SAMSON ABRAMSKY, «Logic and categories as a tool for building theories. Second edition, 25/01/12.» <https://arxiv.org/abs/1201.5342v1> [accessed 31 August 2018]. First Edition in: Journal of Indian Council of Philosophical Research. Issue on Logic and Philosophy Today, 27 (2010), 277-304.

Logic and Categories As Tools For Building Theories

Samson Abramsky
Oxford University Computing Laboratory

1 Introduction

My aim in this short article is to provide an impression of some of the ideas emerging at the interface of logic and computer science, in a form which I hope will be accessible to philosophers.

Why is this even a good idea? Because there has been a huge interaction of logic and computer science over the past half-century which has not only played an important rôle in shaping Computer Science, but has also greatly broadened the scope and enriched the content of logic itself.

This huge effect of Computer Science on Logic over the past five decades has several aspects: new ways of *using* logic, new attitudes to logic, new questions and methods. These lead to new perspectives on the question:

What logic is — and should be!

Our main concern is with method and attitude rather than matter; nevertheless, we shall base the general points we wish to make on a case study: *Category theory*. Many other examples could have been used to illustrate our theme, but this will serve to illustrate some of the points we wish to make.

2 Category Theory

Category theory is a vast subject. It has enormous potential for any serious version of 'formal philosophy' — and yet this has hardly been realized.

We shall begin with introduction to some basic elements of category theory, focussing on the fascinating conceptual issues which arise even at the most elementary level of the subject, and then discuss some its consequences and philosophical ramifications.

2.1 Some Basic Notions of Category Theory

We briefly recall the basic definitions. A category has a collections of *objects* A, B, C, ..., and a collection of *arrows* (or *morphisms*) f, g h .... Each arrow has specified objects as its *domain* and *codomain*.[1] We write $f : A \to B$ for an arrow with domain A and codomain B. For any triple of objects A, B, C there is an operation of *composition*: given $f : A \to B$ and $g : B \to C$, we can form $g \circ f : A \to C$. Note that the codomain of f has to match with the domain of g. Moreover, for each object A, there is an *identity arrow* $\mathsf{id}_A : A \to A$. These data are subject to the following axioms:

$$h \circ (g \circ f) = (h \circ g) \circ f \qquad f \circ \mathsf{id}_A = f = \mathsf{id}_B \circ f$$

whenever the indicated compositions make sense, *i.e.* the domains and codomains match appropriately.

These definitions appear at first sight fairly innocuous: some kind of algebraic structure, reminiscent of monoids (groups without inverses), but with the clumsy-looking apparatus of objects,

[1] More formally, there are operations dom, cod from arrows to objects.

domains and codomains restricting the possibilities for composition of arrows. These first appearances are deceptive, as we shall see, although in a few pages we can only convey a glimpse of the richness of the notions which arise as the theory unfolds.

Let us now see some first examples of categories.

- The most basic example of a category is **Set**: the objects are sets, and the arrows are functions. Composition and identities have their usual meaning for functions.
- Any kind of mathematical structure, together with structure preserving functions, forms a category. E.g.
  - **Mon** (monoids and monoid homomorphisms)
  - **Grp** (groups and group homomorphisms)
  - $\mathbf{Vect}_k$ (vector spaces over a field k, and linear maps)
  - **Pos** (partially ordered sets and monotone functions)
  - **Top** (topological spaces and continuous functions)
- **Rel**: objects are sets, arrows $R : X \to Y$ are *relations* $R \subseteq X \times Y$. Relational composition:

$$R;S(x,z) \iff \exists y.\, R(x,y) \,\wedge\, S(y,z)$$

- Let k be a field (for example, the real or complex numbers). Consider the following category $\mathbf{Mat}_k$. The objects are natural numbers. A morphism $M : \mathbf{n} \to \mathbf{m}$ is an $\mathbf{n} \times \mathbf{m}$ matrix with entries in k. Composition is matrix multiplication, and the identity on $\mathbf{n}$ is the $\mathbf{n} \times \mathbf{n}$ diagonal matrix.
- Monoids are one-object categories. Arrows correspond to the elements of the monoid, composition of arrows to the monoid multiplication, and the identity arrow to the monoid unit.
- A category in which for each pair of objects A, B there is at most one morphism from A to B is the same thing as a *preorder*, *i.e.* a reflexive and transitive relation. Note that the identity arrows correspond to reflexivity, and composition to transitivity.

2.1.1 Categories as Contexts and as Structures

Note that our first class of examples illustrate the idea of categories as *mathematical contexts*; settings in which various mathematical theories can be developed. Thus for example, **Top** is the context for general topology, **Grp** is the context for group theory, etc.

This issue of "mathematics in context" should be emphasized. The idea that any mathematical discussion is relative to the category we happen to be working in is pervasive and fundamental. It allows us simultaneously to be both properly specific and general: specific, in that statements about mathematical structures are not really precise until we have specified which structures we are dealing with, *and* which morphisms we are considering — *i.e.* which category we are working in. At the same time, the awareness that we are working in some category allows us to extract the proper generality for any definition or theorem, by identifying exactly which properties of the ambient category we are using.

On the other hand, the last two examples illustrate that many important mathematical structures *themselves appear as categories of particular kinds.* The fact that two such different kinds of structures as monoids and posets should appear as extremal versions of categories is also rather striking.

This ability to capture mathematics both "in the large" and "in the small" is a first indication of the flexibility and power of categories.

2.1.2 Arrows vs. Elements

Notice that the axioms for cateories are formulated purely in terms of the algebraic operations on arrows, without any reference to 'elements' of the objects. Indeed, in general elements are not available in a category. We will refer to any concept which can be defined purely in terms of composition and identities as *arrow-theoretic*. We will now take a first step towards learning to "think with arrows" by seeing how we can replace some familiar definitions for functions between sets couched in terms of elements by arrow-theoretic equivalents.

We say that a function $f : X \longrightarrow Y$ is:

$$\begin{array}{lll} \textit{injective} & \text{if} & \forall x, x' \in X.\, f(x) = f(x') \implies x = x' \,, \\ \textit{surjective} & \text{if} & \forall y \in Y.\, \exists x \in X.\, f(x) = y \,, \\ & & \\ \textit{monic} & \text{if} & \forall g, h : Z \to X.\, f \circ g = f \circ h \implies g = h \,, \\ \textit{epic} & \text{if} & \forall g, h : Y \to Z.\, g \circ f = h \circ f \implies g = h \,. \end{array}$$

Note that injectivity and surjectivity are formulated in terms of elements, while epic and monic are arrow-theoretic.

