What is category theory?

What we are probably seeking is a “purer” view of functions: a theory of functions in themselves, not a theory of functions derived from sets. What, then, is a pure theory of functions? Answer: category theory.

Dana Scott, *Relating theories of the λ-calculus*, p406

set theory gives an “element-oriented” account of mathematical structure, whereas

category theory takes a ‘function-oriented” view – understand structures not via their elements, but by how they transform, i.e. via morphisms.

(Both theories are part of logic, broadly construed.)

Category Theory emerges

1945 Eilenberg† and MacLane†
General Theory of Natural Equivalences,
Trans AMS 58, 231–294

(algebraic topology, abstract algebra)

1950s Grothendieck† (algebraic geometry)

1960s Lawvere† (logic and foundations)

1970s Johnstone, Joyal and Tierney†

(elementary topos theory)

1980s Dana Scott, Plotkin

(semantics of programming languages)

Lambek† (linguistics)

Category Theory and Computer Science

"Category theory has…become part of the standard "tool-box" in many areas of theoretical informatics, from programming languages to automata, from process calculi to Type Theory."

Dagstuhl Perpectives Workshop on *Categorical Methods at the Crossroads*
April 2014

See `http://www.appliedcategorytheory.org/events` for recent examples of category theory being applied (not just in computer science).

This course

basic concepts of category theory

adjunction ← **natural transformation**

↑

category → **functor**

applied to:

- **typed lambda-calculus**
- **functional programming**

Sets

- Examples

  Empty set: $0 = \emptyset = \{\ \}$

  Singleton sets: $1 = \{0\}$, $\{*\}$

  Natural numbers: $\mathbb{N}$

- Products

  The **cartesian product** of sets X and Y is the set of all *ordered pairings* (x, y) for $x \in X$ and $y \in Y$:

  $$X \times Y = \{(x, y) \mid x \in X \wedge y \in Y\}$$

  The equality for ordered pairs is pointwise:

  $$(x, y) = (x', y') \Leftrightarrow x = x' \wedge y = y'$$

The cartesian product comes equipped with first and second projection operations π_1 and π_2 satifying:

1. for all $x \in X$ and $y \in Y$,

$$\pi_1(x, y) = x \quad , \quad \pi_2(x, y) = y$$

2. for all $p \in X \times Y$,

$$(\pi_1\, p\, ,\ \pi_2\, p) = p$$

Example:

For $n \in \mathbb{N}$,

$$X^n = \begin{cases} \{\,()\,\} & \text{, if } n = 0 \\ X \times X^m & \text{, if } n = m + 1 \end{cases}$$

and, for $x_1, x_2, \ldots, x_{n-1}, x_n \in \mathbb{N}$, one writes $(x_1, \ldots, x_n)$ for $(x_1, (x_2, \ldots (x_{n-1}, x_n) \ldots)) \in X^n$.

- Functions

  The **set of functions** from a set X to a set Y, for which we write $(X \Rightarrow Y)$ or Y^X, consists of all the single-valued and total relations from X to Y:

  $$(X \Rightarrow Y) = \{f \subseteq X \times Y \mid f \text{ is single-valued and total}\}$$

  Single-valued:

  $$\forall x \in X, \forall y, y' \in Y, (x, y) \in f \wedge (x, y') \in f \Rightarrow y = y'$$

  Total:

  $$\forall x \in X, \exists y \in Y, (x, y) \in f$$

Notation: We write $f : X \to Y$ or $X \xrightarrow{f} Y$ for $f \in (X \Rightarrow Y)$ and, for $x \in X$, we write $f\,x$ or $f(x)$ for the unique element y of Y such that $(x, y) \in f$.

The equality for functions is extensional:

$$f = g : X \to Y \;\Leftrightarrow\; \forall x \in X,\; f\,x = g\,x$$

This is because

1. Assuming $f = g$, we have, for all $x \in X$, $(x, f\,x) \in g$ and so $f\,x = g\,x$.
2. Assuming $\forall x \in X,\; f\,x = g\,x$, we have

$$\begin{aligned} f &= \{(x,y) \mid (x,y) \in f\} = \{(x, f\,x) \mid x \in X\} = \{(x, g\,x) \mid x \in X\} \\ &= \{(x,y) \mid (x,y) \in g\} = g \end{aligned}$$

In other words, function extensionality reduces to the extensionality property of sets: two sets are equal iff they have the same elements.

