# Corecursion and Coinduction

Gabriel Field

15/Aug/2025

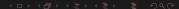
### Goals

### 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
- 3 Vanilla Coinduction
- 4 Structural Induction and Co-structural(?) Coinduction
- 5 Abstract Nonsense

# Outline

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

# 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.

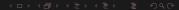
#### Loose Definition

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

$$u: 0 \mapsto x_0$$
  
 $u: \operatorname{succ}(n) \mapsto f(u(n))$ 



# 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

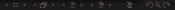
Let  $lhs, rhs : \mathbb{N} \to X$  be two functions out of  $\mathbb{N}$ . If

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

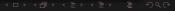
then lhs = rhs.

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

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

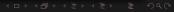
then lhs = rhs.

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

### 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.

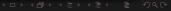
$$0 + y := y \qquad \operatorname{succ}(x) + y := \operatorname{succ}(x + y)$$

### 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

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

## The Conatural Numbers

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

## 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

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

# The Conatural Numbers

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

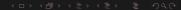
#### Definition: Conatural numbers, Predecessor

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

The **predecessor** operation is the bijection

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

 $\mathbb{N}$  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:

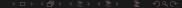
- $f: x \mapsto 1/x : \mathbb{R} \to \mathbb{R}$ ; f(0) = no.
- "Get input from user": ComputerState → String;
   Fails when the user doesn't give input.

#### Definition: Partial function

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

### Examples:

- $f: x \mapsto 1/x : \mathbb{R} \to \mathbb{R}$ ; 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  $u: X \to co\mathbb{N}$  by

• 
$$u(x) = 0 \iff f(x) = \text{no}$$

### Corecursion principle

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

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

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

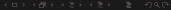
Write u := wait(f).

#### 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

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

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

### Example:

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

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

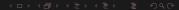
- wait(c)(x) is the length of  $x \stackrel{c}{\mapsto} \cdots \stackrel{c}{\mapsto} 1$ , possibly  $\infty$ .
- E.g.  $5 \stackrel{c}{\mapsto} 16 \stackrel{c}{\mapsto} 8 \stackrel{c}{\mapsto} 4 \stackrel{c}{\mapsto} 2 \stackrel{c}{\mapsto} 1 \stackrel{c}{\mapsto} \text{no, so wait}(c)(5) = 5.$

### 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  $\infty$ .
- 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

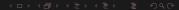
The function  $(-+-): co\mathbb{N} \times co\mathbb{N} \to co\mathbb{N}$  is defined by

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

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

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

By the pred-wait lemma,



## Corecursion into coN

#### Definition: Addition in coN

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

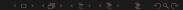
$$(-+-) := wait(f)$$

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

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

By the pred-wait lemma,

• 
$$\operatorname{pred}(0+0) = \operatorname{no}$$
, so  $0+0=0$ ;



### Corecursion into coN

#### Definition: Addition in coN

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

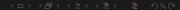
$$(-+-) := wait(f)$$

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

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

By the pred-wait lemma,

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



### Corecursion into coN

#### 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

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

By the pred-wait lemma,

- 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  $\operatorname{pred}(x) \neq \operatorname{no}$ .



## Outline

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

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

Given  $f,g:X\to Y$ , 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  $Y_{/f=g}$ .

Given  $f,g:X\to Y$ , 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  $lhs, rhs : X \to co\mathbb{N}$  be two functions into  $co\mathbb{N}$ .

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

lf

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

(with 
$$[no] := no$$
)

Given  $f,g:X\to Y$ , 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  $lhs, rhs : X \to co\mathbb{N}$  be two functions into  $co\mathbb{N}$ .

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

lf

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

(with [no] := no), then lhs = rhs.

# Coinduction into coN: proof sketch

### Theorem: Simple coinduction

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

#### Proof sketch.

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

### Theorem: Simple coinduction

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

#### Proof sketch.

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

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

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

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

### Theorem: Simple coinduction

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

#### Proof sketch.

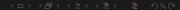
It follows that  $[\operatorname{pred}^n(\operatorname{lhs}(x))] = [\operatorname{pred}^n(\operatorname{rhs}(x))]$  for all n (with  $\operatorname{pred}(\operatorname{no}) := \operatorname{no}$ ).

