Journey Three: Successors of Peano

The axioms of natural numbers and their relation to iterators and the nonlinear traversal of binary trees.

Successors of Peano

Lecture 1

Axioms and Computers

- Formalization of Mathematics led to work of Gödel and Turing.
- Specifying software is based on axioms.
- *Concepts* are a way of grouping axioms together.

Euclid’s Axiomatic Method

Euclid is the founder of the axiomatic method:

- Definitions
- Postulates
- Common notions

23 Definitions

1. A *point* is that which has not parts.

…

23. *Parallel* straight lines are straight lines which, being in the same plane and being produced indefinitely in both directions, do not meet one another in either direction.

Common Notions

1. Things which are equal to the same thing are also equal to one another.
2. If equals be added to equals, the whole are equal.
3. If equals be subtracted from equals, the remainders are equal.
4. Things which coincide with one another are equal to one another.
5. The whole is greater than the part.

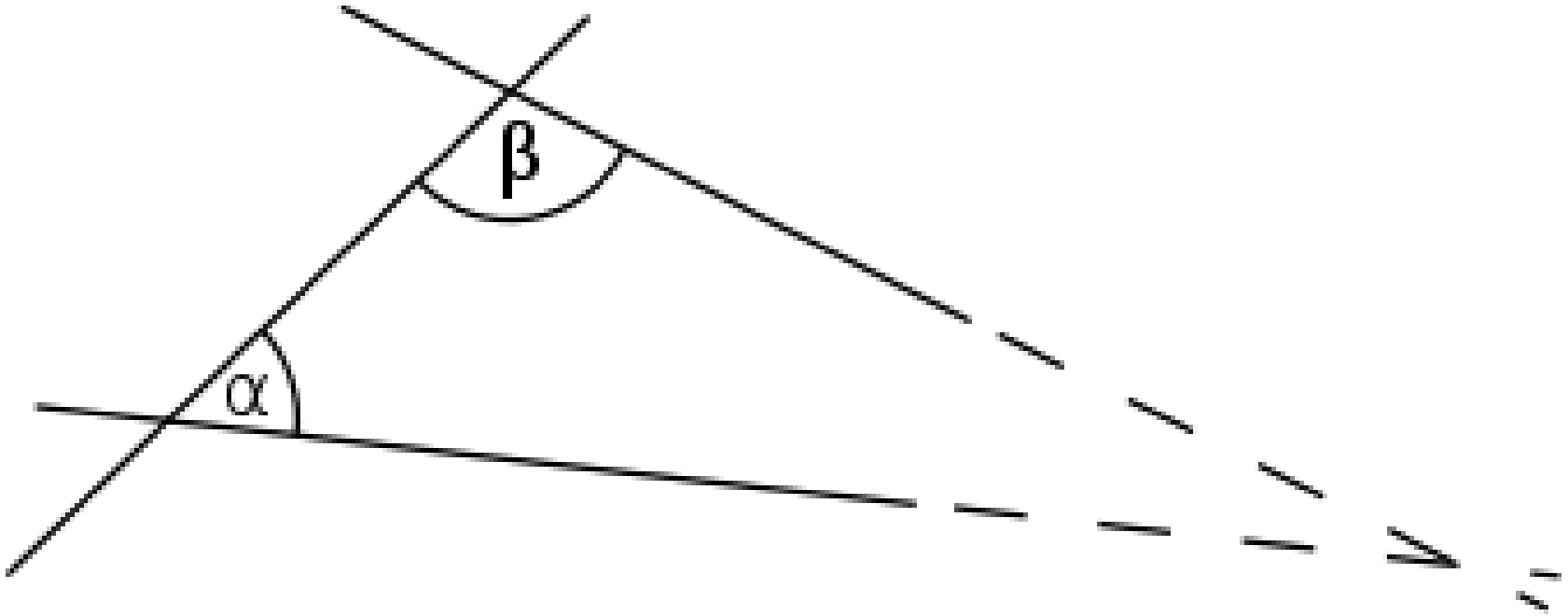
Modern Restatement of Common Notions

1. $a = c \wedge b = c \implies a = b$
2. $a = b \wedge c = d \implies a + c = b + d$
3. $a = b \wedge c = d \implies a - c = b - d$
4. $a \equiv b \implies a = b$
5. $a < a + b$

Postulates

I. To draw a straight line from any point to any point.
II. To produce a finite straight line continuously in a straight line.
III. To describe a circle with any centre and distance.
IV. That all right angles are equal to one another.
V. That, if a straight line falling on two straight lines make the interior angles on the same side less than two right angles, the two straight lines, if produced indefinitely, meet on that side on which are the angles less than the two right angles.

$\alpha + \beta < 180°$



Equivalent Formulations

- Given a line and a point not on it, at most one parallel to the given line can be drawn through the point.
  - Playfair's axiom
- There exists a triangle whose angles add up to 180°.
- There exist two similar triangles that are not congruent.

Proving Vth postulate

- Ptolemy (90 – 168)
- Omar Khayyam (1050–1123)
- Girolamo Saccheri (1667-1733)

Nikolai Ivanovich Lobachevsky (1792 -1856)

Lobachevsky's Work

- Taught by Johann Bartels
  - Professor of Gauss
- Kazan University
  - Professor, rector (president)
- Non-Euclidean Geometry
  - First reported in 1826
  - Rejected by Philistines (1832 – 1834)
  - Elected to Göttingen Academy of Sciences (1842)
  - Last major book (Pangeometry) in 1855

Lobachevsky's Impact

"It is no exaggeration to call Lobatchewsky the Copernicus of Geometry, for geometry is only a part of the vaster domain which he renovated; it might even be just to designate him as a Copernicus of all thought."

E.T. Bell, *Men of Mathematics,* page 306

Janos Bolyai
(1802 – 1860)

Bolyai Tragedy

"If I commenced by saying that I am unable to praise this work, you would certainly be surprised for a moment. But I cannot say otherwise. To praise it would be to praise myself. Indeed the whole contents of the work, the path taken by your son, the results to which he is led, coincide almost entirely with my meditations, which have occupied my mind partly for the last thirty or thirty-five years."

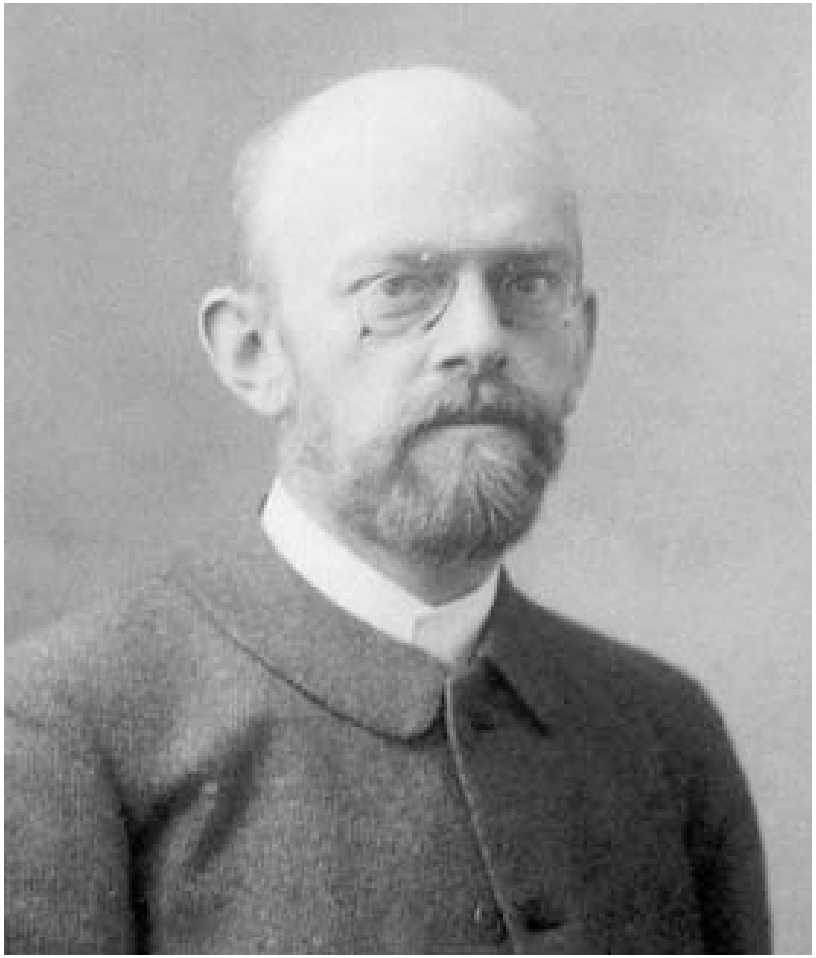
Letter from Carl Gauss to Farkas Bolyai

Geometry and Reality

- Who is right, Euclid or Lobachevsky?
- Gauss's experiment
- Independence of the V postulate
  - Beltrami, Poincare and Klein
- "It does not matter if we call the things *chairs*, *tables* and *beer mugs* or *points*, *lines* and *planes*."

