

CATEGORIES

ARTHUR RYMAN

ABSTRACT. This article is part of the `mathz` library. It contains formal definitions for mathematical concepts related to categories. It uses Z Notation and has been type checked by `fUZZ`.

CONTENTS

| | |
|--|----|
| Introduction | 1 |
| 1. Categories, Functors, and Natural Transformations | 2 |
| References | 11 |

INTRODUCTION

This article is part of the `mathz` library. It contains formal definitions for mathematical concepts related to categories. It uses Z Notation[3] and has been type checked by `fUZZ`[4].

Category theory provides a unifying organizational framework for mathematics. It abstracts and generalizes many concepts and constructions that occur throughout mathematics. For example, maps of sets, homomorphisms of groups, and linear transformations of vector spaces all form categories. Without these, and other important motivating examples, category theory might seem to be merely “general abstract nonsense”.

A formal definition in Z Notation may depend on other formal definitions. However, any formal definition may only depend on previous definitions, i.e. the dependencies must be noncircular. The dependency relation between definitions induces a dependency relation between articles, namely an article depends on another article if any of its definitions depend on definitions contained in the other article. Furthermore, the articles in the `mathz` library are organized such that the dependencies between articles are also noncircular.

The requirement for noncircular dependencies between articles raises the question of where to position categories within the `mathz` library. Should categories be foundational or advanced? Here the distinction between being foundational versus advanced is that a foundational article should depend on very few other articles whereas an advanced article may depend on many others. In the present case, the

question is whether an article on categories should stand on its own or depend on articles that define the motivating examples?

I have taken the position that categories should be foundational and should therefore not depend on any other advanced articles. Accordingly, the definitions presented here will be *informally* illustrated with examples that are formally defined in later articles. For example, although this article states that groups and homomorphisms form a category, it does not use the formal definition of group or homomorphism. Such formal definitions are deferred to later articles. This approach allows later, advanced articles to reference the foundational definitions of categories contained here without introducing circularity in the article dependency graph.

The structure of this article follows that of Mac Lane[1], which the reader should consult for motivating examples. The section numbers here correspond to those in that reference.

1. CATEGORIES, FUNCTORS, AND NATURAL TRANSFORMATIONS

1.1. Axioms for Categories. Mac Lane[1] begins by giving logical axioms for *metagraph* and *metacategory* without reference to set theory. The motivation for doing so is to avoid the well-known paradoxes of set theory that would follow when trying to make sense of concepts such as “the category of all categories”.

Russell[2, Appendix B] introduced *types* precisely to avoid these paradoxes and to put mathematics on a more consistent foundation. Z Notation is based on *typed set theory* and so the introduction of these “meta” concepts is unnecessary. However, there is no free lunch. The tax one pays when using Z Notation is to make definitions explicitly depend on formal generic parameters that represent arbitrary given sets. Specifically, a category generically depends on two arbitrary sets, one for its objects and another for its arrows.

A *graph* consists of a set of *objects* a, b, c, \dots , a set of *arrows* f, g, h, \dots , and two operations, *domain* and *codomain*, that assign objects to arrows. Arrows are also referred to as *morphisms*. The domain and codomain of an arrow are also referred to as its *source* and *target*. The domain of the arrow f is denoted $\text{dom } f$ and its codomain is denoted $\text{cod } f$. The set of all arrows with domain a and codomain b is denoted $a \rightarrow b$ and is referred to as a *hom* set.

| $\text{Category_Graph_Data}[\mathbf{o}, \mathbf{a}]$ |
|--|
| $\text{objects} : \mathbb{P} \mathbf{o}$ |
| $\text{arrows} : \mathbb{P} \mathbf{a}$ |
| $\text{domain}, \text{dom} : \mathbf{a} \leftrightarrow \mathbf{o}$ |
| $\text{codomain}, \text{cod} : \mathbf{a} \leftrightarrow \mathbf{o}$ |
| $\text{hom}, _ \rightarrow _ : \mathbf{o} \times \mathbf{o} \leftrightarrow \mathbb{P} \mathbf{a}$ |
| $\text{dom} = \text{domain}$ |
| $\text{cod} = \text{codomain}$ |
| $(_ \rightarrow _) = \text{hom}$ |