Convention: We typically define functions $f : X \longrightarrow Y$ by a *well-defined rule* that to each element $x \in X$ assigns a unique element $f(x) \in Y$.

Examples:

1. We define $\mathtt{id}_X : X \longrightarrow X$ by:
$$\mathtt{id}_X(x) = x \text{ for all } x \in X$$
2. For $f : X \longrightarrow Y$ and $g : Y \longrightarrow Z$, we define $g \circ f : X \longrightarrow Z$ by:
$$(g \circ f)(x) = g(f(x)) \text{ for all } x \in X$$
3. We define $\mathtt{app} : (X \Rightarrow Y) \times X \longrightarrow Y$ by:
$$\mathtt{app}(f, x) = f(x) \text{ for all } f \in (X \Rightarrow Y) \text{ and } x \in X$$
4. For $f : Z \times X \longrightarrow Y$, we define $\mathtt{cur}(f) : Z \longrightarrow (X \Rightarrow Y)$ by:
$$(\mathtt{cur}(f)\, z)(x) = f(z, x) \text{ for all } z \in Z \text{ and } x \in X$$

- Sums

  The **sum** of sets X and Y is their *disjoint union*:

$$X + Y = \{\iota_1(x) \mid x \in X\} \cup \{\iota_2(y) \mid y \in Y\}$$

The sum comes equipped with first and second tagging operations $\iota_1 : X \longrightarrow X + Y$ and $\iota_2 : Y \longrightarrow X + Y$.

The equality for *tagged elements* is:

$$\iota_i(x) = \iota_j(y) \iff (i = j) \wedge (x = y)$$

Sets in Grothendieck universes

A **Grothendieck universe** $\mathscr{U}$ is a set of sets satisfying

- $X \in Y \in \mathscr{U} \Rightarrow X \in \mathscr{U}$
- $X, Y \in \mathscr{U} \Rightarrow \{X, Y\} \in \mathscr{U}$
- $X \in \mathscr{U} \Rightarrow \mathscr{P} X \triangleq \{Y \mid Y \subseteq X\} \in \mathscr{U}$
- $X \in \mathscr{U} \ \wedge \ F : X \longrightarrow \mathscr{U} \Longrightarrow \{y \mid \exists x \in X,\ y \in F\, x\} \in \mathscr{U}$
  (hence also $X, Y \in \mathscr{U} \ \Rightarrow \ X \times Y \in \mathscr{U} \ \wedge \ (X \Rightarrow Y) \in \mathscr{U}$)

The above properties are satisfied by $\mathscr{U} = \emptyset$, but we will always assume

- $\mathbb{N} \in \mathscr{U}$

Monoids

A **monoid** is a structure $\underline{M} = (M, \bullet, \imath)$ consisting of a set M equipped with a binary operation $\_ \bullet \_ : M \times M \to M$ and an element $\imath \in M$ that satisfy:

- the associativity law:
  $\forall x, y, z \in M,\ (x \bullet y) \bullet z = x \bullet (y \bullet z)$
- the unit laws:
  $\forall x \in M,\ \imath \bullet x = x = x \bullet \imath$

Examples:

1. Lists $(\texttt{List}\, X, @, \texttt{nil})$:

$$
\begin{aligned}
\texttt{List}\, X &= \text{set of finite lists of elements of } X \\
@ &= \text{append} \\
& \quad \left(\begin{aligned} \texttt{nil} \mathbin{@} \ell &= \ell \\ (x :: \ell) \mathbin{@} \ell' &= x :: (\ell \mathbin{@} \ell') \end{aligned} \right) \\
\texttt{nil} &= \text{empty list}
\end{aligned}
$$

2. Sequences $(X^\star, \cdot, \varepsilon)$:

$$
\begin{aligned}
X^\star &= \textstyle\bigcup_{n \in \mathbb{N}} X^n \\
\cdot &= \text{concatenation} \\
& \quad \left((x_1, \ldots, x_m) \cdot (y_1, \ldots, y_n) = (x_1, \ldots, x_m, y_1, \ldots, y_n) \right) \\
\varepsilon &= (\,)
\end{aligned}
$$

3. The set of **endomorphisms** on a set:

$$\mathrm{End}(X) = (\, X \Rightarrow X\,, \circ, \mathtt{id}_X\,)$$

is a monoid.

In particular, the monoid $\mathrm{End}(1)$ is trivial.