  David Hilbert

David Hilbert (1862 – 1943)



Hilbert's Work

- Invariant theory
- Theory of algebraic integers
- Foundations of geometry
- Hilbert Spaces
- Mathematical Physics
  - General Relativity Theory
- Foundations of Mathematics

Hilbert's Axioms

1. 7 axioms of connection
   - If two points lie on a plane, all points on the line going through these points are on this plane.
2. 4 axioms of order
   - There is a point between any two points on a line.
3. 1 axiom of parallels
4. 6 axioms of congruence
   - Two triangles are congruent if side-angle-side…
5. 1 Archimedes' axiom

(6.) 1 Completeness axiom

Hilbert’s 23 Problems

1. Continuum hypothesis
2. Consistency of arithmetic

…

10. Existence of a solution to Diophantine equation

…

Hilbert's Program

To formalize mathematics:

- every proposition is written in a formal language
- complete: every true proposition is provable
- consistent: no contradiction can be derived

Giuseppe Peano (1858 -1932)

Peano Work

- Space filling curve (Peano curve) – 1890
- *Formulario Mathematico* – 1891 till 1908
- *Latine sine flexione* – 1903 till about 1930

Peano on “tables, chairs and beer mugs”

“Certainly it is permitted to anyone to put forward whatever hypotheses he wishes, and to develop the logical consequences contained in those hypotheses. But in order that this work merit the name of Geometry, it is necessary that these hypotheses or postulates express the result of the more simple and elementary observations of physical figures.”

Peano Axioms

There is a set $\mathbb{N}$ called the *natural numbers*:

1. $\exists 0 \in \mathbb{N}$
2. $\forall n \in \mathbb{N} \ : \ \exists n' \in \mathbb{N}$ – called its *successor*
3. $\forall \mathbb{S} \subset \mathbb{N} \ : \ (0 \in \mathbb{S} \wedge \forall n \ : \ n \in \mathbb{S} \implies n' \in \mathbb{S}) \implies \mathbb{S} = \mathbb{N}$
4. $\forall n, m \in \mathbb{N} \ : \ n' = m' \implies n = m$
5. $\forall n \in \mathbb{N} \ : \ n' \neq 0$

Axioms in Interlingua

0. N_0 es classe, vel “numero” es nomen commune.
1. Zero es numero.
2. Si *a* es numero, tunc suo successivo es numero.
3. N_0 es classe minimo, que satisfac ad conditione 0, 1, 2; […]
4. Duo numero, que habe successivo aequale, es aequale inter se.
5. 0 non seque ullo numero.

Modern Variations

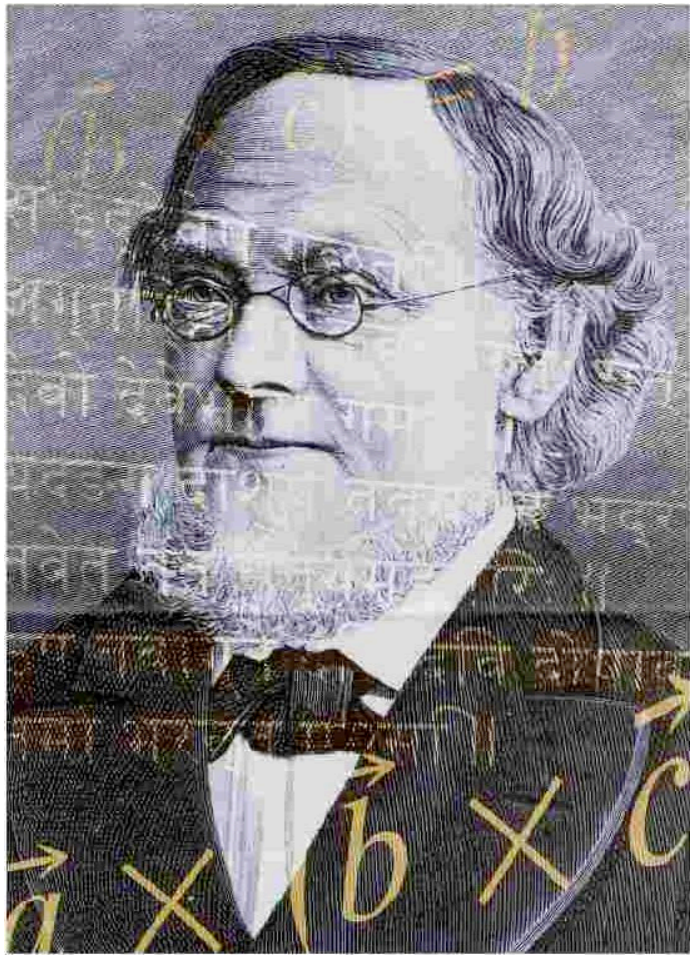
- Many modern texts start natural numbers with 1 and not 0.
- They often put the induction axiom last.
- Sometimes they replace *second order* induction axiom with a *first order* induction axiom schema.

Predecessors of Peano

- Hermann Grassmann (1809 – 1877)
  - *Lehrbuch der Arithmetik* (1861)

- Richard Dedekind
  - *Was sind und was sollen die Zahlen?* (1888)

Hermann Grassmann



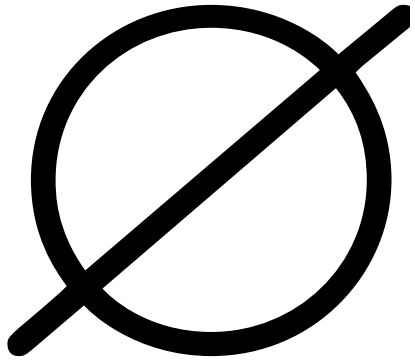
Independence of Peano Axioms

To prove that every axiom is needed, we need to remove each one from the set of axioms and demonstrate that the remaining set has models that do not meet our intent. (In other words, they are not isomorphic to our natural numbers.)

Removing *Existence of 0* Axiom

1. $\forall n \in \mathbb{N} \ : \ \exists n' \in \mathbb{N}$ – called its *successor*
2. $\forall \mathbb{S} \subset \mathbb{N} \ : \ (\forall n \ : \ n \in \mathbb{S} \implies n' \in \mathbb{S}) \implies \mathbb{S} = \mathbb{N}$
3. $\forall n, m \in \mathbb{N} \ : \ n' = m' \implies n = m$

Trivial model



Removing *Totality of Successor* Axiom

1. $\exists 0 \in \mathbb{N}$
2. $\forall \mathbb{S} \subset \mathbb{N} \ : \ (0 \in \mathbb{S} \wedge \forall n \ : \ n \in \mathbb{S} \implies n' \in \mathbb{S}) \implies \mathbb{S} = \mathbb{N}$
3. $\forall n, m \in \mathbb{N} \ : \ n' = m' \implies n = m$
4. $\forall n \in \mathbb{N} \ : \ n' \neq 0$

Finite linear models

- {0}
- {0, 1}
- {0, 1, 2}
- …

Removing *Induction* Axiom

1. $\exists 0 \in \mathbb{N}$
2. $\forall n \in \mathbb{N} \ : \ \exists n' \in \mathbb{N} -$ called its *successor*
3. $\forall n, m \in \mathbb{N} \ : \ n' = m' \implies n = m$
4. $\forall n \in \mathbb{N} \ : \ n' \neq 0$

Transfinite ordinals

- {0, 1, 2, 3, …, ω, ω+1, ω+2, …}
- {0, 1, 2, 3, …, ω_1, ω_1+1, ω_1+2, …, ω_2, ω_2+1, ω_2+2, …}
- …

Removing *Invertibility of Successor* Axiom

1. $\exists 0 \in \mathbb{N}$
2. $\forall n \in \mathbb{N} \ : \ \exists n' \in \mathbb{N}$ – called its *successor*
3. $\forall \mathbb{S} \subset \mathbb{N} \ : \ (0 \in \mathbb{S} \wedge \forall n \ : \ n \in \mathbb{S} \implies n' \in \mathbb{S}) \implies \mathbb{S} = \mathbb{N}$
4. $\forall n \in \mathbb{N} \ : \ n' \neq 0$