Table 1 gives the L^AT_EX commands for these notations.

| Notation | Command | Description |
|-------------------|--------------------------|-------------|
| $\text{dom } f$ | <code>\domCat f</code> | domain |
| $\text{cod } f$ | <code>\codCat f</code> | codomain |
| $a \rightarrow b$ | <code>a \arrCat b</code> | arrows |

TABLE 1. Notation for domain, codomain, and arrows.

Figure 1 shows an arrow f with domain a and codomain b diagrammed as an arrow pointing from a to b .

$$a \xrightarrow{f} b$$

FIGURE 1. An arrow $f : a \rightarrow b$.

| |
|---|
| $\text{Category_Graph}[\mathbf{o}, \mathbf{a}]$ |
| $\text{Category_Graph_Data}[\mathbf{o}, \mathbf{a}]$ |
| $\text{domain} \in \text{arrows} \rightarrow \text{objects}$ |
| $\text{codomain} \in \text{arrows} \rightarrow \text{objects}$ |
| $\text{hom} = (\lambda a, b : \text{objects} \bullet$ $\{ f : \text{arrows} \mid \text{dom } f = a \wedge \text{cod } f = b \})$ |

Recall that the term *graph* has two distinct uses in mathematics, namely as a set of nodes (objects) connected by arcs (arrows) as above or as a set of ordered pairs that define a binary relation. Indeed, Z Notation identifies a binary relation with its graph. The intended meaning of the term “graph” should be clear from context.

A *category* is a graph with two additional operations, *identity* and *composition*.

The identity operation maps each object a to its *identity arrow* in $a \rightarrow a$. The identity arrow for the object a is denoted $\text{id } a$.

| |
|--|
| $\text{Category_Identity_Data}[\mathbf{o}, \mathbf{a}]$ |
| $\text{identity}, \text{id} : \mathbf{o} \leftrightarrow \mathbf{a}$ |
| $\text{id} = \text{identity}$ |

Table 2 gives the L^AT_EX command for this notation.

| Notation | Command | Description |
|----------------|-----------------------|-------------|
| $\text{id } a$ | <code>\idCat a</code> | identity |

TABLE 2. Notation for identity.

| |
|---|
| $Category_Identity[o, a]$ $Category_Graph[o, a]$ $Category_Identity_Data[o, a]$ |
| $id \in objects \rightarrow arrows$ $\forall a : objects \bullet$ $id\ a \in a \rightarrow a$ |

A pair of arrows (g, f) is *composable* if the domain of g is the codomain of f . The composition operation maps each composable pair of arrows (g, f) where $g : b \rightarrow c$ and $f : a \rightarrow b$ to some arrow $h : a \rightarrow c$ called their *composite*. If (g, f) is a composable pair of arrows then their composite is denoted $g \circ f$.

| |
|---|
| $Category_Composition_Data[a]$ $composable : a \leftrightarrow a$ $composition, _ \circ _ : a \times a \leftrightarrow a$ |
| $(_ \circ _) = composition$ |

Table 3 gives the L^AT_EX command for this notation.

| Notation | Command | Description |
|-------------|-------------------------|-------------|
| $g \circ f$ | <code>\compCat f</code> | composition |

TABLE 3. Notation for composition.

| |
|--|
| $Category_Composition[o, a]$ $Category_Graph[o, a]$ $Category_Composition_Data[a]$ |
| $composable = \{g, f : arrows \mid \text{dom } g = \text{cod } f\}$ $composition \in composable \rightarrow arrows$ $\forall f, g : arrows \mid$ $(g, f) \in composable \bullet$ $g \circ f \in \text{dom } f \rightarrow \text{cod } g$ |

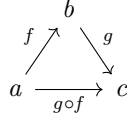
Figure 2 is a diagram of a composable pair of arrows (g, f) .

$$a \xrightarrow{f} b \xrightarrow{g} c$$

FIGURE 2. A composable pair of arrows (g, f) .

