The Real Continuum in Cubical Agda: A Formal and Syntactic Taxonomy of 33 Constructive Definitions

Formalization Taxonomy Report

November 22, 2025

Abstract

In the classical mathematical tradition, the real numbers $\mathbb{R}$ are regarded as a monolithic, canonical structure. However, when transposed into the constructive framework of Cubical Agda, this unity fractures into a diverse ecosystem of distinct mathematical objects. This report provides an exhaustive analysis of thirty-three distinct definitions of real numbers, classifying them by their structural properties (Dedekind-type, Cauchy/HIT-type, Stream-type, etc.) and examining the subtle obstructions that prevent their collapse into a single entity in the absence of the Axiom of Choice.

Contents

1 Introduction: The Constructive Fracture of the Continuum

In the classical mathematical tradition, the real numbers $\mathbb{R}$ are regarded as a monolithic, canonical structure. Whether one approaches them via the geometric intuition of the complete line, the analytic rigor of Cauchy sequences, or the algebraic precision of Dedekind cuts, the result is provably isomorphic. In the classical view, these definitions differ only in implementation details, all converging to the unique complete Archimedean ordered field. This isomorphism is so deeply ingrained that working mathematicians rarely distinguish between "the real number defined by a cut" and "the real number defined by a sequence."

However, this uniqueness is an artifact of classical logic, specifically dependent on the Law of Excluded Middle (LEM) and the Axiom of Choice (AC). When we transpose the foundations of analysis into a constructive framework—specifically, the computational and homotopy-theoretic setting of Cubical Agda—this unity fractures. Without the non-constructive principles that smooth over the gaps, the "Real Numbers" dissolve into a diverse ecosystem of distinct mathematical objects.

The research materials provided identify thirty-three distinct definitions of real numbers. The user's query strikes at the heart of this constructive diversity: *Are these definitions syntactically different, and are they implementable in Cubical Agda?*

The answer to both questions is a definitive yes, but with profound qualifications. Syntactically, these definitions range from logical predicates and coinductive streams to higher inductive types (HITs) and algebraic homomorphisms. They are not merely different "views" of the same object; in Agda, they are distinct types with incompatible introduction and elimination rules. While they are almost all implementable, their equivalence is often unprovable. This report provides an exhaustive, expert-level analysis of this fracture. We will dissect the syntax, semantics, and implementation details of all 33 definitions, classifying them by their structural properties and examining the subtle obstructions that prevent their collapse into a single entity.

We proceed by categorizing these definitions into the families identified in the source literature: the Dedekind-type completions (Class A), the Cauchy/HIT completions (Class B), the

Representation/Stream types (Class C), the Coalgebraic and Domain-theoretic variants (Classes D and F), and the axiomatic or synthetic outliers (Classes G and H).

2 Foundations: Syntax and Logic in Cubical Agda

Before analyzing the specific definitions, it is necessary to establish the metatheoretical context of Cubical Agda. This system is based on Martin-Löf Type Theory (MLTT) extended with the principles of Homotopy Type Theory (HoTT), specifically the Univalence Axiom and Higher Inductive Types (HITs).

2.1 The Syntactic Reality

In Agda, a definition is a Type. A "syntactic difference" is not a trivial formatting distinction; it implies a difference in the memory layout and the operational semantics of the object.

- A **function type** ($\mathbb{N} \to \mathbb{Q}$) represents a process: given a natural number, it computes a rational.
- A **sigma type** ($\Sigma(x : A)P(x)$) represents a pair: a value and a proof.
- A **data type** (inductive) represents a tree structure built from constructors.
- A **record type** (coinductive) represents a stream of observations.

The 33 definitions utilize the full spectrum of these constructs. The difference between a Dedekind real (a predicate) and a Cauchy real (a sequence) is the difference between a logical specification and an algorithmic process.

