{card } B$, i.e.,
 $(A \preceq B) \wedge (B \preceq A) \Rightarrow A \approx B$.

Cantor-Schröder-Bernstein Theorem

Let $f : A \rightarrow B$ and $g : B \rightarrow A$ be injective.

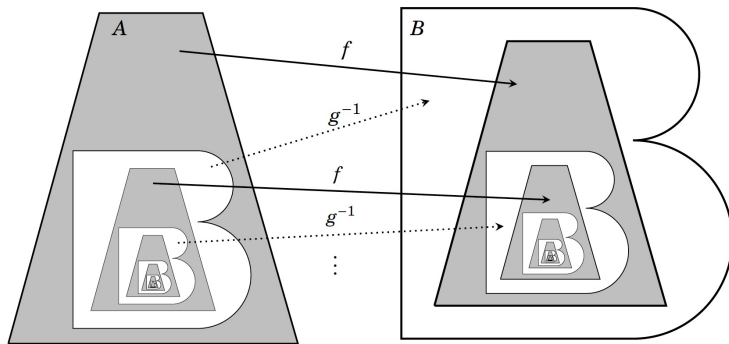


Alternating f and g , we get a chain of injections.



Cantor-Schröder-Bernstein Theorem

Iterate to get a bijection $h : A \rightarrow B$.



$$h(x) := \begin{cases} f(x), & x \in \bigcup_{k \in \mathbb{N}} (g \circ f)^k (A - g(B)) \\ g^{-1}(x), & \text{otherwise} \end{cases}$$

Table of Contents

1. Sets
2. Logic
3. Induction
4. Relations and Functions
5. Numbers and Equinumerosity
6. Cardinality
7. Finite Sets and Pigeonhole Principle

Finite Sets

For any $n \in \mathbb{N}$, let $[n] := \{1, \dots, n\}$, with $[0] = \emptyset$.

Definition

A set A is **finite** if it is equinumerous to $[n]$ for some n . A set is **infinite** if it is not finite.

Example

Any natural number is itself a finite set. Recall that for any $n \in \mathbb{N}$

$$n = \{0, \dots, n-1\}$$

Theorem (Pigeonhole Principle)

No set of the form $[n]$ is equinumerous to a proper subset of itself, where $n \in \mathbb{N}$.

Pigeonhole Principle

Pigeonhole Principle (that we know)

If there are $n + 1$ pigeons in n holes, then some hole contains at least 2 pigeons.



Pigeonhole Principle

Theorem (Pigeonhole Principle)

No set of the form $[n]$ is equinumerous to a proper subset of itself, where $n \in \mathbb{N}$.

Proof (Take 1).

Note that any function F is a surjection onto its image $\text{im } F$, we need to show that

$$\begin{aligned} & (\nexists f : [n] \rightarrow [n]) \underbrace{(f \text{ injective} \wedge \overbrace{f([n])}^{\text{im } f} \subsetneq [n])}_{f \text{ not surjective}} \\ \Leftrightarrow & (\forall f : [n] \rightarrow [n]) (\neg(f \text{ injective} \wedge \neg(f \text{ surjective}))) \\ \Leftrightarrow & (\forall f : [n] \rightarrow [n]) (\neg f \text{ injective} \vee f \text{ surjective}) \\ \Leftrightarrow & (\forall f : [n] \rightarrow [n]) (f \text{ injective} \rightarrow f \text{ surjective}) \end{aligned}$$

See (Gallier, p. 133) for the rest of the proof (by induction).



Pigeonhole Principle

Proof (by induction).

We want to show that for all $m, n \in \mathbb{N}$,

$$(m > n) \rightarrow (\nexists f : [m] \rightarrow [n] \text{ bijective})$$

It suffices to show that for all $m, n \in \mathbb{N}$,

$$(m > n) \rightarrow (\nexists f : [m] \rightarrow [n] \text{ injective})$$

or, for all $m, n \in \mathbb{N}$,

$$(m > n) \rightarrow (\neg \exists f : [m] \rightarrow [n] \text{ injective})$$

or equivalently, by considering the contrapositive, for all $m, n \in \mathbb{N}$,

$$(\exists f : [m] \rightarrow [n] \text{ injective}) \rightarrow (m \leq n)$$

Pigeonhole Principle

Proof by Induction.

