UNIVERSITY OF BIRMINGHAM

Category Theory

Midlands Graduate School 2026

Sergey Goncharov

April 13–17, 2026

Big Picture / Motivation

- Category theory is a “theory”, like set theory, like group theory, like type theory
- Types in functional programming $\rightsquigarrow$ sets (or set-like objects)
- Functions $\rightsquigarrow$ arrows/morphisms between types
- Category theory gives language to talk about how we compose computations, abstract patterns, and universal constructions that mirror programming constructs (pairs, function types, currying)
- It provides means of abstraction to reason and prove things generically, instead of on case-by-case basis

Course Outline

Part I: Sets and Warm-Up

Sets, functions, isomorphisms, singletons, products, natural numbers

Part II: Categories & Functors

Definitions and examples, products, coproducts, commutative diagrams, universal properties

Part III: Natural Transformations

Natural transformations, exponentials, Yoneda Theorem

Part IV: Adjunctions & Free Objects

Adjunctions, free objects

These slides:

www.sergey-goncharov.org/mgs-2026

Recommended Sources

- **Steve Awodey** — *Category Theory*, 2nd ed., Oxford University Press, 2010
  - [accessible introduction]
- **Emily Riehl** — *Category Theory in Context*, Dover, 2016
  - [freely available at https://math.jhu.edu/~eriehl/161/context.pdf]
- **Roy Crole** — *Categories for Types*, Cambridge University Press, 1993
  - [CS-oriented; Category theory for types & denotational semantics]
- **Saunders Mac Lane** — *Categories for the Working Mathematician*, 2nd ed., Springer, 1998
  - [timeless classics]

Sets and Warm-Up

Why Begin with Sets?

- Mathematics builds on sets
- Types in programming follow set intuition
- Category theory (standardly) also builds on sets
- But: what exactly is a set?
  - OK, that is too ambitious — what is it roughly?
- We start with sets to discover something deeper is hiding underneath

Naïve (Philosophical) Sets

- Set is collection of objects, like

$$X = \left\{ \text{⚽}, \text{🐕}, \pi, \mathbb{N}, \text{❤} \right\}$$

- We write $x \in X$ to say that x is member of X, e.g. $\pi \in X$
- Empty set $\varnothing = \{\,\}$ is a set of no elements
- There are various mechanisms to construct larger sets
- Set comprehension

$$\{x \in S \mid P(x)\} \qquad \text{e.g.} \qquad \text{evens} = \{n \in \mathbb{N} \mid \exists m \in \mathbb{N}.\, n = 2m\}$$

Intuition: Sets are containers

Finite Sets v.s. Finite Lists

- (Finite) lists over set X: $\mathrm{List}(X) = \{[x_1, \ldots, x_n] \mid n \in \mathbb{N}, x_1 \in X, \ldots, x_n \in X\}$
- **Notation:** $[\,]$ — empty list, $x :: xs$ — list with head x and tail xs. E.g. $x :: [\,] = [x]$
- For contrast/analogy: Finite sets are lists, where we additionally enable
  - Commutativity: $[\ldots, x, y, \ldots] \equiv [\ldots, y, x, \ldots]$
  - Idempotence: $[\ldots, x, x, \ldots] \equiv [\ldots, x, \ldots]$
- (Finite) Multisets = lists with commutativity, but not idempotence. Thus

$$\text{Lists} \quad < \quad \text{Multisets} \quad < \quad \text{Sets}$$

Sets (Slightly More) Formally

- Set theory postulates: everything is a set
- Only primitive relation between sets is '$\in$'

We then derive:

- Natural numbers: $0 = \varnothing$, $1 = \{0, \varnothing\} = \{\varnothing\}$, $2 = \{1, \varnothing\} = \{\{\varnothing\}, \varnothing\}, \ldots$
- Set inclusion: $X \subseteq Y$ if $\forall x \in X. x \in Y$
  - Extensionality axiom: $X = Y$ if and only if $X \subseteq Y$ and $Y \subseteq X$
- Powerset: $Y \in \mathcal{P}(X)$ if and only if $Y \subseteq X$
- Relations: $R \subseteq X \times Y$
- Functions: $f : X \to Y$ — such relations $R \subseteq X \times Y$ that for every $x \in X$ there is unique $y \in Y$ with $x\,R\,y$
  - So, in set theory: Sets $\rightsquigarrow$ Relations $\rightsquigarrow$ Functions

Empty Set and Singleton Sets

- No controversy about empty set: $\varnothing = \{\,\}$ — set with no elements
- But what is one-element set? $\{$⚽$\}$? $\{$🐕$\}$? $\{\pi\}$?
- $\{\varnothing\}$ is good candidate
- For every set S, $\{S\}$ is also singleton set
- Singleton sets are characterized by property: they are sets with exactly one element
- $\{\varnothing\}$ is a concrete representative in the class of all singletons

Note on Classes

- All sets do not form a set, but a class

$$\text{Set} < \text{Class}$$

- Division into sets and proper classes is way to avoid paradoxes
- Other approach: universes

$$\text{Set}_0 < \text{Set}_1 < \text{Set}_2 < \dots$$

  where the union of all sets in Set_i is in Set_{i+1}
- By saying “set” we will imagine set from Set_i, and by saying “class” we will imagine set from Set_{i+1}

Natural Numbers as Sets

- In computer science, we count from 0: $\mathbb{N} = \{0, 1, \ldots\}$!
- Von Neumann numerals: $0 := \varnothing, n + 1 := n \cup \{n\}$ So:

$$0 = \varnothing, \quad 1 = \{\varnothing\}, \quad 2 = \{\varnothing, \{\varnothing\}\}, \quad 3 = \{\varnothing, \{\varnothing\}, \{\varnothing, \{\varnothing\}\}\}, \ldots$$

  This is very convenient, because $n + 1 = \{0, \ldots, n\}$
- Thus, we can — and we will! — identify natural numbers with sets
- Alternatively Zermelo numerals: $0 := \varnothing, n + 1 := \{n\}$. This is less convenient

Ordered Pairs and Products

We would like to define Cartesian products $X \times Y = \{(x, y) \mid x \in X, y \in Y\}$. But what is (x, y)?

- Kuratowski definition: $(x, y) := \big\{\{x\}, \{x, y\}\big\}$
- Then (PROVE IT):

$$(x, y) = (x', y') \iff x = x' \quad \text{and} \quad y = y'.$$

- Using this, we can define:

$$X \times Y := \Big\{\{\{x\}, \{x, y\}\} \subseteq \mathcal{P}(X \cup Y) \mid x \in X,\ y \in Y\Big\}$$

Pairs and Cartesian products are constructed objects!

Disjoint Unions

- Union of sets: $x \in X \cup Y$ iff $x \in X$ or $x \in Y$
- How to define Disjoint Union $X + Y$ (= $X \uplus Y$)?
- Maybe $X + Y = \{(x, 🎸) \mid x \in X\} \cup \{(x, 🔨) \mid y \in Y\}$?

Disjoint Unions

- Union of sets: $x \in X \cup Y$ iff $x \in X$ or $x \in Y$
- How to define Disjoint Union $X + Y$ (= $X \uplus Y$)?
- Maybe $X + Y = \{(x, 🎸) \mid x \in X\} \cup \{(x, 🔨) \mid y \in Y\}$?
- Maybe $X + Y = \{(x, 🍔) \mid x \in X\} \cup \{(x, 🐎) \mid y \in Y\}$?