ρ-shaped structures

- {0, 1, 1, 1, …}
- {0, 1, 2, 1, 2, …}
- {0, 1, 2, 3, 4, 5, 3, 4, 5, …}
- …

Removing *Nothing has 0 as its Successor* Axiom

1. $\exists 0 \in \mathbb{N}$
2. $\forall n \in \mathbb{N} \ : \ \exists n' \in \mathbb{N}$ – called its *successor*
3. $\forall \mathbb{S} \subset \mathbb{N} \ : \ (0 \in \mathbb{S} \wedge \forall n \ : \ n \in \mathbb{S} \implies n' \in \mathbb{S}) \implies \mathbb{S} = \mathbb{N}$
4. $\forall n, m \in \mathbb{N} \ : \ n' = m' \implies n = m$

Circular structures

- {0, 0, …}
- {0, 1, 0, 1, …}

Definition of addition

$$a + 0 = a$$

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

0 is left additive identity

base step :

$$0 + 0 = 0$$

induction step :

$$0 + a = a \implies 0 + a' = (0 + a)' = a'$$

Definition of Multiplication

$$a \cdot 0 = 0$$

$$a \cdot b' = (a \cdot b) + a$$

$0 \cdot a$

base step :

$$0 \cdot 0 = 0$$

induction step :

$$0 \cdot a = 0 \implies 0 \cdot a' = 0 \cdot a + 0 = 0$$

Definition of 1

$$1 = 0'$$

Adding 1

$$a + 1 = a + 0' = (a + 0)' = a'$$

Multiplying by 1

$$a \cdot 1 = a \cdot 0' = a \cdot 0 + a = 0 + a = a$$

Associativity of addition

base step :

$$(a+b)+0 = a+b = a+(b+0)$$

induction step :

$$(a+b)+c = a+(b+c) \implies$$

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

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

$a + 1 = 1 + a$

base step :

$$1 + 0 = 1 + 0 = 1$$

induction step :

$$a + 1 = 1 + a \implies$$

$$a' + 1 = a' + 0' =$$

$$(a' + 0)' = ((a + 1) + 0)' =$$

$$(a + 1)' = (1 + a)' = 1 + a'$$

Commutativity of addition

base step :

$$a + 0 = a = 0 + a$$

induction step :

$$a + b = b + a \implies$$

$$a + b' = a + (b + 1) =$$

$$(a + b) + 1 = (b + a) + 1 =$$

$$b + (a + 1) = b + (1 + a) =$$

$$(b + 1) + a = b' + a$$

Problem 52

Using induction prove

- associativity and commutativity of multiplication
- distributivity of multiplication over addition

Problem 53

Using induction define total ordering between natural numbers.

Problem 54

Using induction define the following partial functions on natural numbers.

- predecessor
- subtraction

Limitation of Peano Axioms

"…the answer is that number (positive integer) cannot be defined (seeing that the ideas of order, succession, aggregate, etc., are as complex as that of number)."

Giuseppe Peano

Utility of Axioms

- Axioms explain, not define.
- Explanation could be really slow
  - defining + from successor
- Or not constructive at all
  - there is an inverse element

Successors of Peano

Lecture 2

Sets

- Peano axioms referred to the set of natural numbers.
- What are sets?
- Does a set of natural numbers exist?

Nicole Oresme (1320 – 1382)

Oresme Accomplishments

- Cartesian coordinates
- Translation of Aristotle into French
- Proof of Merton theorem
  - The distance traveled in any given period by a body moving under uniform acceleration is the same as if the body moved at a uniform speed equal to its speed at the midpoint of the period
  - Thomas Bradwardine and Oxford Calculators
- Economics
  - Monetary policy

Convergent infinite series

$$\sum_{i=1}^{\infty} \frac{i}{2^{i-1}} = 1 + 2 \cdot \frac{1}{2} + 3 \cdot \frac{1}{4} + \cdots + n \cdot \frac{1}{2^{n-1}} + \cdots = 4$$

step 1

$$1 + 2 \cdot \frac{1}{2} + 3 \cdot \frac{1}{4} + \cdots + n \cdot \frac{1}{2^{n-1}} + \cdots =$$

step 2

$$
\begin{aligned}
&(1+\frac{1}{2}+\frac{1}{4}+\cdots+\frac{1}{2^{n-1}}+\ldots)+\\
&\quad(\frac{1}{2}+\frac{1}{4}+\cdots+\frac{1}{2^{n-1}}+\ldots)+\\
&\quad\quad(\frac{1}{4}+\cdots+\frac{1}{2^{n-1}}+\ldots)+\cdots=
\end{aligned}
$$

step 3

$$1 \cdot (1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{n-1}} + \ldots) +$$
$$\frac{1}{2} \cdot (1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{n-1}} + \ldots) +$$
$$\frac{1}{4} \cdot (1 + \frac{1}{2} + \frac{1}{4} + \cdots + \frac{1}{2^{n-1}} + \ldots) + \cdots =$$

step 4

$$(1+\frac{1}{2}+\frac{1}{4}+\dots)\cdot(1+\frac{1}{2}+\frac{1}{4}+\dots)=2\cdot 2=4$$

Divergence of Harmonic series

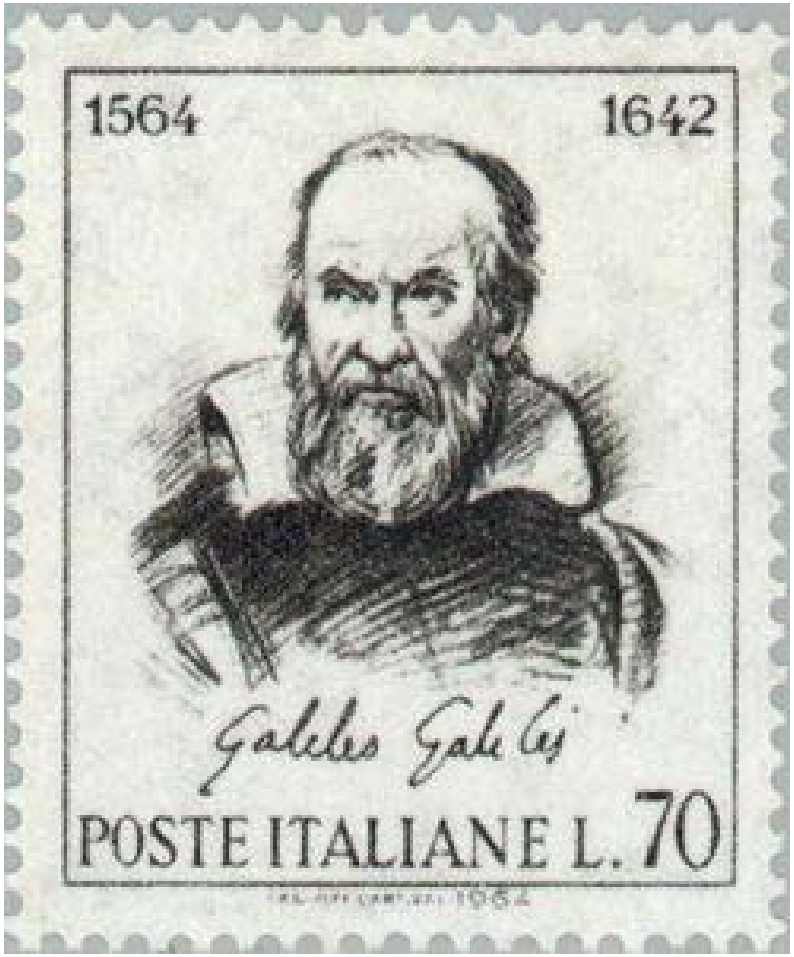
$$\sum_{i=1}^{2^n} \frac{1}{i} = 1 + \frac{1}{2}$$
$$+ (\frac{1}{3} + \frac{1}{4})$$
$$+ (\frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8})$$
$$+ \dots$$
$$+ (\frac{1}{2^{n-1}+1} + \dots + \frac{1}{2^n}) \geq \frac{n}{2}$$

One-to-one Correspondence

"Oresme shows by the principle of one-to-one correspondence that the collection of odd natural numbers is not smaller than the collection of natural numbers, because it is possible to count the odd natural numbers by the natural numbers."

Stanford Encyclopedia of Philosophy

Galileo Galilei



Main Books
(20 volumes total)

[1632] *Dialogue Concerning the Two Chief World Systems.*

[1638] *Dialogues Concerning Two New Sciences.*

Two New Sciences (1638)

DISCORSI
E
DIMOSTRAZIONI
MATEMATICHE,
intorno à due nuoue ſcienze
Attenenti alla
MECANICA & I MOVIMENTI LOCALI,
del Signor
GALILEO GALILEI LINCEO,
Filoſofo e Matematico primario del Sereniſſimo
Grand Duca di Toſcana.
Con vna Appendice del centro di grauità d'alcuni Solidi.

IN LEIDA,
Appreſſo gli Elſevirii. M. D. C. XXXVIII.

