

Corecursion and Coinduction

Gabriel Field

15/Aug/2025

Goals

Actual goals:

- Introduce the **conatural numbers**
- Introduce **corecursion** to specifying functions into $\text{co}\mathbb{N}$
- Introduce **coinduction** to reason about corecursive functions
- **Generalise** to other corecursive structures
- Give **useful examples** of coinductive proofs

Notable omissions:

- Exhaustive examples (no time)
- Solid background theory (too much category theory)
- Initial algebras and terminal coalgebras in categories other than **Set** (prerequisites not met)

Outline

- 1 Vanilla Induction
- 2 Vanilla Corecursion
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense
- 6 Outroduction

Outline

- 1 Vanilla Induction
- 2 Vanilla Corecursion
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense
- 6 Outroduction

The Natural Numbers

Coinduction \leftarrow Corecursion \leftarrow Recursion \leftarrow \mathbb{N} .

The Natural Numbers

Coinduction \leftarrow Corecursion \leftarrow Recursion $\leftarrow \mathbb{N}$.

Loose Definition

The set of **natural numbers** is $\mathbb{N} := \{0, 1, \dots\}$, “freely generated” by the constant 0 and the successor operation $\text{succ} : x \mapsto x + 1$.

The Natural Numbers

Coinduction \leftarrow Corecursion \leftarrow Recursion \leftarrow \mathbb{N} .

Loose Definition

The set of **natural numbers** is $\mathbb{N} := \{0, 1, \dots\}$, “freely generated” by the constant 0 and the successor operation $\text{succ} : x \mapsto x + 1$.

Recursion Principle

Given a set X , a constant $x_0 \in X$, and an operation $f : X \rightarrow X$, there is a unique map $u : \mathbb{N} \rightarrow X$ with

$$\begin{aligned}u : 0 &\mapsto x_0 \\ u : \text{succ}(n) &\mapsto f(u(n))\end{aligned}$$

Recursion from \mathbb{N}

Example: Powers

The function $u : \mathbb{N} \rightarrow \mathbb{R}$, $u : n \mapsto 3^n$ is defined recursively by

$$u : 0 \mapsto 1$$

$$3^0 := 1$$

$$u : \text{succ}(n) \mapsto 3^n \cdot n$$

$$3^{\text{succ}(n)} := 3^n \cdot n$$

Example: Addition on \mathbb{N}

The function $(- + -) : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ is defined recursively by

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Example: $\forall n \in \mathbb{N}, n + 0 = n$

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Example: $\forall n \in \mathbb{N}, n + 0 = n$

Let $\text{lhs} : n \mapsto n + 0$ and $\text{rhs} : n \mapsto n$. Then,

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Example: $\forall n \in \mathbb{N}, n + 0 = n$

Let $\text{lhs} : n \mapsto n + 0$ and $\text{rhs} : n \mapsto n$. Then,

- $\text{lhs}(0) = 0 + 0 = 0 = \text{rhs}(0)$;

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Example: $\forall n \in \mathbb{N}, n + 0 = n$

Let $\text{lhs} : n \mapsto n + 0$ and $\text{rhs} : n \mapsto n$. Then,

- $\text{lhs}(0) = 0 + 0 = 0 = \text{rhs}(0)$;
- $\text{lhs}(\text{succ}(n)) = \text{succ}(n) + 0 = \text{succ}(n + 0) = \text{succ}(n) = \text{rhs}(\text{succ}(n))$;

Induction from \mathbb{N}

Theorem: Simple induction

Let $\text{lhs}, \text{rhs} : \mathbb{N} \rightarrow X$ be two functions out of \mathbb{N} . If

- $\text{lhs}(0) = \text{rhs}(0)$, and
- $(\forall n \in \mathbb{N}, \text{lhs}(n) = \text{rhs}(n) \implies \text{lhs}(\text{succ}(n)) = \text{rhs}(\text{succ}(n)))$

then $\text{lhs} = \text{rhs}$.

$$0 + y := y$$

$$\text{succ}(x) + y := \text{succ}(x + y)$$

Example: $\forall n \in \mathbb{N}, n + 0 = n$

Let $\text{lhs} : n \mapsto n + 0$ and $\text{rhs} : n \mapsto n$. Then,

- $\text{lhs}(0) = 0 + 0 = 0 = \text{rhs}(0)$;
- $\text{lhs}(\text{succ}(n)) = \text{succ}(n) + 0 = \text{succ}(n + 0) = \text{succ}(n) = \text{rhs}(\text{succ}(n))$;

so $\text{lhs} = \text{rhs}$.

Outline

- 1 Vanilla Induction
- 2 **Vanilla Corecursion**
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense
- 6 Outroduction

The Conatural Numbers

- $\mathbb{N} \rightarrow \text{Recursion} \rightarrow \text{Induction}$
- $\text{co}\mathbb{N} \rightarrow \text{Corecursion} \rightarrow \text{Coinduction}$

The Conatural Numbers

- $\mathbb{N} \rightarrow \text{Recursion} \rightarrow \text{Induction}$
- $\text{co}\mathbb{N} \rightarrow \text{Corecursion} \rightarrow \text{Coinduction}$

Definition: Conatural numbers, Predecessor

The set of **conatural numbers** is $\text{co}\mathbb{N} := \mathbb{N} \sqcup \{\infty\}$.

The **predecessor** operation is the bijection

$$\text{pred} : \text{co}\mathbb{N} \longrightarrow \{\text{no}\} \sqcup \text{co}\mathbb{N}$$

$$0 \longmapsto \text{no}$$

$$\text{succ}(n) \longmapsto n$$

$$\infty \longmapsto \infty$$

The Conatural Numbers

- $\mathbb{N} \rightarrow \text{Recursion} \rightarrow \text{Induction}$
- $\text{co}\mathbb{N} \rightarrow \text{Corecursion} \rightarrow \text{Coinduction}$

Definition: Conatural numbers, Predecessor

The set of **conatural numbers** is $\text{co}\mathbb{N} := \mathbb{N} \sqcup \{\infty\}$.