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

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

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

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



### Theorem: Simple coinduction

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

#### Proof sketch.

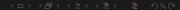
It follows that  $[\operatorname{pred}^n(\operatorname{lhs}(x))] = [\operatorname{pred}^n(\operatorname{rhs}(x))]$  for all n (with  $\operatorname{pred}(\operatorname{no}) := \operatorname{no}$ ).

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

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

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

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



### Theorem: Simple coinduction

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

#### Proof sketch.

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

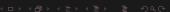
• If  $lhs(x) = n \in \mathbb{N}$ , then

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

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

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

Therefore, lhs = rhs.



## Coinduction into coN: $\forall x \in coN$ , x + 0 = x

Theorem: Simple coinduction

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

- (\*)  $\operatorname{pred}(x+0) = \operatorname{no} \text{ whenever } \operatorname{pred}(x) = \operatorname{no}$
- (\*\*)  $\operatorname{pred}(x+y) = \operatorname{pred}(x) + y$  whenever  $\operatorname{pred}(x) \neq \operatorname{no}$

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

Theorem: Simple coinduction

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

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

Proof that  $\forall x \in \text{coN}, x + 0 = x$ .

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

## Coinduction into coN: $\forall x \in coN$ , x + 0 = x

### Theorem: Simple coinduction

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

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

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

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

• If pred(x) = no, then

$$[\operatorname{pred}(\operatorname{lhs}(x))] = [\operatorname{pred}(x+0)]$$

$$= [\operatorname{no}]$$

$$= [\operatorname{pred}(x)]$$

$$[\operatorname{pred}(\operatorname{lhs}(x))] = [\operatorname{pred}(\operatorname{rhs}(x))]$$

#### Theorem: Simple coinduction

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

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

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

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

• If instead  $\operatorname{pred}(x) \neq \operatorname{no}$ , then

$$[\operatorname{pred}(\operatorname{lhs}(x))] = [\operatorname{pred}(x+0)]$$

$$= [\operatorname{pred}(x)+0]$$

$$= [\operatorname{lhs}(\operatorname{pred}(x))]$$

$$= [\operatorname{rhs}(\operatorname{pred}(x))]$$

$$= [\operatorname{pred}(x)]$$

$$[\operatorname{pred}(\operatorname{lhs}(x))] = [\operatorname{pred}(\operatorname{rhs}(x))]$$

Theorem: Simple coinduction

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

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 lhs = rhs.



### Theorem: Simple coinduction

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

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

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

Theorem: Simple coinduction

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

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

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

### Theorem: Simple coinduction

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

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

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

• If pred(x) = no, then

### Theorem: Simple coinduction

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

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

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

• If pred(x) = no, then

$$[\operatorname{pred}(\operatorname{lhs}(x,y))] = [\operatorname{pred}(x+y)]$$
$$= [\operatorname{pred}(y)]$$
$$= [\operatorname{pred}(y+x)]$$
$$[\operatorname{pred}(\operatorname{lhs}(x,y))] = [\operatorname{pred}(\operatorname{rhs}(x,y))]$$

### Theorem: Simple coinduction

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

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

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

• If  $pred(x) \neq no$ , then

### Theorem: Simple coinduction

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

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

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

• If  $pred(x) \neq no$ , then

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

### Theorem: Simple coinduction

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

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

• The theorem asserts lhs = rhs.



# Other properties of $co\mathbb{N}$

### These can be proven coinductively:

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

# Other properties of $co\mathbb{N}$

Coinductive proofs only involve  $\leq 2$  cases:

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

and sometimes no splits at all!

# Other properties of $co\mathbb{N}$

Coinductive proofs only involve  $\leq 2$  cases:

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

and sometimes no splits at all!

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

- (something) = 0
- (something) = succ(something else)
- (something) =  $\infty$

Why induct when you could coinduct?

## Outline

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

### 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.

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

### Lists

Let A be a set.

#### Definition: Lists on A

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

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

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

cons : 
$$A \times \text{List}(A) \longrightarrow \text{List}(A)$$
  
 $(a_0, [a_1, \ldots]) \longmapsto [a_0, a_1, \ldots]$ 

## Colists

Let  $\overline{A}$  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.

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

### Colists

Let A be a set.

