The countable reals

Exposition by Jean Abou Samra
(Eötvös Loránd University, Budapest)
of work by Andrej Bauer and James E. Hanson:
The countable reals, arXiv:2404.01256

November 10th 2025

Introduction

Cantor proved that $\mathbb{R}$ is uncountable in two ways.

First proof: Take $u : \mathbb{N} \to \mathbb{R}$. Construct nested intervals

$$[x_0, y_0] \supseteq [x_1, y_1] \supseteq [x_2, y_2] \supseteq \cdots$$

such that $[x_n, y_n]$ avoids u_n. Then u misses any point in the intersection, so is not surjective.

Introduction

Construction: cut $[x_n, y_n]$ in three equal parts $[x_n, a_n], [a_n, b_n], [b_n, y_n]$ and set

$$[x_{n+1}, y_{n+1}] = \begin{cases} [x_n, a_n] \text{ if } u_n > a_n \\ [b_n, y_n] \text{ if } u_n < b_n \end{cases}$$

Introduction

Construction: cut $[x_n, y_n]$ in three equal parts $[x_n, a_n], [a_n, b_n], [b_n, y_n]$ and set

$$[x_{n+1}, y_{n+1}] = \begin{cases} [x_n, a_n] \text{ if } u_n > a_n \\ [b_n, y_n] \text{ if } u_n < b_n \end{cases}$$

Constructively, we can obtain "if $a_n < b_n$ then $u_n > a_n$ or $u_n < b_n$", but "$u_n > a_n$ or $u_n \leq a_n$" is the analytic limited principle of omniscience, a constructive taboo.

Introduction

Construction: cut $[x_n, y_n]$ in three equal parts $[x_n, a_n], [a_n, b_n], [b_n, y_n]$ and set

$$[x_{n+1}, y_{n+1}] = \begin{cases} [x_n, a_n] \text{ if } u_n > a_n \\ [b_n, y_n] \text{ if } u_n < b_n \end{cases}$$

Constructively, we can obtain “if $a_n < b_n$ then $u_n > a_n$ or $u_n < b_n$”, but “$u_n > a_n$ or $u_n \leq a_n$” is the analytic limited principle of omniscience, a constructive taboo.

Alternatively, use dependent choice: If for all s there exists s' such that $R(s, s')$ then there is a sequence (s_n) such that $R(s_n, s_{n+1})$.

Introduction

Construction: cut $[x_n, y_n]$ in three equal parts $[x_n, a_n], [a_n, b_n], [b_n, y_n]$ and set

$$[x_{n+1}, y_{n+1}] = \begin{cases} [x_n, a_n] \text{ if } u_n > a_n \\ [b_n, y_n] \text{ if } u_n < b_n \end{cases}$$

Constructively, we can obtain "if $a_n < b_n$ then $u_n > a_n$ or $u_n < b_n$", but "$u_n > a_n$ or $u_n \leq a_n$" is the analytic limited principle of omniscience, a constructive taboo.

Alternatively, use dependent choice: If for all s there exists s' such that $R(s, s')$ then there is a sequence (s_n) such that $R(s_n, s_{n+1})$.

Or, by making all choices in advance, just countable choice: If for all $n \in \mathbb{N}$ there exists s such that $R(n, s)$ then there exists f such that $R(n, f(n))$.

Introduction

Second proof: Take $u : \mathbb{N} \to 2^{\mathbb{N}}$. The sequence $n \mapsto \mathrm{flip}(u_n(n))$ is not in the image of u.

Constructively valid, but proves that $2^{\mathbb{N}}$ is uncountable, a different theorem!

Building the decimal expansion of a real requires the analytic limited principle of omniscience "$x \geq y$ or $x < y$" again.

Which flavor of constructive mathematics?

Technically: the internal language of an “elementary 1-topos with natural numbers object”. Traditionally higher-order logic, but can model extensional type theory as well.

Which flavor of constructive mathematics?

Technically: the internal language of an "elementary 1-topos with natural numbers object". Traditionally higher-order logic, but can model extensional type theory as well.

Practically speaking:

- Constructive
- Impredicative: type Ω of truth values, powerset of A is Ω^A
- Quotients and propositional truncation
- Equality reflection, function extensionality, propositional extensionality, unique choice

Which flavor of constructive mathematics?

Technically: the internal language of an "elementary 1-topos with natural numbers object". Traditionally higher-order logic, but can model extensional type theory as well.

Practically speaking:

- Constructive
- Impredicative: type Ω of truth values, powerset of A is Ω^A
- Quotients and propositional truncation
- Equality reflection, function extensionality, propositional extensionality, unique choice

Terminological points:

- Existence means mere existence ($\exists$ not Σ).
- A set X is **inhabited** when there exists an element in it ($\|X\| \equiv \exists x : X, \top$).
- A truth value $p \in \Omega$ is **decidable** when $p \vee \neg p$ ($\longrightarrow$ decidable subset, decidable equality), and **classical** when $\neg\neg p \Rightarrow p$.