The **predecessor** operation is the bijection

$$\text{pred} : \text{co}\mathbb{N} \longrightarrow \{\text{no}\} \sqcup \text{co}\mathbb{N}$$

$$0 \longmapsto \text{no}$$

$$\text{succ}(n) \longmapsto n$$

$$\infty \longmapsto \infty$$

\mathbb{N} comes with $\{\text{no}\} \sqcup \mathbb{N} \xrightarrow{(0, \text{succ})} \mathbb{N}$.

$\text{co}\mathbb{N}$ comes with $\text{co}\mathbb{N} \xrightarrow{\text{pred}} \{\text{no}\} \sqcup \mathbb{N}$.

Partial Functions

Definition: Partial function

A **partial function** $X \rightharpoonup Y$ is a function $X \rightarrow \{\text{no}\} \sqcup Y$.

Examples:

Partial Functions

Definition: Partial function

A **partial function** $X \rightharpoonup Y$ is a function $X \rightarrow \{\text{no}\} \sqcup Y$.

Examples:

- $f : x \mapsto 1/x : \mathbb{R} \rightharpoonup \mathbb{R};$
 $f(0) = \text{no}.$

Partial Functions

Definition: Partial function

A **partial function** $X \rightharpoonup Y$ is a function $X \rightarrow \{\text{no}\} \sqcup Y$.

Examples:

- $f : x \mapsto 1/x : \mathbb{R} \rightharpoonup \mathbb{R};$
 $f(0) = \text{no}.$
- “Get input from user” : `ComputerState` \rightharpoonup `String`;
Fails when the user doesn’t give input.

Partial Functions

Definition: Partial function

A **partial function** $X \rightharpoonup Y$ is a function $X \rightarrow \{\text{no}\} \sqcup Y$.

Examples:

- $f : x \mapsto 1/x : \mathbb{R} \rightharpoonup \mathbb{R};$
 $f(0) = \text{no}.$
- “Get input from user” : `ComputerState` \rightharpoonup `String`;
Fails when the user doesn’t give input.
- $\text{pred} : \text{co}\mathbb{N} \rightharpoonup \text{co}\mathbb{N};$
 $\text{pred}(0)$ fails.

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightharpoonup X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightarrowtail X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

- $u(x) = 0 \iff f(x) = \text{no}$

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightharpoonup X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

- $u(x) = 0 \iff f(x) = \text{no}$
- $u(x) = 1 \iff f(x) \neq \text{no} \text{ and } f(f(x)) = \text{no}$

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightharpoonup X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

- $u(x) = 0 \iff f(x) = \text{no}$
- $u(x) = 1 \iff f(x) \neq \text{no} \text{ and } f(f(x)) = \text{no}$
- $u(x) = 2 \iff f(x) \neq \text{no} \text{ and } f(f(x)) \neq \text{no} \text{ and } f^3(x) = \text{no}$
- etc.

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightharpoonup X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

- $u(x) = 0 \iff f(x) = \text{no}$
- $u(x) = 1 \iff f(x) \neq \text{no} \text{ and } f(f(x)) = \text{no}$
- $u(x) = 2 \iff f(x) \neq \text{no} \text{ and } f(f(x)) \neq \text{no} \text{ and } f^3(x) = \text{no}$
- etc.
- $u(x) = \infty \iff f(x) \neq \text{no} \text{ and } f^2(x) \neq \text{no} \text{ and } \dots$

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightharpoonup X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

- $u(x) = 0 \iff f(x) = \text{no}$
- $u(x) = 1 \iff f(x) \neq \text{no} \text{ and } f(f(x)) = \text{no}$
- $u(x) = 2 \iff f(x) \neq \text{no} \text{ and } f(f(x)) \neq \text{no} \text{ and } f^3(x) = \text{no}$
- etc.
- $u(x) = \infty \iff f(x) \neq \text{no} \text{ and } f^2(x) \neq \text{no} \text{ and } \dots$

Write $u := \text{wait}(f)$.

Corecursion into $\text{co}\mathbb{N}$

Corecursion principle

Let $f : X \rightharpoonup X$ be a partial function. Construct $u : X \rightarrow \text{co}\mathbb{N}$ by

- $u(x) = 0 \iff f(x) = \text{no}$
- $u(x) = 1 \iff f(x) \neq \text{no} \text{ and } f(f(x)) = \text{no}$
- $u(x) = 2 \iff f(x) \neq \text{no} \text{ and } f(f(x)) \neq \text{no} \text{ and } f^3(x) = \text{no}$
- etc.
- $u(x) = \infty \iff f(x) \neq \text{no} \text{ and } f^2(x) \neq \text{no} \text{ and } \dots$

Write $u := \text{wait}(f)$.

Lemma: Pred-wait

Let $f : X \rightharpoonup X$ be a partial function and $x \in X$.

- If $f(x) = \text{no}$, then $\text{pred}(\text{wait}(f)(x)) = \text{no}$.
- If $f(x) = x' \neq \text{no}$, then $\text{pred}(\text{wait}(f)(x)) = \text{wait}(f)(x')$.

Abstract nonsense: pred is the terminal partial endo-function.