Disjoint Unions

- Union of sets: $x \in X \cup Y$ iff $x \in X$ or $x \in Y$
- How to define Disjoint Union $X + Y$ ($= X \uplus Y$)?
- Maybe $X + Y = \{(x, 🎸) \mid x \in X\} \cup \{(x, 🔨) \mid y \in Y\}$?
- Maybe $X + Y = \{(x, 🍔) \mid x \in X\} \cup \{(x, 🐎) \mid y \in Y\}$?
- More seriously: $X + Y = \{(x, 0) \mid x \in X\} \cup \{(x, 1) \mid y \in Y\}$

Now compute with von Neumann numerals: $1 = \{0\}$, $2 = \{0, 1\}$. Then

$$1 + 1 = \{(1, 0), (1, 1)\} = \big\{\{\{1\}, \{1, 0\}\}, \{\{1\}\}\big\}$$

This is not equal to $2 = \{0, 1\}$! So

$$1 + 1 \neq 2$$

Functions between Sets

- Function $f : X \to Y$, assigns each $x \in X$ a unique $f(x) \in Y$
- Functions $f : X \to Y$ are special relations $f \subseteq X \times Y$
- Identity function $id_X : X \to X$, $id_X(x) = x$
- Equality of functions $f = g$ if for all x, $f(x) = g(x)$
  - This is provable principle (PROVE IT), called function extensionality

Composition:

$$(g \circ f)(x) = g(f(x))$$

Proposition: $f \circ id = f$, $id \circ f = f$ and $f \circ (g \circ h) = (f \circ g) \circ h$ (associativity)

Bijection vs Isomorphism

Function $f : X \to Y$ is

- Injection if $f(x) = f(x')$ entails $x = x'$
- Surjection if for every $y \in Y$ there is $x \in X$, such that $f(x) = y$
- Bijection if it is both injection and surjection

Alternative view: $f : X \to Y$ is isomorphism if for some $f^{-1} : Y \to X$,

$$f \circ f^{-1} = id_Y \quad \text{and} \quad f^{-1} \circ f = id_X$$

Proposition: f is bijection if and only if f is isomorphism

Some Properties of Isomorphisms

It follows that (PROVE IT)

- f^{-1} is unique if exists
- If f is isomorphism, so is f^{-1}
- id is isomorphism and $id^{-1} = id$
- Composition of isomorphisms $f \circ g$ is isomorphism and $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$

These are derivable from definition, without using that isomorphism = bijection!

Notation: $X \cong Y$ if there is isomorphism $f : X \to Y$

Up-to Isomorphism

- Big innovation of category theory: work with things via their descriptions
- We narrow down objects of interest as those that “fit best” to such description
- Such descriptions together with their best-fit condition are called universal properties
- They determine objects not uniquely, but up-to isomorphism

Example. A singleton set 1 is characterised by: there is exactly one function $X \to 1$ for every set X

- $\{\varnothing\}$, $\{\{\varnothing\}\}$, $\{⚽\}$ all satisfy this — and are all isomorphic
- Any two singletons are isomorphic, so we speak of the singleton up to isomorphism

Induction/Recursion for Natural Numbers

- Set theory postulates existence of natural numbers $\mathbb{N}$ as smallest set that contains $0 = \varnothing$ and closed under $n \mapsto n + 1 := n \cup \{n\}$
- Induction is paraphrasing this: if $P \subseteq \mathbb{N}$ with $0 \in P$ and $n \in P \Rightarrow n{+}1 \in P$, then $P = \mathbb{N}$
  - Because predicates over $\mathbb{N} \iff$ subsets of $\mathbb{N}$:

$$P(0) \wedge (\forall n.\, P(n) \Rightarrow P(n+1)) \Rightarrow \forall n.\, P(n)$$

- (Structural) Recursion: for any $a \in A$ and $h : A \to A$, there is unique $f : \mathbb{N} \to A$ with

$$f(0) = a \qquad f(n{+}1) = h(f(n))$$

- Primitive recursion is derivable from recursion (PROVE IT): for any $g : X \to A$ and $h : A \times X \times \mathbb{N} \to A$, there is unique $f : X \times \mathbb{N} \to A$ with

$$f(x, 0) = g(x) \qquad f(x, n{+}1) = h(f(x, n), x, n)$$

Induction/Recursion for Lists

- Given set X, $\mathrm{List}(X)$ is constructable as smallest set that contains $[\,]$ and closed under $xs \mapsto x :: xs$ for $x \in X$
- Induction: $P([\,]) \wedge (\forall x \in X, xs.\, P(xs) \Rightarrow P(x :: xs)) \Rightarrow \forall xs.\, P(xs)$
- (Structural) Recursion: for any $a \in A$ and $h : X \times A \to A$, there is unique $f : \mathrm{List}(X) \to A$ with

$$f([\,]) = a \qquad f(x :: xs) = h(x, f(xs))$$

Questions:

- What is primitive recursion for lists?
- Why it is derivable (PROVE IT)
- Define *length* and concatenation recursively

Remarks:

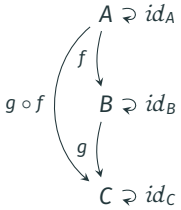
- Structural recursion $\neq$ general recursion (no arbitrary self-calls)
- Induction — proof principle; recursion — definition principle

Categories & Functors

Definition of Category

Definition: Category $\mathcal{C}$ consists of:

- Class of objects $\mathrm{Ob}(\mathcal{C})$
- For every A, B set of morphisms $\mathrm{Hom}_{\mathcal{C}}(A, B)$. Write $f : A \to B$
- Composition: if $f : A \to B$ and $g : B \to C$ then $g \circ f : A \to C$
- Identity morphisms $id_A : A \to A$



Subject to associativity $(h \circ g) \circ f = h \circ (g \circ f)$ and identity laws $id \circ f = f = f \circ id$

- We can imagine categories as (multi)graphs: objects = nodes, morphisms = edges
- We defined locally small categories, but see Slide 9 for comment on size issues!

Sets as Category

Category of sets **Set**

- **Objects:** sets
- **Morphisms:** functions between sets
- Composition: usual function composition
- Identities: identity functions

Further set-like categories (objects = sets, identities = identity functions):

- Relations (**Rel**) — morphisms are relations
- Partial functions
- Injections
- Isomorphisms

...

Category Pos

- **Objects:** partially ordered sets $(P, \leq)$
- **Morphisms:** monotone functions

$$f : (P, \leq_P) \to (Q, \leq_Q)$$

such that

$$x \leq_P y \Rightarrow f(x) \leq_Q f(y)$$

Example:

Category Mon

- **Objects:** monoids $(M, \cdot, e)$
- **Morphisms:** monoid homomorphisms are maps $f : M \to N$, such that unit and multiplication are preserved:

$$f(x \cdot y) = f(x) \cdot f(y) \qquad f(e_M) = e_N$$

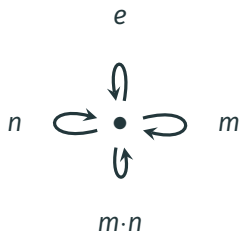
Example: $length : (\mathsf{List}(A), +\!\!+, [\,]) \to (\mathbb{N}, +, 0)$ — preserves unit and multiplication:

$$\begin{aligned} length([\,]) &= 0 \\ length(xs +\!\!+ ys) &= length(xs) + length(ys) \end{aligned}$$

Monoid as One-Object Category

Fix some monoid $(M, \cdot, e)$. We can turn it into category!

- **Objects:** — exactly one, say $\bullet$
- **Morphisms:** $\mathrm{Hom}(\bullet, \bullet) = M$
- Composition is given by monoid multiplication
- Identity given by e



This is equivalence: Monoids $\iff$ one-object categories (PROVE IT)

Monoid homomorphisms $\iff$ functors between one-object categories (PROVE IT)

Posets as Categories

Fix poset $(P, \leq)$. It is also category:

- **Objects:** elements of P
- **Morphisms:** $\mathrm{Hom}(x, y) = \{\star\}$ if $x \leq y$ and $\mathrm{Hom}(x, y) = \varnothing$ otherwise
  (at most one morphism between two objects)
- Unit of composition = reflexivity
- Associativity of composition = transitivity

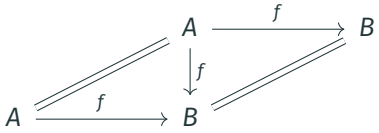
Hint: Categorical notions become order-theoretic when instantiated to poset-categories.
To understand a complex categorical notion, instantiate it to poset-category!