Countability in constructive mathematics

X is **countable** when there is a surjection $\mathbb{N} \to X + 1$.

If a countable set surjects into X then X is countable.

A countable union $\bigcup_{i \in I} A_i$ can be rewritten as $\bigcup_{n \in \mathbb{N}} B_i$.

Countability in constructive mathematics

X is **countable** when there is a surjection $\mathbb{N} \to X + 1$.

If a countable set surjects into X then X is countable.

A countable union $\bigcup_{i \in I} A_i$ can be rewritten as $\bigcup_{n \in \mathbb{N}} B_i$.

X is **subcountable** when there is an injection $X \to \mathbb{N}$.

A countable set may contain an uncountable subset! Countability and subcountability are unrelated constructively.

Countability in constructive mathematics

X is **countable** when there is a surjection $\mathbb{N} \to X + 1$.

If a countable set surjects into X then X is countable.

A countable union $\bigcup_{i \in I} A_i$ can be rewritten as $\bigcup_{n \in \mathbb{N}} B_i$.

X is **subcountable** when there is an injection $X \to \mathbb{N}$.

A countable set may contain an uncountable subset! Countability and subcountability are unrelated constructively.

Bauer previously exhibited a model where $\mathbb{R}$ and even $2^{\mathbb{N}}$ are subcountable.

The real numbers in constructive mathematics

The construction of the field $\mathbb{Q}$ is unproblematic. It is countable and has decidable equality and ordering.

Three constructions of $\mathbb{R}$ from $\mathbb{Q}$:

- Cauchy reals: using Cauchy sequences to "add missing limits" in $\mathbb{Q}$. Sub-variants: quotient in one go, or add sequences and quotient them at the same time, quotient-inductive-inductively.
- Dedekind reals: next slide.
- MacNeille reals: a different order completion construction. Always uncountable, but "$x < 1$ or $x > 0$" not provable.

The real numbers in constructive mathematics

The construction of the field $\mathbb{Q}$ is unproblematic. It is countable and has decidable equality and ordering.

Three constructions of $\mathbb{R}$ from $\mathbb{Q}$:

- Cauchy reals: using Cauchy sequences to "add missing limits" in $\mathbb{Q}$. Sub-variants: quotient in one go, or add sequences and quotient them at the same time, quotient-inductive-inductively.
- Dedekind reals: next slide.
- MacNeille reals: a different order completion construction. Always uncountable, but "$x < 1$ or $x > 0$" not provable.

Even classically, it is interesting to consider all three because they generalize in different ways: Cauchy reals $\longrightarrow$ p-adic numbers, Dedekind reals $\longrightarrow$ surreal numbers, MacNeille reals $\longrightarrow$ Dedekind-MacNeille completion of a poset.

The Dedekind reals

Intuition: Represent $r \in \mathbb{R}$ by $L := \{x \in \mathbb{Q} \mid x < r\}$ and $U := \{x \in \mathbb{Q} \mid x > r\}$.

A **Dedekind cut** is a pair (L, U) such that:

- L is inhabited, i.e., $\|L\|$
- L is downwards-closed, i.e., for all $y \in \mathbb{Q}$, if there exists $x \in L$ such that $y < x$, then $y \in L$
- L is open, which in view of downwards-closedness is equivalent to the converse: if $y \in L$ then there exists $x \in L$ such that $y < x$
- Symmetrically, U is inhabited and $y \in U \Leftrightarrow (\exists x \in U, x < y)$
- L is below U: if $x \in L$ and $y \in U$ then $x < y$
- L and U are located: if $x < y$ then $x \in L$ or $y \in U$

Realizability

The counter-model is a variant of realizability models.

Curry-Howard: A proof is a program...

Realizability: A proof is a program...

Realizability

The counter-model is a variant of realizability models.

Curry-Howard: A proof is a program… in an expressive typed programming language with propositions-as-types

Realizability: A proof is a program… in a model of computation that may be completely untyped

Realizability

The counter-model is a variant of realizability models.

Curry-Howard: A proof is a program… in an expressive typed programming language with propositions-as-types

Realizability: A proof is a program… in a model of computation that may be completely untyped

A “Curry-Howard model” is just a syntactic model/term model.

A realizability model is more complicated: sets with computational representations, maps which have a computational counterpart

Realizability was invented by Kleene in 1945, ~25 years earlier than Curry-Howard. Variants: Dialectica (Gödel), linear realizability, classical realizability, …

Realizability

The counter-model is a variant of realizability models.