Two Sciences

- Strength of materials
  - Square-cube law
- Laws of motion
  - Uniform acceleration of falling bodies
  - Trajectory of projectiles

Galileo on infinity

“This is one of the difficulties which arise when we attempt, with our finite minds, to discuss the infinite, assigning to it those properties which we give to the finite and limited; but this I think is wrong, for we cannot speak of infinite quantities as being the one greater or less than or equal to another.”

Mapping from numbers to squares

$$n \longleftrightarrow n^2$$

$$\{1,2,3,\ldots\} \longleftrightarrow \{1,4,9,\ldots\}$$

Paradox

“The proportionate number of squares diminishes as we pass to larger numbers, Thus up to 100 we have 10 squares, that is, the squares constitute 1/10 part of all the numbers; up to 10000, we find only 1/100 part to be squares; and up to a million only 1/1000 part; on the other hand in an infinite number, if one could conceive of such a thing, he would be forced to admit that there are as many squares as there are numbers taken all together.”

Littlewood's Paradox

(5) *An infinity paradox.* Balls numbered 1, 2, . . . (or for a mathematician the numbers themselves) are put into a box as follows. At 1 minute to noon the numbers 1 to 10 are put in, and the number 1 is taken out. At $\frac{1}{2}$ minute to noon numbers 11 to 20 are put in and the number 2 is taken out. At $\frac{1}{3}$ minute 21 to 30 in and 3 out; and so on. How many are in the box at noon?

A Mathematician's Miscellany, page 5

Bernard Bolzano

Contributions

- ε-δ definition of continuity
- Bolzano-Weierstrass Theorem
  - every bounded sequence has a convergent subsequence.
- Intermediate value theorem
- Uniform convergence
- *Paradoxien des Unendlichen*
- *Wissenschaftslehre*

Paradoxes of the infinite

§19 Not all infinite sets are *equal with respect to their multiplicity*

- One could say that all infinite sets are infinite and thus one cannot compare them, but most people will agree that an interval in the real line is certainly a part and thus agree to a comparison of infinite sets.

§20 There are distinct infinite sets between which there is 1-1 correspondence. It is possible to have a 1-1 correspondence between an infinite set and a proper subset of it.

- y=12/5x and y=5/12x gives a 1-1 correspondence between [0,5] and [0,12].

§21 If two sets *A* and *B* are **infinite**, one can not conclude anything about the equality of the sets even if there is a 1-1 correspondence.

- If A and B are **finite** and A is a subset of B such that there is a 1-1 correspondence, then indeed A=B
- The above property is thus characteristic of infinite sets.

“There are truths, at least one”

“That no proposition has truth disproves itself because it is itself a proposition and we should have to call it false in order to call it true. For, if all propositions were false, then this proposition itself, namely that all propositions are false, would be false. Thus, not all propositions are false, but there also true propositions.”

There is at least one true proposition

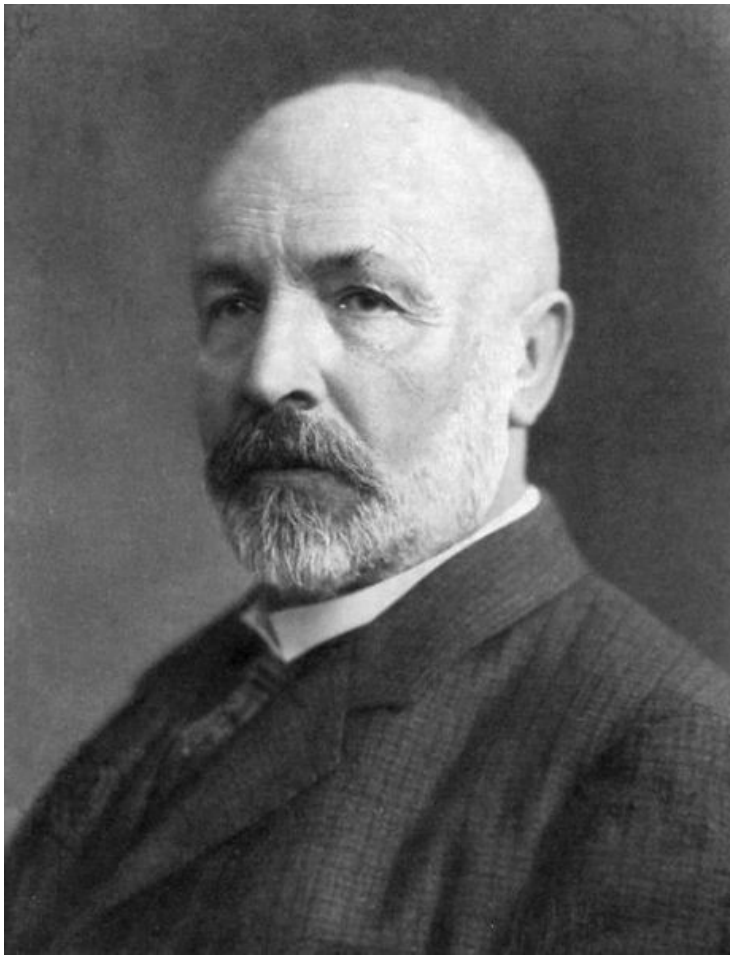
Consider the following propositions:

1. There are no true propositions.
2. If 1 is true then 1 is false.
3. 1 is false.
4. 3 is true.
5. There is at least one true proposition.

Problem 81

Using Bolzano method, prove that there are infinitely many true propositions.

Georg Cantor
(1845 – 1918)



Cantor-Dedekind Correspondence (November - December 1873)

[Cantor] Could we enumerate positive real numbers?

[Dedekind] It is not an interesting question… But here is the proof that we can enumerate the algebraic numbers.

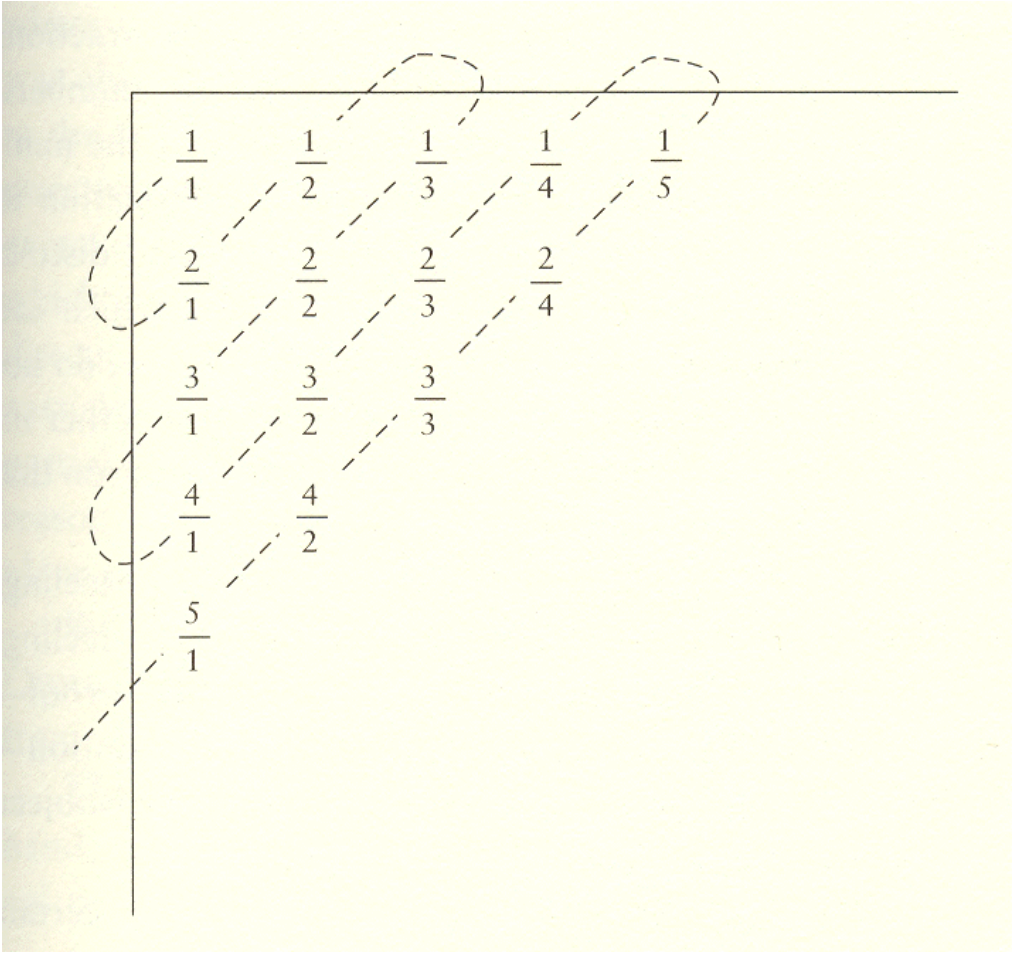
[Cantor] Then if the answer to my question is *no*, it would prove the existence of transcendental numbers…

[Cantor] The answer is *NO*!