Proposition 1 *Let $f : X \to Y$. Then:*

1. *f is injective iff f is monic.*
2. *f is surjective iff f is epic.*

Proof We show 1. Suppose $f : X \to Y$ is injective, and that $f \circ g = f \circ h$, where $g, h : Z \to X$. Then for all $z \in Z$:

$$f(g(z)) = f \circ g(z) = f \circ h(z) = f(h(z)) \,.$$

Since f is injective, this implies $g(z) = h(z)$. Hence we have shown that

$$\forall z \in Z.\, g(z) = h(z) \,,$$

and so we can conclude that $g = h$. So f injective implies f monic.
For the converse, fix a one-element set $\mathbf{1} = \{\bullet\}$. Note that elements $x \in X$ are in 1–1 correspondence with functions $\bar{x} : \mathbf{1} \to X$, where $\bar{x}(\bullet) := x$. Moreover, if $f(x) = y$ then $\bar{y} = f \circ \bar{x}$. Writing injectivity in these terms, it amounts to the following:

$$\forall x, x' \in X.\, f \circ \bar{x} = f \circ \bar{x}' \implies \bar{x} = \bar{x}'.$$

Thus we see that being injective is a *special case* of being monic. □

The reader will enjoy — and learn from — proving the equivalence for functions of the conditions of being surjective and epic.

2.1.3 Generality of Notions

Since the concepts of monic and epic are defined in purely arrow-theoretic terms, *they make sense in any category*. This possibility for making definitions in vast generality by formulating them in purely arrow-theoretic terms can be applied to virtually all the fundamental notions and constructions which pervade mathematics.

As an utterly elementary, indeed "trivial" example, consider the notion of isomorphism. What is an isomorphism *in general*? On might try a definition at the level of generality of model theory, or Bourbaki-style structures, but this is really both unnecessarily elaborate, and still insufficiently general. Category theory has exactly the language needed to give a perfectly general answer to the question, in *any mathematical context, as specified by a category*. An isomorphism in a category $\mathcal{C}$ is an arrow $f : A \to B$ with a two-sided inverse: an arrow $g : B \to A$ such that

$$g \circ f = \mathsf{id}_A, \qquad f \circ g = \mathsf{id}_B.$$

One can check that in **Set** this yields the notion of bijection; in **Grp** it yields isomorphism of groups; in **Top** it yields homeomorphism; in $\mathbf{Mat}_k$, it yields the usual notion of invertible matrix; and so on throughout the range of mathematical structures. In a monoid considered as a category, an isomorphism is an invertible element. Thus a group is exactly a one-object category in which every arrow is an isomorphism! This cries out for generalization; and the notion of a category in which every arrow is an isomorphism is indeed significant — it is the idea of a *groupoid*, which plays a key rôle in modern geometry and topology.

We also see here a first indication of the *prescriptive nature* of categorical concepts. Having defined a category, what the notion of isomorphism means inside that category is now *fixed* by the general definition. We can observe and characterize what that notion is; if it isn't right for our purposes, we need to work in a different category.

2.1.4 Replacing Coding by Intrinsic Properties

We now consider one of the most common constructions in mathematics: the formation of "direct products". Once again, rather than giving a case-by-case construction of direct products in each mathematical context we encounter, we can express once and for all a general notion of product, meaningful in any category — and such that, if a product exists, it is characterised uniquely up to unique isomorphism. Given a particular mathematical context, *i.e.* a category, we can then verify whether on not the product exists in that category. The concrete construction appropriate to the context will enter only into the proof of *existence*; all of the useful *properties* of the product follow from the general definition. Moreover, the categorical notion of product has a *normative* force; we can test whether a concrete construction works as intended by verifying that it satisfies the general definition.

In set theory, the cartesian product is defined in terms of the ordered pair:

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

It turns out that ordered pairs can be *defined* in set theory, e.g. as

$$(x, y) := \{\{x, y\}, y\}.$$

Note that in no sense is such a definition canonical. The essential *properties* of ordered pairs are:

1. We can retrieve the first and second components x, y of the ordered pair (x, y), allowing *projection functions* to be defined:

$$\pi_1 : (x, y) \mapsto x, \qquad \pi_2 : (x, y) \mapsto y\,.$$

2. The information about first and second components completely determines the ordered pair:

$$(x_1, x_2) = (y_1, y_2) \iff x_1 = y_1 \ \wedge \ x_2 = y_2.$$

The categorical definition expresses these properties in arrow-theoretic terms, meaningful in any category.

Let A, B be objects in a category $\mathcal{C}$. A *product* of A and B is an object $A \times B$ together with a pair of arrows $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ such that for *every* such triple $A \xleftarrow{f} C \xrightarrow{g} B$ there exists a *unique* morphism

$$\langle f, g \rangle : C \longrightarrow A \times B$$

such that the following diagram commutes.

$$\begin{tikzcd}
A & A \times B \arrow[l, "\pi_1"'] \arrow[r, "\pi_2"] & B \\
& C \arrow[ul, "f"] \arrow[u, dashed, "\langle f, g \rangle"] \arrow[ur, "g"'] &
\end{tikzcd}$$

Writing the equations corresponding to this commuting diagram explicitly:

$$\pi_1 \circ \langle f, g \rangle = f, \qquad \pi_2 \circ \langle f, g \rangle = g.$$

Moreover, $\langle f, g \rangle$ is *the unique morphism* $h : C \to A \times B$ satisfying these equations.

To relate this definition to our earlier discussion of definitons of pairing for sets, note that a 'pairing' $A \xleftarrow{f} C \xrightarrow{g} B$ offers a decomposition of C into components in A and B, at the level of arrows rather than elements. The fact that pairs are uniquely determined by their components is expressed in arrow-theoretic terms by the *universal property* of the product; the fact that for *every* candidate pairing, there is a unique arrow into the product, which commutes with taking components.

As immediate evidence that this definition works in the right way, we note the following properties of the categorical product (which of course hold in *any* category):

- The product is determined *uniquely up to unique isomorphism.* That is, if there are two pairings satisfying the universal property, there is a unique isomorphism between them which commutes with taking components. This sweeps away all issues of coding and concrete representation, and shows that we have isolated the essential content of the notion of product. We shall prove this property for the related case of terminal objects in the next subsection.

- We can also express the universal property in purely equational terms. This equational specification of products requires that we have a pairing $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ satisfying the equation
$$\pi_1 \circ \langle f, g \rangle = f, \qquad \pi_2 \circ \langle f, g \rangle = g$$
as before, and additionally, for any $h : C \to A \times B$:
$$h = \langle \pi_1 \circ h, \pi_2 \circ h \rangle.$$
This says that any map into the product is uniquely determined by its components. This equational specification is equivalent to the definition given previously.

We look at how this definition works in some of our example categories.

- In **Set**, products are the usual cartesian products.
- In **Pos**, products are cartesian products with the pointwise order.
- In **Top**, products are cartesian products with the product topology.
- In $\mathbf{Vect}_k$, products are direct sums.
- In a poset, seen as a category, products are *greatest lower bounds.*

2.1.5 Terminal Objects

Our discussion in the previous sub-section was for *binary* products. The same idea can be extended to define the product of any family of objects in a category. In particular, the apparently trivial idea of the product of an empty family of objects turns out to be important. The product of an empty family of objects in a category $\mathcal{C}$ will be an object $\mathbf{1}$; there are no projections, since there is nothing in the family to project to! The universal property turns into the following: for each object A in $\mathcal{C}$, there is a unique arrow from C to $\mathbf{1}$. Note that compatibility with the projections trivially holds, since there are no projections! This 'empty product' is the notion of *terminal object*, which again makes sense in any category.