Corecursion into coN

Example:

- Let $c : \mathbb{Z}_{>0} \rightarrow \mathbb{Z}_{>0}$ by

$c(1) := \text{no}$; $c(n) := n/2$ if n is even; $c(n) := 3n + 1$ otherwise

Corecursion into coN

Example:

- Let $c : \mathbb{Z}_{>0} \rightarrow \mathbb{Z}_{>0}$ by

$c(1) := \text{no}$; $c(n) := n/2$ if n is even; $c(n) := 3n + 1$ otherwise

- $\text{wait}(c)(x)$ is the length of $x \xrightarrow{c} \cdots \xrightarrow{c} 1$, possibly ∞ .

Corecursion into $\text{co}\mathbb{N}$

Example:

- Let $c : \mathbb{Z}_{>0} \rightarrow \mathbb{Z}_{>0}$ by

$c(1) := \text{no}$; $c(n) := n/2$ if n is even; $c(n) := 3n + 1$ otherwise

- $\text{wait}(c)(x)$ is the length of $x \xrightarrow{c} \dots \xrightarrow{c} 1$, possibly ∞ .
- E.g. $5 \xrightarrow{c} 16 \xrightarrow{c} 8 \xrightarrow{c} 4 \xrightarrow{c} 2 \xrightarrow{c} 1 \xrightarrow{c} \text{no}$, so $\text{wait}(c)(5) = 5$.

Corecursion into coN

Example:

- Let $c : \mathbb{Z}_{>0} \rightarrow \mathbb{Z}_{>0}$ by

$c(1) := \text{no}$; $c(n) := n/2$ if n is even; $c(n) := 3n + 1$ otherwise

- $\text{wait}(c)(x)$ is the length of $x \xrightarrow{c} \dots \xrightarrow{c} 1$, possibly ∞ .
- E.g. $5 \xrightarrow{c} 16 \xrightarrow{c} 8 \xrightarrow{c} 4 \xrightarrow{c} 2 \xrightarrow{c} 1 \xrightarrow{c} \text{no}$, so $\text{wait}(c)(5) = 5$.
- The **Collatz conjecture** asks whether $\text{wait}(c)(n) \neq \infty$ for all $n \in \mathbb{Z}_{>0}$.

Corecursion into $\text{co}\mathbb{N}$

Definition: Addition in $\text{co}\mathbb{N}$

The function $(- + -) : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N}$ is defined by

$$(- + -) := \text{wait}(f)$$

where $f : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N} \times \text{co}\mathbb{N}$ is given by

$$f : (x, y) \mapsto \begin{cases} \text{no} & \text{if } \text{pred}(x) = \text{pred}(y) = \text{no} \\ (x, y') & \text{if } \text{pred}(x) = \text{no} \text{ and } \text{pred}(y) = y' \neq \text{no} \\ (x', y) & \text{if } \text{pred}(x) = x' \neq \text{no} \end{cases}$$

Corecursion into $\text{co}\mathbb{N}$

Definition: Addition in $\text{co}\mathbb{N}$

The function $(- + -) : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N}$ is defined by

$$(- + -) := \text{wait}(f)$$

where $f : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N} \times \text{co}\mathbb{N}$ is given by

$$f : (x, y) \mapsto \begin{cases} \text{no} & \text{if } \text{pred}(x) = \text{pred}(y) = \text{no} \\ (x, y') & \text{if } \text{pred}(x) = \text{no} \text{ and } \text{pred}(y) = y' \neq \text{no} \\ (x', y) & \text{if } \text{pred}(x) = x' \neq \text{no} \end{cases}$$

By the pred-wait lemma,

Corecursion into coN

Definition: Addition in coN

The function $(- + -) : \text{coN} \times \text{coN} \rightarrow \text{coN}$ is defined by

$$(- + -) := \text{wait}(f)$$

where $f : \text{coN} \times \text{coN} \rightarrow \text{coN} \times \text{coN}$ is given by

$$f : (x, y) \mapsto \begin{cases} \text{no} & \text{if } \text{pred}(x) = \text{pred}(y) = \text{no} \\ (x, y') & \text{if } \text{pred}(x) = \text{no} \text{ and } \text{pred}(y) = y' \neq \text{no} \\ (x', y) & \text{if } \text{pred}(x) = x' \neq \text{no} \end{cases}$$

By the pred-wait lemma,

- $\text{pred}(0 + 0) = \text{no}$, so $0 + 0 = 0$;

Corecursion into $\text{co}\mathbb{N}$

Definition: Addition in $\text{co}\mathbb{N}$

The function $(- + -) : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N}$ is defined by

$$(- + -) := \text{wait}(f)$$

where $f : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N} \times \text{co}\mathbb{N}$ is given by

$$f : (x, y) \mapsto \begin{cases} \text{no} & \text{if } \text{pred}(x) = \text{pred}(y) = \text{no} \\ (x, y') & \text{if } \text{pred}(x) = \text{no} \text{ and } \text{pred}(y) = y' \neq \text{no} \\ (x', y) & \text{if } \text{pred}(x) = x' \neq \text{no} \end{cases}$$

By the pred-wait lemma,

- $\text{pred}(0 + 0) = \text{no}$, so $0 + 0 = 0$;
- $\text{pred}(0 + y) = \text{pred}(y)$, so $0 + y = y$;

Corecursion into $\text{co}\mathbb{N}$

Definition: Addition in $\text{co}\mathbb{N}$

The function $(- + -) : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N}$ is defined by

$$(- + -) := \text{wait}(f)$$

where $f : \text{co}\mathbb{N} \times \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N} \times \text{co}\mathbb{N}$ is given by

$$f : (x, y) \mapsto \begin{cases} \text{no} & \text{if } \text{pred}(x) = \text{pred}(y) = \text{no} \\ (x, y') & \text{if } \text{pred}(x) = \text{no} \text{ and } \text{pred}(y) = y' \neq \text{no} \\ (x', y) & \text{if } \text{pred}(x) = x' \neq \text{no} \end{cases}$$

By the pred-wait lemma,

- $\text{pred}(0 + 0) = \text{no}$, so $0 + 0 = 0$;
- $\text{pred}(0 + y) = \text{pred}(y)$, so $0 + y = y$;
- $\text{pred}(x + y) = \text{pred}(x) + y$ whenever $\text{pred}(x) \neq \text{no}$.