Countable Sets

- Hilbert’s hotel

Counting Rational Numbers



Counting Algebraic Numbers

- An real number is called *algebraic* if it is a root of a polynomial with integer coefficients.
- A polynomial has *weight k* if *k* is the maximum of absolute values of its coefficients and exponents.
- There is a finite number of polynomials of a given weight and every one of them has finitely many roots.

Cantor's 1874 Theorem

Given a closed interval of real numbers and
a sequence of real numbers, the interval contains
a number that is not in the sequence.

Formally:
If $a_0, b_0 \in \mathbb{R} \wedge a_0 < b_0$ and $\{r_i\}_{i=0}^{\infty} \subseteq \mathbb{R}$
then $\exists c \in [a_0, b_0]$ such that $c \notin \{r_i\}$.

Informally:
You cannot enumerate every point in a line segment.

1874 proof of *uncountability of continuum*

1. Find the first pair of distinct elements in the sequence $\{r_i\}$ that is in the interval $[a_0, b_0]$. (If there is no such pair, then, obviously there is an element in $[a_0, b_0]$ such that it is not in the sequence; therefore; we are done with the proof.)

2. Designate the smaller element of the pair a_1 and the larger one b_1, and repeat steps 1 and 2 with the interval $[a_0, b_0]$ being replaced with the $[a_1, b_1]$ and the sequence $\{r_i\}$ with the ”unused” elements in the sequence $\{r_i\}$.

3. Observe that r_i does not belong to the inside of the interval $[a_{2i}, b_{2i}]$.

4. So if the sequence of intervals $[a_i, b_i]$ is finite, the middle of the last interval is not in the sequence $\{r_i\}$.

5. If the sequence of intervals $\{[a_i, b_i]\}$ is infinite, no element r_i could belong to its intersection $\bigcap[a_i, b_i]$. And since we know that the intersection is not empty, an element from it will not be in $\{r_i\}$.

Problem 89

Prove that the intersection of nested, closed intervals is not empty.

Two Kinds of Infinity

1. Sets equipotent to $\mathbb{N}$ have cardinality $\aleph_0$.
2. Sets equipotent to $\mathbb{R}$ have cardinality $\mathfrak{c}$.

Transcendental Numbers

- Since there are “more” real numbers than algebraic numbers, most real numbers are *transcendental*.

There are very few algebraic numbers.

Every countable set $\{a_i\}$ of reals is of measure 0.
Take the union of open intervals

$$\bigcup_{i=0}^{\infty}(a_i - \frac{\epsilon}{2^{i+2}},\ a_i + \frac{\epsilon}{2^{i+2}})$$

The size of the union is ϵ.
Since we can make ϵ however small, the measure of the set is 0.

Power set: 2^S or $\mathcal{P}(S)$

$$x \in 2^S \iff x \subseteq S$$

Problem 94

Prove that a power set of a finite set with n elements contains 2^n elements.

$$|2^{\aleph_0}| = |\mathfrak{C}|$$

1. $|\mathbb{R}| = |\mathbb{R}^+|$ through $f(x) = \ln(x)$
2. $|(0,1)| = |\mathbb{R}^+|$ through $f(x) = \frac{1}{x} - 1$
3. $|(0,1)| = |\text{Seq}\{0,1\}|$ through binary (?)
4. $|2^{\aleph_0}| = |\text{Seq}\{0,1\}|$ through the characteristic function

Continuum Hypothesis (CH)

There are no cardinalities between $\aleph_0$ and $\mathfrak{c}$.

In other words, $\mathfrak{c} = 2^{\aleph_0} = \aleph_1$

History of CH

- Georg Cantor (1878) states it.
- David Hilbert (1900) makes it famous.
- Kurt Gödel (1940) shows that it cannot be disproved in ZF or ZFC.
- Paul Cohen (1963) shows that it cannot be proved in ZF or ZFC.
  - but Cohen believed it to be false!

Onto or Surjective Functions

A function $f : X \to Y$ is called *onto* if

$$\forall y \in Y \; \exists x \in X \; : \; f(x) = y$$

Cantor’s Theorem (1891)

There is no onto function from a set to its power set.

Diagonalization Proof

Assume the contrary:

there exist a set $\mathbb{S}$ and a surjection $f : \mathbb{S} \to 2^{\mathbb{S}}$.

Take $\mathbb{D} = \{x \in \mathbb{S} | x \notin f(x)\}$.

$\mathbb{D} \subset \mathbb{S} \implies \mathbb{D} \in 2^{\mathbb{S}}$.

f is surjective $\implies \exists d \in \mathbb{S} : f(d) = \mathbb{D}$.

$d \in \mathbb{D} \vee d \notin \mathbb{D}$.

$d \in \mathbb{D} \implies d \notin \mathbb{D} \implies \neg(d \in \mathbb{D})$.

$d \notin \mathbb{D} \implies d \in \mathbb{D} \implies \neg(d \notin \mathbb{D})$.

Contradiction.

Infinity of Infinities

Power set sequence: $\aleph_0, 2^{\aleph_0}, 2^{2^{\aleph_0}}, \ldots$.

Designate the ith element of the power set sequence as $\beth_i$.

Use a *limit* operation: $\bigcup_{i=0}^{\infty} \beth_i$ to go beyond the first sequence.

Combining power set and limit operations gets into really big sets...

Russell's Paradox (1903)

Consider the set $S = \{x \mid x \notin x\}$ of all sets that do not contain themselves.

Is $S \in S$?

If it is, it is not; if it is not, it is.

Successors of Peano

Lecture 3

Zermelo's Axioms (1907)

I. Axiom of extensionality. If every element of set M is also an element of N and the other way around, then $M = N$.
II. Axiom of elementary sets. There is a set, the *empty set* $\varnothing$, that contains no element. If a is an object of the domain, there exists a set $\{a\}$, that contains a and only a as an element. If a and b are two objects of the domain, there always exists a set $\{a, b\}$ containing as elements a and b but no object x distinct from them both.
III. Axiom of separation. Whenever the propositional function $E(x)$ is defined for all elements of a set M, M possesses a subset containing all the elements x of M for which $E(x)$ is true and no other elements.
IV. Axiom of the power set. To every set T there corresponds a set or $\mathcal{P}(T)$, the power set of T, that contains all the subsets of T and no other elements.
V. Axiom of the union. To every set T there corresponds a set $\cup_T$, the union of T, that contains all the elements of the elements of T and no other elements.
VI. Axiom of choice.
VII. Axiom of infinity.

Axiom of Choice (AC)

If T is a set whose elements all are sets that are different from $\varnothing$, and mutually disjoint, its union $\cup_T$ includes at least one subset C_T having one and only one element in common with each element of T.

Banach-Tarski Paradox

Using the axiom of choice, one can cut a sphere into a finite number of pieces that can be so rearranged that one obtains two spheres of the same size as the original sphere.

Axiom of Infinity

There exists in the domain at least one set Z that contains the empty set as an element and is so constituted that to each of its elements a there corresponds a further element of the form $\{a\}$.

Problem 108

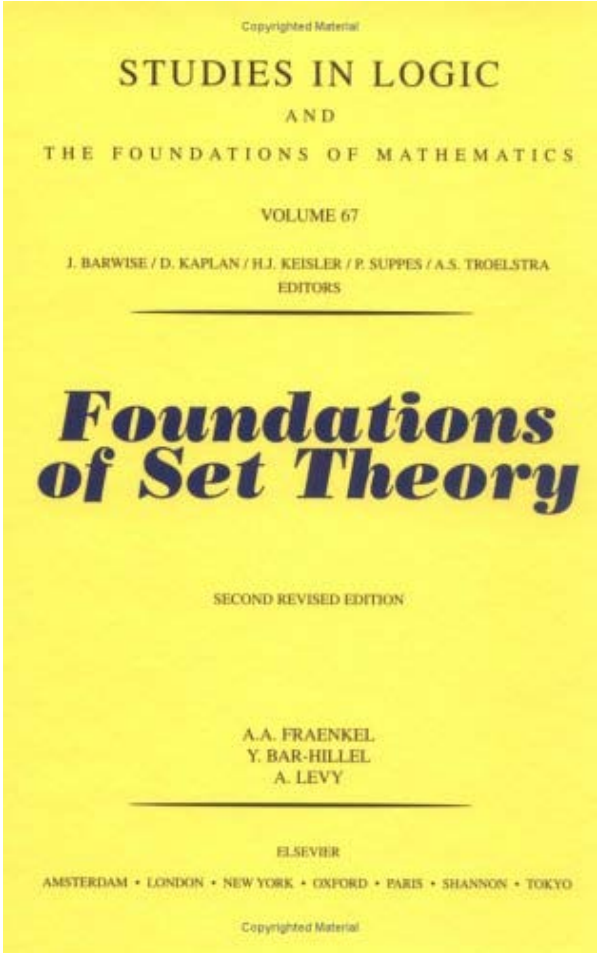
Prove the following theorem (Zermelo):

Every set M possesses at least one subset M_0 that is not an element of M.

