Introduction

- Example: "a stream is a natural number followed by a stream." Circular defn?
- Only understood by category theorists, (some) theoretical computer scientists
- Popular slogans: "(categorical) dual of induction", "induction without base cases" / "non-well-founded induction", "induction is smallest fixed-point, coinduction is greatest fixed-point"

(Co)inductive types

- Types represent classes of mathematical objects. e.g. $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{R}$, $\mathbb{N} \to \mathbb{N}$, $\mathbb{N}^2$, $2^{\mathbb{N}}$
- Some basic types are $\mathbf{0} = \bot$ (empty type/False) and $\mathbf{1} = \top$ (unit type/True)
- Given types A, B, can construct $A \times B$, $A + B$, $A \to B$
- Inductive types have the form

$$\text{Inductive } X :=\quad \left|\begin{array}{l} c_1 : A_1(X) \to X \\ \vdots \\ c_n : A_n(X) \to X \end{array}\right.$$

$$\text{Inductive } \mathbb{N} :=\quad \left|\begin{array}{l} 0 : \mathbb{N} \\ \text{succ} : \mathbb{N} \to \mathbb{N} \end{array}\right.$$

$$\text{Inductive List}[A] :=\quad \left|\begin{array}{l} \text{nil} : \text{List}[A] \\ \text{cons} : \text{List}[A] \times A \to \text{List}[A] \end{array}\right.$$

  where $A_i(X)$ are types that may depend on X. The c_i are the constructors of X.

- Can wrap them all into a *single* constructor $c_X : A_1(X) + \cdots + A_n(X) \to X$
- The "inverse" of c_X is the destructor $d_X : X \to A_1(X) + \cdots + A_n(X)$
  - E.g. $d_{\mathbb{N}}$ is the predecessor, $d_{\text{List}[A]}$ is the head and tail functions.
- Coinductive types have the same syntax, but define a different type
  - Inductive = defined by **con**structors, coinductive = defined by **de**structors.
- Can view elements of a (co)inductive type as trees: branching is output of destructor
  - Give example for lists, natural numbers
  - Inductive: trees must be well-founded. Coinductive: don't have to be
  - Use trees to work out the elements of $\overline{\mathbb{N}}$, Colist$[A]$, Str$[A]$

$$\text{Coinductive } \overline{\mathbb{N}} :=\quad \left|\begin{array}{l} \overline{0} : \overline{\mathbb{N}} \\ \text{cosucc} : \overline{\mathbb{N}} \to \overline{\mathbb{N}} \end{array}\right.$$

$$\text{Coinductive Str}[A] :=\quad \left|\begin{array}{l} \text{prepend} : A \times \text{Str}[A] \to \text{Str}[A] \end{array}\right.$$

$$\text{Coinductive Colist}[A] :=\quad \left|\begin{array}{l} \text{conil} : \text{Colist}[A] \\ \text{cocons} : \text{Colist}[A] \times A \to \text{Colist}[A] \end{array}\right.$$

$$\text{Coinductive BinTree} :=\quad \left|\begin{array}{l} \text{niltree} : \text{BinTree} \\ \text{pair} : \text{BinTree} \times \text{BinTree} \to \text{BinTree} \\ \text{left}, \text{right} : \text{BinTree} \to \text{BinTree} \end{array}\right.$$

Categorical semantics

- We will work exclusively in **Set**, the category of sets.
- $\mathbf{0}$ initial if $\forall A\ \exists! f\colon \mathbf{0} \to A$. $\quad$ $\mathbf{1}$ terminal/final if $\forall A\ \exists! f\colon A \to \mathbf{1}$.
- Endofunctor F: maps sets $A \mapsto FA$, functions $(f\colon A \to B) \mapsto Ff\colon FA \to FB$ so that $F(\mathrm{id}_X) = \mathrm{id}_{FX}$, $F(g \circ f) = Fg \circ Ff$.
- F-algebra is pair $(A, \alpha\colon FA \to A)$, while F-coalgebra is pair $(A, \alpha\colon A \to FA)$
- Morphism between F-(co)algebras (A, α) and (B, β) is $f\colon A \to B$ such that

$$\begin{tikzcd} A \arrow[r, "f"] \arrow[d, dash, "\alpha"] & B \arrow[d, dash, "\beta"] \\ FA \arrow[r, "Ff"] & FB \end{tikzcd}$$

- Inductive types are initial algebras for the endofunctor $X \mapsto A_1(X) + \cdots + A_n(X)$
  - E.g. $F\colon X \mapsto \mathbf{1} + X$. $\quad$ $\mathbb{N}$ is initial F-algebra with $* \mapsto 0$, $n \mapsto n+1$.
  - Unpack definition: for any $a \in A$ and $h\colon A \to A$, there is unique $f\colon \mathbb{N} \to A$ such that $f(0) = a$, $f(n+1) = h(f(n))$. Recursion!
- Coinductive types are terminal coalgebras for this endofunctor
- Adámek's theorem: general way to construct initial F-algebras.
  - Start with $\mathbf{0}$, there is unique $f\colon \mathbf{0} \to F(\mathbf{0})$. Keep applying F to get chain

    $$\mathbf{0} \longrightarrow F(\mathbf{0}) \longrightarrow F^2(\mathbf{0}) \longrightarrow F^3(\mathbf{0}) \longrightarrow \cdots$$

    and the colimit ($\approx$ union) C of this chain is initial F-algebra, if $F(C) \cong C$.
  - Dually: final F-coalgebra by taking limit ($\approx$ compatible sequences in product) of

    $$\mathbf{1} \longleftarrow F(\mathbf{1}) \longleftarrow F^2(\mathbf{1}) \longleftarrow F^3(\mathbf{1}) \longleftarrow \cdots$$