Outline

- 1 Vanilla Induction
- 2 Vanilla Corecursion
- 3 **Vanilla Coinduction**
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense
- 6 Outroduction

Coinduction into coN

Given $f, g : X \rightarrow Y$, let $Y_{/f=g}$ be the quotient of Y by the smallest equivalence relation \sim with $f(x) \sim g(x)$ for each $x \in X$.

Coinduction into coN

Given $f, g : X \rightarrow Y$, let $Y_{/f=g}$ be the quotient of Y by the smallest equivalence relation \sim with $f(x) \sim g(x)$ for each $x \in X$.

So, $[f(x)] = [g(x)]$ in $Y_{/f=g}$.

Coinduction into $\text{co}\mathbb{N}$

Given $f, g : X \rightarrow Y$, let $Y_{/f=g}$ be the quotient of Y by the smallest equivalence relation \sim with $f(x) \sim g(x)$ for each $x \in X$.

So, $[f(x)] = [g(x)]$ in $Y_{/f=g}$.

Theorem: Simple coinduction

Let $\text{lhs}, \text{rhs} : X \rightarrow \text{co}\mathbb{N}$ be two functions into $\text{co}\mathbb{N}$.

Let $[-] : \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$ be the quotient map.

If

$$\forall x \in X, \quad [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$$

(with $[\text{no}] := \text{no}$)

Coinduction into $\text{co}\mathbb{N}$

Given $f, g : X \rightarrow Y$, let $Y_{/f=g}$ be the quotient of Y by the smallest equivalence relation \sim with $f(x) \sim g(x)$ for each $x \in X$.

So, $[f(x)] = [g(x)]$ in $Y_{/f=g}$.

Theorem: Simple coinduction

Let $\text{lhs}, \text{rhs} : X \rightarrow \text{co}\mathbb{N}$ be two functions into $\text{co}\mathbb{N}$.

Let $[-] : \text{co}\mathbb{N} \rightarrow \text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$ be the quotient map.

If

$$\forall x \in X, \quad [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$$

(with $[\text{no}] := \text{no}$), then $\text{lhs} = \text{rhs}$.

Coinduction into coN : proof sketch

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{coN}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof sketch.

It follows that $[\text{pred}^n(\text{lhs}(x))] = [\text{pred}^n(\text{rhs}(x))]$ for all n (with $\text{pred}(\text{no}) := \text{no}$).

Coinduction into $\text{co}\mathbb{N}$: proof sketch

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof sketch.

It follows that $[\text{pred}^n(\text{lhs}(x))] = [\text{pred}^n(\text{rhs}(x))]$ for all n (with $\text{pred}(\text{no}) := \text{no}$).

- If $\text{lhs}(x) = n \in \mathbb{N}$, then

$$[\text{pred}^{n+1}(\text{rhs}(x))] = [\text{pred}^{n+1}(\text{lhs}(x))] = [\text{no}] = \text{no}$$

so $\text{pred}^{n+1}(\text{rhs}(x)) = \text{no}$, and hence $\text{rhs}(x) = n = \text{lhs}(x)$.

Coinduction into $\text{co}\mathbb{N}$: proof sketch

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof sketch.

It follows that $[\text{pred}^n(\text{lhs}(x))] = [\text{pred}^n(\text{rhs}(x))]$ for all n (with $\text{pred}(\text{no}) := \text{no}$).

- If $\text{lhs}(x) = n \in \mathbb{N}$, then

$$[\text{pred}^{n+1}(\text{rhs}(x))] = [\text{pred}^{n+1}(\text{lhs}(x))] = [\text{no}] = \text{no}$$

so $\text{pred}^{n+1}(\text{rhs}(x)) = \text{no}$, and hence $\text{rhs}(x) = n = \text{lhs}(x)$.

- If $\text{rhs}(x) \in \mathbb{N}$, do the same thing.

Coinduction into $\text{co}\mathbb{N}$: proof sketch

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof sketch.

It follows that $[\text{pred}^n(\text{lhs}(x))] = [\text{pred}^n(\text{rhs}(x))]$ for all n (with $\text{pred}(\text{no}) := \text{no}$).

- If $\text{lhs}(x) = n \in \mathbb{N}$, then

$$[\text{pred}^{n+1}(\text{rhs}(x))] = [\text{pred}^{n+1}(\text{lhs}(x))] = [\text{no}] = \text{no}$$

so $\text{pred}^{n+1}(\text{rhs}(x)) = \text{no}$, and hence $\text{rhs}(x) = n = \text{lhs}(x)$.

- If $\text{rhs}(x) \in \mathbb{N}$, do the same thing.
- The only remaining case is $\text{lhs}(x) = \infty = \text{rhs}(x)$.

Coinduction into $\text{co}\mathbb{N}$: proof sketch

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof sketch.

It follows that $[\text{pred}^n(\text{lhs}(x))] = [\text{pred}^n(\text{rhs}(x))]$ for all n (with $\text{pred}(\text{no}) := \text{no}$).

- If $\text{lhs}(x) = n \in \mathbb{N}$, then

$$[\text{pred}^{n+1}(\text{rhs}(x))] = [\text{pred}^{n+1}(\text{lhs}(x))] = [\text{no}] = \text{no}$$

so $\text{pred}^{n+1}(\text{rhs}(x)) = \text{no}$, and hence $\text{rhs}(x) = n = \text{lhs}(x)$.