2.2 The Role of the Axiom of Choice

The primary driver of the separation between these definitions is the absence of Countable Choice (AC_ω). In classical mathematics, we freely convert between "existence" and "data." If we know that for every n, there exists a rational q_n close to a real number x, AC_ω allows us to collect these witnesses into a sequence (q_n). In constructive type theory, purely logical existence (propositional truncation) does not grant access to the witness. Thus, we cannot extract a Cauchy sequence from a Dedekind cut without additional logical assumptions. This renders the types $\mathbb{R}_D$ and $\mathbb{R}_C$ distinct: one contains "more" information (or rather, less accessible information) than the other.

2.3 Higher Inductive Types (HITs)

Cubical Agda introduces a crucial tool: the HIT. Standard constructive analysis often struggles with quotients (e.g., defining reals as Cauchy sequences modulo convergence to zero). In standard Agda, one uses "Setoids"—sets equipped with an equivalence relation—which complicates every function definition. In Cubical Agda, HITs allow us to define quotients as true types where the equality path is generated by the relation. This allows definitions like the Cauchy reals and Eudoxus reals to be implemented as first-class citizens, rather than second-class setoids.

3 Class A: The Dedekind-Type Completions

The first and most robust class of definitions treats the real number as a "cut" in the rational line. This is the topological view: a real number is a *place* in the order.

3.1 Definition 1: $\mathbb{R}_D$ (Dedekind Reals)

The Dedekind real is widely considered the standard definition for constructive mathematics when "correctness" (theoretical properties like completeness) is prioritized over computational efficiency.

Syntactic Structure in Agda: Syntactically, $\mathbb{R}_D$ is implemented as a record containing two predicates, L (Lower cut) and U (Upper cut), acting on the rational numbers $\mathbb{Q}$.

```
record DedekindReal : Type where
  field
    L : Q -> Type
    U : Q -> Type
    isPropL : (q : Q) -> isProp (L q)
    isPropU : (q : Q) -> isProp (U q)
    inhabitedL : exists q, L q
    inhabitedU : exists q, U q
    roundedL : forall q p, q < p -> L p -> L q
    roundedU : forall q p, q < p -> U q -> U p
    disjoint : forall q, L q -> U q -> Bottom
    located : forall q p, q < p -> (L q) \/ (U p)
```

Insight and Nuance: The presence of `isProp` is vital. In Homotopy Type Theory, "Sets" are types where equality is a proposition. If $L(q)$ were a general type (containing data), a single real number would contain multiple distinct proofs of being "greater than 0," potentially making the real number strictly more complex than a simple point. The propositional truncation ensures $\mathbb{R}_D$ is a Set. The **Locatedness** property ($L(q) \vee U(p)$) is the computational engine of the Dedekind real; it allows us to decide, for any two rationals, where the real number lies relative to them. However, it does not allow us to compare the real number to a *single* rational q (i.e., we cannot decide $x < q \vee x > q \vee x = q$).

Implementation Status: Fully implementable in Cubical Agda. It forms a Dedekind-complete Archimedean ordered field.

3.2 Definition 14: $\mathbb{R}_{formal}$ (Formal/Locale Reals)

While mathematically isomorphic to $\mathbb{R}_D$ in topos theory, the syntax of the **Formal Real** differs significantly. It arises from Point-free Topology (Locale Theory).

Syntactic Structure: Instead of defining points directly, one defines the "Frame" of open sets of the real line, denoted $\mathcal{O}(\mathbb{R})$. A "real number" is then defined as a *point* of this locale.

- **Agda Syntax:** A point is a completely prime filter on the frame of formal opens. Syntactically, this looks like a predicate on the set of "basic open intervals" (p, q) satisfying axioms that mirror the topology (e.g., if the point is in (p, q), it must be in some smaller interval covering it).

Comparison: $\mathbb{R}_D$ defines the real via arithmetic inequalities ($x < q$). $\mathbb{R}_{formal}$ defines the real via topological membership ($x \in U$). While the underlying data (rational bounds) is identical, the library structure needed to support them (Order Theory vs. Formal Topology) is distinct.

3.3 Definitions 22 & 23: Sheaf-Theoretic Reals and the RNO

The research material distinguishes between "Sheaf-theoretic reals" and the "Real Numbers Object" (RNO) in a topos.

