The Reals as a Higher Coinductive Type?

Thorsten Altenkirch
University of Nottingham

jww chatgpt

Martin Escardó 60th Birthday Seminar

Motivation / setting

- Explore *higher coinductive types* (HCIT).
- Used in HOTT (Narya) to define the fibrant universe.
- Goal: define *signed-digit reals* as a (truncated) HCIT
- Work in progress: I want feedback on what the *right observations* / *equations* should be.
- Meta-goal: a general framework for (truncated) HCITs, analogous to HITs / QIITs.

The Real Martin

Old papers on signed reals

- M.H. Escardó. *PCF extended with real numbers.*
  Theoretical Computer Science 162(1), 79–115 (1996).
- M.H. Escardó. *Effective and sequential definition by cases on the reals via infinite signed-digit numerals.*
  Electronic Notes in Theoretical Computer Science 13, 53–68 (1998).

Recent papers on midpoint algebras / interval objects

- M.H. Escardó and A. Simpson. *A universal characterization of the closed Euclidean interval* (extended abstract).
  In: Proc. 16th IEEE Symposium on Logic in Computer Science (LICS 2001), pp. 115–125.
- A.B. Booij. *The HoTT reals coincide with the Escardó–Simpson reals.*
  CoRR abs/1706.05956 (2017). (arXiv:1706.05956).

Reminder: QIITs and HoTT reals

- In HoTT, reals can be defined as the *Cauchy completion* of rationals.
- QIITs (quotient inductive-inductive types): simultaneously generate points and equations.
- Attractive feature: avoid (countable) choice by internalising limits / Cauchy structure.

Signed-digit reals

- We define a higher coinductive type $\mathbb{I}$ of *signed reals*.
- Digits $d \in \{-1, 0, +1\}$ with constructor $d :: x : \mathbb{I}$.
- The semantics of $\mathbb{I}$ is the interval object

$$[\![\mathbb{I}]\!] \;=\; \{\, x : \mathbb{R} \mid -1 \leq x \leq 1 \,\}.$$

Higher coinductive types (HCITs): idea

- Dual flavour to HIT/QIIT:
  - HCIT $=$ *terminal object* in a category of algebras/coalgebras with equations.
  - In this talk: *truncated setting* (equalities are propositions).
- Guiding question: under what conditions does the *terminal algebra/coalgebra* exist?

Why not just streams with destructors?

- If we specify a coinductive type by destructors `hd` and `tl`, we recover ordinary streams.

$$\mathtt{hd} : \mathbb{I} \to \mathtt{Digit}$$
$$\mathtt{tl} : \mathbb{I} \to \mathbb{I}$$

- But for signed reals, this gives *too many observations*:
  - observation of head/tail should not be definable from the real equality.
  - we want controlled ambiguity (redundant digits) rather than full stream extensionality.

Start too small: only $::$

```
_::_ : Digit → 𝕀 → 𝕀
```

- Suppose the only constructor/operation is `cons` (written `d :: x`).
- Terminal algebra can collapse to a *trivial one-point solution* (no observations).
- To get non-triviality we need *some* equations/observations.

Start too large: add full injectivity + generation

If we add:

- full injectivity of `cons`
- and a generator (every element is `d :: x`)

then we essentially recover streams (and hence `hd` and `tl`).

$$\texttt{gen} : (\texttt{y} : \mathbb{I}) \rightarrow \exists\ \texttt{d} : \texttt{Digit}\ ,\ \exists\ \texttt{x} : \mathbb{I}\ ,\ \texttt{y} \equiv \texttt{d} :: \texttt{x}$$

$$\texttt{cons-inj}_1 : \texttt{d} :: \texttt{x} \equiv \texttt{e} :: \texttt{y} \rightarrow \texttt{d} \equiv \texttt{e}$$
$$\texttt{cons-inj}_2 : \texttt{d} :: \texttt{x} \equiv \texttt{e} :: \texttt{y} \rightarrow \texttt{x} \equiv \texttt{y}$$

A first attempt at "separating" equalities

Keep gen but only add *Horn clauses* that separate:

```
no-conf    : (-1 :: x ≡ +1 :: y) → ⊥
cons-inj₁  : (d :: x ≡ e :: x) → d ≡ e
cons-inj₂  : (d :: x ≡ d :: y) → x ≡ y
```

- prevents extreme head confusion (-1 vs $+1$),
- allows cancellation when heads match,
- still allows "cross-head" equalities via 0 (needed for carry/borrow),
- but may still permit semantically unwanted cross-head equalities.