- If $\text{rhs}(x) \in \mathbb{N}$, do the same thing.
- The only remaining case is $\text{lhs}(x) = \infty = \text{rhs}(x)$.

Therefore, $\text{lhs} = \text{rhs}$.



Coinduction into $\text{co}\mathbb{N}$: $\forall x \in \text{co}\mathbb{N}, x + 0 = x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

(*) $\text{pred}(x + 0) = \text{no}$ whenever $\text{pred}(x) = \text{no}$

(**) $\text{pred}(x + y) = \text{pred}(x) + y$ whenever $\text{pred}(x) \neq \text{no}$

Proof that $\forall x \in \text{co}\mathbb{N}, x + 0 = x$.

Coinduction into $\text{co}\mathbb{N}$: $\forall x \in \text{co}\mathbb{N}, x + 0 = x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

(*) $\text{pred}(x + 0) = \text{no}$ whenever $\text{pred}(x) = \text{no}$

Proof that $\forall x \in \text{co}\mathbb{N}, x + 0 = x$.

Set $\text{lhs} : x \mapsto x + 0$ and $\text{rhs} : x \mapsto x$.

Coinduction into $\text{co}\mathbb{N}$: $\forall x \in \text{co}\mathbb{N}, x + 0 = x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

(*) $\text{pred}(x + 0) = \text{no}$ whenever $\text{pred}(x) = \text{no}$

Proof that $\forall x \in \text{co}\mathbb{N}, x + 0 = x$.

Set $\text{lhs} : x \mapsto x + 0$ and $\text{rhs} : x \mapsto x$.

- If $\text{pred}(x) = \text{no}$, then

$$\begin{aligned} [\text{pred}(\text{lhs}(x))] &= [\text{pred}(x + 0)] \\ &= [\text{no}] \\ &= [\text{pred}(x)] \\ [\text{pred}(\text{lhs}(x))] &= [\text{pred}(\text{rhs}(x))] \end{aligned}$$

Coinduction into $\text{co}\mathbb{N}$: $\forall x \in \text{co}\mathbb{N}, x + 0 = x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

(**) $\text{pred}(x + y) = \text{pred}(x) + y$ whenever $\text{pred}(x) \neq \text{no}$

Proof that $\forall x \in \text{co}\mathbb{N}, x + 0 = x$.

Set $\text{lhs} : x \mapsto x + 0$ and $\text{rhs} : x \mapsto x$.

- If instead $\text{pred}(x) \neq \text{no}$, then

$$\begin{aligned} [\text{pred}(\text{lhs}(x))] &= [\text{pred}(x + 0)] \\ &= [\text{pred}(x) + 0] \\ &= [\text{lhs}(\text{pred}(x))] \\ &= [\text{rhs}(\text{pred}(x))] \\ &= [\text{pred}(x)] \\ [\text{pred}(\text{lhs}(x))] &= [\text{pred}(\text{rhs}(x))] \end{aligned}$$

Coinduction into $\text{co}\mathbb{N}$: $\forall x \in \text{co}\mathbb{N}, x + 0 = x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof that $\forall x \in \text{co}\mathbb{N}, x + 0 = x$.

Set $\text{lhs} : x \mapsto x + 0$ and $\text{rhs} : x \mapsto x$.

- The theorem asserts $\text{lhs} = \text{rhs}$.



Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

- $\text{pred}(a + b) = \text{pred}(b)$ whenever $\text{pred}(a) = \text{no}$
- $\text{pred}(a + b) = \text{pred}(a)$ whenever $\text{pred}(b) = \text{no}$
- $\text{pred}(a + b) = \text{pred}(a) + b$ whenever $\text{pred}(a) \neq \text{no}$
- $\text{pred}(a + b) = a + \text{pred}(b)$ whenever $\text{pred}(b) \neq \text{no}$ (possible)

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

- Set $X := \text{co}\mathbb{N}^2$.
- Set $\text{lhs}(x, y) := x + y$.
- Set $\text{rhs}(x, y) := y + x$.

Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

- $\text{pred}(a + b) = \text{pred}(b)$ whenever $\text{pred}(a) = \text{no}$
- $\text{pred}(a + b) = \text{pred}(a)$ whenever $\text{pred}(b) = \text{no}$

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

- If $\text{pred}(x) = \text{no}$, then

Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}/_{\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

- $\text{pred}(a + b) = \text{pred}(b)$ whenever $\text{pred}(a) = \text{no}$
- $\text{pred}(a + b) = \text{pred}(a)$ whenever $\text{pred}(b) = \text{no}$

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

- If $\text{pred}(x) = \text{no}$, then

$$\begin{aligned} [\text{pred}(\text{lhs}(x, y))] &= [\text{pred}(x + y)] \\ &= [\text{pred}(y)] \\ &= [\text{pred}(y + x)] \\ [\text{pred}(\text{lhs}(x, y))] &= [\text{pred}(\text{rhs}(x, y))] \end{aligned}$$

Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

- $\text{pred}(a + b) = \text{pred}(a) + b$ whenever $\text{pred}(a) \neq \text{no}$
- $\text{pred}(a + b) = a + \text{pred}(b)$ whenever $\text{pred}(b) \neq \text{no}$

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

- If $\text{pred}(x) \neq \text{no}$, then

Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

- $\text{pred}(a + b) = \text{pred}(a) + b$ whenever $\text{pred}(a) \neq \text{no}$
- $\text{pred}(a + b) = a + \text{pred}(b)$ whenever $\text{pred}(b) \neq \text{no}$

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

- If $\text{pred}(x) \neq \text{no}$, then

$$\begin{aligned} [\text{pred}(\text{lhs}(x, y))] &= [\text{pred}(x + y)] \\ &= [\text{pred}(x) + y] \\ &= [\text{lhs}(\text{pred}(x), y)] \\ &= [\text{rhs}(\text{pred}(x), y)] \\ &= [y + \text{pred}(x)] \\ &= [\text{pred}(y + x)] = [\text{pred}(\text{rhs}(x, y))] \end{aligned}$$

Coinduction into $\text{co}\mathbb{N}$: $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$

Theorem: Simple coinduction

If $\forall x \in X, [\text{pred}(\text{lhs}(x))] = [\text{pred}(\text{rhs}(x))]$ in $\text{co}\mathbb{N}_{/\text{lhs}=\text{rhs}}$, then $\text{lhs} = \text{rhs}$.

