Corecursion and Coinduction

Gabriel Field

15/Aug/2025

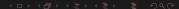
Goals

Actual goals:

- Introduce the conatural numbers
- ullet Introduce **corecursion** to specifying functions into $\mathrm{co}\mathbb{N}$
- 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

Outline

1 Vanilla Induction

(2) Vanilla Corecursion

The Natural Numbers

Coinduction \leftarrow Corecursion \leftarrow Recursion \leftarrow N.

The Natural Numbers

Coinduction \leftarrow Corecursion \leftarrow Recursion \leftarrow N.

Loose Definition

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

The Natural Numbers

Coinduction \leftarrow Corecursion \leftarrow Recursion \leftarrow N.

Loose Definition

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

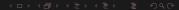
Recursion Principle

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

$$u: 0 \mapsto x_0$$

 $u: \operatorname{succ}(n) \mapsto f(u(n))$

We'll write $u := iterate(f, x_0)$.



Recursion from N

Example: Powers

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

$$u: 0 \mapsto 1$$
 $u: \operatorname{succ}(n) \mapsto 3^n \cdot n$
 $3^0 := 1$ $3^{\operatorname{succ}(n)} := 3^n \cdot n$

Example: Addition on \mathbb{N}

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

$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

Theorem: Simple induction

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

- lhs(0) = 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 lhs = rhs.

Theorem: Simple induction

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

- lhs(0) = 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 lhs = rhs.

$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

Theorem: Simple induction

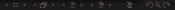
Let $lhs, rhs : \mathbb{N} \to X$ be two functions out of \mathbb{N} . If

- lhs(0) = 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 lhs = rhs.

$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

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



Theorem: Simple induction

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

- lhs(0) = 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 lhs = rhs.

$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

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

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

Theorem: Simple induction

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

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

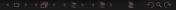
then lhs = rhs.

$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

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

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

• lhs(0) = 0 + 0 = 0 = rhs(0);



Theorem: Simple induction

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

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

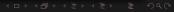
then lhs = rhs.

$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

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

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

- lhs(0) = 0 + 0 = 0 = rhs(0);
- lhs(succ(n)) = succ(n) + 0 = succ(n + 0) = succ(n) = rhs(succ(n));



Theorem: Simple induction

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

- lhs(0) = 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 lhs = rhs.

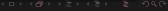
$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

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

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

- lhs(0) = 0 + 0 = 0 = rhs(0);
- lhs(succ(n)) = succ(n) + 0 = succ(n + 0) = succ(n) = rhs(succ(n));

so lhs = rhs.



Outline

Vanilla Induction

2 Vanilla Corecursion

The Conatural Numbers

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

The Conatural Numbers

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

Definition: Conatural numbers, Predecessor

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

The **predecessor** operation is the bijection

$$\operatorname{pred}: \operatorname{co}\mathbb{N} \longrightarrow \{\operatorname{no}\} \sqcup \operatorname{co}\mathbb{N}$$
$$0 \longmapsto \operatorname{no}$$
$$\operatorname{succ}(n) \longmapsto n$$
$$\infty \longmapsto \infty$$

The Conatural Numbers

- $\mathbb{N} \to \mathsf{Recursion} \to \mathsf{Induction}$
- $co\mathbb{N} \to \mathsf{Corecursion} \to \mathsf{Coinduction}$

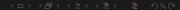
Definition: Conatural numbers, Predecessor

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

The **predecessor** operation is the bijection

$$\operatorname{pred}: \operatorname{co}\mathbb{N} \longrightarrow \{\operatorname{no}\} \sqcup \operatorname{co}\mathbb{N}$$
$$0 \longmapsto \operatorname{no}$$
$$\operatorname{succ}(n) \longmapsto n$$
$$\infty \longmapsto \infty$$

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



Definition: Partial function

A partial function $X \nrightarrow Y$ is a function $X \to \{no\} \sqcup Y$.

Examples:

Definition: Partial function

A partial function $X \nrightarrow Y$ is a function $X \to \{no\} \sqcup Y$.