We proceed by induction on n .

- ▶ **base case.** ($n = 0$): If $n = 0$, $[0] = \emptyset$. If $f : [m] \rightarrow [n]$ is injective, then the only possibility is that $[m] = \emptyset$, hence $m = 0$.
- ▶ **inductive case.** ($n \geq 1$): Assume the IH that for all $m \in \mathbb{N}$

$$(\exists f : [m] \rightarrow [n - 1] \text{ injective}) \rightarrow (m \leq n - 1)$$

We want to show that for all $m \in \mathbb{N}$

$$(\exists f : [m] \rightarrow [n] \text{ injective}) \rightarrow (m \leq n)$$

Suppose that for all $m \in \mathbb{N}$, there exists an injective $f : [m] \rightarrow [n]$,

- ▶ If $f(i) < n$ for all $i \in [m]$, then consider $g : [m] \rightarrow [n - 1]$, $i \mapsto f(i)$, which is also injective. Hence by IH $m \leq n - 1 \leq n$.

Pigeonhole Principle

Proof by Induction (Cont.)

- ▶ If $n \in f([m])$, say, $f(i_0) = n$ for some $i_0 \in [m]$, $m \neq 0$, then $n \notin f([m] \setminus \{i_0\})$ (since f is injective). Define

$$g : [m-1] \rightarrow [n-1]$$
$$i \mapsto \begin{cases} f(i), & i < i_0 \\ f(i+1), & i \geq i_0 \end{cases}$$

which is also injective, since for $i, j \in [m-1]$,

- ▶ $i, j < i_0$. $g(i) = g(j) \Rightarrow f(i) = f(j) \Rightarrow i = j$;
- ▶ $i, j \geq i_0$. $g(i) = g(j) \Rightarrow f(i+1) = f(j+1) \Rightarrow i = j$;
- ▶ $i < i_0 \leq j$. $g(i) = g(j) \Rightarrow f(i) = f(j+1) \Rightarrow i = j+1 \Rightarrow i > j$. Impossible!
- ▶ $j < i_0 \leq i$. Also impossible.

Therefore by IH $m-1 \leq n-1$, hence $m \leq n$.



Pigeonhole Principle

Proof by Induction.

We proceed by induction on m .

- ▶ **base case.** ($m = 0$): If $m = 0$, $[0] = \emptyset$. Since $f : \emptyset \rightarrow [n]$ is injective for all $n \in \mathbb{N}$, then trivially $m \leq n$.
- ▶ **inductive case.** ($m \geq 1$): Assume the IH that for all $n \in \mathbb{N}$,

$$(\exists f : [m-1] \rightarrow [n] \text{ injective}) \rightarrow (m-1 \leq n)$$

We want to show that for all $n \in \mathbb{N}$

$$(\exists f : [m] \rightarrow [n] \text{ injective}) \rightarrow (m \leq n)$$

Suppose that for all $m \in \mathbb{N}$, there exists an injective $f : [m] \rightarrow [n]$,

- ▶ If $f(i) < n$ for all $i \in [m]$, define $g : [m-1] \rightarrow [n-1]$ as $g = f|_{[m-1]}$. Then g is also injective, hence $m-1 \leq n-1$, and $m \leq n$.

Pigeonhole Principle

Proof by Induction (Cont.)