A **monoid homomorphism** $h : \underline{M}_1 \to \underline{M}_2$ from a monoid $\underline{M}_1 = (M_1, \bullet_1, \iota_1)$ to a monoid $\underline{M}_2 = (M_2, \bullet_2, \iota_2)$ is a function $h : M_1 \to M_2$ such that

- for all $x, y \in M_1$, $h(x \bullet_1 y) = h(x) \bullet_2 h(y)$
- $h(\iota_1) = \iota_2$

Example: For all $f : X \to Y$,

$$\mathtt{map}\, f : \mathtt{List}\, X \to \mathtt{List}\, Y$$

is a monoid homomorphism. (Check it.)

The **product** $\underline{M}_1 \times \underline{M}_2$ of monoids $\underline{M}_1 = (M_1, \bullet_1, \imath_1)$ and $\underline{M}_2 = (M_2, \bullet_2, \imath_2)$ is the structure

$$(M_1 \times M_2 \, , \; \bullet \, , \; \imath \,)$$

with

$$(x_1, x_2) \bullet (y_1, y_2) = (x_1 \bullet_1 y_1, x_2 \bullet_2 y_2)$$

$$\imath = (\imath_1, \imath_2)$$

and projections homomorphisms

$$\underline{M}_1 \xleftarrow{\pi_1} \underline{M}_1 \times \underline{M}_2 \xrightarrow{\pi_2} \underline{M}_2$$

given by $\pi_1(x_1, x_2) = x_1$ and $\pi_2(x_1, x_2) = x_2$.

Explore

? Is there a monoid of homomorphisms between monoids?

? What is the sum $\underline{M}_1 +_{\mathbf{Mon}} \underline{M}_2$ of monoids $\underline{M}_1$ and $\underline{M}_2$?

Can you make sense of the following?

$$\mathtt{List}(X) +_{\mathbf{Mon}} \mathtt{List}(Y) = \mathtt{List}(X +_{\mathbf{Set}} Y)$$

Groups

A **group** is a structure $\underline{G} = (G, \overline{}, \bullet, \imath)$ consisting of a monoid $(G, \bullet, \imath)$ and a unary operation $\overline{\_} : M \to M$ that satisfies:

- the inverse laws:

$$\forall x \in G,\ x \bullet \overline{x} = \imath = \overline{x} \bullet x$$

A **group homomorphism** $h : \underline{G}_1 \to \underline{G}_2$ from a group $\underline{G}_1 = (G_1, \overline{}^1, \bullet_1, \imath_1)$ to a group $\underline{G}_2 = (G_2, \overline{}^2, \bullet_2, \imath_2)$ is a monoid homomorphism $h : (G_1, \bullet_1, \imath_1) \to (G_2, \bullet_2, \imath_2)$ such that

- for all $x \in G_1$, $h(\overline{x}^1) = \overline{h(x)}^2$

Examples:

- The set of integers modulo a prime p

$$\mathbb{Z}_p$$

has a group structure.

- The set of **automorphisms** on a set

$$\mathrm{Aut}(X) = \{\, f : X \to X \mid f \text{ is a bijection} \,\}$$

has a group structure.

Definitions:

- A function $f : X \to Y$ is a **bijection** whenever
$$\forall y \in Y,\ \exists! x \in X,\ f(x) = y$$
- A function $f : X \to Y$ is an **isomorphism** whenever there exists a (necessarily unique) function $g : Y \to X$ (typically denoted f^{-1}) such that $g \circ f = \mathtt{id}_X$ and $f \circ g = \mathtt{id}_Y$.

Proposition. A function is a *bijection* if, and only if, it is an *isomorphism*.

Explore the above for homomorphisms between monoids and between groups.

Universal problems

- **Vague problem**: To manufacture a monoid out of a set in the most general or least constrained possible way.

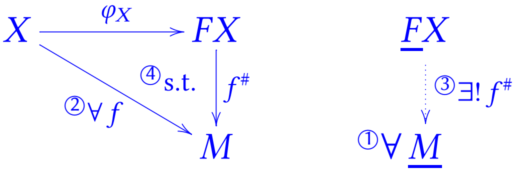
  This is typically referred to as *freely generating* a monoid from a set.

- **Mathematical problem**: For a set X, consider the *data* of interest to be given by a monoid $\underline{M} = (M, \bullet, \iota)$ together with a function $f : X \to M$. Given a set X,
   1. construct data $\underline{F}X = (FX, \bullet_X, \iota_X)$ and $\varphi_X : X \to FX$

  such that

   1. for all data $\underline{M} = (M, \bullet, \iota)$ and $f : X \to M$, there exists a unique monoid homomorphism $f^\# : FX \to \underline{M}$ such that $f^\# \circ \varphi_X = f$.