Zermelo-Fraenkel (ZF)

- In addition to Zermelo axioms
  - Axiom of regularity
    - Every non-empty set contains an element disjoint from it.
  - Axiom of replacement
    - An image of every set is a set.

Fraenkel, Bar-Hillel, Levy



Emile Borel
(1871 – 1956)

- *Les Nombres Inaccessibles* (1952)
  - There are countably many accessible numbers.

Mathematical Paradise

“No one shall expel us from the Paradise that Cantor has created.”

David Hilbert, *Über das Unendliche*
Mathematische Annalen 95 (1925)

Hilbert's Program

- Formalization of mathematics
- Complete
  - if a proposition is true it is provable
    - either a proposition or its negation is provable
- Consistent
  - it is impossible to prove both a proposition and its negation
- Completeness and consistency should be proven with *finitary* methods

Kurt Gödel (1906 – 1978)

Gödel's results

- Completeness of first-order predicate calculus
- Incompleteness theorems (2)
  - Gödel numbering
- Recursive functions
- Length of proofs
- Consistency of axiom of choice and continuum hypothesis with ZF

Incompleteness Theorems

1. Any consistent theory containing Peano arithmetic contains a proposition that is true, but not provable.
2. Any consistent theory containing Peano arithmetic cannot prove its own consistency.

Magic Proposition

This proposition is not provable.

Alonzo Church (1903 – 1995)

Field of Symbolic Logic

- *A Bibliography of Symbolic Logic* (1936)
- Founding of the Association of Symbolic Logic
- Journal of Symbolic Logic
- Introduction to Mathematical Logic. Part 1. (1944, 1956)

Church's Students

31 students including:

- Stephen Kleene
- John Rosser
- Alan Turing
- Leon Henkin
- Martin Davis
- Hartley Rogers
- Michael Rabin
- Dana Scott

Church's Thesis

Whatever can be computed by a modern computer (or a Turing machine) includes everything that can ever be computed.

Alan Turing (1912 – 1954)

Turing's Contributions

- Turing machine
- Halting problem
- Turing-completeness
- Enigma code
- Turing test
- Biology

Diagonalization

- Cantor’s proof of the eponymous theorem led to multiple similar proofs in mathematical logic and computer science.
- Self-referential propositions have a long history.

Liar’s Paradox

Insolubilia

What I am saying is false.

- Thomas Bradwardine of Merton
- Jean Buridan of Paris

Abstracting Diagonalization

It is instructive to try to use the method of abstraction that we learned in the Journey 2 to find a generic form of Cantor's proof.

Unary-binary Family

A set $\mathbb{F}$ of (possibly partial) functions of one ($\mathbb{F}_1$) and two ($\mathbb{F}_2$) variables from domain T to codomain T' is called a *unary-binary family*.

Examples of Unary-binary Families

- total functions from $\mathbb{N}$ to $\mathbb{N}$ and from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$
- continuous functions from $\mathbb{R}$ to $\mathbb{R}$ and from $\mathbb{R} \times \mathbb{R}$ to $\mathbb{R}$
- computable functions from $\mathbb{N}$ to $\mathbb{N}$ and from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$

Interpreter

In a unary-binary family $\mathbb{F}$ with the domain T
a binary function $\mathfrak{I} \in \mathbb{F}_2$ is called an *interpreter* if

$$\forall f \in \mathbb{F}_1 \;\; \exists c \in T \;:\; \forall x \in T \;\; \mathfrak{I}(c, x) = f(x)$$

Example of an interpreter

eval1 in Lisp:

```
(defun eval1(x y)
        (eval (cons x y)))
```

- takes two S-expressions
- treats first as code, the second as data

Another interpreter

Let us take a unary-binary family L containing unary functions $f_k(x) = x + k$ and binary functions $w_k(x, y) = x + y + k$.

Then, $w_0(x, y) = x + y$ is an interpreter in L, since $f_k(x) = w(k, x)$.

Anti-interpreter

In a unary-binary family $\mathbb{F}$ with the domain T, a function $\mathfrak{A} \in \mathbb{F}_2$ is called an *anti-interpreter* if

$$\forall f \in \mathbb{F}_1 \;\; \exists c \in T \;:\; \forall x \in T \;\; f(x) \neq \mathfrak{A}(c, x)$$

An example of an anti-interpreter

As before, L is a unary-binary family containing unary functions $f_k(x) = x + k$ and binary functions $w_k(x, y) = x + y + k$.

Then, $w_1(x, y) = x + y + 1$ is an anti-interpreter in L, since $w_1(k, x) = x + k + 1 = f_k(x) + 1 \neq f_k(x)$.

eval

Could we make an analogous anti-interpreter for Lisp based on eval?

Diagonalizable Families

A unary-binary family $\mathbb{D}$ from T to T' is called a *diagonalizable family* if

$$\forall g \in \mathbb{D}_2 \;\; \exists f \in \mathbb{D}_1 \;\; : \;\; \forall x \in T \;\; g(x,x) = f(x)$$

Example of diagonalizable family

Lisp functions are diagonalizable.

For any function foo we can define a function bar:

```
(defun bar (x) (foo x x))
```

Anti-interpreter Theorem

A diagonalizable family does not have an anti-interpreter.

Proof of the Anti-interpreter Theorem

Let us assume that there is an anti-interpreter $\mathfrak{A}$. Then let $\mathfrak{a}(x) = \mathfrak{A}(x, x)$. Since $\mathfrak{A}$ is an anti-interpreter

$$\exists c \in T \ : \ \forall x \ \ \mathfrak{a}(x) \neq \mathfrak{A}(c, x)$$

But then

$$\mathfrak{a}(c) = \mathfrak{A}(c, c) \wedge \mathfrak{a}(c) \neq \mathfrak{A}(c, c)$$

Contradiction.

How about *L* ?

L has an interpreter and an anti-interpreter.

It is not diagonalizable because a function $w_k(x, x) = 2x + k$ is not in the family.

Composable-diagonalizable (C-D) family

A diagonalizable family C with domain T and co-domain T' is *composable* if:

$$\exists(\mathfrak{f} : T' \to T')\ \forall(w \in C)\ :$$
$$\mathfrak{f}(x) \neq x \ \ \wedge \ \ \mathfrak{f}(w(x,y)) \in C$$

$\mathfrak{f}$ is called the *non-fixed-point function* of the family since it always returns something different from its argument.

Non-existence of interpreter theorem

A C-D family does not contain an interpreter.

Proof of non-existence of interpreter

Let us assume that an interpreter $\mathfrak{I}$ exists.
We now can compose it with our function $\mathfrak{f}$.
Then $\mathfrak{A}(x, y) = \mathfrak{f}(\mathfrak{I}(x, y))$ is an anti-interpreter.

Indeed, for any unary function g in the family,
there is a $c \in T$ such that $\mathfrak{I}(c, x) = g(x)$.
But $\mathfrak{A}(c, x) = \mathfrak{f}(\mathfrak{I}(c, x)) = \mathfrak{f}(g(x)) \neq g(x)$.

Interpreter for any Turing-complete system

Turing constructed a universal Turing machine: a Turing machine which is an interpreter for Turing machines.

If we accept Church's thesis, such an interpreter exists in any equivalent formulation.

Interpreter Thesis

Any non-trivial set of computable functions that contains an interpreter is Turing-complete.

Examples of C-D families

- Total computable functions over integers
- Polynomial time functions over integers
- Continuous functions over real numbers
- Differentiable functions over real numbers

Halting problem

There is no computable function *halt(c, x)* that returns true if code *c* terminates on input *x* and false otherwise.

Indeed, if such a function existed we would be able to construct an anti-interpreter:

Anti-interpreter Code

```
integer anti_interpreter(integer c,
                         integer x) {
  if (halts(c, x)) {
      return interpreter(c, x) + 1;
  } else {
    return 0;
  }
}
```

Successors of Peano

Lecture 4

The School of Athens

Aristotle (384 BC – 322 BC)

The Summary

πάντες ἄνθρωποι τοῦ εἰδέναι ὀρέγονται φύσει.

All humans naturally desire to know.

