

---

$$\left\| \Sigma q. \text{rat } q \underset{[\varepsilon]}{\sim} x \right\|$$

# Kraus's Magic Trick for Computing with Constructive Reals

---

Riccardo Borsetto · University of Verona

*joint work with Lorenzo Molena & Marcin Jan Turek-Grzybowski*

# A real number is abstract, yet we want its digits

## TWO VIEWS OF A REAL NUMBER

- an **abstract object**: fixed by the axioms of a Cauchy-complete Archimedean ordered field;
- a **concrete representation**: one that computes approximations.

## THE CONTRAST

*classical*: function exists, not an algorithm  
*constructive*: no uniform extractor, closed terms compute

How can an abstract  
real  
ever reveal a digit?

*One way to get computation back  
without changing the type of reals.*

# Why constructive, and why cubical?

## WHY CONSTRUCTIVE

A constructive existence proof is a **witness**: an actual term, not merely the impossibility of its absence.

→ for a real, that witness is a rational approximation we can feed to a computation.

## WHY CUBICAL

Equality is a **path that computes**: closed proofs reduce to values. Completions are higher inductive types, with no setoid bookkeeping.

→ so a closed approximation proof actually evaluates to a rational.

| Together: a witness **exists**, and the proof **computes**. That turns the magic trick into a digit.

# The reals are an internal completion of the rationals

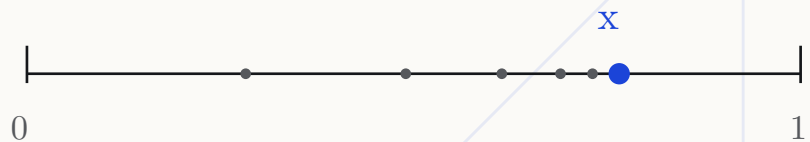
## A HIGHER-INDUCTIVE CAUCHY COMPLETION

The HoTT-Book reals are  $\mathbb{R}$ , built from  $\mathbb{Q}$  with the limits and identifications **inside** the type.

**rat**  $\mathbb{Q} \rightarrow \mathbb{R}$

**lim** Cauchy approximations give points

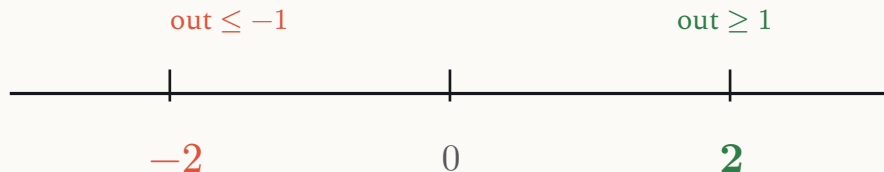
**eq** arbitrarily close points are equal



*a limit point of rational approximations*

Rational density follows from completion:  $\iota\text{-dense} : \forall y \in \mathbb{R} \rightarrow \exists [q \in \mathbb{Q}] (y \sim [\varepsilon] \text{rat } q)$

# No uniform operation reads off a rational approximation



## WHAT WE CANNOT ASK FOR

A uniform extractor approx :  $\mathbb{R} \rightarrow \mathbb{Q}_+ \rightarrow \mathbb{Q}$ .

At fixed precision  $x \mapsto \text{approx}(x, 1)$  would be nonconstant:  $-2$  gives  $\leq -1$ ,  $2$  gives  $\geq 1$ . A nonconstant map into the discrete rationals would violate the path / closeness structure of  $\mathbb{R}$ .

Constant maps exist. The issue is a uniform, nontrivial observation.

# Rational density gives a hidden witness

## WHAT DENSITY GIVES

For every real and every precision, rational density supplies an approximation that **merely exists**.

$$\text{approx} : (x : \mathbb{R}) \rightarrow (\varepsilon : \mathbb{Q}_+) \rightarrow \left\| \sum_{q: \mathbb{Q}} \text{rat}(q) \underset{[\varepsilon]}{\sim} x \right\|$$

 $(q, p)$ 
 $\Sigma$ 

witness + proof

 $\| ? \|$ 
 $\| \cdot \|$ 

mere existence

## THE CATCH

All inhabitants are identified. The witness is **deliberately hidden**. No generic projection back to a rational.

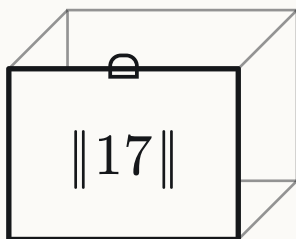
THE MAGIC TRICK

# The magic trick

*recovering a hidden value by normalisation*

# Kraus's **recover** returns a hidden value definitionally

SEALED · FORGOTTEN

*truncated · name out of scope***recover***along a path*

REVEALED · IN SCOPE

17

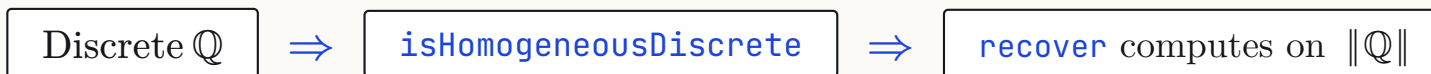
*a closed term, recovered*

**recover**  $|x| \equiv x$  *definitionally,*  
by **refl**

*after Nicolai Kraus (2013)*

```
recover | hidden | ↗ 17
```

# The trick works because the rationals are homogeneous



## THE IDEA

A type is **homogeneous** when every element is reachable from a chosen basepoint by an automorphism. Decidable equality on  $\mathbb{Q}$  provides exactly those automorphisms.

```
discreteQ : Discrete  $\mathbb{Q}$   
  
isHomogeneousDiscrete discreteQ  
  : isHomogeneous ( $\mathbb{Q}$  , 0)
```

# Recovery computes, but is not a function out of the truncation

## NOT AN INVERSE

Not a map  $\|Q\| \rightarrow Q$ : the result type **depends** on the input. Along the squash path `cong recover` is a **dependent** path over changing types (*Gross*). It is not an equality of values.

