

# Category Theory

Andrew Pitts

University of Cambridge  
Computer Science Tripos  
Part II Unit of Assessment  
Part III and MPhil. ACS Module L108  
Michaelmas Term 2021

# Course web page

Go to

<https://www.cl.cam.ac.uk/teaching/2122/CatTheory/>

<https://www.cl.cam.ac.uk/teaching/2122/L108/>

for

- ▶ these slides and lecture recordings
  - ▶ exercise sheets and details of examples classes  
(trying the exercises is essential!)
  - ▶ pointers to some additional material
- + office hours  
(see Moodle)

Recommended text for the course is:

[Awodey] Steve Awodey, *Category theory*,  
Oxford University Press (2nd ed.), 2010.

# Assessment

- ▶ **A graded exercise sheet** (25% of the final mark).  
issued in lecture 10 with a one week deadline
- ▶ **A take-home test** (75% of the final mark).  
issued after the end of the course

There will be two one-hour example class sessions to provide help with the exercises.

**See course web page for dates and deadlines.**

# Lecture 1

# 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  $\lambda$ -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.)

# GENERAL THEORY OF NATURAL EQUIVALENCES

BY

SAMUEL EILENBERG AND SAUNDERS MACLANE

## CONTENTS

	Page
Introduction . . . . .	231
I. Categories and functors . . . . .	237
1. Definition of categories . . . . .	237
2. Examples of categories . . . . .	239
3. Functors in two arguments . . . . .	241
4. Examples of functors . . . . .	242
5. Slicing of functors . . . . .	245
6. Foundations . . . . .	246
II. Natural equivalence of functors . . . . .	248
7. Transformations of functors . . . . .	248
8. Categories of functors . . . . .	250
9. Composition of functors . . . . .	250
10. Examples of transformations . . . . .	251
11. Groups as categories . . . . .	256
12. Construction of functors by transformations . . . . .	257
13. Combination of the arguments of functors . . . . .	258
III. Functors and groups . . . . .	260
14. Subfunctors . . . . .	260
15. Quotient functors . . . . .	262
16. Examples of subfunctors . . . . .	263
17. The isomorphism theorems . . . . .	265
18. Direct products of functors . . . . .	267
19. Characters . . . . .	270
IV. Partially ordered sets and projective limits . . . . .	272
20. Quasi-ordered sets . . . . .	272
21. Direct systems as functors . . . . .	273
22. Inverse systems as functors . . . . .	276
23. The categories $\mathfrak{Dir}$ and $\mathfrak{Inv}$ . . . . .	277
24. The lifting principle . . . . .	280
25. Functors which commute with limits . . . . .	281
V. Applications to topology . . . . .	283
26. Complexes . . . . .	283
27. Homology and cohomology groups . . . . .	284
28. Duality . . . . .	287
29. Universal coefficient theorems . . . . .	288
30. Čech homology groups . . . . .	290
31. Miscellaneous remarks . . . . .	292
Appendix. Representations of categories . . . . .	292

**Introduction.** The subject matter of this paper is best explained by an example, such as that of the relation between a vector space  $L$  and its “dual”

Presented to the Society, September 8, 1942; received by the editors May 15, 1945.

# Category Theory emerges

1945 Eilenberg<sup>†</sup> and MacLane<sup>†</sup>  
*General Theory of Natural Equivalences*,  
Trans AMS 58, 231–294

(algebraic topology, abstract algebra)

1950s Grothendieck<sup>†</sup> (algebraic geometry)

1960s Lawvere (logic and foundations)

1970s Joyal and Tierney<sup>†</sup> (elementary topos theory)

1980s Dana Scott, Plotkin

(semantics of programming languages)

Lambek<sup>†</sup> (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 Perspectives 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**

Lent Term ACS module on  
Advanced Topics in Category Theory

# Definition

A **category**  $\mathbf{C}$  is specified by

- ▶ a set  $\text{obj } \mathbf{C}$  whose elements are called  **$\mathbf{C}$ -objects**
- ▶ for each  $X, Y \in \text{obj } \mathbf{C}$ , a set  $\mathbf{C}(X, Y)$  whose elements are called  **$\mathbf{C}$ -morphisms from  $X$  to  $Y$**

(so far, that is just what some people call a **directed graph**)

# Definition

A **category**  $\mathbf{C}$  is specified by

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

# Definition, continued

satisfying...

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

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

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

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

# Example: category of sets, **Set**

- ▶ **obj Set** = some fixed universe of sets

(more on universes later)

- ▶ **Set**( $X, Y$ ) =

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

**Cartesian product** of sets  $X$  and  $Y$  is the set of all ordered pairs  $(x, y)$  with  $x \in X$  and  $y \in Y$ .