Examples

- In **Set**, any one-element set $\{\bullet\}$ is terminal.
- In **Pos**, the poset $(\{\bullet\}, \{(\bullet, \bullet)\})$ is terminal.

- In **Top**, the space $(\{\bullet\}, \{\varnothing, \{\bullet\}\})$ is terminal.
- In $\mathbf{Vect}_k$, the one-element space $\{0\}$ is terminal.
- In a poset, seen as a category, a terminal object is a greatest element.

We shall now prove that terminal objects are *unique up to (unique) isomorphism.* This property is characteristic of all such "universal" definitions. For example, the apparent arbitrariness in the fact that any singleton set is a terminal object in **Set** is answered by the fact that what counts is the property of being terminal; and this suffices to ensure that any two objects having this property must be isomorphic to each other.

The proof of the proposition, while elementary, is a first example of distinctively categorical reasoning.

Proposition 2 *If T and T' are terminal objects in the category $\mathcal{C}$ then there exists a unique isomorphism $T \cong T'$.*

Proof Since T is terminal and T' is an object of $\mathcal{C}$, there is a unique arrow $\tau_{T'} : T' \to T$. We claim that $\tau_{T'}$ is an isomorphism.
Since T' is terminal and T is an object in $\mathcal{C}$, there is an arrow $\tau'_T : T \to T'$. Thus we obtain $\tau_{T'} \circ \tau'_T : T \to T$, while we also have the identity morphism $\mathsf{id}_T : T \to T$. But T is terminal, and therefore there exists a *unique* arrow from T to T, which means that $\tau_{T'} \circ \tau'_T = \mathsf{id}_T$. Similarly, $\tau'_T \circ \tau_{T'} = \mathsf{id}_{T'}$, so $\tau_{T'}$ is indeed an isomorphism. □

One can reduce the corresponding property for binary products to this one, since the definition of binary product is equivalently expressed by saying that the pairing $A \xleftarrow{\pi_1} A \times B \xrightarrow{\pi_2} B$ is terminal in the category of such pairings, where the morphisms are arrows preserving the components.

It is straightforward to show that if a category has a terminal object, and all binary products, then it has products of all finite families of objects. Thus these are the two cases usually considered.

2.1.6 Natural Numbers

We might suppose that category theory, while suitable for formulating general notions and structures, would not work well for specific mathematical objects such as the number systems. In fact, this is not the case, and the idea of *universal definition*, which we just caught a first glimpse of in the categorical notion of product, provides a powerful tool for specifying the basic discrete number systems of mathematics. We shall illustrate this with the most basic number system of all — the natural numbers (*i.e.* the non-negative integers).

Suppose that $\mathcal{C}$ is a category with a terminal object $\mathbf{1}$. We define a *natural numbers object* in $\mathcal{C}$ to be an object N together with arrows $z : \mathbf{1} \to N$ and $s : N \to N$ such that, for every such triple of an object A and arrows $c : \mathbf{1} \to A$, $f : A \to A$, *there exists a unique arrow* (note this characteristic property of universal definitions again) $h : N \to A$ such that the following diagram commutes:

$$\begin{tikzcd}
\mathbf{1} \arrow[r, "z"] \arrow[dr, "c"'] & N \arrow[r, "s"] \arrow[d, "h"] & N \arrow[d, "h"] \\
& A \arrow[r, "f"'] & A
\end{tikzcd}$$

Equivalently, this means that the following equations hold:

$$h \circ z = c, \qquad h \circ s = f \circ h.$$

Once again, the universal property implies that if a natural numbers object exists in $\mathcal{C}$, it is unique up to unique isomorphism. We are not committed to any particular representation of natural numbers; we have specified the properties a structure with a constant and a unary operation must have in order to function as the natural numbers in a particular mathematical context.

In **Set**, we can verify that $\mathbb{N} = \{0, 1, 2, \ldots\}$ equipped with

$$z : \{\bullet\} \to \mathbb{N} :: \bullet \mapsto 0, \qquad s : \mathbb{N} \to \mathbb{N} :: n \mapsto n + 1$$

does indeed form a natural numbers object. But we are not committed to any particular set-theoretic representation of $\mathbb{N}$: whether as von Neumann ordinals, Zermelo numerals [10] or anything else. Indeed, any countable set X with a particular element x picked out by a map z, and a unary operation $s : X \to X$ which is injective and has $X \setminus \{x\}$ as its image, will fulfil the definition; and any two such systems will be canonically isomorphic.

Note that, if we are given a natural numbers object (N, z, s) in an abstract category $\mathcal{C}$, the resources of *definition by primitive recursion* are available to us. Indeed, we can define *numerals relative to* N: $\bar{n} : \mathbf{1} \to N := s^n \circ z$. Here s^n is defined inductively: $s^1 = s$, $s^{n+1} = s \circ s^n$.[2] Given any (A, c, f), with the unique arrow $h : N \to A$ given by the universal property, we can check that $h \circ \bar{n} = f^n \circ c$. In fact, if we assume that $\mathcal{C}$ has finite products, and refine the definition of natural numbers object to allow for parameters, or if we keep the definition of natural numbers object as it is but assume that $\mathcal{C}$ is *cartesian closed* [20], then all primitive recursive function definitions can be interpreted in $\mathcal{C}$, and will have their usual equational properties.

2.1.7 Functors: category theory takes its own medicine

Part of the "categorical philosophy" is:

Don't just look at the objects; take the morphisms into account too.

We can also apply this to categories! A "morphism of categories" is a *functor*. A functor $F : \mathcal{C} \to \mathcal{D}$ is given by:

- An object map, assigning an object FA of $\mathcal{D}$ to every object A of $\mathcal{C}$.
- An arrow map, assigning an arrow $Ff : FA \to FB$ of $\mathcal{D}$ to every arrow $f : A \to B$ of $\mathcal{C}$, in such a way that composition and identities are preserved:

$$F(g \circ f) = Fg \circ Ff\,, \qquad F\mathsf{id}_A = \mathsf{id}_{FA]}.$$

Note that we use the same symbol to denote the object and arrow maps; in practice, this never causes confusion. The conditions expressing preservation of composition and identities are called *functoriality*.

As a first glimpse as to the importance of functoriality, note the following:

Proposition 3 *Functors preserve isomorphisms; if $f : A \to B$ is an isomorphism, so is Ff.*

Proof Suppose that f is an isomorphism, with inverse f^{-1}. Then

$$F(f^{-1}) \circ F(f) = F(f^{-1} \circ f) = F(\mathsf{id}_A) = \mathsf{id}_{FA}$$

and similarly $F(f) \circ F(f^{-1}) = \mathsf{id}_{FB}$. So $F(f^{-1})$ is a two-sided inverse for Ff, which is thus an isomorphism. □

[2] There is a metainduction going on here, using a natural number object *outside* the category under discussion. This is not essential, but is a useful device for seeing what is going on.