Works

- Organon
- Physics
- Metaphysics
- Ethics
- Poetics
- Politics
- …

Organon

1. *Categories*
2. *On Interpretation*
3. *Prior Analytics*
4. *Posterior Analytics*
5. *Topics*
6. *Sophistical Refutations*

Individual, Species, Genus

- individual

- species
- genus - differentia

- definition
- proprium
- accident

Theories

- A theory is a set of true propositions.
- A theory could be generated by a set of axioms plus the set of inference rules.
- A theory is finitely-axiomatizable if it can be generated from a finite set of axioms.
- A set of axioms is independent if removing one will decrease the set of true propositions.
- A theory is complete is for any proposition, either it or its negation is in the theory.
- A theory is consistent if for no proposition it contains it and its negation.

Theory of Groups

operations: $x \circ y, \;\; x^{-1}$

constant: $\quad e$ (identity)

axioms:

$x \circ (y \circ z) = (x \circ y) \circ z$

$x \circ e = e \circ x = x$

$x \circ x^{-1} = x^{-1} \circ x = e$

theorems:

$\forall y \, \forall x \;:\; x \circ y = x \implies y = e$

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

$\ldots$

Model of a theory

A set of elements that satisfies all the propositions in the theory is called its model.

Categorical Theories

A (consistent) theory is *categorical* or *univalent* if all of its models are isomorphic.

(This is an original definition of Oswald Veblen. Modern logicians talk about κ-categorical theories: all models of the cardinality κ are isomorphic.)

Groups of order < 13

order	abelian	non-abelian
1	1	0
2	1	0
3	1	0
4	2	0
5	1	0
6	1	1
7	1	0
8	3	2
9	2	0
10	1	1
11	1	0
12	2	3

Non-isomorphic groups of order 4

	e	**a**	**b**	**c**
e	e	a	b	c
a	a	b	c	e
b	b	c	e	a
c	c	e	a	b

Cyclic group Z_4

	e	**a**	**b**	**c**
e	e	a	b	c
a	a	e	c	b
b	b	c	e	a
c	c	b	a	e

Klein group

Distinguishing proposition (*differentia*) for groups of order 4

$$\forall x \in G \ : \ x^2 = e$$

is true for Klein group but false for $\mathbb{Z}_4$

Different models of Z_4

- Additive group of remainders modulo 4
  - 0, **1**, 2, **3**
- Multiplicative group of non-zero remainders modulo 5
  - 1, **2**, **3**, 4
- What are possible mappings between them?

Different models of Klein group

- Multiplicative group of units modulo 8
  - 1, 3, 5, 7
- Group of isometries transforming a rectangle into itself
- What are possible mappings between them?

Propositions vs. models

- The more propositions there are in a theory, the fewer different models there are.
  - More propositions imply more axioms
- The more models there are in a theory, the fewer propositions there are.

Value and its type

- A *datum* is a sequence of bits.
  - 101
- A *value* is a datum together with its interpretation.
  - it is 5.
  - it is -1 (how?)
- A value-type is a set of values sharing a common interpretation.

Value-types in Programming Languages

- C++, Java, etc, do not provide mechanisms for defining value-type.
- Every type resides in memory and is an object type.

Object

- An object is collection of bits in memory that contain a value of a given type.
- An object is *immutable* if the value never changes.
- An object is *mutable* otherwise.
- An object is *unrestricted* if it can contain any value of its value type.

Object type

- An object type is a uniform method of storing and retrieving values of a given value type from a particular object when given its *address*.
- In programming languages, types are object types.

Concepts and their types

- Concept is a way of describing a family of related object types.

Integer Concepts and Their Types

- Integral
  - int8_t, uint8_t, int16_t,…
- UnsignedIntegral
  - uint8_t, uint16_t, …
- SignedIntegral
  - int8_t, int16_t, …

Concepts as predicates on types

Concept is a predicate on type that assures that a given type satisfies a set of requirements

- operations
- semantics
- time/space complexity

Type-functions

A function that given a type, returns an affiliated type.

- value_type(Sequence)
- coefficient_type(Polynomial)
- ith_element_type(Tuple, size_t)

Type-attribute

A function that given a type, returns one of its numerical attributes

- sizeof
- alignment_of
- number of members in a struct

Regular

- copy-constructor
  - default constructor
    - T a(b) equivalent to T a; a = b;
- assignment
- equality
- destructor

Semantics of Regular

$$\forall a\ \forall b\ \forall c\ : T\ a(b) \implies (b = c \implies a = c)$$
$$\forall a\ \forall b\ \forall c\ : a \leftarrow b \implies (b = c \implies a = c)$$
$$\forall f \in \text{RegularFunction}\ :\ a = b \implies f(a) = f(b)$$
$$\ldots \text{axioms of destructors}$$

Semiregular

- Semiregular is like Regular except that equality is not explicitly defined.
- It is assumed to be implicitly defined so that axioms that control copying and assignment are still valid.

Swap

```
template <Semiregular T>
void swap(T& x, T& y) {
  T tmp(x);
  x = y;
  y = tmp;
}
```

Iterator Concepts

Fundamental operations

- regular type operations
- successor
- dereferencing

Dereferencing

- Dereferencing is assumed to be “fast.”
  - There is not faster way of getting to data than through the iterator.
  - Iterators might be bigger in size than pointers to allow for navigation.
- Past-the-end values
- Singular values

Separating successor and dereferencing

- It is possible (as does *EoP*) to separate successor from dereferencing.
- Another way of getting similar results is to assure that dereferencing is defined for all objects

Default dereferencing (illegal in C++)

```
template <typename T>
T& operator*(T& x) {
  return x;
}

template <typename T>
const T& operator*(const T& x) {
  return x;
}
```

Connection of dereferencing and successor

- Dereferencing is defined on an iterator if and only if successor is defined.
- There are no non-empty ranges containing no values.
- If you are not at the end of the range, you can dereference.

Forward Iterators

- equality
- dereferencing
- successor

All operations are constant time

- algorithms written in terms of these operations are expected to be as fast as possible

Complexity is part of the interface!

Successors of Peano

Lecture 5

Three Fundamental Problems

- swap
  - copy, assignment, (implied) equality
- minimum, maximum
  - strict ordering
- linear search
  - successor

Three fundamental concepts

- regular
  - semi-regular
- total ordering
  - weak ordering
- iterator concepts

Iterator Categories

- Input iterators
  - one directional traversal, single-pass algorithms
  - model: input stream
- Forward iterators
  - one directional traversal, multi-pass algorithms
  - model: singly-linked list
- Bidirectional iterators
  - bidirectional traversal, multi-pass algorithms
  - model: doubly-linked list
- Random access iterators
  - random-access algorithms
  - model: array

Output Iterators

- No equality
- dereferencing only as an l-value
- alternating ++ and *

Other Categories of Iterators

- Linked iterators
  - successor function is mutable
- Segmented iterators
  - std::deque would immediately benefit

distance for input iterators

```
template <InputIterator I>
DifferenceType(I) distance(I f, I l,
                  std::input_iterator_tag) {
  // precondition: valid_range(f, l)
  DifferenceType(I) n(0);
  while (f != l) {
    ++f;
    ++n;
  }
     return n;
}
```

distance for random access iterators

```
template <RandomAccessIterator I>
DifferenceType(I) distance(I f, I l,
        std::random_access_iterator_tag) {
    // precondition: valid_range(f, l)
    return l - f;
}
```

difference type

difference type of iterator is an integral type that is large enough to encode the longest possible range

Iterator Traits

- value_type
- reference
- pointer
- difference_type
- iterator_category

type functions in C++

```
#define DifferenceType(X) typename \
std::iterator_traits<X>::difference_type
```

Category dispatch

```
template <InputIterator I>
inline
DifferenceType(I) distance(I f, I l) {
  return     distance(f, l,
                 IteratorCategory(I)());
}
```

Un-implementability of valid_range

- Linked data structures
- Pointers

Laws of valid ranges

$$\text{container}(c) \implies \text{valid}(\text{begin}(c), \text{end}(c))$$
$$\text{valid}(x, y) \wedge x \neq y \implies \text{valid}(\text{successor}(x), y)$$

advance (input iterators)

```
template <InputIterator I>
inline
void advance(I& x, DifferenceType(I) n,
             std::input_iterator_tag) {
  while (n) {
    --n;
    ++x;
  }
}
```

advance (random access iterators)

```
template <RandomAccessIterator I>
inline
void advance(I& x, DifferenceType(I) n,
             std::random_access_iterator_tag) {
  x += n;
}
```

advance

```
template <InputIterator I>
inline
void advance(I& x, DifferenceType(I) n) {
  advance(x, n, IteratorCategory(I)());
}
```

Open, Semi-open and Closed Ranges

- open: does not include either end

  (i, j)

- semi-open: includes value at i but not at j

  [i, j)

- closed: includes value at i and at j

  [i, j]

There are also semi-open ranges (i, j]

Semi-open Ranges

- Algorithmically, semi-open ranges are superior
  - search
  - insert
  - rotation
  - partition

Positions in a sequence

- A sequence of n elements has n + 1 (insertion, rotation, etc) positions.