Figure 3 illustrates the composition operation.

The composition operation for arrows is inspired by the composition operation for functions. It is an accident of history that we happen to write a function application $f(x)$ with the function f on the left and the argument x on the right. Consequently,

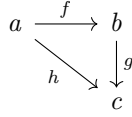
FIGURE 3. Composition of g and f .

the order of functions in a composition is opposite to the order in which they are applied, e.g. $(g \circ f)(x) = g(f(x))$. One can visually eliminate this order mismatch by drawing arrows that point from right to left, as in Figure 4.

$$c \xleftarrow{g} b \xleftarrow{f} a$$

FIGURE 4. Composition of g and f with right-to-left arrows.

The composition $h : a \rightarrow c$ of arrows $g : b \rightarrow c$ and $f : a \rightarrow b$ is diagrammed as a directed graph whose vertices are labelled by the objects a, b, c and whose edges are labelled by the arrows f, g, h . In such a diagram of objects and arrows, any directed path between two objects defines an arrow by composition of the edge labels. If all directed paths between any given pair of objects define the same arrow then the diagram is said to be *commutative*. Figure 5 shows the commutative diagram of the composition $h = g \circ f$.

FIGURE 5. Commutative diagram of $h = g \circ f$.

Consider the configuration of objects and arrows shown in Figure 6.

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{k} d$$

FIGURE 6. The composition $k \circ g \circ f$.

The composition operation must satisfy *associativity*.

| | |
|---|-------|
| $Category_Associativity[o, a]$ | _____ |
| $Category_Composition[o, a]$ | |
| $\forall a, b, c, d : objects; f, g, k : arrows \mid$ $f \in a \rightarrow b \wedge$ $g \in b \rightarrow c \wedge$ $k \in c \rightarrow d \bullet$ $k \circ (g \circ f) = (k \circ g) \circ f$ | |

Figure 7 illustrates associativity.

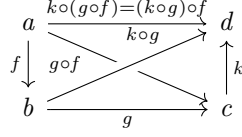


FIGURE 7. Commutative diagram for associativity.

The identity operation must satisfy the *unit law*.

| |
|--|
| $Category_UnitLaw[o, a]$ |
| $Category_Identity[o, a]$ |
| $Category_Composition[o, a]$ |
| $\forall a, b, c : objects \bullet$ $\quad \forall f : a \rightarrow b; g : b \rightarrow c \bullet$ $\quad \quad id\ b \circ f = f \wedge$ $\quad \quad g \circ id\ b = g$ |

Figure 8 illustrates the unit law.

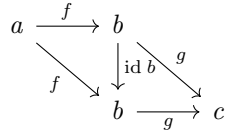


FIGURE 8. Commutative diagram for the unit law.

A *category* is a graph with a composition operation that satisfies associativity and an identity operation that satisfies the unit law.

| |
|---------------------------------|
| $Category[o, a]$ |
| $Category_Associativity[o, a]$ |
| $Category_UnitLaw[o, a]$ |

Remark. The identity operation of a category is an injection from the set of objects into the set of arrows.

$\forall Category[O, A] \bullet$
 $\quad id \in objects \rightarrow arrows$

Proof. Let a and b be arbitrary objects. It suffices to show that

$$id\ a = id\ b \Rightarrow a = b$$

a

$$= \text{dom}(id\ a)$$

$[Category]$

$$\begin{aligned}
&= \text{dom}(\text{id } b) && [\text{By assumption}] \\
&= b && [\text{Category}]
\end{aligned}$$

□

The significance of this remark is that the identity operation is an isomorphism from the set of objects to the set of identity arrows. This implies that a category can be specified, up to an isomorphism of the set of objects, by its set of arrows, its subset of identity arrows, and its composition operation, with all mentions of objects replaced by mentions of identity arrows. Therefore, any true statement about a category can, in principle, be made without reference to its objects. That being said, replacing objects by identity arrows would be awkward. It is more intuitive to think of a category in terms of both its objects and arrows, even though its objects are, technically, somewhat redundant.