Context: These definitions are categorical. In the internal language of a topos, the RNO is the object that plays the role of $\mathbb{R}$.

- **Definition 22 (Sheaf Reals):** Specifically refers to the sheaf of continuous functions (or Dedekind cuts) on a topological space (or site). In Agda, if we work within a specific modality or cohesive topos structure, this defines the reals "globally" over the base category.
- **Definition 23 (RNO):** This is the abstract interface. Any object satisfying the axioms of a Dedekind-complete field in the internal language is *an* RNO.

Implementation: In plain Cubical Agda, these collapse into $\mathbb{R}_D$. However, if one formalizes Category Theory *within* Agda, Definition 23 becomes a record type describing a strictly categorical object (an object R, morphisms $+, *$, etc.), whereas Definition 1 is a concrete instance of that object.

4 Class B: The Cauchy and HIT-Type Completions

This class defines real numbers as the limit of a process. This is the definition most familiar to computer scientists, as it directly models computation: a real number is a program that outputs rational approximations.

4.1 Definition 2: $\mathbb{R}_C$ (Cauchy Reals)

The classical definition: equivalence classes of Cauchy sequences.
Syntactic Structure:

```
record CauchySequence : Type where
  field
    seq : Nat -> Rat
    cauchy : forall (eps : Rat) -> eps > 0 -> exists (N : Nat) ->
             forall (n m : Nat) -> n > N -> m > N ->
             abs(seq n - seq m) < eps
```

The Quotient Problem: The type above defines *sequences*, not numbers. To get $\mathbb{R}_C$, we must quotient by the relation $(x_n) \sim (y_n) \iff \lim |x_n - y_n| = 0$.

- **In Standard Agda:** This is a Setoid (a pair of a Type and a Relation). Functions must be proven to respect the relation (congruence).
- **In Cubical Agda:** We use a Higher Inductive Type (HIT) to define the quotient $\mathbb{R}_C$ as a true type.

  ```
  data CauchyReal : Type where
    inc : CauchySequence -> CauchyReal
    eq : (s1 s2 : CauchySequence) -> (s1 ~ s2) -> inc s1 == inc s2
  ```

Implementation Status: Implementable, but computationally awkward. The `cauchy` field uses an existential quantifier ($\exists N$). In constructive logic, extracting this N to compute bounds for addition or multiplication is difficult without a constructive form of choice or a specific "modulus of convergence."

4.2 Definition 4: $\mathbb{R}_{FC}$ / $\mathbb{R}_I$ (Fast Cauchy / Interval Reals)

To fix the computational awkwardness of $\mathbb{R}_C$, we introduce **Fast Cauchy Reals**.
Syntactic Structure: We replace the existential `cauchy` proof with a specific, uniform modulus.

- **Standard Modulus:** $|x_n - x_m| < \frac{1}{n} + \frac{1}{m}$.
- **Geometric Modulus:** $|x_n - x_{n+1}| < 2^{-n}$.

Syntactic Difference: This is a massive syntactic shift. The type no longer contains a logic-heavy proof object; it contains a purely arithmetical proof that the sequence stays within fixed bounds.

- **Agda Implementation:**

  ```
  record FastCauchy : Type where
    field
      seq : Nat -> Rat
      valid : (n : Nat) -> abs (seq n - seq (n+1)) < 1/2^n
  ```

Insight: $\mathbb{R}_{FC}$ is the workhorse of Exact Real Arithmetic. Addition is defined simply as pointwise addition with a shift in index. However, $\mathbb{R}_{FC}$ is technically a *subset* of the raw Cauchy sequences. The map from $\mathbb{R}_{FC}$ to $\mathbb{R}_C$ is trivial; the reverse map requires finding a subsequence, which is computable but syntactically nontrivial.

4.3 Definition 12: $\mathbb{R}_H$ (HoTT/HIT Reals)

This definition is native to Cubical Agda and Homotopy Type Theory. It seeks to define $\mathbb{R}$ not as a quotient of $\mathbb{Q}^{\mathbb{N}}$, but as the free Cauchy-complete object generated by $\mathbb{Q}$.