Examples

- Let $(P, \leq)$, $(Q, \leq)$ be preorders (seen as categories). A functor $F : (P, \leq) \longrightarrow (Q, \leq)$ is specified by an object-map, say $F : P \to Q$, and an appropriate arrow-map. The arrow-map corresponds to the condition

$$\forall p_1, p_2 \in P.\, p_1 \leq p_2 \;\Rightarrow\; F(p_1) \leq F(p_2)\,,$$

i.e. to monotonicity of F. Moreover, the functoriality conditions are trivial since in the codomain $(Q, \leq)$ all hom-sets are singletons.
Hence, a functor between preorders is just a monotone map.

- Let $(M, \cdot, 1)$, $(N, \cdot, 1)$ be monoids. A functor $F : (M, \cdot, 1) \longrightarrow (N, \cdot, 1)$ is specified by a trivial object map (monoids are categories with a single object) and an arrow-map, say $F : M \to N$. The functoriality conditions correspond to

$$\forall m_1, m_2 \in M.\, F(m_1 \cdot m_2) = F(m_1) \cdot F(m_2)\,, \qquad F(1) = 1\,,$$

i.e. to F being a monoid homomorphism.
Hence, a functor between monoids is just a monoid homomorphism.

Some further examples:

- The *covariant* powerset functor $\mathcal{P} : \mathbf{Set} \to \mathbf{Set}$:

$$X \mapsto \mathcal{P}(X)\,, \qquad (f : X \to Y) \mapsto \mathcal{P}(f) := S \mapsto \{f(x) \mid x \in S\}.$$

- More sophisticated examples: e.g. *homology*. The basic idea of algebraic topology is that there are functorial assignments of algebraic objects (e.g. groups) to topological spaces. The fact that functoriality implies that isomorphisms are preserved shows that these assignments are *topological invariants*. Variants of this idea ('(co)homology theories') are pervasive throughout modern pure mathematics.

2.1.8 The category of categories

There is a category **Cat** whose objects are categories, and whose arrows are functors. Identities in **Cat** are given by identity functors:

$$\mathsf{Id}_{\mathcal{C}} : \mathcal{C} \longrightarrow \mathcal{C} := A \mapsto A,\; f \mapsto f.$$

Composition of functors is defined in the evident fashion. Note that if $F : \mathcal{C} \to \mathcal{D}$ and $G : \mathcal{D} \to \mathcal{E}$ then, for $f : A \to B$ in $\mathcal{C}$,

$$G \circ F(f) := G(F(f)) : G(F(A)) \longrightarrow G(F(B))$$

so the types work out. A category of categories sounds (and is) circular, but in practice is harmless: one usually makes some size restriction on the categories, and then **Cat** will be too 'big' to be an object of itself.

2.1.9 Universality and Adjoints

Universality arises when we are interested in finding *canonical solutions* to problems of construction: that is, we are interested not just in the *existence* of a solution but in its *canonicity*. This canonicity should guarantee uniqueness, in the sense we have become familiar with: a canonical solution should be *unique up to (unique) isomorphism*.

The notion of canonicity has a simple interpretation in the case of posets, as an *extremal solution*: one that is the least or the greatest among all solutions. Such an extremal solution is obviously unique. For example, consider the problem of finding a lower bound of a pair of elements A, B in a poset P: a *greatest lower bound* of A and B is an extremal solution to this problem. As we have seen, this is the specialisation to posets of the problem of constructing a product:

- A product of A, B in a poset is an element C such that $C \leq A$ and $C \leq B$, (C is a lower bound);
- and for any other solution C', *i.e.* C' such that $C' \leq A$ and $C' \leq B$, we have $C' \leq C$. (C is a greatest lower bound.)

The ideas of universality and adjunctions have an appealingly simple form in the case of posets, which is, moreover, useful in its own right, and in particular has some striking applications to logic. We shall develop the ideas in that special case.

Suppose $g : Q \to P$ is a monotone map between posets. Given $x \in P$, a *g-approximation of x* (from above) is an element $y \in Q$ such that $x \leq g(y)$.
A *best g-approximation of x* is an element $y \in Q$ such that

$$x \leq g(y) \ \wedge \ \forall z \in Q. \, (\, x \leq g(z) \ \Rightarrow \ y \leq z \,) \, .$$

If a best g-approximation exists then it is clearly unique.

Discussion It is worth clarifying the notion of best g-approximation. If y is a best g-approximation to x, then in particular, by monotonicity of g, $g(y)$ is the least element of the set of all $g(z)$ where $z \in Q$ and $x \leq g(z)$. However, the property of being a best approximation is much *stronger* than the mere existence of a least element of this set. We are asking for *y itself* to be the least, *in Q*, among all elements z such that $x \leq g(z)$. Thus, even if g is *surjective*, so that for every x there is a $y \in Q$ such that $g(y) = x$, there need not exist a *best* g-approximation to x. This is exactly the issue of having a *canonical choice* of solution.

Exercise Give an example of a surjective monotone map $g : Q \to P$, and an element $x \in P$, such that there is no best g-approximation to x in Q.

If such a best g-approximation $f(x)$ exists for all $x \in P$ then we have a function $f : P \to Q$ such that, for all $x \in P$, $z \in Q$:

$$x \leq g(z) \iff f(x) \leq z \, . \tag{1}$$

We say that f is the *left adjoint* of g, and g is the *right adjoint* of f. It is immediate from the definitions that the left adjoint of g, if it exists, is uniquely determined by g.

Proposition 4 *If such a function f exists, then it is monotone. Moreover,*

$$\mathsf{id}_P \leq g \circ f, \qquad f \circ g \leq \mathsf{id}_Q, \qquad f \circ g \circ f = f, \qquad g \circ f \circ g = g.$$

Proof If we take $z = f(x)$ in equation (1), then since $f(x) \leq f(x)$, $x \leq g \circ f(x)$. Similarly, taking $x = g(z)$ we obtain $f \circ g(z) \leq z$. Now, the ordering on functions $h, k : P \longrightarrow Q$ is the *pointwise order*:

$$h \leq k \iff \forall x \in P. \, h(x) \leq k(x).$$

This gives the first two equations.

Now, if $x \leq_P x'$ then $x \leq x' \leq g \circ f(x')$, so $f(x')$ is a g-approximation of x, and hence $f(x) \leq f(x')$. Thus, f is monotone.

Finally, using the fact that composition is monotone with respect to the pointwise order on functions, and the first two equations:

$$g = \mathsf{id}_P \circ g \leq g \circ f \circ g \leq g \circ \mathsf{id}_Q = g,$$

and hence $g = g \circ f \circ g$. The other equation is proved similarly. $\square$

Examples

- Consider the inclusion map

$$i : \mathbb{Z} \hookrightarrow \mathbb{R}\,.$$

This has both a left adjoint f^L and a right adjoint f^R, where $f^L, f^R : \mathbb{R} \to \mathbb{Z}$. For all $z \in \mathbb{Z}$, $r \in \mathbb{R}$:

$$z \leq f^R(r) \iff i(z) \leq r\,, \qquad f^L(r) \leq z \iff r \leq i(z)\,.$$

We see from these defining properties that the right adjoint maps a real r to the *greatest integer below it* (the extremal solution to finding an integer below a given real). This is the standard *floor function*.