Proof that $\forall x, y \in \text{co}\mathbb{N}, x + y = y + x$.

- The theorem asserts $\text{lhs} = \text{rhs}$.



Other properties of coN

These can be proven coinductively:

- $\text{pred}(x + y) = x + \text{pred}(y)$ whenever $\text{pred}(y) \neq \text{no.}$
- $+$ is associative.
- If x, y are (finite or infinite) sequences, then $\text{length}(x ++ y) = \text{length}(x) + \text{length}(y)$.

Other properties of coN

Coinductive proofs only involve ≤ 2 cases:

- $\text{pred}(\text{something}) = \text{no}$
- $\text{pred}(\text{something}) \neq \text{no}$

and sometimes no splits at all!

Other properties of coN

Coinductive proofs only involve ≤ 2 cases:

- $\text{pred}(\text{something}) = \text{no}$
- $\text{pred}(\text{something}) \neq \text{no}$

and sometimes no splits at all!

Contrast against inductive-plus-infinity proofs with 3 cases:

- $(\text{something}) = 0$
- $(\text{something}) = \text{succ}(\text{something else})$
- $(\text{something}) = \infty$

Why induct when you could coinduct?

Outline

- 1 Vanilla Induction
- 2 Vanilla Corecursion
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense
- 6 Outroduction

Lists

Let A be a set.

Definition: Lists on A

A **list** on A is a finite-length sequence of elements in A .

Lists

Let A be a set.

Definition: Lists on A

A **list** on A is a finite-length sequence of elements in A .

$$\text{List}(A) := \{[], [a_0], [a_0, a_1], \dots \mid a_0, a_1, \dots \in A\}$$

Lists

Let A be a set.

Definition: Lists on A

A **list** on A is a finite-length sequence of elements in A .

$$\text{List}(A) := \{[], [a_0], [a_0, a_1], \dots \mid a_0, a_1, \dots \in A\}$$

$\text{List}(A)$ comes with a constant $[] \in \text{List}(A)$, and an operation

$$\begin{aligned} \text{cons} : A \times \text{List}(A) &\longrightarrow \text{List}(A) \\ (a_0, [a_1, \dots]) &\longmapsto [a_0, a_1, \dots] \end{aligned}$$

Colists

Let A be a set.

Definition: Colists on A

A **colist** on A is a finite-or-infinite sequence of elements in A .

Colists

Let A be a set.

Definition: Colists on A

A **colist** on A is a finite-or-infinite sequence of elements in A .

$$\text{coList}(A) := \text{List}(A) \sqcup \{[a_0, a_1, \dots] \mid a_0, a_1, \dots \in A\}$$

Colists

Let A be a set.

Definition: Colists on A

A **colist** on A is a finite-or-infinite sequence of elements in A .

$$\text{coList}(A) := \text{List}(A) \sqcup \{[a_0, a_1, \dots] \mid a_0, a_1, \dots \in A\}$$

$\text{coList}(A)$ comes with an operation

$$\text{pop} : \text{coList}(A) \longrightarrow \{\text{no}\} \sqcup (A \times \text{coList}(A))$$

$$[] \longmapsto \text{no}$$

$$[a_0, a_1, \dots] \longmapsto (a_0, [a_1, \dots])$$

Recursion/Corecursion

Recursion principle

To define a function $u : \text{List}(A) \rightarrow X$, it is enough to specify $u([])$ and to specify $u(\text{cons}(a, \text{as}))$ in terms of a and $u(\text{as})$.

Recursion/Corecursion

Recursion principle

To define a function $u : \text{List}(A) \rightarrow X$, it is enough to specify $u([])$ and to specify $u(\text{cons}(a, \text{as}))$ in terms of a and $u(\text{as})$.

Corecursion principle

To define a function $u : X \rightarrow \text{coList}(A)$, it is enough to specify a partial function $f : X \rightharpoonup A \times X$.

Recursion/Corecursion

Recursion principle

To define a function $u : \text{List}(A) \rightarrow X$, it is enough to specify $u([])$ and to specify $u(\text{cons}(a, \text{as}))$ in terms of a and $u(\text{as})$.

Corecursion principle

To define a function $u : X \rightarrow \text{coList}(A)$, it is enough to specify a partial function $f : X \rightarrowtail A \times X$.

- $\text{pop}(u(x)) = \text{no}$ whenever $f(x) = \text{no}$
- $\text{pop}(u(x)) = (a, u(x'))$ whenever $f(x) = (a, x')$

Recursion/Corecursion

Recursion principle

To define a function $u : \text{List}(A) \rightarrow X$, it is enough to specify $u([])$ and to specify $u(\text{cons}(a, \text{as}))$ in terms of a and $u(\text{as})$.

Corecursion principle

To define a function $u : X \rightarrow \text{coList}(A)$, it is enough to specify a partial function $f : X \rightharpoonup A \times X$.