Equality of ordered pairs:

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

# Example: category of sets, **Set**

- ▶ **obj Set** = some fixed universe of sets

(more on universes later)

- ▶ **Set**( $X, Y$ ) =

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

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

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

# Example: category of sets, **Set**

- ▶ **obj Set** = some fixed universe of sets

(more on universes later)

- ▶ **Set**( $X, Y$ ) =

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

- ▶  $\text{id}_X = \{(x, x) \mid x \in X\}$

- ▶ composition of  $f \in \text{Set}(X, Y)$  and  $g \in \text{Set}(Y, Z)$  is

$$g \circ f = \{(x, z) \mid \exists y \in Y, (x, y) \in f \wedge (y, z) \in g\}$$

(check that associativity and unity properties hold)

# Example: category of sets, **Set**

**Notation.** Given  $f \in \mathbf{Set}(X, Y)$  and  $x \in X$ , it is usual to write  $f x$  (or  $f(x)$ ) for the unique  $y \in Y$  with  $(x, y) \in f$ .

Thus

$$\text{id}_X x = x$$

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



# Domain and codomain

Given a category  $\mathbf{C}$ ,

write  $f : X \rightarrow Y$  or  $X \xrightarrow{f} Y$

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

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

$$X = \text{dom } f \quad Y = \text{cod } f$$

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

# Commutative diagrams

in a category  $\mathbf{C}$ :

a **diagram** is

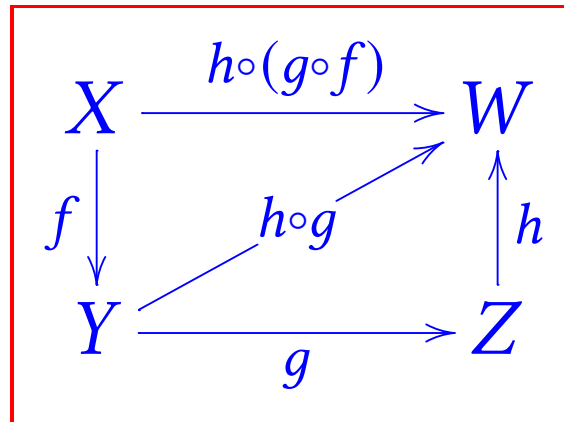
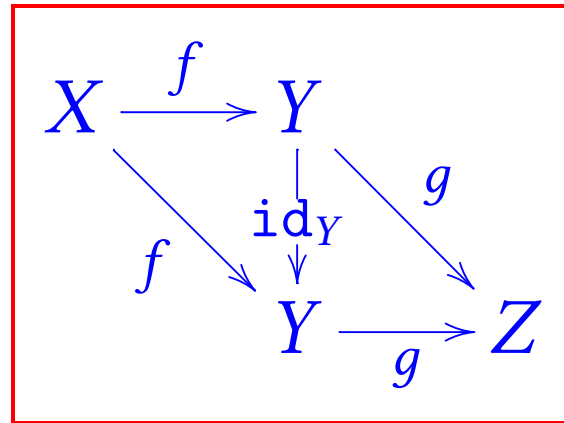
a directed graph whose vertices are  $\mathbf{C}$ -objects  
and whose edges are  $\mathbf{C}$ -morphisms

and the diagram is **commutative** (or **commutes**) if  
any two finite paths in the graph between any  
two vertices determine equal morphisms in the  
category under composition

(N.B.  $\exists$  another kind of diagram, Euler dual  
of this kind)

# Commutative diagrams

Examples:



# Alternative notations

I will often just write

$\mathbf{C}$  for  $\text{obj } \mathbf{C}$

$\text{id}$  for  $\text{id}_X$

Some people write

$\text{Hom}_{\mathbf{C}}(X, Y)$  for  $\mathbf{C}(X, Y)$

$1_X$  for  $\text{id}_X$

$gf$  for  $g \circ f$

I use “applicative order” for morphism composition;  
other people use “diagrammatic order” and write

$f;g$  (or  $fg$ ) for  $g \circ f$

# Alternative definition of category

The definition given here is “dependent-type friendly”.

See [Awodey, Definition 1.1] for an equivalent formulation:

One gives the whole set of morphisms  $\text{mor } \mathbf{C}$   
(in bijection with  $\sum_{X, Y \in \text{obj } \mathbf{C}} \mathbf{C}(X, Y)$  in my definition)  
plus functions

$$\text{dom}, \text{cod} : \text{mor } \mathbf{C} \rightarrow \text{obj } \mathbf{C}$$

$$\text{id} : \text{obj } \mathbf{C} \rightarrow \text{mor } \mathbf{C}$$

and a *partial* function for composition

$$\_ \circ \_ : \text{mor } \mathbf{C} \times \text{mor } \mathbf{C} \rightarrow \text{mor } \mathbf{C}$$

defined at  $(f, g)$  iff  $\text{cod } f = \text{dom } g$

and satisfying the associativity and unity equations.