Similarly, the left adjoint maps a real to the least integer above it, yielding the *ceiling function*. Thus:

$$f^R(r) = \lfloor r \rfloor\,, \qquad f^L(r) = \lceil r \rceil\,.$$

- Consider a relation $R \subseteq X \times Y$. R induces a function:

$$f_R : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y) := S \mapsto \{y \in Y \mid \exists x \in S.\, xRy\}\,.$$

This has a right adjoint $[R] : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X)$:

$$S \subseteq [R]T \iff f_R(S) \subseteq T\,.$$

The definition of $[R]$ which satisfies this condition is:

$$[R]T := \{x \in X \mid \forall y \in Y.\, xRy \;\Rightarrow\; y \in T\}\,.$$

If we consider a set of *worlds* W with an *accessibility relation* $R \subseteq W \times W$ as in Kripke semantics for modal logic, we see that $[R]$ gives the usual Kripke semantics for the modal operator $\Box$, seen as a propositional operator mapping the set of worlds satisfied by a formula ϕ to the set of worlds satisfied by $\Box\phi$.

On the other hand, if we think of the relation R as the denotation of a (possibly non-deterministic) program, and T as a predicate on *states*, then $[R]T$ is exactly the *weakest precondition* $\mathbf{wp}(R, T)$ [14]. In *Dynamic Logic* [28], the two settings are combined, and we can write expressions such as $[R]T$ directly, where T will be (the denotation of) some formula, and R the relation corresponding to a program.

2.1.10 Logical notions as adjunctions

We shall look at some examples of adjunctions arising from logic [23], which also give a first impression of the deep connections which exist between category theory and logic.

We begin with implication. Implication and conjunction — whether classical or intuitionistic — are related by the following bidirectional inference rule:

$$\frac{\phi \,\wedge\, \psi \vdash \theta}{\phi \vdash \psi \to \theta}\,.$$

If we form the preorder of formulas related by entailment as a category, this rule becomes a relationship between arrows which holds in this category. In fact, it can be shown that this *uniquely characterizes* implication, and is a form of universal definition. Note that it gives the essence of what implication is. The way one proves an implication—essentially the *only* way—is to add the antecedent to one's assumptions and then prove the consequent. This is justified by the above rule.

In terms of the boolean algebra of sets, define $X \Rightarrow Y = X^c \cup Y$, where X^c is the set complement. Then we have, for any sets X, Y, Z:

$$X \cap Y \subseteq Z \iff X \subseteq Y \Rightarrow Z.$$

This says precisely that the function $f_X : Z \mapsto Z \cap X$ is left adjoint to the function $g_X : Y \mapsto X \Rightarrow Y$.

The same algebraic relation holds in any Heyting algebra, and defines intuitionistic implication.

Now we show that this same formal structure of adjoints underpins quantification. This is the fundamental insight due to Lawvere [23], that *quantifiers are adjoints to substitution.*

Consider a function $f : X \to Y$. This induces a function

$$f^{-1} : \mathcal{P}(Y) \longrightarrow \mathcal{P}(X) :: T \mapsto \{x \in X \mid f(x) \in T\}.$$

This function f^{-1} has both a left adjoint $\exists(f) : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$, and a right adjoint $\forall(f) : \mathcal{P}(X) \longrightarrow \mathcal{P}(Y)$. These adjoints are uniquely specified by the following conditions. For all $S \subseteq X$, $T \subseteq Y$:

$$\exists(f)(S) \subseteq T \iff S \subseteq f^{-1}(T), \qquad f^{-1}(T) \subseteq S \iff T \subseteq \forall(f)(S).$$

The unique functions satisfying these conditions can be defined explicitly as follows:

$$\begin{array}{lcl} \exists(f)(S) & := & \{y \in Y \mid \exists x \in X. \, f(x) = y \;\wedge\; x \in S\}\,, \\ \forall(f)(S) & := & \{y \in Y \mid \forall x \in X. \, f(x) = y \;\Rightarrow\; x \in S\}\,. \end{array}$$

Given a formula ϕ with free variables in $\{v_1, \ldots, v_{n+1}\}$, it will receive its Tarskian denotation $[\![\phi]\!]$ in $\mathcal{P}(A^{n+1})$ as the set of satisfying assignments:

$$[\![\phi]\!] = \{s \in A^{n+1} \mid s \models_X \phi\}\,.$$

We have a projection function

$$\pi : A^{n+1} \longrightarrow A^n \;:: (a_1, \ldots, a_{n+1}) \mapsto (a_1, \ldots, a_n)\,.$$

Note that this projection is the Tarskian denotation of the tuple of terms $(v_1, \ldots, v_n)$. We can characterize the standard quantifiers as *adjoints to this projection*:

$$[\![\forall v_{n+1}. \, \phi]\!] = \forall(\pi)([\![\phi]\!]), \qquad [\![\exists v_{n+1}. \, \phi]\!] = \exists(\pi)([\![\phi]\!])\,.$$

More explicitly, the Tarski semantics over a structure $\mathcal{M} = (A, \ldots)$ assigns such formulas values in $\mathcal{P}(A^{n+1})$. We can regard the quantifiers $\exists v_{n+1}$, $\forall v_{n+1}$ as functions

$$\exists(\pi), \forall(\pi) : \mathcal{P}(A^{n+1}) \longrightarrow \mathcal{P}(A^n)$$

$$\begin{array}{lcl} \exists(\pi)(S) & = & \{s \in A^n \mid \exists a \in A. \, s[v_{n+1} \mapsto a] \in S\} \\ \forall(\pi)(S) & = & \{s \in A^n \mid \forall a \in A. \, s[v_{n+1} \mapsto a] \in S\} \end{array}$$

If we unpack the adjunction conditions for the universal quantifier, they yield the following bidirectional inference rule:

$$\frac{\Gamma \vdash_X \phi}{\Gamma \vdash_X \forall v_{n+1}. \, \phi} \qquad X = \{v_1, \ldots, v_n\}\,.$$

Here the set X keeps track of the free variables in the assumptions Γ. Note that the usual "eigenvariable condition" is automatically taken care of in this way.

Since adjoints are uniquely determined, this characterization completely captures the meaning of the quantifiers.

2.2 Discussion: the significance of category theory

We turn from this all too brief glimpse at the basics of category theory to discuss its conceptual significance, and why it might matter to philosophy.

The basic feature of category theory which makes it conceptually fascinating and worthy of philosophical study is that it is not just another mathematical theory, but a way of mathematical thinking, and of doing mathematics, which is genuinely distinctive, and in particular very different to the prevailing set-theoretic style which preceded it. If one wanted a clear-cut example of a paradigm-shift in the Kuhnian sense within mathematics, involving a new way of looking at the mathematical universe, then the shift from the set-theoretic to the categorical perspective provides the most dramatic example we possess.