In diagrammatic form:



? Is it a well-posed problem?

? Does it have a solution?

? If so, how can we construct it?

Categories

A category $\mathbf{C}$ is an algebraic structure specified by

- a set $\boxed{\mathtt{obj}\,\mathbf{C}}$ whose elements are called $\mathbf{C}$-objects
- for each $X, Y \in \mathtt{obj}\,\mathbf{C}$, a set $\boxed{\mathbf{C}(X, Y)}$ whose elements are called $\mathbf{C}$-morphisms from X to Y
- a function assigning to each $X \in \mathtt{obj}\,\mathbf{C}$ an element $\boxed{\mathtt{id}_X \in \mathbf{C}(X, X)}$ called the identity morphism for the $\mathbf{C}$-object X
- a function assigning to each $f \in \mathbf{C}(X, Y)$ and $g \in \mathbf{C}(Y, Z)$ (where $X, Y, Z \in \mathtt{obj}\,\mathbf{C}$) an element $\boxed{g \circ f \in \mathbf{C}(X, Z)}$ called the composition of $\mathbf{C}$-morphisms f and g

satisfying

- associativity: for all $X, Y, Z, W \in \mathtt{obj}\,\mathbf{C}$, $f \in \mathbf{C}(X, Y)$, $g \in \mathbf{C}(Y, Z)$ and $h \in \mathbf{C}(Z, W)$

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

- unity: for all $X, Y \in \mathtt{obj}\,\mathbf{C}$ and $f \in \mathbf{C}(X, Y)$

$$\mathtt{id}_Y \circ f = f = f \circ \mathtt{id}_X$$

Category of sets: **Set**

- $\mathtt{obj}\,\mathbf{Set}$ = a fixed universe of sets
- **Set**-morphisms are functions:
  $$\mathbf{Set}(X, Y) = (X \Rightarrow Y)$$
- Identities:
  $$\mathtt{id}_X$$
- Composition of $f \in \mathbf{Set}(X, Y)$ and $g \in \mathbf{Set}(Y, Z)$ is:
  $$g \circ f$$

NB: Associativity and unity laws hold. (Check it.)

Category of monoids: **Mon**

- objects are monoids,
- morphisms are monoid homomorphisms,
- identities and composition are as for sets and functions.

Q: why is this well-defined?
A: because the set of functions that are monoid homomorphisms contains identity functions and is closed under composition.

Category of groups: **Grp**

- objects are groups,
- morphisms are group homomorphisms,
- identities and composition are as for sets and functions.

Q: why is this well-defined?
A: because the set of functions that are group homomorphisms contains identity functions and is closed under composition.

Conventions:

- Given a category $\mathbf{C}$, one writes

$$f : X \to Y \quad \text{or} \quad X \xrightarrow{f} Y \quad \text{or} \quad Y \xleftarrow{f} X$$

for

$$f \in \mathbf{C}(X, Y)$$

in which case one says

object X is the domain of the morphism f

object Y is the codomain of the morphism f

and writes

$$X = \mathtt{dom}\, f \ , \quad Y = \mathtt{cod}\, f$$

NB: Which category $\mathbf{C}$ we are referring to is left implicit with this notation.

- The sets $\mathbf{C}(X, Y)$ are typically referred to as hom-sets and sometimes also denoted $\hom_{\mathbf{C}}(X, Y)$ or simply $\hom(X, Y)$ when $\mathbf{C}$ is clear from the context.
- One often abbreviates $g \circ f$ as $g\,f$.
- Because of the associativity law, one unambigously writes

$$h \circ g \circ f \quad \text{or} \quad h\,g\,f$$

  for either of the equal composites $h \circ (g \circ f) = h\,(g\,f)$ and $(h \circ g) \circ f = (h\,g)\,f$.

Alternative notations

Some people write

- $\mathbf{C}$ for $\mathtt{obj}\,\mathbf{C}$
- $\mathtt{id}$ for $\mathtt{id}_X$
- $\mathtt{Hom}_{\mathbf{C}}(X, Y)$ for $\mathbf{C}(X, Y)$
- X or 1_X for $\mathtt{id}_X$

Most people use "applicative order" for morphism composition; some people use "diagrammatic order" and write

- $f ; g$ for $g \circ f$