## CLOSED TERMS ONLY

It reduces for **specific closed** terms, not for an arbitrary real handed in by the environment. No global digit extractor appears.

| **Safe:** No excluded middle, no choice, no univalence. Just definitional normalisation.

# Computing the digits of concrete reals

## PROTOTYPE WORKFLOW

1. a closed real expression, e.g.  $\exp e$
2. ask for a rational at precision  $\varepsilon$ ;
3. normalise the hidden witness;
4. read off the rational / digits.

$\exp e \mid \varepsilon \rightsquigarrow [1/64]$

## HONEST STATUS

In the library `ℝrationalApprox` exists, but `recover` is not yet wired to it. This is the principle / a prototype.

```

exp : ℝ → ℝ
_√_ : ℕ → ℚ → ℝ

👤 : (x : ℝ) → (ε : ℚ) → x / ε
👤 : (x : ℝ) → (N : ℕ) → x d / N

- : ℚ
- = { 👤 (7 √ 11 + r e) [ 1 / 100000000 ]+ }1

- : ℕ × ℕ
- = { 👤 (7 √ 11 + r e) 15 }2

- : ℚ
- = { 👤 (exp e) [ 1 / 64 ]+ }3

```

# Digits computed inside Agda

```

exp : ℝ → ℝ
_√_ : ℕ+ → ℚ+ → ℝ

🧙 : (x : ℝ) → (ε : ℚ+) → x ✂ ✨ ε
🎲 : (x : ℝ) → (N : ℕ) → x d ✂ ✨ N

_ : ℚ
_ = { 🧙 (7 √ 11 +r e) [ 1 / 100000000 ]+ }1

_ : ℕ × ℕ
_ = { 🎲 (7 √ 11 +r e) 15 }2

_ : ℚ
_ = { 🧙 (exp e) [ 1 / 64 ]+ }3

```

## RECORDED NORMALISATION

A closed expression is normalised, through the approximation code, to a rational, then to digits.

*Static slide: stills stand in for the video.*

External playback: [Pacman/images/magic.mov](#)

## ON SCREEN

Closed terms only. Never an arbitrary real.

# The trick sits atop a from-algebra-to-analysis library

## THE CUBICAL AGDA DEVELOPMENT

- ordered commutative & Archimedean rings:  
 $\mathbb{Z}$ ,  $\mathbb{Z}[\frac{1}{2}]$ ,  $\mathbb{Q}$ ,  $\mathbb{R}$
- setoid-style equality and **choice-free** completion of premetric spaces;
- constructive analysis on the HoTT-Book reals: derivatives, MVT, integration, FTC,  $\exp / \sin / \cos$ ,  $\pi$ .

## THE BRIDGE

Computation reappears **term by term**, even though no uniform operation extracts digits from arbitrary reals.

The bridge: propositional truncation + definitional normalisation.

# Computation does what no uniform map can

## THE OBSTRUCTION

No uniform  $\mathbb{R} \rightarrow \mathbb{Q}$   
approximation extractor.

## THE TRICK

Kraus's recover: a hidden  
value returns **definitionally**.

## THE PAYOFF

Closed reals compute without  
choice or LEM.

Computation does, term by term,  
**what no definable map can.**

# References and further reading

- 1 Nicolai Kraus. *The truncation map  $|\cdot| : \mathbb{N} \rightarrow \|\mathbb{N}\|$  is nearly invertible*. Homotopy Type Theory blog, 2013.
- 2 Jason Gross. *Composition is not what you think it is! Why “nearly invertible” isn’t*. HoTT blog, 2014.
- 3 The Univalent Foundations Program. *Homotopy Type Theory*, Chapter 11.3. 2013.
- 4 Gaëtan Gilbert. *Formalising Real Numbers in Homotopy Type Theory*. CPP 2017.
- 5 *The Cubical Agda library*: `Cubical.HITs.PropositionalTruncation.MagicTrick` PR #1182.

# Backup: the higher-inductive shape of $\mathbb{R}$

$$\text{rat} : \mathbb{Q} \rightarrow \mathbb{R}$$

$$\text{lim} : ((\varepsilon : \mathbb{Q}_+) \rightarrow \mathbb{R}) \rightarrow \text{isCauchy} \rightarrow \mathbb{R}$$

$$\text{eq} : \left( \forall \varepsilon. x \underset{[\varepsilon]}{\sim} y \right) \rightarrow x = y$$

*Mutually defined with the closeness relation  $\sim$ ; constructor names vary between presentations.*

## Backup: why recover is safe

```
recover-squash1 : (x y : A) →  
  PathP (λ i → typ (B· (squash1 | x |1 | y |1 i))) x y
```

### THE CODOMAIN CHANGES

The path from the truncation's squash constructor lives over a **family of types**. It is not a proof that  $x = y$  in  $A$ .