- ▶ If $f(i_0) = n$ for some $i_0 \in [m]$, thus $f([m] \setminus \{i_0\}) \subset [n-1]$ (since for any other $i \neq i_0$, $f(i) \neq f(i_0) = n$). Define

$$g : [m-1] \rightarrow [n-1]$$
$$i \mapsto \begin{cases} f(i), & i \neq i_0 \\ f(m), & i = i_0 \end{cases}$$

then g is also injective, since

- ▶ If $i, j \neq i_0$, then $g(i) = g(j) \Rightarrow f(i) = f(j) \Rightarrow i = j$.
- ▶ If $i \neq i_0$, then $g(i) = f(i) \neq f(m) = g(i_0)$ (since $i < m \Rightarrow f(i) \neq f(m)$).

Therefore by IH $m-1 \leq n-1$, hence $m \leq n$.



Finite Sets

Caveat

Given function $f : A \rightarrow B$,

- ▶ If $A = \emptyset$, then any function, $f : \emptyset \rightarrow B$ is (trivially) injective.
- ▶ If $B = \emptyset$, then f is the **empty function** from \emptyset to itself, and is (trivially) surjective, hence also bijective.

Corollary

- ▶ *No finite set is equinumerous to a proper subset of itself.*
- ▶ \mathbb{N} *is infinite.*
- ▶ *Every finite set is equinumerous to a **unique** natural number.*
- ▶ *Any subset of a finite subset is finite.*

Finite Sets

Corollary (Pigeonhole Principle for Finite Sets)

No finite set is equinumerous to a proper subset of itself.

Proof.

Since A is finite, then there exists a bijection $g : A \rightarrow [n]$ for some $n \in \mathbb{N}$.

Assume that there exists a bijection f between A and some proper subset of A .

Then, consider the function $g \circ f \circ g^{-1}$, from $[n]$ to itself.

$$\begin{array}{ccc} A & \xleftarrow{g^{-1}} & [n] \\ f \downarrow & & \downarrow g \circ f \circ g^{-1} \\ A & \xrightarrow{g} & [n] \end{array}$$

Then, note that $g(a) \in [n] \setminus \text{ran } g \circ f \circ g^{-1}$ for some $a \in A \setminus \text{ran } f$, therefore $g \circ f \circ g^{-1}$ is a bijection from $[n]$ to some proper subset of itself, which is a contradiction. □

Finite Sets

Corollary

- ▶ *Given non-empty finite sets A, B , if there exists an injection $f : A \rightarrow B$, then $|A| \leq |B|$*
- ▶ *Suppose that $f : A \rightarrow [n]$ is an injection, then A is a finite set and $|A| \leq n$.*
- ▶ *Given a finite set A and a surjective function $f : A \rightarrow B$, then $|B| \leq |A|$.*
- ▶ *Given a finite set A and a function $f : A \rightarrow B$, then $|f(A)| \leq |A|$.*

Pigeonhole Principle

Other versions of pigeonhole principle

Let $r, s \in \mathbb{N} - \{0\}$, if a set containing at least $rs + 1$ elements is partitioned into r subsets, then some subsets contains at least $s + 1$ elements.

Example

- ▶ In any group of $12 \cdot 2 + 1 = 25$ people, at least three were born in the same month.
- ▶ At least two people in London have the same number of hairs on their heads.

Application of Pigeonhole Principle

Example

Let $S \subset [200]$ with $|S| = 101$, then S contains two consecutive integers.

Proof.

Consider the following sets,

$$S_1 = \{1, 2\}$$

$$S_2 = \{3, 4\}$$

$$\vdots$$

$$S_{99} = \{197, 198\}$$

$$S_{100} = \{199, 200\}$$

The rest follows by applying the Pigeonhole principle.



Application of Pigeonhole Principle

Example

Let $S \subset [200]$ with $|S| = 101$, then S contains two integers that one divides the other.

Proof.