This has been widely misunderstood. Category theory has been portrayed, sometimes by its proponents, but more often by its detractors, as offering an alternative foundational scheme for mathematics to set theory. But this is to miss the point. What category theory offers is an alternative to *foundational schemes in the traditional sense themselves.* This point has been argued with great clarity and cogency in a forceful and compelling essay by Steve Awodey [7]. We shall not attempt to replicate his arguments, but will just make some basic observations.

Firstly, it must be emphasized that the formalization of mathematics within the language of set theory, as developed in the first half of the twentieth century, has been extremely successful, and has enabled the formulation of mathematical definitions and arguments with a previously unparalleled degree of precision and rigour. However, the set-theoretical paradigm has some deficiencies.

The set-theoretical formalization of mathematics rests on the idea of representing mathematical objects as sets which can be defined within a formal set theory, typically ZFC. It is indeed a significant empirical observation, as remarked by Blass [11], that mathematical objects can be thus represented, and mathematical proofs carried out using the axioms of set theory. This leads to claims such as the recent one by Kunen [19] (p. 14), that

> All abstract mathematical concepts are set-theoretic. All concrete mathematical objects are specific sets.

This claim fails to ring true, for several reasons.

- Firstly, the set-theoretic representation is *too concrete.* it involves irrelevant details and choices — it is a coding rather than a structural representation of the concepts at hand. We saw this illustrated with the issue of defining ordered pairs in set theory, and the contrast with the categorical definition of product, which extracted the essential structural features of pairing at the right level of abstraction. Even if we think of number systems, the representation say of the natural numbers as the finite ordinals in set theory is just a particular coding — there are many others. The essential features of the natural numbers are, rather, conveyed by the universal definition of *natural numbers object* — which makes sense in any category. We should not ask what natural numbers *are*, but rather what they *do* — or what we can do with them. Set theoretic representations of mathematical objects give us too much information — and information of the wrong kind.

- Furthermore, by being too specific, set theoretic representations lose much of the generality that mathematical concepts, as used by mathematicians, naturally have. Indeed, the notion of natural numbers object makes sense in any category with a terminal object. Moreover, as a universal consruction, if it exists in a given category, it is unique up to unique isomorphism. Once we are in a particular mathematical context specified by a category, we can *look and see* what the natural numbers object is — while knowing that the standard reasoning principles such as proof by induction and definition by primitive recursion will hold.

- When one passes to more inherently structural notions, such as 'cohomology theory' or 'coalgebra' the assertion that 'all abstract mathematical concepts are set-theoretic' becomes staggeringly implausible, unless we replace 'are' by 'are codable into'. The crudity of the pure set-theoretic language becomes all too apparent. One might indeed say that insensitivity to the distortions of coding is a tell-tale feature of the set-theoretic cast of thought.

  It may be useful to draw an analogy here with geometry. A major theme of 20th century geometry was the replacement of coordinate-based definitions of geometrical notions (such as tensors or varieties) with 'intrinsic' definitions. Coordinates are still very useful for calculations, but the intrinsic definitions are more fundamental and more illuminating

— and ultimately more powerful. The move from set-theoretical encodings, which identify mathematical structures with specific entities in the set-theoretical universe, to universal characterizations which make sense in any mathematical context (category) satisfying some given background conditions, similarly leads to greater insight and technical power.

The foundationalist critique of category theory proceeds as follows:

1. Category theory cannot emancipate itself completely from set theory, and indeed relies on set theory at certain points.

2. Hence it is not truly fundamental, and cannot serve as a foundation for mathematics.

On the first point, one can discern two main arguments.

- Firstly the very definition of category and functor presuppose the notion of a *collection* of things, and of *operations* on these things. So one needs an underlying theory of collections and operations as a substrate for category theory.

  This is true enough; but the required 'theory of collections and operations' is quite rudimentary. Certainly nothing like formal set theory is presupposed. In fact, the basic notions of categories are *essentially algebraic* in form [15]; that is, they can be formalized as partial algebras, in which the domains of definition of the operations can themselves be defined equationally, in terms of operations which have already been specified. For example, if we consider composition as a partial binary operation comp on arrows, then $\mathsf{comp}(g, f)$ is defined just when $\mathsf{cod}(f) = \mathsf{dom}(g)$.

- The second argument is that at various points, issues of *size* enter into category theory. We saw an example of this in considering the category **Cat** of categories and functors. Is **Cat** an object of **Cat**? To avoid such issues, one usually defines a version of **Cat** with some size restriction; for example, one only considers categories whose underlying collection of arrows form a set in Zermelo-Fraenkel set theory. Then **Cat** will be too large (a proper class) to be an object of itself.

  There are various technical elaborations of this point. One can consider categories of arbitrary size in a stratified fashion, by assuming a sufficient supply of inaccessible cardinals (and hence of 'Grothendieck universes' [11]). One can also formalize notions of size relative to an ambient category one is 'working inside'; which actually describes what one is doing when formalizing category-thoretic notions in set theory.

  Again, the point that in practice category theory is not completely emancipated from set theory is fair enough. What should be borne in mind, though, is how innocuous this residue of set theory in category theory actually is in practice. The strongly typed nature of category theory means that one rarely — one is tempted to say 'never' — stumbles over these size issues; they serve more as a form of type-checking than as a substantial topic in their own right. Moreover, category theoretic arguments typically work generically in relation to size; thus in practice, one argument fits all cases, despite the stratification.

All this is to say that, while category theory is not completely disentangled from set theory, it is quite misleading to see this as the main issue in considering the philosophical significance of categories. The temptation to do so comes from the foundationalist attitude expressed in (2) above.

The form of categorical structuralism sketched by Awodey in [7] stands in contrast to this set-theoretic foundationalism. It is a much better representation of mathematical practice, and it directs attention towards the kind of issues we have been discussing, and away from the well-worn tracks of traditional thought in the philosophy of mathematics, which after more than a century have surely reached, and passed, the point of diminishing returns.

2.2.1 Categories and Logic

Our brief introduction to category theory did not reach the rich and deep connections which exist between category theory and logic. Categorical logic is a well-developed area, with several different branches. The most prominent of these is *topos theory*.

Topos theory is an enormous field in its own right, now magisterially presented in Peter Johnstone's *magnum opus* [17]. Because, among other things, it provides a categorical formulation of a form of set theory, it is often seen as the main or even the only part of category theory relevant to philosophy. Topos theory is seen as an alternative or rival to standard versions of set theory, and the relevance of category theory to the foundations of mathematics is judged in these terms.

There are many things within topos theory of great conceptual interest; but topos theory is far from covering all of categorical logic, let alone all of category theory. From our perspective, there is a great deal of 'logic' in the elementary parts of category theory which we have diacussed. The overemphasis on topos theory in this context arises from the wish to understand the novel perspectives of category theory in terms of the traditional concepts of logic and set theory. This impulse is understandable, but misguided. As we have already argued, learning to look at mathematics from a category-theoretic viewpoint is a real and deep-seated paradigm shift. It is only by embracing it that we will reap the full benefits.