Curry-Howard: A proof is a program… in an expressive typed programming language with propositions-as-types

Realizability: A proof is a program… in a model of computation that may be completely untyped

A "Curry-Howard model" is just a syntactic model/term model.

A realizability model is more complicated: sets with computational representations, maps which have a computational counterpart

Realizability was invented by Kleene in 1945, ~25 years earlier than Curry-Howard. Variants: Dialectica (Gödel), linear realizability, classical realizability, …

Realizability can work with a very general class of models of computation: models of a variant of the SKI combinator calculus where application is a partial operation.

Partial combinatory algebras

Let A be a set with a partial operation $A \times A \dashrightarrow A$. An expression is, inductively, a variable, a constant or the application of an expression to an expression.

A is a partial combinatory algebra when for all expression e in variables $x_0, \ldots, x_n$, there exists $a \in A$ such that for all $a_0, \ldots, a_n \in A$:

1. $a a_0 \ldots a_{n-1} a_n \simeq e[a_0/x_0, \ldots, a_n/x_n]$ where $\simeq$ means one is defined iff the other is, and then they have the same value
2. $a a_0 \ldots a_{n-1}$ is defined

Partial combinatory algebras

Let A be a set with a partial operation $A \times A \dashrightarrow A$. An expression is, inductively, a variable, a constant or the application of an expression to an expression.

A is a partial combinatory algebra when for all expression e in variables $x_0, \ldots, x_n$, there exists $a \in A$ such that for all $a_0, \ldots, a_n \in A$:

1. $a a_0 \ldots a_{n-1} a_n \simeq e[a_0/x_0, \ldots, a_n/x_n]$ where $\simeq$ means one is defined iff the other is, and then they have the same value
2. $a a_0 \ldots a_{n-1}$ is defined

Reason for (2): $p \wedge q \Rightarrow r$ implies $p \Rightarrow q \Rightarrow r$ intuitionistically, so we need currification that gives a defined function if we apply it with fewer arguments.

Partial combinatory algebras

Let A be a set with a partial operation $A \times A \dashrightarrow A$. An expression is, inductively, a variable, a constant or the application of an expression to an expression.

A is a partial combinatory algebra when for all expression e in variables $x_0, \ldots, x_n$, there exists $a \in A$ such that for all $a_0, \ldots, a_n \in A$:

1. $a a_0 \ldots a_{n-1} a_n \simeq e[a_0/x_0, \ldots, a_n/x_n]$ where $\simeq$ means one is defined iff the other is, and then they have the same value
2. $a a_0 \ldots a_{n-1}$ is defined

Reason for (2): $p \wedge q \Rightarrow r$ implies $p \Rightarrow q \Rightarrow r$ intuitionistically, so we need currification that gives a defined function if we apply it with fewer arguments.

For SKI fans, this is equivalent to having $k, s \in A$ such that $kab = a$, $sabc = ac(bc)$ and sab defined.

Examples of partial combinatory algebras

Fundamental example: "Kleene's first algebra" $\mathcal{K}_1$ is $\mathbb{N}$ — Gödel codes of Turing machines — with application by execution of Turing machines

More pcas:

- Untyped λ-terms
- Better: every model of the untyped λ-calculus
- Infinite-time Turing machines, used to make $\mathbb{R}$ and $2^{\mathbb{N}}$ subcountable
- Turing machines with a fixed oracle

The maze of higher-order computability

Giving a computational interpretation to higher-order logic involves some choices.

What is a computable function $\mathbb{N} \to \mathbb{N}$? Answered by the pca, e.g., Church-Turing model.

What is a computable function $2^{\mathbb{N}} \to \mathbb{N}$? Several ideas:

The maze of higher-order computability

Giving a computational interpretation to higher-order logic involves some choices.

What is a computable function $\mathbb{N} \to \mathbb{N}$? Answered by the pca, e.g., Church-Turing model.

What is a computable function $2^{\mathbb{N}} \to \mathbb{N}$? Several ideas:

- A Turing machine which receives a program and is guaranteed to terminate provided that the program defines a total function $f : \mathbb{N} \to 2$
- Same, but the output must only depend on f, not on the program ($\longrightarrow$ function extensionality)

The maze of higher-order computability

Giving a computational interpretation to higher-order logic involves some choices.

What is a computable function $\mathbb{N} \to \mathbb{N}$? Answered by the pca, e.g., Church-Turing model.

What is a computable function $2^{\mathbb{N}} \to \mathbb{N}$? Several ideas:

- A Turing machine which receives a program and is guaranteed to terminate provided that the program defines a total function $f : \mathbb{N} \to 2$
- Same, but the output must only depend on f, not on the program ($\longrightarrow$ function extensionality)
- A Turing machine with an oracle
- A Turing machine with an oracle, but only required to terminate provided the oracle is computable (not equivalent!!)