#### Definition: Colists on A

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

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

coList(A) comes with an operation

$$\operatorname{pop} : \operatorname{coList}(A) \longrightarrow \{\operatorname{no}\} \sqcup (A \times \operatorname{coList}(A))$$
$$[] \longmapsto \operatorname{no}$$
$$[a_0, a_1, \ldots] \longmapsto (a_0, [a_1, \ldots])$$

## Recursion/Corecursion

### Recursion principle

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

### Recursion/Corecursion

#### Recursion principle

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

#### Corecursion principle

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

### Recursion/Corecursion

#### Recursion principle

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

#### Corecursion principle

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

- pop(u(x)) = no whenever f(x) = no
- pop(u(x)) = (a, u(x')) whenever f(x) = (a, x')

### Recursion/Corecursion

#### Recursion principle

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

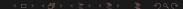
#### Corecursion principle

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

- pop(u(x)) = no whenever f(x) = no
- pop(u(x)) = (a, u(x')) whenever f(x) = (a, x')

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

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



### Induction principle

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

- lhs([]) = rhs([])
- lhs(cons(a, as)) = rhs(cons(a, as)) whenever lhs(as) = rhs(as)

then lhs = rhs.

#### Induction principle

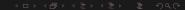
Let lhs, rhs : List(A)  $\rightarrow X$ . If

- lhs([]) = rhs([])
- lhs(cons(a, as)) = rhs(cons(a, as)) whenever lhs(as) = rhs(as)

then lhs = 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



#### Coinduction principle

Let  $lhs, rhs : X \to coList(A)$ .

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

lf

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

(with [no] := no), then lhs = rhs.

#### Coinduction principle

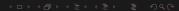
Let  $\mathrm{lhs}, \mathrm{rhs}: X \to \mathrm{coList}(A)$ . Let  $[-]: \mathrm{coList}(a) \to \mathrm{coList}(A)_{/\mathrm{lhs}=\mathrm{rhs}}$  be the quotient map. If

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

(with [no] := no), then lhs = rhs.

#### Example theorems:

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



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

•  $\mathbb{N} \leadsto \mathrm{co}\mathbb{N}$ 

- $\mathbb{N} \leadsto \mathrm{co}\mathbb{N}$
- $List \leadsto coList$

- $\mathbb{N} \leadsto \operatorname{co}\mathbb{N}$
- List  $\rightsquigarrow$  coList
- Finite-size binary trees → potentially infinite binary trees

- $\mathbb{N} \leadsto \operatorname{co}\mathbb{N}$
- List → coList
- Finite-size binary trees → 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

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

### Initial algebras

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

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

### Initial algebras

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

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

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

### Initial algebras

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

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

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

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

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

### Initial algebras

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

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

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

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

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

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

### Initial algebras

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

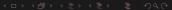
An **algebra** for T is an arrow  $a \xrightarrow{\alpha} T(a)$  in  $\mathfrak{C}$ .

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

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

With identity and composite arrows as in  $\ensuremath{\mathfrak{C}}$ , 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**.

All of the inductive data types are initial algebras.

#### Recursion

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

The recursion principle is its universal property.

#### Induction

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

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

Suppose the equaliser  $\iota: e \rightarrowtail i$  of f and g exists in  $\mathcal{C}$ , and that

 $f \circ \cos \circ T(\iota) = g \circ \cos \circ T(\iota)$ :

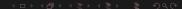
$$T(e) \xrightarrow{T(\iota)} e$$

$$T(i) \xrightarrow{\text{cons}} i$$

$$f \downarrow g$$

$$a$$

Then, f = g.



# Corecursion/Coinduction

Turn all the arrows around.

### Corecursion/Coinduction

#### Turn all the arrows around.

#### Coinduction

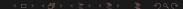
Let  $T(j) \xrightarrow{\text{des}} j$  be an initial algebra.

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

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

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

Then, f = g.



 $T:\mathbf{Set}\to\mathbf{Set}$  given by  $X\mapsto \{\mathrm{no}\}\sqcup X$  has:

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

 $T:\mathbf{Set}\to\mathbf{Set}$  given by  $X\mapsto \{\mathrm{no}\}\sqcup X$  has:

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

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

 $T:\mathbf{Set}\to\mathbf{Set}$  given by  $X\mapsto \{\mathrm{no}\}\sqcup X$  has:

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

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

The initial algebra exists, and is N with discrete topology.

 $T: \mathbf{Set} \to \mathbf{Set}$  given by  $X \mapsto \{\mathrm{no}\} \sqcup X$  has:

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

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

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

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