Semantics to guide the equations

Define a semantics map (into Cauchy reals, say):

$$[\![d :: x]\!] \;=\; \frac{d}{2} + \frac{[\![x]\!]}{2}$$

and aim for:

- **Soundness:** $x \equiv y \Rightarrow [\![x]\!] = [\![y]\!]$.
- **Completeness (conjecture):** $[\![x]\!] = [\![y]\!] \Rightarrow x \equiv y$.

Motivate `inc` and `dec` semantically

The carry/borrow laws suggest affine maps on tails:

$$[\![\mathrm{inc}(y)]\!] = \frac{1}{2} + \frac{[\![y]\!]}{2}, \quad [\![\mathrm{dec}(y)]\!] = -\frac{1}{2} + \frac{[\![y]\!]}{2}.$$

```
inc : 𝕀 → 𝕀    -- tail-carry increment
dec : 𝕀 → 𝕀    -- tail-carry decrement
```

(These are not global +1/-1 on reals; they are “shifted” operations forced by carry/borrow.)

Corecursive clauses for inc and dec

```
inc : 𝕀 → 𝕀
inc (d-1 :: x) = d0  :: inc x
inc (d0  :: x) = d+1 :: (d0 :: x)
inc (d+1 :: x) = d+1 :: inc x

dec : 𝕀 → 𝕀
dec (d+1 :: x) = d0  :: dec x
dec (d0  :: x) = d-1 :: (d0 :: x)
dec (d-1 :: x) = d-1 :: dec x
```

Restrict cross-head equalities: carry/borrow completeness clauses

Add *completeness* directions as Horn clauses:

```
carry-compl  : (0 :: x ≡ inc y) → (-1 :: x ≡ 0 :: y)
borrow-compl : (0 :: x ≡ dec y) → (+1 :: x ≡ 0 :: y)
```

- Together with symmetry/transitivity these imply the corresponding “sep” directions.
- In particular, one can then derive the usual carry/borrow equations.

```
carry  : d+1 :: (d-1 :: x) ≡ d0 :: inc x
borrow : d-1 :: (d+1 :: x) ≡ d0 :: dec x
```

Defining the semantic map (I)

- Goal: $[\![\_]\!] : \mathbb{I} \to [\![\mathbb{I}]\!]$ with
$$[\![d :: x]\!] = \frac{d}{2} + \frac{[\![x]\!]}{2}.$$
- No structural recursion on the HCIT $\mathbb{I}$.
- Define a corecursive map in the opposite direction:

$$q : [\![\mathbb{I}]\!] \to \mathbb{I}, \qquad q\left(\frac{d}{2} + \frac{x}{2}\right) = d :: q(x) \;\; (d \in \{-1, 0, +1\}).$$

Defining the semantic map (II)

- Use terminality to obtain an evaluation witness:

$$\text{eval}(x) : \exists\, y : [\![\mathbb{I}]\!].\ q(y) = x \qquad \text{(an h-proposition).}$$

- Define semantics by projection:

$$[\![x]\!] := \pi_1(\text{eval}(x)) \qquad \Rightarrow \qquad [\![d :: x]\!] = \frac{d}{2} + \frac{d}{2} + \frac{[\![x]\!]}{2}.$$

(...and simplify the RHS to get the intended equation.)

Completeness conjecture and the "head lemma"

Conjecture:

$$[\![x]\!] = [\![y]\!] \Rightarrow x \equiv y.$$

Key technical ingredient (informal):

- A *head lemma*: if $[\![d \cdot x]\!] = [\![e \cdot y]\!]$ then
  - either $d = e$ and recurse on tails, or
  - $(d, e) = (-1, 0)$ and reduce to $0 \cdot x \equiv \mathrm{inc}(y)$, or
  - $(d, e) = (+1, 0)$ and reduce to $0 \cdot x \equiv \mathrm{dec}(y)$,
  - but never $(-1, +1)$ (boundedness).

Conditional completeness and reverse directions

To get an "iff"-style characterisation of cross-head equalities, add reverse directions:

```
sep-L : (-1 :: x ≡ 0 :: y) → (0 :: x ≡ inc y)
sep-R : (+1 :: x ≡ 0 :: y) → (0 :: x ≡ dec y)
```

- Together with `carry-compl`/`borrow-compl`: cross-head equalities are exactly those explained by carry/borrow.
- This addresses "conditional completeness" (no spurious cross-head equalities).

Summary and further work

- Formally verify:
  - soundness of the Horn clauses w.r.t. semantics,
  - completeness conjecture (using the head lemma),
  - conditional completeness with both directions.
- Compare with HoTT/QIIT reals:
  - use midpoint algebras instead of signed reals.
  - equivalence to Cauchy completion (extending Auke Booij's work).
- Develop a general framework for *(truncated) HCITs*:
  - existence of terminal objects,
  - modular presentation of observation/equation principles.

Thank you

Questions / suggestions welcome!