Syntactic Structure: The definition typically follows the "Cauchy Approximation" inductive family pattern.

```
data Real : Type where
  rat : Rat -> Real
  lim : (s : Nat -> Real) -> IsCauchy s -> Real
  eq  : (u v : Real) -> isClose u v -> u == v
```

Distinctness: $\mathbb{R}_H$ is "inductively generated." It is the smallest type containing rationals and closed under limits. This makes it syntactically distinct from $\mathbb{R}_C$ (which is a projection from the function space $\mathbb{N} \to \mathbb{Q}$).

4.4 Definition 13: $\mathbb{R}_{ES}$ (Escardó-Simpson Reals)

Definition: The interval domain definition of reals, or specifically the "least Cauchy-complete subobject of $\mathbb{R}_D$." **Syntax:** A real is a Dedekind real $x : \mathbb{R}_D$ combined with a proof that x is in the inductive closure of $\mathbb{Q}$ under limits.

- **Agda:** $\Sigma(x : \mathbb{R}_D)(\text{InClosure}(x))$.

This definition bridges Class A and Class B. It explicitly selects the "computable" Dedekind reals (those reachable by limits) and bundles them into a type.

4.5 Definitions 31 & 32: Apartness and Filter Reals

- **Definition 31 (Apartness Reals):** Bishop-style constructive analysis focuses heavily on the "Apartness" relation ($x\#y \iff |x - y| > 0$). Bishop reals are essentially $\mathbb{R}_{FC}$ but the entire theory is built around ensuring operations respect apartness.
- **Definition 32 (Filter Reals):** A real is a Cauchy Filter on $\mathbb{Q}$. A filter is a set of sets of rationals. This is the topological dual to the sequence definition.

5 Class C: Representation and Stream Types

These definitions are grounded in computer science and digital representation. They view real numbers as infinite streams of data. Syntactically, they are **Coinductive Types**.

5.1 Definition 6: $\mathbb{R}_b$ (Base-b Reals)

The standard decimal (or binary) expansion.

Syntactic Structure:

```
record Stream (A : Type) : Type where
  coinductive
  field
    head : A
    tail : Stream A

Definition Base10Real := Integer x Stream (Fin 10)
```

The Problem: While fully implementable, this type is pathological for arithmetic. Addition is not computable due to the "carry propagation" problem. To know the first digit of $0.333... + 0.666...$, one may need to inspect an arbitrary number of future digits. Thus, $\mathbb{R}_b$ is not a field. It is a "Pre-Real".

5.2 Definition 30: Decimal Base-10 Cauchy Reals

To fix the arithmetic problem of $\mathbb{R}_b$, one must quotient the stream type by the equivalence relation $0.999... \sim 1.000...$.

- **Syntax:** $\mathbb{R}_{dec} = \mathbb{R}_b / \sim$.

In Cubical Agda, this is a HIT quotient of the stream type. This restores the field properties (abstractly), but it doesn't fix the computational blockage of addition on the raw streams.

5.3 Definition 7: $\mathbb{R}_{SD}$ (Signed-Digit Reals)

This is the standard solution for **Exact Real Arithmetic** in computer science. **Syntactic Structure:** A stream of digits from the set $\{-1, 0, 1\}$ (often denoted $\bar{1}, 0, 1$). **Insight:** The redundancy (e.g., $0.1\bar{1}\cdots = 0.01\ldots$ in binary) allows arithmetic algorithms to "delay" decisions and absorb carries. Addition becomes a stream-processing function with limited lookahead.

- **Agda Implementation:** Fully implementable as a coinductive record. This is distinct from $\mathbb{R}_C$ because it fixes the "basis" of convergence (powers of 2), whereas $\mathbb{R}_C$ allows any rational approximation.

5.4 Definition 5: $\mathbb{R}_{CF}$ (Continued Fraction Reals)

Syntactic Structure: A stream of natural numbers (and one integer) $[a_0; a_1, a_2, \ldots]$. **Properties:** Efficient for certain operations (Möbius transformations). However, standard addition is surprisingly complex.