- $\text{pop}(u(x)) = \text{no}$ whenever $f(x) = \text{no}$
- $\text{pop}(u(x)) = (a, u(x'))$ whenever $f(x) = (a, x')$

Think of $u(x) = [a_0, a_1, \dots]$ as tracing the execution

$$x \xrightarrow{f} (a_0, x') \xrightarrow{f} (a_0, (a_1, x'')) \xrightarrow{f} \dots$$

Induction/Coinduction

Induction principle

Let $\text{lhs}, \text{rhs} : \text{List}(A) \rightarrow X$. If

- $\text{lhs}([]) = \text{rhs}([])$
- $\text{lhs}(\text{cons}(a, \text{as})) = \text{rhs}(\text{cons}(a, \text{as}))$ whenever $\text{lhs}(\text{as}) = \text{rhs}(\text{as})$

then $\text{lhs} = \text{rhs}$.

Induction/Coinduction

Induction principle

Let $\text{lhs}, \text{rhs} : \text{List}(A) \rightarrow X$. If

- $\text{lhs}([]) = \text{rhs}([])$
- $\text{lhs}(\text{cons}(a, \text{as})) = \text{rhs}(\text{cons}(a, \text{as}))$ whenever $\text{lhs}(\text{as}) = \text{rhs}(\text{as})$

then $\text{lhs} = \text{rhs}$.

Example theorems:

- Concatenation of lists is associative
- Length of concatenation is sum of lengths (induct from List)
- List is a functor
- List has the “list comprehension” applicative from Haskell

Induction/Coinduction

Coinduction principle

Let $\text{lhs}, \text{rhs} : X \rightarrow \text{coList}(A)$.

Let $[-] : \text{coList}(A) \rightarrow \text{coList}(A)_{/\text{lhs}=\text{rhs}}$ be the quotient map.

If

$$\forall x \in X, [\text{pop}(\text{lhs}(x))] = [\text{pop}(\text{rhs}(x))]$$

(with $[\text{no}] := \text{no}$), then $\text{lhs} = \text{rhs}$.

Induction/Coinduction

Coinduction principle

Let $\text{lhs}, \text{rhs} : X \rightarrow \text{coList}(A)$.

Let $[-] : \text{coList}(A) \rightarrow \text{coList}(A)_{/\text{lhs}=\text{rhs}}$ be the quotient map.

If

$$\forall x \in X, [\text{pop}(\text{lhs}(x))] = [\text{pop}(\text{rhs}(x))]$$

(with $[\text{no}] := \text{no}$), then $\text{lhs} = \text{rhs}$.

Example theorems:

- Concatenation of colists is associative
- Length of concatenation is sum of lengths (coinduct into $\text{co}\mathbb{N}$)
- coList is a functor
- coList has the “ZipList” applicative from Haskell

Other structures

Every (non-dependent) inductive data you can make in Haskell/Lean/Agda/etc. has a coinductive version.

Other structures

Every (non-dependent) inductive data you can make in Haskell/Lean/Agda/etc. has a coinductive version.

- $\mathbb{N} \rightsquigarrow \text{coN}$

Other structures

Every (non-dependent) inductive data you can make in Haskell/Lean/Agda/etc. has a coinductive version.

- $\mathbb{N} \rightsquigarrow \text{coN}$
- $\text{List} \rightsquigarrow \text{coList}$

Other structures

Every (non-dependent) inductive data you can make in Haskell/Lean/Agda/etc. has a coinductive version.

- $\mathbb{N} \rightsquigarrow \text{coN}$
- $\text{List} \rightsquigarrow \text{coList}$
- Finite-size binary trees \rightsquigarrow potentially infinite binary trees

Other structures

Every (non-dependent) inductive data you can make in Haskell/Lean/Agda/etc. has a coinductive version.

- $\mathbb{N} \rightsquigarrow \text{coN}$
- $\text{List} \rightsquigarrow \text{coList}$
- Finite-size binary trees \rightsquigarrow potentially infinite binary trees
- Etc.

Outline

- 1 Vanilla Induction
- 2 Vanilla Corecursion
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 **Abstract Nonsense**
- 6 Outroduction

All the categories

All of the inductive data types are **initial algebras**.

Initial algebras

Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor.

All the categories

All of the inductive data types are **initial algebras**.

Initial algebras

Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor.

An **algebra** for T is an arrow $T(a) \xrightarrow{\alpha} a$ in \mathcal{C} .

All the categories

All of the inductive data types are **initial algebras**.

Initial algebras

Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor.

An **algebra** for T is an arrow $T(a) \xrightarrow{\alpha} a$ in \mathcal{C} .

A **morphism** from $T(a) \xrightarrow{\alpha} a$ to $T(b) \xrightarrow{\beta} b$ is an arrow $f : a \rightarrow b$ in \mathcal{C} such that this commutes:

$$\begin{array}{ccc} T(a) & \xrightarrow{\alpha} & a \\ T(f) \downarrow & & \downarrow f \\ T(b) & \xrightarrow{\beta} & b \end{array}$$

All the categories

All of the inductive data types are **initial algebras**.

Initial algebras

Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor.

An **algebra** for T is an arrow $T(a) \xrightarrow{\alpha} a$ in \mathcal{C} .

A **morphism** from $T(a) \xrightarrow{\alpha} a$ to $T(b) \xrightarrow{\beta} b$ is an arrow $f : a \rightarrow b$ in \mathcal{C} such that this commutes:

$$\begin{array}{ccc} T(a) & \xrightarrow{\alpha} & a \\ T(f) \downarrow & & \downarrow f \\ T(b) & \xrightarrow{\beta} & b \end{array}$$

With identity and composite arrows as in \mathcal{C} , these form a category.

All the categories

All of the inductive data types are **initial algebras**.

Initial algebras

Let $T : \mathcal{C} \rightarrow \mathcal{C}$ be an endofunctor.

An **algebra** for T is an arrow $T(a) \xrightarrow{\alpha} a$ in \mathcal{C} .