Thus while we heartily recommend learning about topos theory, this should build on having already absorbed the lessons to be learnt from category theory in general, and with the awareness that there are other important connections between category theory and logic, in particular *categorical proof theory* and *type theory*.

2.2.2 Applications of Category Theory

As we have argued, category theory has a great deal of intrinsic conceptual interest. Beyond this, it offers great potential for applications in formal philosophy, as a powerful and versatile tool for building theories. The best evidence for this comes from Theoretical Computer Science, which has seen an extensive development of applications of category theory over the past four decades.

Some of the main areas where category theory has been applied in Computer Science include:

- **Semantics of Computation**. Denotational semantics of programming languages makes extensive use of categories. In particular, categories of domains have been widely studied [30, 5]. An important topic has been the study of *recursive domain equations* such as

  $$D \cong [D \to D]$$

  which is a space isomorphic to its own function space. Such spaces do not arise in ordinary mathematics, but are just what is needed to provide models for the type-free λ-calculus [9], in which one has self-application, leading to expressions such as the $\mathbf{Y}$ combinator

  $$\lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$$

  which produces fixpoints from arbitrary terms: $\mathbf{Y}M = M(\mathbf{Y}M)$.

  The solution of such domain equations is expressed in terms of fixpoints of functors:

  $$FX \cong X.$$

  This approach to the consistent interpretation of a large class of recursive data types has proved very powerful and expressive, in allowing a wide range of reflexive and recursive behaviours to be modelled.

  Another form of categorical structure which has proved very useful in articulating the semantic structure of programs are monads. Various 'notions of computation' can be encapsulated as monads [26]. This has proved a fruitful idea, not only in semantics, but also in the development of functional programming languages.

- **Type Theories**. An important point of contact between category theory and logic is in the realm of proof theory and type theory. Logical systems can be represented as categories in which formulas are objects, proofs are arrows, and equality of arrows reflects equality of proofs [20]. There are deep connections between cut-elimination in proof systems, and coherence theorems in category theory. Moreover, this paradigm extends to type theories of various kinds, which have played an important rôle in computer science as core calculi for programming languages, and as the basis for automated proof systems.

- **Coalgebra**. Over the past couple of decades, a very lively research area has developed in the field of *coalgebra*. In particular, 'universal coalgebra' has been quite extensively developed as a very attractive theory of systems [29]. This entire area is a good witness to the possibilities afforded by categorical thinking. The idea of an algebra as a set equipped with some operations is familiar, and readily generalizes to the usual setting for universal algebra. Category theory allows us to *dualize* the usual discussion of algebras to obtain a very general notion of *coalgebras of an endofunctor*. Coalgebras open up a new and quite unexpected territory, and provides an effective abstraction and mathematical theory for a central class of computational phenomena:

  - Programming over *infinite data structures*, such as streams, lazy lists, infinite trees, etc.
  - A novel notion of *coinduction*
  - Modelling *state-based computations* of all kinds
  - A general notion of *observation equivalence* between processes.
  - A general form of *coalgebraic logic*, which can be seen as a wide-ranging generalization of modal logic.

  In fact, coalgebra provides the basis for a very expressive and flexible theory of discrete, state-based dynamical systems, which seem ripe for much wider application than has been considered thus far; for a recent application to the representation of physical systems, see [2].

- **Monoidal Categories**.

  Monoidal categories impart a geometrical flavour to category theory. They have a beautiful description in terms of 'string diagrams' [31], which allows equational proofs to be carried out in a visually compelling way. There are precise correspondences between free monoidal categories of various kinds, and constructions of braids, tangles, links, and other basic structures in knot theory and low-dimensional topology. Monoidal categories are also the appropriate general setting for the discussion of multilinear algebra, and, as has recently been shown, for much of the basic apparatus of quantum mechanics and quantum information: tensor products, traces, kets, bras and scalars, map-state duality, Bell states, teleportation and more [3, 4]. There are also deep links to linear logic and other substructural, 'resource-sensitive' logics, and to diagrammatic representations of proofs. For a paper showing links between all these topics, see [1]. Monoidal categories are used in the modelling of concurrent processes [25], and are beginning to be employed in 'computational systems biology' [18].

  Altogether, the development of structures based on monoidal categories, and their use in modelling a wide range of computational, physical, and even biological phenomena, is one of the liveliest areas in current logically and semantically oriented Theoretical Computer Science.

  It is interesting to compare and contrast the two rich realms of monoidal categories and the structures built upon them, on the one hand; and topos theory, on the other. One might say: the *linear* world, and the *cartesian* world. It is still not clear how these two worlds should be related. A clearer understanding of the mathematical and structural issues here may shed light on difficult questions such as the relation of quantum and classical in physics.

Having surveyed some of the ways in which category theory has been used within Computer Science, we shall now consider some of the features and qualities of category theory which have made it particularly suitable for these applications, and which may suggest a wider range of possible applications within the scope of formal philosophy.

Modelling at the right level of abstraction As we have discussed, category theory goes beyond coding to extract the essential features of concepts in terms of universal characterizations, which are then uniquely specified up to isomorphism. This is not just aesthetically pleasing; as experience in Computer Science has shown, working at the right level of abstraction is *essential* if large and complex systems are to be described and reasoned about in a managable fashion. Formal philosophy will benefit enormously by learning this lesson — among others! — from Computer Science.

Compositionality Another deep lesson to be learned from Computer Science is the importance of *compositionality*, in the general sense of a form of description of complex systems in terms of their parts. This notion originates in logic, but has been greatly widened in scope and applicability in its use in computer science.

The traditional approach to systems modelling in the sciences has been *monolithic*; one considers a whole system, models it with a system of differential equations or some other formalism, and then analyzes the model.

In the compositional approach, one starts with a fixed set of basic, simple building blocks, and *constructions* for building new, more complex systems out of given sub-systems, and builds up the required complex system with these. This typically leads to some form of algebraic description of complex systems:

$$S \;=\; \omega(S_1, \ldots, S_n)$$

where ω is an operation corresponding to one of the system-building constructions.

In order to understand the logical properties of such a system, one can develop a matching compositional view:

$$\frac{S_1 \models \phi_1, \ldots, S_n \models \phi_n}{\omega(S_1, \ldots, S_n) \models \phi}$$

One searches for a rule will allows one to reduce the verification of a property of a complex system to verifications of sub-properties for its components.

The compositional methods for description and analysis of systems which have been developed in Computer Science are ripe for application in a much wider range of scientific contexts — and in formal philosophy.

Mappings between representations Another familiar theme in Computer Science is the need for multiple levels of abstraction in describing and analyzing complex systems, and for mappings between them. Functorial methods provide the most general and powerful basis for such mappings. Particular cases, such as Galois connections, which specialize the categorical notion of adjoint functors to posets, are widely used in abstract interpretation [13].

Normative criteria for definitions As we have already remarked on a couple of occasions, category theory has a strong normative force. If we devise a mapping from one kind of structure to another, category theory tells us that we should demand that it maps morphisms as well as objects, and that it should be *functorial*. Similarly, if we devise some kind of product for a certain type of structures, category theory tells us which properties our construction should satisfy to indeed be a product in the corresponding category. More generally, constructions, if they are 'canonical', should satisfy a suitable universal property; and if they do, then they are unique up to isomorphism. There are other important criteria too, such as *naturality* (which we have not discussed).