Example Categories: Summary

Category	Objects	Morphisms	Composition
Set	sets	functions	function composition
Pos	posets	monotone maps	function composition
Mon	monoids	homomorphisms	function composition
Rel	sets	relations $R \subseteq X \times Y$	relational composition
$(P, \leq)$	elements of P	$\star$ if $x \leq y$	transitivity
$(M, \cdot, e)$	$\{\star\}$	elements of M	monoid multiplication

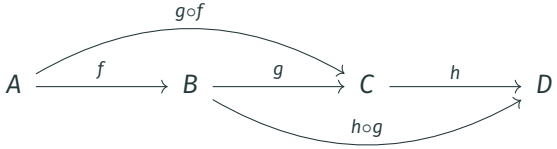
Commutative Diagrams

- For $f : A \to B$, we can express $f \circ id_A = f$ and $id_B \circ f = f$ diagrammatically:



('$=$' denotes '$\xrightarrow{id}$')

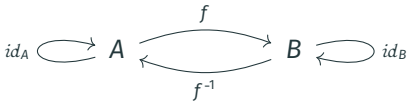
- Associativity:



But that is redundant, because we read sequence f, g as $g \circ f$ — associativity is baked in!

Isomorphisms

Morphism $f : A \to B$ is an isomorphism if there exists $f^{-1} : B \to A$ with $f^{-1} \circ f = id_A$ and $f \circ f^{-1} = id_B$:



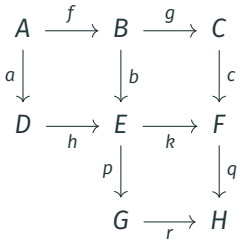
- A and B are isomorphic, written $A \cong B$, if such f exists
- f^{-1} is unique: if $g \circ f = id_A$ and $f \circ g = id_B$, then

$$g = g \circ id_B = g \circ (f \circ f^{-1}) = (g \circ f) \circ f^{-1} = id_A \circ f^{-1} = f^{-1}$$

- In **Set**: isomorphism = bijection; in **Pos**: order-isomorphism; in **Mon**: monoid isomorphism
- Group is one-object category in which every morphism is isomorphism (PROVE IT)

Commutative Diagrams and Equational Reasoning

A diagram commutes if every directed path with the same start and end gives the same composite.



If all three squares commute:

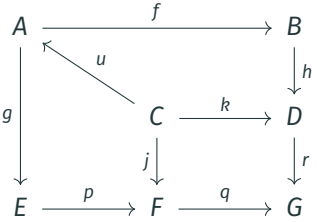
$$b \circ f = h \circ a, \quad c \circ g = k \circ b, \quad q \circ k = r \circ p$$

then the outer paths from A to H agree:

$$\begin{aligned} q \circ c \circ g \circ f &= q \circ (k \circ b) \circ f \\ &= (q \circ k) \circ (b \circ f) \\ &= (r \circ p) \circ (h \circ a) \\ &= r \circ p \circ h \circ a \end{aligned}$$

Thus, commutative diagrams = pictorial representations of equational proofs!

Not Every Diagram Proves Something



Commuting cells (by assumption):

$$h \circ f \circ u = k$$
$$p \circ g \circ u = j$$
$$r \circ k = q \circ j$$

Chaining all three:

$$\begin{aligned}(r \circ h \circ f) \circ u &= r \circ k \\ &= q \circ j \\ &= (q \circ p \circ g) \circ u\end{aligned}$$

But this does not imply $r \circ h \circ f = q \circ p \circ g$, because we generally cannot discard u!

Terminal and Initial Objects

In category $\mathcal{C}$:

- Terminal object 1: for every object A there is exactly one morphism $A \to 1$
- Initial object 0: for every object A there is exactly one morphism $0 \to A$

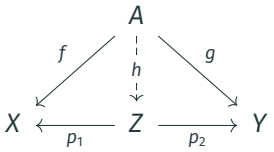
Examples:

- In **Set**: any singleton $\{*\}$ is terminal; $\varnothing$ is initial
- In **Pos**: any one-element poset is terminal; $\varnothing$ is initial
- In **Mon**: trivial monoid $\{e\}$ is terminal and initial
- In every poset category: terminal = greatest element; initial = least element

Key viewpoint: elements of X in **Set** correspond to arrows $1 \to X$, i.e. $\text{elements}(X) \cong \text{Hom}(1, X)$. So, $f(x)$ must be interpreted as composition $1 \xrightarrow{x} X \xrightarrow{f} Y$!

Product Universal Property

- **Object:** a product of X and Y is an object Z
- **Diagram (span):** morphisms $p_1 : Z \to X$ and $p_2 : Z \to Y$
- **Universal property:** for any $f : A \to X$ and $g : A \to Y$ there is a unique $h : A \to Z$ with $p_1 \circ h = f$ and $p_2 \circ h = g$:



Selected product $X \times Y$**:** specific choice of product for every pair (X, Y); all products of X and Y are uniquely isomorphic

Notation: With $X \times Y$, use $\mathit{fst}, \mathit{snd}$ for the chosen projections; $\langle f, g \rangle$ for the induced h

Examples of Products

- **Set:** $X \times Y = \{(x, y) \mid x \in X,\ y \in Y\}$; projections $(x, y) \mapsto x$ and $(x, y) \mapsto y$; pairing $\langle f, g \rangle(x) = (f(x), g(x))$
- **Pos:** $X \times Y$ with pointwise order: $(x, y) \leq (x', y')$ iff $x \leq x'$ and $y \leq y'$
- **Mon:** componentwise monoid $(M \times N,\ \cdot,\ (e_M, e_N))$ where

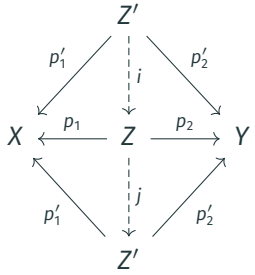
$$(m, n) \cdot (m', n') = (m \cdot m',\ n \cdot n')$$

- **Poset category:** product of x and y = meet $x \wedge y$ (=greatest lower bound), when it exists

Products are Up-To-Isomorphism

Let (Z, p_1, p_2) and (Z', p_1', p_2') both be products of X and Y:

- Universal property of Z: $\exists!\, i \colon Z' \to Z$ with $p_1 \circ i = p_1'$, $p_2 \circ i = p_2'$
- Universal property of Z' on span (p_1, p_2): $\exists!\, j \colon Z \to Z'$ with $p_1' \circ j = p_1$, $p_2' \circ j = p_2$
- By uniqueness: $j \circ i = id_{Z'}$
- By symmetric argument: $i \circ j = id_Z$, so $Z \cong Z'$



Example: Take selected product $(X \times Y, \mathit{fst}, \mathit{snd})$. Then $(Y \times X, \mathit{snd}, \mathit{fst})$ is also product of X and Y — isomorphism swap $= \langle \mathit{snd}, \mathit{fst} \rangle \colon X \times Y \xrightarrow{\sim} Y \times X$

Axioms of Binary Products

For selected products, one can show (PROVE IT) :

(β_1) $fst \circ \langle f, g \rangle = f$

(β_2) $snd \circ \langle f, g \rangle = g$

(η) $\langle fst, snd \rangle = id_{X \times Y}$

(nat) $\langle f, g \rangle \circ h = \langle f \circ h,\ g \circ h \rangle$

Theorem: This is a sound and complete axiomatization of binary products (PROVE IT)

Example (Uniqueness of pairing): if $fst \circ k = f$ and $snd \circ k = g$, then

$$k \ = \ id \circ k \overset{(\eta)}{=} \langle fst, snd \rangle \circ k \overset{(\mathrm{nat})}{=} \langle fst \circ k,\ snd \circ k \rangle \overset{(\mathrm{assm.})}{=} \langle f, g \rangle$$

Dual Category

Given category $\mathcal{C}$, the opposite category $\mathcal{C}^{\mathrm{op}}$ has:

- Same objects as $\mathcal{C}$
- Morphisms reversed: $f : A \to B$ in $\mathcal{C}$ becomes $f^{\mathrm{op}} : B \to A$ in $\mathcal{C}^{\mathrm{op}}$
- Composition: $g^{\mathrm{op}} \circ f^{\mathrm{op}} = (f \circ g)^{\mathrm{op}}$

Principle of Duality: any theorem proved in $\mathcal{C}$ yields a dual theorem in $\mathcal{C}^{\mathrm{op}}$ — just reverse all arrows

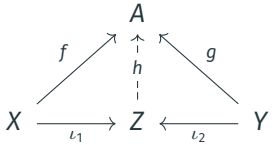
Examples:

- Poset $(P, \leq)^{\mathrm{op}}$ = poset with reversed order $(P, \geq)$
- Initial objects in $\mathcal{C}$ = terminal objects in $\mathcal{C}^{\mathrm{op}}$