Bounded and Counted Ranges

- A range (open, semi-open, or closed) can be specified in two ways
  - *bounded*: two iterators
  - *counted*: an iterator and length

Ranges

	semi-open	closed
Bounded: two iterators	[*i*, *j*)	[*i*, *j*]
Counted: iterator and integer	[*i*, *n*)	[*i*, *n*]

Linear search

```
template <InputIterator I,
          Predicate P>
I find_if(I f, I l, P p) {
  while (f != l && !p(*f)) ++f;
  return f;
}
```

Input iterators

- Single-pass algorithms
  - you cannot step into the same river twice
- i == j does not imply ++i == ++j
- modifying one copy invalidates the rest

Linear search (counted)

```
template <InputIterator I,
          Predicate P>
std::pair<I, difference_type(I)>
find_if_n(I f, difference_type(I) n, P p) {
  while (n && !p(*f)) { ++f; --n; }
  return std::make_pair(f, n);
}

// Why do we return a pair?
```

Binary Search

- John Mauchly (1946)
  - discussion
- D.H. Lehmer (1960)
  - “correct” implementation

bsearch()

“The bsearch() function shall return a pointer to a matching member of the array, or a null pointer if no match is found. If two or more members compare equal, which member is returned is unspecified.”

http://www.unix.com/man-page/POSIX/3posix/bsearch/

Intermediate value theorem

f is a continuous function in an interval $[a, b]$ such that $f(a) < f(b)$.
Then $\forall u \in [f(a), f(b)]$ there is $c \in [a, b]$ such that $u = f(c)$.

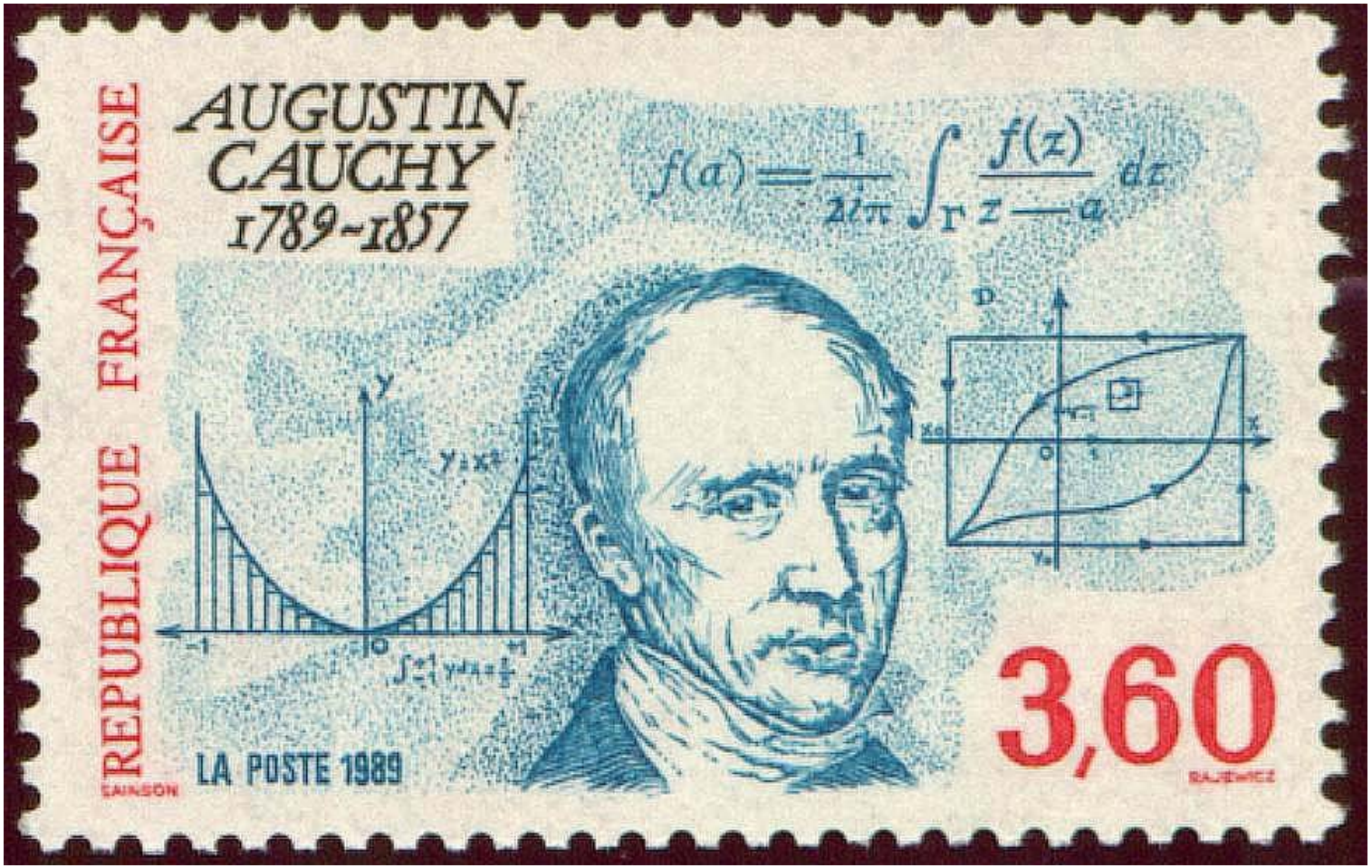
Proof

Do the binary search!

History of IVT

- Simon Stevin (1594)
  - polynomials
- Joseph Louis Lagrange (1795)
  - polynomials
- Bernard Bolzano (1817)
- Augustin-Louis Cauchy (1821)

Augustin-Louis Cauchy (1789 -1857)



partition_point_n

```
template <ForwardIterator I, Predicate P>
I partition_point_n(I f, DifferenceType(I) n, P p);
```

Partition point semantics

precondition:
$\exists m \in [f, n) \ : \ (\forall i \in [f, m) \ : \ p(i)) \ \wedge \ (\forall i \in [m, f + n) \ : \ \neg p(i))$

postcondition: return value is m from the precondition

The sad story of partition

- Should elements that satisfy the predicate precede the elements that do not?

```
template <ForwardIterator I, Predicate P>
I partition_point_n(I f, DifferenceType(I) n, P p) {
  while (n) {
    I middle(f);
    DifferenceType(I) half(n >> 1);
    advance(middle, half);
    if (!p(*middle)) {
      n = half;
    } else {
      f = ++middle;
      n = n - (half + 1);
    }
  }
  return f;
}
```

partition_point

```
template <ForwardIterator I, Predicate P>
I partition_point(I f, I l, P p) {
  return partition_point_n(f,
                           std::distance(f, l),
                           p);
}
```

Sorted ranges

Read section 6.5 of EoP

Binary Search Lemma

For any sorted range $[v_i, v_j)$ and a value a, there are two iterators, lower bound b_l and upper bound b_u such that

1. $\forall k \in [i, b_l) \quad : \; v_k < a$
2. $\forall k \in [b_l, b_u) \; : \; v_k = a$
3. $\forall k \in [b_u, j) \quad : \; v_k > a$

Problem 224

Prove the Binary Search Lemma.

lower_bound

```
template <ForwardIterator I>
I lower_bound(I f, I l, ValueType(I) a) {
  std::less<ValueType(I)> cmp;
  return std::partition_point(f, l,
                    std::bind2nd(cmp, a));
}
```

lower_bound (C++11)

```
template <ForwardIterator I>
I lower_bound(I f, I l, ValueType(I) a) {
  return std::partition_point(f, l,
    [=](ValueType(I) x) { return x < a; });
}
```

upper_bound

```
template <ForwardIterator I>
I upper_bound(I f, I l, ValueType(I) a) {
  std::less_equal<ValueType(I)> comp;
  return std::partition_point(f, l,
                    std::bind2nd(cmp, a));
}
```

upper_bound (C++11)

```
template <ForwardIterator I>
I lower_bound(I f, I l, ValueType(I) a) {
  return std::partition_point(f, l,
    [=](ValueType(I) x) { return x <= a; });
}
```

Practical theories

Iterator theories are as important to Computer Science as theories of groups and rings are to Algebra.

Knowing theories implies knowing their algorithms.

Lessons of the Journey

- Investigations of foundational issues of arithmetic led to the design of modern computers.
- Simple theories of *successor* allow us to express a large body of algorithms.
- Concentrate on deep understanding of a few central things!