Consider the following sets,

$$S_1 = \{1, 2, 4, 8, \dots, 64, 128\} = \{1, 2, 2^2, 2^3, \dots, 2^7\}$$

$$S_3 = \{3, 6, 12, 24, \dots, 96, 192\} = \{3, 3 \cdot 2, 3 \cdot 2^2, 3 \cdot 2^3, \dots, 3 \cdot 2^6\}$$

$$S_5 = \{5, 10, 20, 40, 80, 160\} = \{5, 5 \cdot 2, 5 \cdot 2^2, 5 \cdot 2^3, 5 \cdot 2^4, 5 \cdot 2^5\}$$

$$\vdots$$

$$S_{49} = \{49, 98, 196\} = \{49, 49 \cdot 2, 49 \cdot 2^2\}$$

$$\vdots$$

$$S_{99} = \{99, 198\} = \{99, 99 \cdot 2\}$$

$$S_{101} = \{101\}, \dots, S_{199} = \{199\}$$

The rest follows by applying the pigeonhole principle.



Erdős–Szekeres Theorem

Theorem (Erdős–Szekeres, 1935)

Let $A = (a_1, \dots, a_n)$ be a sequence of n **different** real numbers. If $n \geq sr + 1$ then either A has an increasing subsequence of $s + 1$ terms or a decreasing subsequence of $r + 1$ terms (or both).

“The slickest and most systematic” proof (Seidenberg 1959).

Proof.

Define a function $f : \mathbb{R} \rightarrow [n]^2$, $a_i \mapsto (x_i, y_i)$, where

- ▶ x_i is the number of terms in the longest **increasing** subsequence **ending** at a_i ,
- ▶ y_i is the number of terms in the longest **decreasing** subsequence **starting** at a_i .

Erdős–Szekeres Theorem

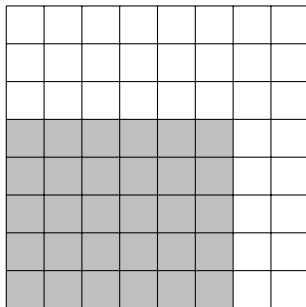
Proof. (Cont.)

Claim: The mapping $a_i \mapsto (x_i, y_i)$ is injective, i.e., $\forall i, j \in [n]$,
 $a_i \neq a_j \Rightarrow (x_i, y_i) \neq (x_j, y_j)$.

Indeed, for a subsequence $\cdots a_i \cdots a_j \cdots$, either

- ▶ $a_i < a_j \Rightarrow x_i < x_j$, or
- ▶ $a_i > a_j \Rightarrow y_i > y_j$,

The rest follows by Pigeonhole principle.



300. Longest Increasing Subsequence⁵

Given an integer array `nums`, return the length of the longest strictly increasing subsequence.

A subsequence is a sequence that can be derived from an array by deleting some or no elements without changing the order of the remaining elements. For example, `[3, 6, 2, 7]` is a subsequence of the array `[0, 3, 1, 6, 2, 2, 7]`.

Example 1

- ▶ **Input:** `nums = [10, 9, 2, 5, 3, 7, 101, 18]`
- ▶ **Output:** 4
- ▶ **Explanation:** The longest increasing subsequence is `[2, 3, 7, 101]`, therefore the length is 4.

Example 2

- ▶ **Input:** `nums = [0, 1, 0, 3, 2, 3]`
- ▶ **Output:** 4

Example 3

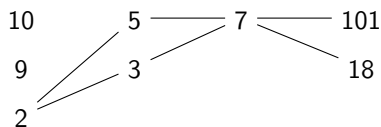
- ▶ **Input:** `nums = [7, 7, 7, 7, 7, 7, 7]`
- ▶ **Output:** 1

5. <https://leetcode.com/problems/longest-increasing-subsequence/>

300. Longest Increasing Subsequence

Example 1 (Patience Sort)

- ▶ **Input:** `nums = [10, 9, 2, 5, 3, 7, 101, 18]`
- ▶ **Output:** 4
- ▶ **Explanation:** A longest increasing subsequence is `[2, 3, 7, 101]`, therefore the length is 4.



Observation

- ▶ Each column is a decreasing subsequence.
- ▶ The length of any increasing subsequence is at most the number of columns (pigeonhole principle).