Coproduct via Duality

Coproduct = product in $\mathcal{C}^{\mathrm{op}}$ — obtained by reversing all arrows in the product definition

- **Object:** a coproduct of X and Y is an object Z
- **Diagram (cospan):** morphisms $\iota_1 \colon X \to Z$ and $\iota_2 \colon Y \to Z$
- **Universal property:** for any $f \colon X \to A$ and $g \colon Y \to A$ there is a unique $h \colon Z \to A$ with $h \circ \iota_1 = f$ and $h \circ \iota_2 = g$:



Selected coproduct $X + Y$**:** choice of coproduct for every pair; notation inl, inr for chosen injections; $[f, g]$ — co-pairing — for the induced h

Example: In **Set**, $X + Y$ = disjoint union (PROVE IT)

Axioms of Binary Coproducts

For selected coproducts, one can show:

(β_1) $[f, g] \circ inl = f$

(β_2) $[f, g] \circ inr = g$

(η) $[inl, inr] = id_{X+Y}$

(nat) $h \circ [f, g] = [h \circ f,\ h \circ g]$

Theorem: This is sound and complete axiomatization of binary coproducts

Proof: Duality

Example: Every $f : X + Y \to Z$ has form $[f_1, f_2]$ for $f_1 : X \to Z$, $f_2 : Y \to Z$. Indeed:

$$f \ = \ f \circ id \ \overset{(\eta)}{=} \ f \circ [inl, inr] \ \overset{(\mathrm{nat})}{=} \ [f \circ inl,\ f \circ inr]$$

Categories & Functors

Functors

Definition of Functor

(Covariant) Functor $F\colon \mathcal{C} \to \mathcal{D}$ acts on objects and morphisms as follows:

$$A \in \mathrm{Ob}(\mathcal{C}) \mapsto F(A) \in \mathrm{Ob}(\mathcal{D}) \qquad \begin{array}{c} A \\ f \downarrow \\ B \end{array} \longmapsto \begin{array}{c} F(A) \\ \downarrow F(f) \\ F(B) \end{array}$$

preserving composition and identities:

$$F(id_A) = id_{F(A)} \qquad F(g \circ f) = F(g) \circ F(f)$$

- Functor $F\colon \mathcal{C} \to \mathcal{C}$ is called endofunctor
- Functors compose: $(GF)(A) = G(F(A))$, $(GF)(f) = G(F(f))$; categories and functors form **Cat** — category of categories, modulo size issues

Examples of Functors

Endofunctors on Set:

- Powerset: $\mathcal{P}(X)$; on $f : X \to Y$ acts as direct image $\mathcal{P}(f)(S) = \{f(x) \mid x \in S\}$
- List: List(X) = finite lists over X; List(f)($[x_1, \ldots, x_n]$) $= [f(x_1), \ldots, f(x_n)]$
- Maybe: $X \mapsto X + \{\text{Nothing}\}$:
  - Maybe(f)($inl\, x$) $= inl(f(x))$
  - Maybe(f)($inr\, x$) $= inr\, x$

Between different categories:

- Forgetful functor G: **Mon** $\to$ **Set**: sends $(M, \cdot, e)$ to M, homomorphisms to functions
- Free monoid functor F: **Set** $\to$ **Mon**: sends X to List(X)

Bifunctors

Bifunctor $F : \mathcal{C} \times \mathcal{D} \to \mathcal{E}$: functor on product category. Unraveling definition:

- $F(id_A, id_B) = id_{F(A,B)}$
- $F(f' \circ f,\ g' \circ g) = F(f', g') \circ F(f, g)$

Example: Binary product $(- \times -) : \mathcal{C} \times \mathcal{C} \to \mathcal{C}$ (if $\mathcal{C}$ has binary products): on morphisms $f : A \to A'$, $g : B \to B'$, gives

$$f \times g = \langle f \circ \mathit{fst},\ g \circ \mathit{snd} \rangle : A \times B \to A' \times B'$$

Contravariant and Mixed-Variance Functors

Contravariant functor $F : \mathcal{C}^{\mathrm{op}} \to \mathcal{D}$: sends $f : A \to B$ to $F(f) : F(B) \to F(A)$; reverses composition: $F(g \circ f) = F(f) \circ F(g)$

Mixed variance $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{D} \to \mathcal{E}$: contravariant in first argument, covariant in second

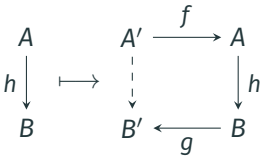
Example: $\mathrm{Hom}(-,-) : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$ is mixed-variance (see next slide)

Duality: contravariant functor $\mathcal{C}^{\mathrm{op}} \to \mathcal{D}$ is the same as covariant functor $\mathcal{C} \to \mathcal{D}^{\mathrm{op}}$; choice of perspective is a matter of convention

Hom-Functor

Hom-functor $\mathrm{Hom}(-,-)\colon \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$: For given $f\colon A' \to A$ and $g\colon B \to B'$, $\mathrm{Hom}(f,g)$ sandwiches input between f and g:

$$\mathrm{Hom}(f,g)\colon \mathrm{Hom}(A,B) \to \mathrm{Hom}(A',B')$$



This specializes in two ways:

- $\mathrm{Hom}(A,-)\colon \mathcal{C} \to \mathbf{Set}$ — covariant; acts by postcomposition: $h \mapsto f \circ h$
- $\mathrm{Hom}(-,B)\colon \mathcal{C}^{\mathrm{op}} \to \mathbf{Set}$ — contravariant; acts by precomposition: $h \mapsto h \circ f$

Contravariant powerset functor sends $X \mapsto \mathcal{P}(X)$ and $f\colon X \to Y$ to $f^{-1}[-]\colon \mathcal{P}(Y) \to \mathcal{P}(X)$

This is isomorphic to $\mathrm{Hom}_{\mathbf{Set}}(X, 2)$ (PROVE IT)

Full and Faithful Functors

Functor $F\colon \mathcal{C} \to \mathcal{D}$ induces maps on hom-sets:

$$F_{A,B}\colon \mathrm{Hom}_{\mathcal{C}}(A, B) \to \mathrm{Hom}_{\mathcal{D}}(FA, FB), \qquad f \mapsto F(f)$$

- F is faithful if every $F_{A,B}$ is injective
- F is full if every $F_{A,B}$ is surjective
- F is fully faithful if every $F_{A,B}$ is a bijection

Examples:

- Forgetful $G\colon$ **Mon** $\to$ **Set**: faithful (distinct homomorphisms remain distinct) but not full (not every function is a homomorphism)
- Inclusion $(P, \leq) \hookrightarrow$ **Set** via poset-as-category: faithful (Hom has at most one element), not full in general

Natural Transformations

Definition: Natural Transformation

For functors $F, G \colon \mathcal{C} \to \mathcal{D}$, natural transformation α assigns to each object X morphism $\alpha_X \colon F(X) \to G(X)$ (component at X) such that for every $f \colon X \to Y$ naturality square commutes:

$$G(f) \circ \alpha_X \;=\; \alpha_Y \circ F(f)$$

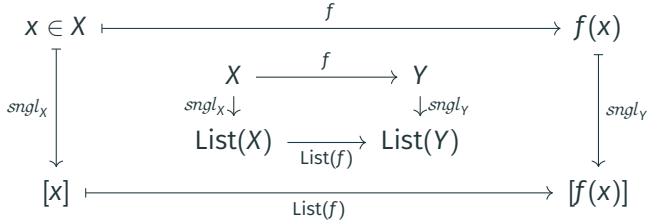
- Write $\alpha \colon F \Rightarrow G$
- α is a natural isomorphism if each α_X is an isomorphism; write $F \cong G$

Programming intuition: Naturality $\approx$ parametric polymorphism — a parametrically polymorphic function $\forall A.\, F(A) \to G(A)$ is automatically natural

Examples: Natural Transformations

Natural Transformations between Endofunctors on Set:

- Identity: $id\colon\ \mathsf{Id} \Rightarrow \mathsf{Id}$ — trivially natural
- Singleton: $sngl\colon\ \mathsf{Id} \Rightarrow \mathsf{List}$, $sngl_X(x) = [x]$:

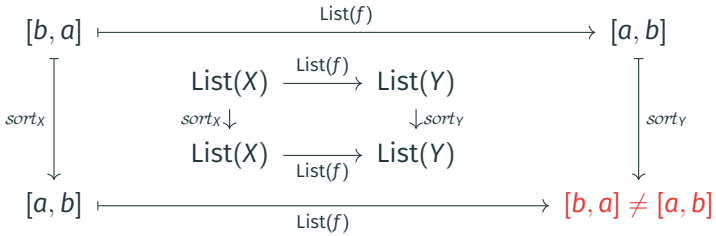


- Flatten: $\mu\colon\ \mathsf{List}\,\mathsf{List} \Rightarrow \mathsf{List}$, $\mu_X([[a,b],[c]]) = [a,b,c]$ — natural (PROVE IT)
- Reverse: $rev\colon\ \mathsf{List} \Rightarrow \mathsf{List}$, $rev_X([x_1,\ldots,x_n]) = [x_n,\ldots,x_1]$ — natural (PROVE IT)

Non-Example: Sort is Not Natural

Let $sort_X$ sort $\mathrm{List}(X)$ by some total order on X

Counterexample: Let $f : \{a, b\} \to \{a, b\}$ swap a and b where $a < b$:



Intuitively, components of natural transformations must be agnostic about properties of objects not preserved by morphisms

Functor Category $[\mathcal{C}, \mathcal{D}]$

- **Objects**: functors $\mathcal{C} \to \mathcal{D}$
- **Morphisms**: natural transformations $F \Rightarrow G$
- **Composition**: for $\alpha \colon F \Rightarrow G$ and $\beta \colon G \Rightarrow H$, define $(\beta \circ \alpha)_X = \beta_X \circ \alpha_X$

$$\begin{array}{ccc} F(X) & \xrightarrow{F(f)} & F(Y) \\ {\scriptstyle \alpha_X}\downarrow & & \downarrow{\scriptstyle \alpha_Y} \\ G(X) & \xrightarrow{G(f)} & G(Y) \\ {\scriptstyle \beta_X}\downarrow & & \downarrow{\scriptstyle \beta_Y} \\ H(X) & \xrightarrow[H(f)]{} & H(Y) \end{array}$$

- **Identity**: $(id_X \colon X \to X)_X$

Natural Isomorphisms

$\alpha : F \Rightarrow G$ is natural isomorphism if each component α_X is isomorphism

- Equivalently: α is isomorphism in $[F, G]$ (PROVE IT)
- F and G are naturally isomorphic, written $F \cong G$, if such α exists

Examples (PROVE IT) :

- Swap: $(- \times -) \cong (- \times -)$ via $swap_{X,Y} : X \times Y \to Y \times X$, $swap_{X,Y} = \langle snd, fst \rangle$
- Products are associative: $(X \times Y) \times Z \cong X \times (Y \times Z)$ naturally in X, Y, Z
- In **Set**: $\mathrm{Hom}(1, X) \cong X$ naturally in X (elements as arrows)

Practical Remarks

- Often, instead "α is natural transformation $F \Rightarrow G$" one says "family $\alpha_X \colon F(X) \to G(X)$ is natural in X"
- Same for bi-functors: $\alpha_{X,Y} \colon F(X,Y) \to G(X,Y)$ is natural in X and Y, etc
- Naturality in $(X,Y) \Leftrightarrow$ naturality in X + naturality in Y (PROVE IT):

$$\begin{array}{ccc} F(X,Y) & \xrightarrow{\alpha_{X,Y}} & G(X,Y) \\ {\scriptstyle F(f,Y)}\downarrow & & \downarrow{\scriptstyle G(f,Y)} \\ F(X',Y) & \xrightarrow[\alpha_{X',Y}]{} & G(X',Y) \end{array} \qquad \begin{array}{ccc} F(X,Y) & \xrightarrow{\alpha_{X,Y}} & G(X,Y) \\ {\scriptstyle F(X,g)}\downarrow & & \downarrow{\scriptstyle G(X,g)} \\ F(X,Y') & \xrightarrow[\alpha_{X,Y'}]{} & G(X,Y') \end{array}$$

Further Remarks

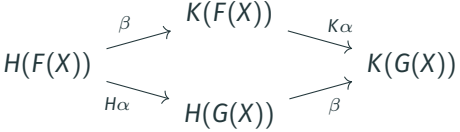
Composition of natural transformations and functors (whiskering):

- $K\alpha \colon (K\alpha)_X = K(\alpha_X)$ (post-whiskering)
- $\beta H \colon (\beta H)_X = \beta_{H(X)}$ (pre-whiskering)

Despite symmetry: $K\alpha$ is α wrapped in K, while βH — alternatively β_H — only varies component of β. So, more suggestively:

$$(K\alpha)_X = K(\alpha_X) \qquad (\beta_H)_X = \beta_{HX}$$

Often, natural transformations are conflated with their components, e.g:



Natural Transformations

Exponentials

Hom's v.s. Exponentials

- External Hom: $\mathrm{Hom}_{\mathcal{C}}(X, Y)$ lives outside $\mathcal{C}$ — $\mathrm{Hom} : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathbf{Set}$
- Internal Hom = Exponential object Y^X — representing same morphisms inside $\mathcal{C}$:

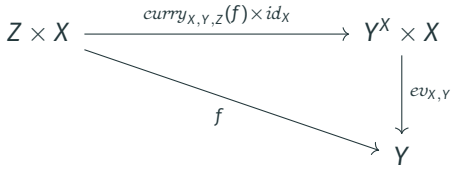
$$(-)^{(-)} : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$$

- **Example in Pos**:
  - $\mathrm{Hom}_{\mathbf{Pos}}(X, Y)$ = monotone maps $X \to Y$ — set, outside **Pos**
  - Y^X in **Pos** = same maps with pointwise order: $f \leq g \iff \forall x. f(x) \leq g(x)$ — inside **Pos**

Exponential Object: Definition

Fix object X in a category $\mathcal{C}$ with binary products

- **Object**: Y^X (also written $[X, Y]$ or $X \Rightarrow Y$)
- **Arrows**: evaluation morphism $ev_{X,Y} : Y^X \times X \to Y$
- **Universal property**: for any $f : Z \times X \to Y$, there exists a unique $curry_{X,Y,Z}(f) : Z \to Y^X$ such that:



Same caveat as before about “an exponential” and “selected exponential” applies

Axioms of Exponentials

For selected exponentials, one can show (PROVE IT):

(β) $ev \circ (curry(f) \times id) = f$

(η) $curry(ev) = id$

(nat) $curry(f) \circ g = curry(f \circ (g \times id))$

Theorem: This axiomatization is sound and complete(PROVE IT)

Exponential Object: Basic Properties

- For $g : Z \to Y^X$, define $uncurry(g)$ as

$$Z \times X \xrightarrow{g \times id_X} Y^X \times X \xrightarrow{ev_{X,Y}} Y$$

- Then, $uncurry = curry^{-1}$, witnessing isomorphism (PROVE IT)

$$\mathrm{Hom}(Z \times X, Y) \cong \mathrm{Hom}(Z, Y^X) :$$

- $curry$, $uncurry$ are natural in all involved parameters, e.g. (nat) is precisely naturality of $curry_{X,Y,Z}$ in Z:

$$\begin{array}{ccc} \mathrm{Hom}(Z \times X,\ Y) & \xrightarrow{curry} & \mathrm{Hom}(Z,\ Y^X) \\ \Big\downarrow{\scriptstyle -\circ (h \times id_X)} & & \Big\downarrow{\scriptstyle -\circ h} \\ \mathrm{Hom}(Z' \times X,\ Y) & \xrightarrow[curry]{} & \mathrm{Hom}(Z',\ Y^X) \end{array}$$

Examples of Exponentials

- **Set**: Y^X = set of functions $X \to Y$; $ev(g, x) = g(x)$; $curry(f)(z)(x) = f(z, x)$
- **Pos**: Y^X = only monotone functions $X \to Y$ (!) with pointwise order:

$$g \leq h \;\Leftrightarrow\; \forall x.\, g(x) \leq h(x)$$

  $curry$ and ev are defined by same expressions
- Poset categories: recall, $x \to y$ iff $x \leq y$
  - exponential $y^x = (x \Rightarrow y)$ is the Heyting implication: largest z with $z \wedge x \leq y$
  - ev is modus ponens $z \wedge (z \Rightarrow y) \leq y$
- **Non-Example:** in **Mon**, exponentials generally do not exist (PROVE IT)

Cartesian Closed Categories

Category $\mathcal{C}$ is Cartesian closed (CCC) if it has:

- terminal object 1
- binary products $X \times Y$ for all X, Y
- exponentials Y^X for all X, Y

CCC are exactly categorical models of typed λ-calculus (with product types)!