  - Explains why inductive types need base cases, but coinductive types don't.
- Example: fix A, consider $F\colon X \mapsto A \times X$. What is terminal F-coalgebra?
  - Use Adámek's. $F^n(\mathbf{1}) = A^n$. The limit is $A^{\mathbb{N}}$ i.e. $\mathrm{Str}[A]$ with head/tail map
- Lambek's theorem: if (A, α) is initial algebra (or final coalgebra), then α is an isomorphism
  - Initial algebras (inductive types) and final coalgebras (coind.) are *fixed points* of F
  - Initial/inductive is *smallest* fixed point, terminal/coinductive is *largest* fixed point
  - Initiality/terminality gives a canonical inclusion $\mathrm{Ind}\, A \hookrightarrow \mathrm{Coind}\, A$

Corecursion

- Given an inductive type X, induction/recursion is the exploitation of the fact that X is an initial algebra for some endofunctor $F\colon \mathbf{Set} \to \mathbf{Set}$
  - To define a function $f\colon X \to A$ by recursion, give A the structure of an F-algebra

- Then initiality of X says there is a unique F-algebra morphism $f\colon X \to A$

- Example: define $f(n) = 2n+3$ by recursion. Informally, $f(0) = 3$, $f(n+1) = f(n)+2$.
  - Draw diagram, unpack: need alg $(c,h)\colon \mathbf{1}+\mathbb{N} \to \mathbb{N}$ with $f(0) = c$, $f(n+1) = h(f(n))$
- For a coinductive type X, <u>corecursion</u> is the exploitation of the fact that X is an *terminal coalgebra* for some endofunctor $F\colon \mathbf{Set} \to \mathbf{Set}$
  - To define $f\colon A \to X$ by **co**recursion, give A the structure of an F-coalgebra
  - Then terminality of X says there is a unique F-coalgebra morphism $f\colon A \to X$
- Example: define $f\colon A \to \mathrm{Str}[A]$, $a \mapsto (a,a,a,\ldots)$ by corecursion
  - $F\colon X \mapsto A \times X$. Need to give F-coalgebra structure to A, i.e. map $\alpha\colon A \to A^2$
  - Draw diagram, pick $\alpha\colon a \mapsto (a,a)$, i.e. $\mathrm{hd}(f(a)) = a$, $\mathrm{tl}(f(a)) = f(a)$
- More informally: to define $f\colon X \to A$ recursively, say what f does on each constructor. To define $f\colon A \to X$ **co**recursively, say what each destructor does on $f(a)$
  - "<u>Simple corecursion</u>": whenever $d_i(f(a)) : X$, then must have $d_i(f(a)) = f(---)$
  - Analogous to "simple recursion" restriction that $f(n+1)$ can depend only on $f(n)$
- More examples of corecursion:

$A^2 \to \mathrm{Str}[A]$	$(a,b) \mapsto (a,b,a,b,a,b,\ldots)$	$\mathrm{hd}(f(a,b)) = a$	$\mathrm{tl}(f(a,b)) = f(b,a)$
$A^{<\omega} \to \mathrm{Str}[A]$	$\sigma \mapsto \sigma\sigma\sigma\sigma\cdots$	$\mathrm{hd}(f(\sigma)) = \mathrm{hd}(\sigma)$	$\mathrm{tl}(f(\sigma)) = f(\tau(\sigma))$
$\mathrm{Str}[A]^2 \to \mathrm{Str}[A]$	$(a_i),(b_i) \mapsto (a_0,b_0,a_1,\ldots)$	$\mathrm{hd}(f(\alpha,\beta)) = \mathrm{hd}(\alpha)$	$\mathrm{tl}(f(\alpha,\beta)) = f(\beta,\mathrm{tl}(\alpha))$
$\mathrm{Str}[A] \to \mathrm{Str}[A]$	$(a_i) \mapsto (a_0,a_2,a_4,\ldots)$	$\mathrm{hd}(f(\alpha)) = \mathrm{hd}(\alpha)$	$\mathrm{tl}(f(\alpha)) = f(\mathrm{tl}(\mathrm{tl}(\alpha)))$

Further topics

- Coinduction as a proof technique
  - Basic idea: two distinct elements of a coinductive type can be distinguished by applying finitely many destructors
  - Can be used to prove two things are equal (e.g. $n+m = m+n$ for conatural n,m)
- There are more powerful forms of coinduction too (I don't know about these)