Examples:

• $f: x \mapsto 1/x : \mathbb{R} \to \mathbb{R};$ f(0) = no.

Definition: Partial function

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

Examples:

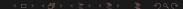
- $f: x \mapsto 1/x : \mathbb{R} \nrightarrow \mathbb{R}$; f(0) = no.
- "Get input from user": ComputerState → String;
 Fails when the user doesn't give input.

Definition: Partial function

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

Examples:

- $f: x \mapsto 1/x : \mathbb{R} \nrightarrow \mathbb{R}$; f(0) = no.
- "Get input from user": ComputerState → String;
 Fails when the user doesn't give input.
- pred : $coN \rightarrow coN$; pred(0) fails.



Corecursion principle

Let $f: X \nrightarrow X$ be a partial function. Construct $u: X \to \mathrm{co}\mathbb{N}$ by

Corecursion principle

Let $f: X \nrightarrow X$ be a partial function. Construct $u: X \to \mathrm{co}\mathbb{N}$ by

• $u(x) = 0 \iff f(x) = \text{no}$

Corecursion principle

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

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

Corecursion principle

Let $f: X \nrightarrow X$ be a partial function. Construct $u: X \to co\mathbb{N}$ by

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

Corecursion principle

Let $f: X \nrightarrow X$ be a partial function. Construct $u: X \to co\mathbb{N}$ by

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

Corecursion principle

Let $f: X \nrightarrow X$ be a partial function. Construct $u: X \to co\mathbb{N}$ by

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

Write u := wait(f).

Corecursion principle

Let $f: X \nrightarrow X$ be a partial function. Construct $u: X \to co\mathbb{N}$ by

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

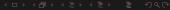
Write u := wait(f).

Lemma: Pred-wait

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

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

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



Example:

• Let $c: \mathbb{Z}_{>0} \nrightarrow \mathbb{Z}_{>0}$ by

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

Example:

• Let $c: \mathbb{Z}_{>0} \nrightarrow \mathbb{Z}_{>0}$ by

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

• wait(c)(x) is the length of $x \stackrel{c}{\mapsto} \cdots \stackrel{c}{\mapsto} 1$, possibly ∞ .

Example:

• Let $c: \mathbb{Z}_{>0} \nrightarrow \mathbb{Z}_{>0}$ by

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

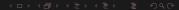
- wait(c)(x) is the length of $x \stackrel{c}{\mapsto} \cdots \stackrel{c}{\mapsto} 1$, possibly ∞ .
- E.g. $5 \stackrel{c}{\mapsto} 16 \stackrel{c}{\mapsto} 8 \stackrel{c}{\mapsto} 4 \stackrel{c}{\mapsto} 2 \stackrel{c}{\mapsto} 1 \stackrel{c}{\mapsto} \text{no, so wait}(c)(5) = 5.$

Example:

• Let $c: \mathbb{Z}_{>0} \nrightarrow \mathbb{Z}_{>0}$ by

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

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



Definition: Addition in coN

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

$$(-+-) := wait(f)$$

where $f: \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N} \nrightarrow \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N}$ is given by

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

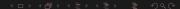
Definition: Addition in coN

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

$$(-+-) := wait(f)$$

where $f: \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N} \nrightarrow \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N}$ is given by

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



Definition: Addition in coN

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

$$(-+-) := wait(f)$$

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

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

•
$$pred(0+0) = no, so 0 + 0 = 0;$$

Definition: Addition in coN

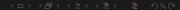
The function $(-+-): co\mathbb{N} \times co\mathbb{N} \to co\mathbb{N}$ is defined by

$$(-+-) := wait(f)$$

where $f: \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N} \nrightarrow \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N}$ is given by

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

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



Definition: Addition in coN

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

$$(-+-) := wait(f)$$

where $f: \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N} \nrightarrow \mathrm{co}\mathbb{N} \times \mathrm{co}\mathbb{N}$ is given by

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

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