Examples:

- **Set** — suitable for semantics of higher-order without divergence (cf. Paul's course)
- Groupoids — like sets, but proof-relevant (cf. Thorsten's course)
- Categories of domains (e.g. Scott domains) for modeling divergence

Natural Transformations

Yoneda Theorem (Lemma)

Polynomial Functors

Polynomial functors are functors of the form:

$$P(X) \;=\; \coprod_{i \in I} X^{n_i} \qquad n_i \in \mathbb{N}$$

where $X^n = \underbrace{X \times \cdots \times X}_{n}$ (powers) and $\coprod_{i \in I}$ are iterated products and coproducts

Example: Polynomial functors model algebraic signatures. For example, monoid $(M, e, \cdot)$ (in **Set**) is represented by morphism (function)

$$f \colon P(M) \to M \qquad \text{where} \qquad P(X) = 1 + X \times X$$

Recall, necessarily $f = [f_1, f_2]$ with $f_1 \colon 1 \to M$, $f_2 \colon M \times M \to M$. $f_1(\star) = e$, $f_2(x, y) = x \cdot y$

Example: $\mathrm{List}\, X = 1 + X + X^2 + \ldots = \coprod_{n \in \mathbb{N}} X^n$ (needs countable coproducts)

“Polymorphic” List Transformations

Problem: Compute $\mathsf{Nat}(\mathsf{List}, \mathsf{List}) =$ all natural transformations $\alpha : \ \mathsf{List} \Rightarrow \mathsf{List}$

For example,

- $rev \in \mathsf{Nat}(\mathsf{List}, \mathsf{List})$
- $(xs \mapsto [\,]) \in \mathsf{Nat}(\mathsf{List}, \mathsf{List})$
- $(\mathit{case}\ xs\ \ \mathit{of}\ \ [\,] \mapsto [\,]; [y \mid ys] \mapsto ys) \in \mathsf{Nat}(\mathsf{List}, \mathsf{List})$

Natural transformations $P \Rightarrow Q$ are “polymorphic operators” from P to Q

Lemma (PROVE IT) :

$$\mathsf{Nat}(P + R, Q) \cong \mathsf{Nat}(P, Q) \times \mathsf{Nat}(R, Q) \qquad \mathsf{Nat}\Big(\coprod_{i \in I} P_i,\ Q\Big) \ \cong \ \prod_{i \in I} \mathsf{Nat}(P_i,\ Q)$$

Yoneda for Powers

- By previous lemma:

$$\mathrm{Nat}(\mathrm{List}, \mathrm{List}) = \mathrm{Nat}\left(\coprod_{n\in\mathbb{N}} \mathrm{Id}^n, \mathrm{List}\right) \cong \prod_{n\in\mathbb{N}} \mathrm{Nat}(\mathrm{Id}^n, \mathrm{List})$$

- But how to compute $\mathrm{Nat}(\mathrm{Id}^n, \mathrm{List})$?
  More generally, how to compute $\mathrm{Nat}(\mathrm{Id}^n, F)$ for $F\colon$ **Set** $\to$ **Set**?
- Take $\alpha\colon \mathrm{Id}^n \Rightarrow F$. Then $\alpha_n\colon n^n \to F(n)$, so we can obtain $\alpha_n(id_n\colon n \to n) \in F(n)$
- Conversely: take $a \in F(n)$. Then $\alpha_X\colon X^n \to F(X)$, sending

$$(f\colon n \to X) \quad \mapsto \quad (F(f)\colon F(n) \to F(X))(a)$$

  is natural transformation (PROVE IT)
- These transitions form isomorhism (PROVE IT) : $\mathrm{Nat}(\mathrm{Id}^n,\ F) \ \cong\ F(n)$

Natural List Transformers

Combine results of the previous slide with $F = \mathsf{List}$:

$$\mathsf{Nat}(\mathsf{List}, \mathsf{List}) \;\cong\; \prod\nolimits_{n \in \mathbb{N}} \mathsf{Nat}(\mathsf{Id}^n, \mathsf{List}) \;\cong\; 0^0 \times 1^1 \times 2^2 \times \ldots$$

Action of $t \in \Big([\,], [1], [k_1^2, k_2^2], [k_1^3, k_2^3, k_3^3], \ldots\Big)$:

$$[x_1, \ldots, x_n] \in \mathsf{List}(X) \quad \mapsto \quad [x_{k_1}, \ldots, x_{k_n}] \in \mathsf{List}(X)$$

Examples (Revisited):

- rev: $t = ([\,], [1], [2, 1], [3, 2, 1], \ldots)$
- $(xs \mapsto [\,])$: $t = ([\,], [\,], [\,], \ldots)$
- $(case\ xs\ \ of\ \ [\,] \mapsto [\,]; [y \mid ys] \mapsto ys)$: $t = ([\,], [\,], [2], [2, 3], \ldots)$

Yoneda Theorem

Theorem: For any category $\mathcal{C}$, functor $F\colon \mathcal{C} \to \mathbf{Set}$ and object $A \in \mathrm{Ob}(\mathcal{C})$:

$$\mathrm{Nat}(\mathrm{Hom}_{\mathcal{C}}(A, -),\ F) \ \cong\ F(A)$$

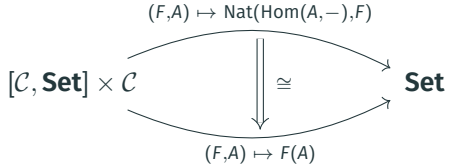
naturally in F and A (p. 49):

$$\begin{array}{ccc} \mathrm{Nat}(\mathrm{Hom}(A,-),F) & \xrightarrow{\cong} & F(A) \\ {\scriptstyle \sigma\circ(-)}\downarrow & \sigma : F \Rightarrow G & \downarrow{\scriptstyle \sigma_A} \\ \mathrm{Nat}(\mathrm{Hom}(A,-),G) & \xrightarrow[\cong]{} & G(A) \end{array} \qquad \begin{array}{ccc} \mathrm{Nat}(\mathrm{Hom}(A,-),F) & \xrightarrow{\cong} & F(A) \\ {\scriptstyle \alpha\mapsto\alpha\circ\mathrm{Hom}(f,-)}\downarrow & f : A \to B & \downarrow{\scriptstyle F(f)} \\ \mathrm{Nat}(\mathrm{Hom}(B,-),F) & \xrightarrow[\cong]{} & F(B) \end{array}$$

Under: $(\alpha\colon \mathrm{Hom}_{\mathcal{C}}(A,-) \Rightarrow F) \ \mapsto\ \alpha_A(id_A)$ $\qquad$ $\big(f : A \to B \mapsto F(f)(a)\big)_{B\in\mathrm{Ob}(\mathcal{C})} \ \mapsfrom\ a \in F(A)$

More on Naturality

The bijection $\mathrm{Nat}(\mathrm{Hom}_{\mathcal{C}}(A,-),F) \cong F(A)$ is jointly natural in $F \in [\mathcal{C},\mathbf{Set}]$ and $A \in \mathcal{C}$:



Both sides define functors $[\mathcal{C},\mathbf{Set}] \times \mathcal{C} \to \mathbf{Set}$; the Yoneda bijection is a natural isomorphism between them

Bottom evaluation functor manifests Cartesian closure of **Cat**!