Remark. *The composition operation of a category is a surjection onto its set of arrows.*

$$\begin{aligned}
&\forall \text{Category}[\mathbf{O}, \mathbf{A}] \bullet \\
&\quad \text{ran}(_ \circ _) = \text{arrows}
\end{aligned}$$

Proof. The proof follows immediately because every arrow is a composition of itself and the identity arrow of its domain (or codomain).

Although the above brief outline of the proof is perfectly acceptable in everyday mathematical writing, we give a more detailed proof here as a warm-up for future formal proofs.

$$\begin{aligned}
&\text{ran}(_ \circ _) = \text{arrows} \\
&\Leftrightarrow \text{ran}(_ \circ _) \subseteq \text{arrows} \wedge \text{arrows} \subseteq \text{ran}(_ \circ _) && [\text{Definition of } =]
\end{aligned}$$

We have

$$\begin{aligned}
&(_ \circ _) \in \text{composable} \rightarrow \text{arrows} && [\text{Category}] \\
&\Rightarrow \text{ran}(_ \circ _) \subseteq \text{arrows} && [\text{Definition of } \rightarrow]
\end{aligned}$$

Therefore, it suffices to show that

$$\text{arrows} \subseteq \text{ran}(_ \circ _)$$

Let f be an arbitrary member of arrows . It suffices to show that

$$f \in \text{ran}(_ \circ _)$$

Let $a = \text{dom } f$. We have

$$\begin{aligned}
&f \\
&= f \circ (\text{id } a) && [\text{Category}] \\
&\in \text{ran}(_ \circ _) && [\text{Definition of ran}]
\end{aligned}$$

□

The significance of this remark is that if we are given the composition operation of a category, then we can recover its set of arrows by taking the range.

If b is any object of a category, the corresponding identity arrow $\text{id } b$ is uniquely determined by the unit law. Consider two categories that have the same composition operations, but possibly different identity operations.

| | |
|---|-------|
| $\text{Category_SameComposition}[\mathbf{o}, \mathbf{a}]$ | _____ |
| $\text{Category}_1[\mathbf{o}, \mathbf{a}]$ | |
| $\text{Category}_2[\mathbf{o}, \mathbf{a}]$ | |
| $\theta \text{Category_Composition}_1 = \theta \text{Category_Composition}_2$ | |

Remark (Identity Is Unique). *Any two categories that have the same composition operation must also have the same identity operation.*

$\forall \text{Category_SameComposition}[\mathbf{O}, \mathbf{A}] \bullet$
 $\text{id}_1 = \text{id}_2$

Proof. Let b be an arbitrary object. It suffices to show that

$$\text{id}_1 b = \text{id}_2 b$$

We have

$$\begin{aligned} \text{id}_1 b &= (\text{id}_1 b) \circ_2 (\text{id}_2 b) && [\text{Category_UnitLaw}_2] \\ &= (\text{id}_1 b) \circ_1 (\text{id}_2 b) && [\text{Category_SameComposition}] \\ &= \text{id}_2 b && [\text{Category_UnitLaw}_1] \end{aligned}$$

□

The significance of this remark is that given an associative composition operation on a graph, there is at most one identity operation that satisfies the unit law. Hence, given a category, the identity operation is uniquely determined.

When defining a specific category, it is useful to base the definition on a structure that collects together all the unconstrained data required in any generic category.

| | |
|---|-------|
| $\text{Category_Data}[\mathbf{o}, \mathbf{a}]$ | _____ |
| $\text{Category_Graph_Data}[\mathbf{o}, \mathbf{a}]$ | |
| $\text{Category_Identity_Data}[\mathbf{o}, \mathbf{a}]$ | |
| $\text{Category_Composition_Data}[\mathbf{a}]$ | |

STOPPED AT p8: An example is the metacategory of sets.