6 Class D: Coalgebraic Definitions

This class abstracts the "Stream" concept using Category Theory.

6.1 Definitions 20 & 21: Unit Interval and Positive Reals as Terminal Coalgebras

Concept: A "Coalgebra" for a functor F is a type X equipped with a map $X \to F(X)$. This models a state machine where F describes the possible transitions/observations.

- The **Terminal Coalgebra** is the "universal" state machine. Every other such machine maps uniquely into it.

Syntactic Difference: In Agda, this is defined via **Copatterns**.

```
record TerminalReal : Type where
  coinductive
  field
    observe : Digit
    next    : TerminalReal
```

While this looks like $\mathbb{R}_b$, the definition asserts a *Universal Property*. The definition is: "The unique object νF such that..."

7 Class E & F: Generalized and Domain-Theoretic Reals

These definitions relax the strictness of Dedekind cuts, creating "larger" spaces.

7.1 Definitions 9 & 10: $\mathbb{R}_L$ / $\mathbb{R}_U$ (Lower and Upper Reals)

Syntactic Structure: A **Lower Real** is defined by a single predicate $L : \mathbb{Q} \to$ Type that is open and inhabited. It lacks the "Upper Cut" and the "Locatedness" property. **Meaning:** $\mathbb{R}_L$ represents numbers that are "approximable from below" but potentially unknowable from above. It includes standard reals and $+\infty$, but also "semicontinuous" values. **Agda:** Trivially implementable. Syntactically, it is half a Dedekind record.

7.2 Definition 11: $\mathbb{R}_M$ (MacNeille Reals)

Concept: MacNeille reals require the cut to be **double-negation closed**: $\neg\neg(q \in L) \to q \in L$. **Logical Insight:** In classical logic, $\neg\neg P \iff P$, so $\mathbb{R}_M \cong \mathbb{R}_D$. In constructive logic, $\mathbb{R}_M$ is the completion of $\mathbb{Q}$ with respect to the $\neg\neg$-topology. **Implementation:** Implementable, but produces a distinct type that behaves differently regarding extracting witnesses.

7.3 Definition 8: $\mathbb{R}_{ID}$ (Interval Domain Reals)

Context: Domain Theory (Scott Domains) models computation with partial information. The **Interval Domain** consists of all rational intervals $[p, q]$ ordered by reverse inclusion ($[a, b] \sqsubseteq [c, d] \iff [c, d] \subseteq [a, b]$). **Definition:** The reals are the **maximal elements** of this domain (the intervals that shrink to a single point). **Syntactic Structure:** In Agda, this is often formalized as a filter on the poset of intervals. **Insight:** This definition naturally supports "partial reals" (intervals that haven't shrunk to 0 width yet). It is the foundation of **Interval Arithmetic**.

8 Class H & Isolated: The Exotic Definitions

8.1 Definition 3: $\mathbb{R}_E$ (Eudoxus Reals)

This definition is marked as "Isolated". It constructs reals directly from Integers $\mathbb{Z}$, skipping $\mathbb{Q}$ entirely. **Definition:** A real number is a function $f : \mathbb{Z} \to \mathbb{Z}$ that is "almost linear." Condition:

The set of "slopes" $\{f(x+y) - f(x) - f(y) \mid x, y \in \mathbb{Z}\}$ must be bounded. **Quotient:** $f \sim g$ if $\{f(x) - g(x) \mid x \in \mathbb{Z}\}$ is bounded. **Intuition:** The function $f(x) = \lfloor \alpha x \rfloor$ represents the real number α. **Agda Implementation:** Requires a HIT for the quotient. The operations are defined purely on $\mathbb{Z}$. Addition is $(f+g)(x) = f(x)+g(x)$. **Syntactic Difference:** Radical. The underlying data is `Integer -> Integer`. No Rationals, no Cuts, no Sequences.