A **morphism** from $T(a) \xrightarrow{\alpha} a$ to $T(b) \xrightarrow{\beta} b$ is an arrow $f : a \rightarrow b$ in \mathcal{C} such that this commutes:

$$\begin{array}{ccc} T(a) & \xrightarrow{\alpha} & a \\ T(f) \downarrow & & \downarrow f \\ T(b) & \xrightarrow{\beta} & b \end{array}$$

With identity and composite arrows as in \mathcal{C} , these form a category.

An **initial algebra** for T is an initial object in this category.

All of the inductive data types are **initial algebras**.

Recursion/Induction

All of the inductive data types are **initial algebras**.

Recursion

An **initial algebra** for T is an initial object $T(i) \xrightarrow{\text{cons}} i$ in the category of T -algebras.

All of the inductive data types are **initial algebras**.

Recursion

An **initial algebra** for T is an initial object $T(i) \xrightarrow{\text{cons}} i$ in the category of T -algebras.

The **recursion principle** is its universal property.

Recursion/Induction

Induction

Let $T(i) \xrightarrow{\text{cons}} i$ be an initial algebra.

Let $f, g : i \rightarrow a$ in \mathcal{C} .

Suppose the equaliser $\iota : e \rightarrow i$ of f and g exists in \mathcal{C} , and that $f \circ \text{cons} \circ T(\iota) = g \circ \text{cons} \circ T(\iota)$:

$$\begin{array}{ccc} T(e) & \xrightarrow{\quad\quad\quad} & e \\ T(\iota) \downarrow & & \downarrow \iota \\ T(i) & \xrightarrow{\text{cons}} & i \\ & & \downarrow f \quad \downarrow g \\ & & a \end{array}$$

Then, $f = g$.

Corecursion/Coinduction

Turn all the arrows around.

Turn all the arrows around.

Coinduction

Let $T(j) \xrightarrow{\text{des}} j$ be an initial algebra.

Let $f, g : a \rightarrow j$ in \mathcal{C} .

Suppose the coequaliser $\pi : j \twoheadrightarrow e$ of f and g exists in \mathcal{C} , and that this commutes:

$$\begin{array}{ccc} a & & \\ f \downarrow & \parallel & \downarrow g \\ j & \xrightarrow{\text{des}} & T(j) \\ \pi \downarrow & & \downarrow T(\pi) \\ e & \dashrightarrow & T(e) \end{array}$$

Then, $f = g$.

Something cool ig

$T : \mathbf{Set} \rightarrow \mathbf{Set}$ given by $X \mapsto \{\text{no}\} \sqcup X$ has:

- Initial algebra $\{\text{no}\} \sqcup \mathbb{N} \xrightarrow{(0, \text{succ})} \mathbb{N}$.
- Terminal coalgebra $\text{co}\mathbb{N} \xrightarrow{\text{pred}} \{\text{no}\} \sqcup \text{co}\mathbb{N}$.

Something cool is

$T : \mathbf{Set} \rightarrow \mathbf{Set}$ given by $X \mapsto \{\text{no}\} \sqcup X$ has:

- Initial algebra $\{\text{no}\} \sqcup \mathbb{N} \xrightarrow{(0, \text{succ})} \mathbb{N}$.
- Terminal coalgebra $\text{co}\mathbb{N} \xrightarrow{\text{pred}} \{\text{no}\} \sqcup \text{co}\mathbb{N}$.

Instead regard $T : \mathbf{Top} \rightarrow \mathbf{Top}$.

Something cool is

$T : \mathbf{Set} \rightarrow \mathbf{Set}$ given by $X \mapsto \{\text{no}\} \sqcup X$ has:

- Initial algebra $\{\text{no}\} \sqcup \mathbb{N} \xrightarrow{(0, \text{succ})} \mathbb{N}$.
- Terminal coalgebra $\text{co}\mathbb{N} \xrightarrow{\text{pred}} \{\text{no}\} \sqcup \text{co}\mathbb{N}$.

Instead regard $T : \mathbf{Top} \rightarrow \mathbf{Top}$.

- The initial algebra exists, and is \mathbb{N} with discrete topology.

Something cool is

$T : \mathbf{Set} \rightarrow \mathbf{Set}$ given by $X \mapsto \{\text{no}\} \sqcup X$ has:

- Initial algebra $\{\text{no}\} \sqcup \mathbb{N} \xrightarrow{(0, \text{succ})} \mathbb{N}$.
- Terminal coalgebra $\text{co}\mathbb{N} \xrightarrow{\text{pred}} \{\text{no}\} \sqcup \text{co}\mathbb{N}$.

Instead regard $T : \mathbf{Top} \rightarrow \mathbf{Top}$.

- The initial algebra exists, and is \mathbb{N} with discrete topology.
- The terminal coalgebra exists, and is $\text{co}\mathbb{N}$ with

$$(0, 1, 2, \dots) \xrightarrow{\text{converges}} \infty$$

Outline

- 1 Vanilla Induction
- 2 Vanilla Corecursion
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense
- 6 **Outroduction**

Thx 4 watch

If you want more, I've written an article at <https://emptylimit.github.io/discussion/coinduction.html>!

- [1] nLab authors. *bisimulation*. URL:
<https://ncatlab.org/nlab/show/bisimulation>.
- [2] nLab authors. *coinduction*. URL:
<https://ncatlab.org/nlab/show/coinduction>.
- [3] nLab authors. *corecursion*. URL:
<https://ncatlab.org/nlab/show/corecursion>.
- [4] nLab authors. *inductive type*. URL:
<https://ncatlab.org/nlab/show/inductive+type>.
- [5] nLab authors. *terminal coalgebra for an endofunctor*. URL:
<https://ncatlab.org/nlab/show/terminal+coalgebra+for+an+endofunctor>.