The maze of higher-order computability

Giving a computational interpretation to higher-order logic involves some choices.

What is a computable function $\mathbb{N} \to \mathbb{N}$? Answered by the pca, e.g., Church-Turing model.

What is a computable function $2^{\mathbb{N}} \to \mathbb{N}$? Several ideas:

- A Turing machine which receives a program and is guaranteed to terminate provided that the program defines a total function $f : \mathbb{N} \to 2$
- Same, but the output must only depend on f, not on the program ($\longrightarrow$ function extensionality)
- A Turing machine with an oracle
- A Turing machine with an oracle, but only required to terminate provided the oracle is computable (not equivalent!!)

What is a computable function $(2^{\mathbb{N}} \to \mathbb{N}) \to \mathbb{N}$?

Assemblies

The primary object in realizability is an assembly: A set X together with a realizability relation $\Vdash$ between the pca A and X, where $a \Vdash x$ is read "a realizes x", such that every x is realized by some a.

Basic assembly: $\mathbb{N}$ where $\lceil n \rceil$ realizes n.

Assemblies

In the category of assemblies over A, the exponential X^Y is made of functions $Y \to X$ which are realized, where a realizes f when for every x and for every b realizing x, the application ab is defined and realizes $f(x)$.

Assemblies

In the category of assemblies over A, the exponential X^Y is made of functions $Y \to X$ which are realized, where a realizes f when for every x and for every b realizing x, the application ab is defined and realizes $f(x)$.

Example:

- $\mathbb{N} \to \mathbb{N}$ is the assembly of computable functions; each f is realized by the Gödel codes of Turing machines computing f.
- Elements of $2^{\mathbb{N}} \to \mathbb{N}$ take computable bit sequences to natural numbers; each f is realized by a when a takes every program computing a bit sequence u to $f(u)$ (no termination guarantee if input does not compute a bit sequence)

Assemblies

In the category of assemblies over A, the exponential X^Y is made of functions $Y \to X$ which are realized, where a realizes f when for every x and for every b realizing x, the application ab is defined and realizes $f(x)$.

Example:

- $\mathbb{N} \to \mathbb{N}$ is the assembly of computable functions; each f is realized by the Gödel codes of Turing machines computing f.
- Elements of $2^{\mathbb{N}} \to \mathbb{N}$ take computable bit sequences to natural numbers; each f is realized by a when a takes every program computing a bit sequence u to $f(u)$ (no termination guarantee if input does not compute a bit sequence)

The category of assemblies is quite well-behaved: it has finite limits, finite colimits and exponentials, and it is regular.

But it is not a topos because we lack a notion of truth values.

Realizability toposes

The realizability topos over a pca A is obtained by a more sophisticated construction with setoid-like objects.

Intuition: The object of truth values Ω is $\mathcal{P}(A)$, and to realize equality of truth values $p, q \subseteq A$, we must provide a realizer that converts an element of p to an element of q and vice-versa.

Realizability toposes

The realizability topos over a pca A is obtained by a more sophisticated construction with setoid-like objects.

Intuition: The object of truth values Ω is $\mathcal{P}(A)$, and to realize equality of truth values $p, q \subseteq A$, we must provide a realizer that converts an element of p to an element of q and vice-versa.

The effective topos is the realizability topos over $\mathcal{K}_1$.

- Internal Church Thesis: every function $\mathbb{N} \to \mathbb{N}$ is computable
- Failure of excluded middle
- Failure of analytic limited principle of omniscience
- All real functions are continuous
- Any non-computably-enumerable subset of $\mathbb{N}$ is subcountable but uncountable
- An immune set is neither finite nor infinite (injection from $\mathbb{N}$)
- But countable choice!

Parameterized realizability toposes

If we "adjoin an oracle" to be able to compute a new sequence, diagonalization kicks in and gives us a real not in the sequence.

Idea: introduce parameters into the pca.

Application: oracle Turing machines but parameterized by the oracle.

A function $X \to Y$ between assemblies is realized by those $a \in A$ such that for all x, for all b realizing x, and for *all* parameter p, the application $a \cdot_p b$ is defined and realizes $f(x)$.

The Miller sequence construction

Notion of oracle representing a real when it encodes a sequence of rationals rapidly converging to the real (rapidly = specified modulus of convergence). Oracle representing a sequence of reals.

Miller sequence: A sequence $\mu : \mathbb{N} \to [0, 1]$ such that if x is a real and n is a program representing x when given *any* oracle representing μ, then actually x appears in μ.

Construction uses Kakutani's fixed point theorem, a generalization of Brouwer's fixed point theorem. Very classical: in the effective topos, there exists a map from the unit square to itself that moves every point by a positive distance.

In the parameterized realizability topos where oracles are those which represent a given Miller sequence, the reals are countable!