8.2 Definition 24: $\mathbb{R}_{SDG}$ (Smooth Reals)

Context: Synthetic Differential Geometry. **Definition:** The reals are a ring R containing an element $d \neq 0$ such that $d^2 = 0$ (a nilpotent infinitesimal). **Implementability: Impossible** in standard Agda arithmetic. The existence of such a d contradicts the field axiom $x \neq 0 \implies x$ invertible. **Solution:** One implements this by working in a **Smooth Topos**.

8.3 Definition 27: Surreal Numbers (No)

Context: Conway's Game Theory. **Definition:** Recursive cut construction where Left and Right sets can be arbitrary sets of previously defined numbers. **Agda:** Requires **Induction-Recursion** because the definition of the type No depends on the order defined *on* No. `data No : Type where cut : (L R : Set) -> (L -> No) -> (R -> No) -> ... -> No.`

9 Table of the 33 Definitions: Class and Syntax

ID	Name	Class	Syntactic Base in Agda	Implementability
1	$\mathbb{R}_D$ (Dedekind)	A	Record of Predicates	**Full**
2	$\mathbb{R}_C$ (Cauchy)	B	HIT (Quotient of Seq)	**Full** (via HITs)
3	$\mathbb{R}_E$ (Eudoxus)	H	HIT (Quotient of $\mathbb{Z} \to \mathbb{Z}$)	**Full** (Isolated)
4	$\mathbb{R}_{FC}$ (Fast Cauchy)	B	Record (Seq + Modulus)	**Full** (Standard)
5	$\mathbb{R}_{CF}$ (Continued Frac)	C	Stream $\mathbb{N}$	**Full**
6	$\mathbb{R}_b$ (Base-b)	C	Stream (Fin b)	**Full** (Pre-Real)
7	$\mathbb{R}_{SD}$ (Signed Digit)	C	Stream $\{-1, 0, 1\}$	**Full**
8	$\mathbb{R}_{ID}$ (Interval Domain)	F	Poset Filter	**Full**
9	$\mathbb{R}_L$ (Lower)	E	Predicate (Open)	**Full**
10	$\mathbb{R}_U$ (Upper)	E	Predicate (Open)	**Full**
11	$\mathbb{R}_M$ (MacNeille)	E	Predicate ($\neg\neg$-closed)	**Full**
12	$\mathbb{R}_H$ (HoTT/HIT)	B	Inductive Type (HIT)	**Native**
13	$\mathbb{R}_{ES}$ (Escardó-Simpson)	B	Σ-Type (Subtype of $\mathbb{R}_D$)	**Full**
14	$\mathbb{R}_{formal}$ (Locale)	A	Point of Frame	**Full**
15	$\mathbb{R}_{init}$	G	Axiomatic Interface	Interface
16	$\mathbb{R}_{term}$	G	Axiomatic Interface	Interface
17	$\mathbb{R}_{DedComp}$	G	Axiomatic Interface	Interface
18	$\mathbb{R}_{CauComp}$	G	Axiomatic Interface	Interface
19	$\mathbb{R}_{Tarski}$	G	Axiomatic Interface	Interface
20	$_{coalg}$	D	Coinductive Record	**Full**
21	$\mathbb{R}^+_{coalg}$	D	Coinductive Record	**Full**

22	Sheaf Reals	A	Category Object	Meta-level
23	RNO	A	Category Object	Meta-level
24	$\mathbb{R}_{SDG}$ (Smooth)	H	Axiomatic (Nilpotents)	**Non-Standard**
25	$^*\mathbb{R}$ (Hyperreals)	H	Ultrafilter Construction	**Limited**
26	Predicative Reals	H	Restricted Cuts	**Full**
27	Surreals (No)	H	Inductive-Recursive	**Complex**
28	Geometric Reals	H	Axiomatic Points	Interface
29	Computable Reals	H	Turing Machines	Meta-level
30	Decimal Reals	C	HIT (Quotient of $\mathbb{R}_b$)	**Full**
31	Apartness Reals	B	Record (Seq + Apartness)	**Full**
32	Filter Reals	B	Cauchy Filters	**Full**
33	Locale Variants	A	Frame Points	**Full**

10 Syntactic Comparison: The Logic of Embeddings

The user asks: "Are they syntactically different?" We have established they are. But are they *convertible*?