Yoneda Embedding

Yoneda Embedding: $\mathsf{y}\colon \mathcal{C}^{\mathrm{op}} \to [\mathcal{C}, \mathbf{Set}]$ by $\mathsf{y}(A) = \mathrm{Hom}_{\mathcal{C}}(A, -)$

Corollary I: *y* is faithful: if $\mathsf{y}(f) = \mathsf{y}(g)$ then $f = g$
Proof: Isomorphism

$$\mathrm{Nat}(\mathrm{Hom}(A, -),\ \mathrm{Hom}(B, -)) \ \cong\ \mathrm{Hom}(B, A) \qquad (*)$$

sends $y(f)$ and $y(g)$ to $\mathrm{Hom}(f, A)(id_A) = f$ and $\mathrm{Hom}(g, A)(id_A) = g$

Corollary II: *y* is full: every $\alpha\colon \mathrm{Hom}(A, -) \Rightarrow \mathrm{Hom}(B, -)$ equals $\mathsf{y}(f)$ for some $f\colon B \to A$
Proof: Isomorphism $(*)$ sends α to $\alpha_A(id_A)$ and $y(f)$ to f where $f = \alpha_A(id_A)$

Dually (co-Yoneda): take $\mathcal{C} := \mathcal{C}^{\mathrm{op}} \quad \rightsquigarrow \quad y'\colon \mathcal{C} \to [\mathcal{C}^{\mathrm{op}}, \mathbf{Set}]$ by $\mathsf{y}'(A) = \mathrm{Hom}_{\mathcal{C}}(-, A)$

Yoneda for Posets

Let $(P, \leq)$ be poset category. Then

- $\mathrm{Hom}(-, a) = {\downarrow}a = \{b \mid b \leq a\} \colon P^{\mathrm{op}} \to \mathbf{Set}$ (principal ideals)
- $\alpha \colon \mathrm{Hom}(-, a) \Rightarrow \mathrm{Hom}(-, b)$ is wittiness of ${\downarrow}a \subseteq {\downarrow}b$

Co-Yoneda Embedding: $\mathrm{y}' \colon P \to [P^{\mathrm{op}}, \mathbf{Set}]$, $a \mapsto {\downarrow}a$, factors through downset completion:

$$P \xrightarrow{a \mapsto {\downarrow}a} [P^{\mathrm{op}}, 2] = \{D \subseteq P \mid D \text{ is down-closed}\} \hookrightarrow [P^{\mathrm{op}}, \mathbf{Set}]$$

Faithfulness and Fullness: ${\downarrow}a = {\downarrow}b \;\Rightarrow\; a = b, \quad {\downarrow}a \subseteq {\downarrow}b \iff a \leq b$

Upshot: $[P^{\mathrm{op}}, 2]$ = free completion of P to complete lattice (all joins freely added); For general categories: $[C^{\mathrm{op}}, \mathbf{Set}]$ = free completion of C under all colimits

Adjunctions & Free Objects

Rounding Up and Down

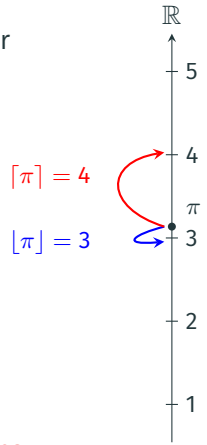
- View $\mathbb{Z}$, $\mathbb{R}$ as poset categories; inclusion $\iota\colon \mathbb{Z} \hookrightarrow \mathbb{R}$ monotone $\Rightarrow$ functor
- For $n \in \mathbb{Z}$, $r \in \mathbb{R}$:

$$r \leq \iota(n) \;\Leftrightarrow\; \lceil r \rceil \leq n \qquad\qquad \iota(n) \leq r \;\Leftrightarrow\; n \leq \lfloor r \rfloor$$

- Categorically:

$$\mathrm{Hom}(r, \iota(n)) \cong \mathrm{Hom}(\lceil r \rceil, n) \quad — \quad \lceil - \rceil \text{ is left adjoint to } \iota$$
$$\mathrm{Hom}(\iota(n), r) \cong \mathrm{Hom}(\lfloor r \rfloor, n) \quad — \quad \lfloor - \rfloor \text{ is right adjoint to } \iota$$

- Adjunctions are pairs (left adjoint, right adjoint)
- Adjunctions between order categories are also called Galois connections



Adjunctions

Functors $F\colon \mathcal{C} \to \mathcal{D}$ and $G\colon \mathcal{D} \to \mathcal{C}$ form adjunction $F \dashv G$ (F left adjoint, G right adjoint) if there is bijection

$$\varphi_{X,Y}\colon \mathrm{Hom}_{\mathcal{D}}(F(X),\, Y) \xrightarrow{\sim} \mathrm{Hom}_{\mathcal{C}}(X,\, G(Y))$$

natural in $X \in \mathcal{C}$ and $Y \in \mathcal{D}$:

$$\begin{array}{ccc} \mathrm{Hom}(F(X), Y) & \xrightarrow{\varphi_{X,Y}} & \mathrm{Hom}(X, G(Y)) \\ {\scriptstyle -\circ F(f)}\downarrow & f\colon X' \to X & \downarrow{\scriptstyle -\circ f} \\ \mathrm{Hom}(F(X'), Y) & \xrightarrow[\varphi_{X',Y}]{} & \mathrm{Hom}(X', G(Y)) \end{array} \qquad \begin{array}{ccc} \mathrm{Hom}(F(X), Y) & \xrightarrow{\varphi_{X,Y}} & \mathrm{Hom}(X, G(Y)) \\ {\scriptstyle g\circ -}\downarrow & g\colon Y \to Y' & \downarrow{\scriptstyle G(g)\circ -} \\ \mathrm{Hom}(F(X), Y') & \xrightarrow[\varphi_{X,Y'}]{} & \mathrm{Hom}(X, G(Y')) \end{array}$$

Mnemonics: F for **F**ree, G for for**G**etful

Examples of Adjunctions

- Exponentials: currying/uncurrying bijection

$$\mathrm{Hom}(Z \times X,\ Y) \cong \mathrm{Hom}(Z,\ Y^X)$$

  witnesses $\boxed{(- \times X) \dashv (-)^X}$ in any CCC

- Coproducts:

$$\mathrm{Hom}_{\mathcal{C}}(A + B,\ C) \cong \mathrm{Hom}_{\mathcal{C}}(A, C) \times \mathrm{Hom}_{\mathcal{C}}(B, C) = \mathrm{Hom}_{\mathcal{C}\times\mathcal{C}}((A, B), (C, C))$$

  witnesses $\boxed{+ \dashv \Delta}$ where $\Delta \colon \mathcal{C} \to \mathcal{C} \times \mathcal{C}$, $\Delta(C) = (C, C)$

- Lists: $\boxed{\mathrm{List} \dashv G}$ for forgetful functor $G \colon \mathbf{Mon} \to \mathbf{Set}$, because

$$\mathrm{Hom}_{\mathbf{Mon}}(\mathrm{List}(X),\ M) \cong \mathrm{Hom}_{\mathbf{Set}}(X,\ G(M))$$

List as Left Adjoint

Check $\mathrm{Hom}_{\mathbf{Mon}}(\mathrm{List}(X), M) \cong \mathrm{Hom}_{\mathbf{Set}}(X, G(M))$:

- Monoid morphism $f : \mathrm{List}(X) \to M$ (for monoid $(M, e, \cdot)$):

$$\begin{aligned} f([\,]) &= e \\ f([x_0, \ldots, x_n]) &= f([x_0]) \cdot \ldots \cdot f([x_n]) \end{aligned}$$

- Thus, f is determined by its action on $x_0, \ldots, x_n \in X$
- Conversely, if $f : X \to M$, we can extend it to entire $\mathrm{List}(X)$:

$$\begin{aligned} f^\star([\,]) &= e \\ f^\star([x_0, \ldots, x_n]) &= f(x_0) \cdot \ldots \cdot f(x_n) \end{aligned}$$

- ... and it is a monoid morphism, because

$$f^\star(xs +\!\!+ ys) = f^\star(xs) \cdot f^\star(ys)$$

Unit and Counit

Considering $F \dashv G$ with $\varphi_{X,Y} \colon \mathrm{Hom}_{\mathcal{D}}(F(X),\, Y) \xrightarrow{\sim} \mathrm{Hom}_{\mathcal{C}}(X,\, G(Y))$:

- apply $\varphi_{X,F(X)}$ to $id_{F(X)}$, get unit $\eta_X = \varphi_{X,F(X)}(id_{F(X)}) \colon X \to GF(X)$
- dually, apply $\varphi^{-1}_{G(Y),Y}$ to $id_{G(Y)}$ to get counit
- Unit and counit are both natural transformations (PROVE IT)

Definition: Equivalence of categories is such adjunction $F \dashv G$, whose unit and counit are natural isomorphisms

In Summary:

$$\text{Isomorphism} \;<\; \text{Equivalence} \;<\; \text{Adjunction}$$

Units and Counits: Examples

- Exponentials: $(- \times X) \dashv (-)^X$ in a CCC,
  - Unit $\eta_Z : Z \to (Z \times X)^X$, $\eta = curry(id)$
  - Counit $\varepsilon_Y : Y^X \times X \to Y$, $\varepsilon = ev$
- Lists: $\mathrm{List} \dashv G$,
  - Unit $\eta_X : X \to G(\mathrm{List}(X))$, $\eta(x) = [x]$
  - Counit $\varepsilon_M : \mathrm{List}(G(M)) \to M$, $\varepsilon([m_0, \ldots, m_n]) = m_0 \cdot \ldots \cdot m_n$
- Coproducts: $+ \dashv \Delta$, where $\Delta(X) = (X, X)$,
  - Unit $\eta_{(X,Y)} : (X, Y) \to (X + Y, X + Y)$ $\eta_{(X,Y)} = (inl, inr)$
  - Counit $\varepsilon_X : X + X \to X$, $\varepsilon = [id, id]$

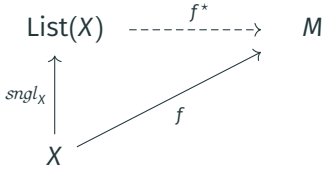
Adjunctions & Free Objects

Free Objects

Lists as Free Monoids

Fix set X

- **Object**: List(X) in **Mon**
- **Arrows**: Unit map $\mathit{sngl}_X : X \to \text{List}(X)$ in **Set**, $\quad x \mapsto [x]$
- **Universal property**: For any monoid M and $f : X \to M$ in **Set**, there exists a unique monoid morphism $f^\star :\ \text{List}(X) \to M$ such that:

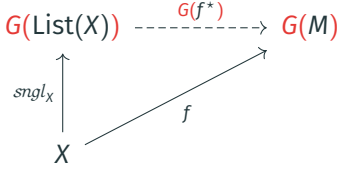


Note that f is from **Set**, but $f^\star$ is from **Mon**; $G :$ **Mon** $\to$ **Set** is the forgetful functor

Lists as Free Monoids

Fix set X

- **Object**: $\mathrm{List}(X)$ in **Mon**
- **Arrows**: Unit map $\mathit{sngl}_X : X \to \mathrm{List}(X)$ in **Set**, $\quad x \mapsto [x]$
- **Universal property**: For any monoid M and $f : X \to M$ in **Set**, there exists a unique monoid morphism $f^\star :\ \mathrm{List}(X) \to M$ such that:

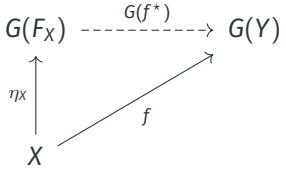


Note that f is from **Set**, but $f^\star$ is from **Mon**; $G :$ **Mon** $\to$ **Set** is the forgetful functor

Free Objects

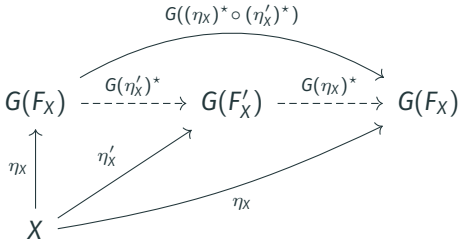
Fix $G\colon \mathcal{D} \to \mathcal{C}$ and $X \in \mathrm{Ob}(\mathcal{C})$

- **Object**: F_X in $\mathcal{D}$
- **Arrows**: morphism $\eta_X : X \to G(F_X)$ in $\mathcal{C}$
- **Universal property**: for any $Y \in \mathcal{D}$ and $f : X \to G(Y)$ in $\mathcal{C}$, there exists unique extension $f^\star : F_X \to Y$ in $\mathcal{D}$ such that:



Free Objects are Up-to-Isomorphism

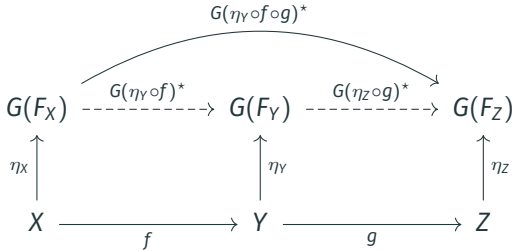
If (F_X, η_X) and (F'_X, η'_X) are both free objects on $X \in \mathcal{C}$, apply universal properties:

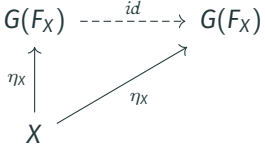


Both $(\eta_X)^\star \circ (\eta'_X)^\star$ and id_{F_X} extend η_X to $G(F_X)$, so by uniqueness: $(\eta_X)^\star \circ (\eta'_X)^\star = id_{F_X}$; other direction analogous, hence $F_X \cong F'_X$

Free Object Functor

If all free objects F_X exist then F extends to functor and η to natural transformation:

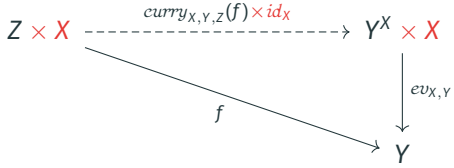




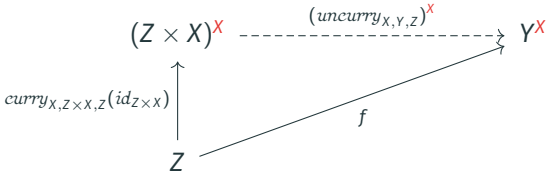
So, we take $F(X) = F_X$ and $F(f) = (\eta_Y \circ f)^\star$

Co-Free Objects

- Recall our definition of exponentials:



- These can be viewed as co-free objects w.r.t. $F = (-) \times X$
- Dually, $FZ = Z \times X$ is free object w.r.t. $G = (-)^X$ (PROVE IT) :



Adjunctions via Free Objects

- Recall that we obtained $uncurry = curry^{-1}$ and adjunction

$$\mathrm{Hom}(Z \times X, Y) \cong \mathrm{Hom}(Z, Y^X) :$$

  from $(Y^X, ev_{X,Y})$ as co-free object
- It is a general principle!

Proposition: Given $G \colon \mathcal{D} \to \mathcal{C}$, $F \dashv G$ exists iff all free objects $F(X)$ w.r.t. G exist

- **Adjunction** $\Rightarrow$ **Free objects**: given $\varphi \colon \mathrm{Hom}(F(X), Y) \cong \mathrm{Hom}(X, G(Y))$, set $\eta_X = \varphi(id_{F(X)})$; for $f : X \to G(Y)$ define $f^\star = \varphi^{-1}(f) \colon F(X) \to Y$
- **Free objects** $\Rightarrow$ **Adjunction**: given $(F(X), \eta_X)$, define

$$\varphi(h) = G(h) \circ \eta_X \qquad \varphi^{-1}(f) = \text{unique } f^\star \text{ with } G(f^\star) \circ \eta_X = f$$

Corollary: Left adjoints are unique up to isomorphism (because so are free objects)

Adjunctions: Summary

We can define adjunctions

1. through natural isomorphism of homs
2. through free objects
3. through co-free objects
4. through units and counits (triangle identities omitted)

Note: 2. and 3. allow us to deal with particular (co-)free object even when adjunction need not exist globally!

Conclusions

Topics covered (or so I hope):

- Sets, functions, naturals, (co-)products in set theory
- Categories, functors, isomorphisms, products, coproducts
- Natural transformations, Yoneda lemma, CCCs, exponentials
- Adjunctions, free objects, units, counits

Further topics:

- Monads, algebras and coalgebras
- Limits and colimits
- Topos theory
- Enriched categories, monoidal categories, 2-categories and higher categories thereof
- ...

Thank You for Your Attention!