These demands and criteria to be satisfied should be seen as providing valuable *guidance*, as we seek to develop a suitable theory to capture some phenomenon. If we have no such guidance, it is all too likely that we may make various ad hoc definitions, not really knowing what we are doing. As it is, once we have specified a category, there are an enormous range of well-posed questions about its structure which we can ask. Does the category have products? Other kinds of limits and colimits? Is it cartesian closed? Is it a topos? And so on. By the time we have answered these questions, we will already know a great deal about the structure of the category, and what we can do with it. We can also then focus on the more distinctive features of the category, which may in turn lead to a characterization of it, or perhaps to a classification of categories of that kind.

2.3 Logic And Category Theory As Tools For Building Theories

The project of scientific or formal philosophy, which seems to be gathering new energy in recent times, can surely benefit from the methods and tools offered by Category theory. Indeed, it can surely not afford to neglect them. Logic has been used as the work-horse of formal philosophy for many years, but the limitations of logic as traditionally conceived become apparent as soon as one takes a wider view of the intellectual landscape. In particular, Computer Science has led the way in finding new ways of applying logic — and new forms of logic and structural mathematics which can be fruitfully applied.

Philosophers and foundational thinkers who are willing and able to grasp these opportunities will find a rich realm of possibilities opening up before them. Perhaps this brief essay, modest in scope as it is, will point someone along this road. If so, the author will feel handsomely rewarded.

3 Guide to Further Reading

The lecture notes [6] are a natural follow-up to this article.

The short book [27] is nicely written and gently paced. A very clear, thorough, and essentially self-contained introduction to basic category theory is given in [8].

Another very nicely written text, focussing on the connections between categories and logic, and especially topos theory, is [16], recently reissued by Dover Books. The book [24] is pitched at an elementary level, but offers insights by one of the key contributors to category theory.

The text [21] is a classic by one of the founders of category theory. It assumes considerable background knowledge of mathematics to fully appreciate its wide-ranging examples, but it provides invaluable coverage of the key topics. The 3-volume handbook [12] provides coverage of a broad range of topics in category theory.

A classic text on categorical logic and type theory is [20]. A more advanced text on topos theory is [22]; while [17] is a comprehensive treatise, of which Volume 3 is still to appear.

References

[1] S. Abramsky. Temperley-Lieb algebra: From knot theory to logic and computation via quantum mechanics. In Goong Chen, Louis Kauffman, and Sam Lomonaco, editors, *Mathematics of Quantum Computing and Technology*, pages 415–458. Taylor and Francis, 2007.

[2] S. Abramsky. Coalgebras, Chu spaces, and representations of physical systems. In J.-P. Jouannaud, editor, *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science: LiCS 2010*, pages 211–220. IEEE Computer Society Press, 2010.

[3] S. Abramsky and B. Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science: LiCS 2004*, pages 415–425. IEEE Computer Society, 2004.

[4] S. Abramsky and B. Coecke. Categorical quantum mechanics. In K. Engesser, D. Gabbay, and D. Lehmann, editors, *Handbook of Quantum Logic and Quantum Structures: Quantum Logic*, pages 261–324. Elsevier, 2009.

[5] S. Abramsky and A. Jung. Domain theory. In S. Abramsky, D. Gabbay, and T. S. E. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 1–168. Oxford University Press, 1994.

[6] S. Abramsky and N. Tzevelekos. Introduction to category theory and categorical logic. In B. Coecke, editor, *New Structures for Physics*, Lecture Notes in Physics, pages 3–94. Springer-Verlag, 2011.

[7] S. Awodey. An answer to G. Hellman's question "Does category theory provide a framework for mathematical structuralism?". *Philosophia Mathematica*, 12:54–64, 2004.

[8] S. Awodey. *Category theory*. Oxford University Press, 2010.

[9] H. P. Barendregt. *The Lambda Calculus*, volume 103 of *Studies in Logic and Foundations of Mathematics*. North-Holland, 1984.

[10] Paul Benacerraf. What numbers could not be. *The Philosophical Review*, 74:47–73, 1965.

[11] Andreas Blass. The interaction between category theory and set theory. In J. Gray, editor, *Mathematical Applications of Category Theory*, volume 30 of *Contemporary Mathematics*, pages 5–29. AMS, 1984.

[12] F. Borceux. *Handbook of Categorical Algebra Volumes 1–3*. Cambridge University Press, 1994.

[13] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In *POPL*, pages 238–252, 1977.

[14] E.W. Dijkstra. *A Discipline of Programming*. Prentice-Hall, 1976.

[15] P. J. Freyd. Aspects of topoi. *Bulletin of the Australian Mathematical Society*, 7(1):1–76, 1972.

[16] R. I. Goldblatt. *Topoi, the Categorial Analysis of Logic*. North-Holland, 1984. Reprinted by Dover Books, 2006.

[17] P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium. I, II.* Oxford University Press, 2002.

[18] Jean Krivine, Robin Milner, and Angelo Troina. Stochastic bigraphs. In *Proceedings of MFPS XXIV: Mathematical Foundations of Programming Semantics*, volume 218 of *ENTCS*, page 7396, 2008.

[19] Kenneth Kunen. *Foundations of Mathematics*. College Publications, 2009.

[20] J. Lambek and P. J. Scott. *Introduction to higher-order categorical logic*. Cambridge University Press, 1986.

[21] S. Mac Lane. *Categories for the Working Mathematician, Second Edition*. Springer, 1998.

[22] S. Mac Lane and I. Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer, 1994.

[23] F. W. Lawvere. Adjointness in foundations. *Dialectica*, 23:281–296, 1969.

[24] F. W. Lawvere and S. Schanuel. *Conceptual Mathematics: A First Introduction to Categories*. Cambridge University Press, 1997.

[25] Robin Milner. *The Space and Motion of Communicating Agents*. Cambridge University Press, 2009.

[26] Eugenio Moggi. Notions of computation and monads. *Inf. Comput.*, 93(1):55–92, 1991.

[27] Benjamin C. Pierce. *Basic Category Theory for Computer Scientists*. MIT Press, 1991.

[28] V.R. Pratt. Semantical considerations on Floyd-Hoare logic. In *Proceedings of 17th Annual Symposium on Foundations of Computer Science, 1976*, pages 109–121. IEEE Press, 1976.

[29] Jan J. M. M. Rutten. Universal coalgebra: a theory of systems. *Theor. Comput. Sci.*, 249(1):3–80, 2000.

[30] D. S. Scott. Outline of a mathematical theory of computation. Technical report, Oxford University Computing Laboratory, 1970. Technical Monograph PRG-2 OUCL.

[31] Peter Selinger. A survey of graphical languages for monoidal categories. In B. Coecke, editor, *New Structures for Physics*, Lecture Notes in Physics, pages 289–355. Springer-Verlag, 2011.