10.1 The One-Way Street: Cauchy $\to$ Dedekind

There exists a canonical embedding $\mathbb{R}_C \hookrightarrow \mathbb{R}_D$.

- **Mechanism:** Given a Cauchy sequence (q_n), define the Lower Cut $L(x) = \exists n, \forall m > n, q_m > x + \epsilon$. (Roughly: the cut is the set of rationals eventually smaller than the sequence).
- **Constructivity:** This direction is purely constructive. It requires no choice, only arithmetic.

10.2 The Blockade: Dedekind $\to$ Cauchy

There is **no** constructive map $\mathbb{R}_D \to \mathbb{R}_C$.

- **The Obstacle:** Countable Choice (AC_ω). To map a Cut to a Sequence, we need to pick a rational q_0 in the cut, then q_1 closer to the boundary, etc. The Cut predicate $L(x)$ gives us the property $\exists q, q \in L \land q \approx$ boundary. It does not give us the *data* of q. Without AC_ω, we cannot extract the sequence.
- **Implication:** In Cubical Agda, $\mathbb{R}_D$ is a strictly "larger" (or more inclusive) type than $\mathbb{R}_C$.

10.3 The Stream Gap: $\mathbb{R}_{SD} \to \mathbb{R}_C$

There is a map from Signed Digits to Cauchy Sequences (summing the series).

- **The Inverse:** Mapping Cauchy to Signed Digits is **discontinuous** (computable only if we allow partiality). We cannot determine the first digit of a Cauchy real until the sequence settles enough, but "settling enough" might take forever if the number is exactly at a boundary.

11 Deep Analysis of the "Cubical" Implementability

The user specifically asks about "Cubical Agda." This implies the use of: 1. **Univalence:** $(A \simeq B) \to (A \equiv B)$. 2. **HITs:** Types with path constructors.

11.1 The Power of HITs for Real Numbers

Standard Type Theory struggles with $\mathbb{R}_C$ because the quotient is a setoid. Cubical Agda solves this perfectly.

- **Implementation:** We define `CauchyReal` as a HIT.
- **Consequence:** We can prove $\forall x, y : \mathbb{R}_C, (x - y \to 0) \to x \equiv y$. The path $x \equiv y$ is now a first-class citizen.

11.2 Univalence and the Fracture

One might hope Univalence heals the fracture. If we could prove $\mathbb{R}_C \simeq \mathbb{R}_D$, Univalence would make them equal.

- **The Verdict:** Univalence does **not** help here. The equivalence $\mathbb{R}_C \simeq \mathbb{R}_D$ is simply *false* without Choice. Univalence only equates equivalent types; it does not force non-equivalent types to be equal. Thus, even in the powerful Cubical setting, the distinction between Definitions 1 and 2 remains rigid.

12 Conclusion: A Landscape of Distinct Objects

The thirty-three definitions of real numbers listed in the research material are not redundant. They represent a fractured landscape where "The Real Numbers" is not a single location, but a category of related objects.

1. **Syntactically Different:** Yes. They use fundamentally different type formers (Inductive vs Coinductive vs Higher Inductive vs Predicates).
2. **Implementable in Cubical Agda:** Yes, almost all are implementable. Cubical Agda is uniquely powerful in implementing the quotient-based definitions ($\mathbb{R}_C, \mathbb{R}_E, \mathbb{R}_{dec}$) correctly via HITs.
3. **The Fracture:** The definitions form a hierarchy.
   - $\mathbb{R}_D$ (Class A) is the standard for geometric/topological correctness.
   - $\mathbb{R}_{FC}$ / $\mathbb{R}_{SD}$ (Class B/C) are the standards for computation and extraction.
   - $\mathbb{R}_H$ (Class B) is the native Homotopy-Theoretic definition.

For the researcher or developer working in Cubical Agda, the choice of definition is an engineering decision. One does not simply "import Real." One selects the Dedekind reals for proving theorems about existence and bounds, and the Signed Digit reals for running computations. The 33 definitions are the distinct tools required to navigate the constructive continuum.