Mac Lane [1, p8] defines the *metacategory of sets* whose objects are all sets and whose arrows are all functions, where here the definition of a function includes its domain and codomain. As mentioned above, we can avoid the introduction of metacategories because we are using Z Notation which prevents the usual paradoxes of set theory. Instead we define the category of sets whose objects are subsets of

a given set that is provided as a generic parameter. We begin by formalizing the definition of function appropriate for this category.

Let \mathbf{t} and \mathbf{u} be given sets. A *function* in $\mathbf{t} \times \mathbf{u}$ consists of a domain, codomain, and graph. The domain is a subset of \mathbf{t} , the codomain is a subset of \mathbf{u} , and the graph is a set of pairs that associate to each element of the domain a unique element of the codomain.

| | |
|---|-------|
| $Function[\mathbf{t}, \mathbf{u}]$ | _____ |
| $domain : \mathbf{P} \mathbf{t}$ $codomain : \mathbf{P} \mathbf{u}$ $graph : \mathbf{t} \leftrightarrow \mathbf{u}$ | |
| $graph \in domain \rightarrow codomain$ | |

Given a function f , we can *apply* it to any element x of its domain to obtain a corresponding element y of its codomain.

| | |
|--|-------|
| $Function_Application[\mathbf{t}, \mathbf{u}]$ | _____ |
| $f : Function[\mathbf{t}, \mathbf{u}]$ $x : \mathbf{t}$ $y : \mathbf{u}$ | |
| $x \in f.domain$ $y = f.graph(x)$ | |

Note that although we can apply a function to an element of its domain, general category-theoretic statements do not assume that objects have elements and arrows are functions.

Let \mathbf{v} be another given set. Let f be a function in $\mathbf{t} \times \mathbf{u}$ and g a function in $\mathbf{u} \times \mathbf{v}$. The pair (g, f) is said to be *composable* if the domain of g is equal to the codomain of f .

| | |
|--|-------|
| $Function_Composable[\mathbf{t}, \mathbf{u}, \mathbf{v}]$ | _____ |
| $f : Function[\mathbf{t}, \mathbf{u}]$ $g : Function[\mathbf{u}, \mathbf{v}]$ | |
| $g.domain = f.codomain$ | |

Let (g, f) be a pair of composable functions. Their *composition* is the function h in $\mathbf{t} \times \mathbf{v}$ whose graph is the usual function composition of the graphs of g and f .

| | |
|--|-------|
| $Function_Composition[\mathbf{t}, \mathbf{u}, \mathbf{v}]$ | _____ |
| $Function_Composable[\mathbf{t}, \mathbf{u}, \mathbf{v}]$ $h : Function[\mathbf{t}, \mathbf{v}]$ | |
| $h.domain = f.domain$ $h.codomain = g.codomain$ $h.graph = g.graph \circ f.graph$ | |

Let S be a subset of \mathbf{t} . The *identity* function id on S has domain and codomain equal to S and has graph equal to the usual identity graph.

| | |
|---|-------|
| $Function_Identity[\mathbf{t}]$ | _____ |
| $S : \mathbb{P} \mathbf{t}$ | |
| $id : Function[\mathbf{t}, \mathbf{t}]$ | |
| $id.domain = S$ | |
| $id.codomain = S$ | |
| $id.graph = id\ S$ | |

Let $SET[\mathbf{t}]$ denote the set of subsets of \mathbf{t} , i.e. its power set.

$$SET[\mathbf{t}] == \mathbb{P} \mathbf{t}$$

Let $FUNCTION[\mathbf{t}]$ denote the set of all functions in $\mathbf{t} \times \mathbf{t}$.

$$FUNCTION[\mathbf{t}] == Function[\mathbf{t}, \mathbf{t}]$$

The objects of the category of sets in \mathbf{t} are elements of $SET[\mathbf{t}]$ and its arrows are elements of $FUNCTION[\mathbf{t}]$.

| | |
|---|-------|
| $Set_Data[\mathbf{t}]$ | _____ |
| $Category_Data[SET[\mathbf{t}], FUNCTION[\mathbf{t}]]$ | |

| | |
|-----------------------------|-------|
| $Set_Objects[\mathbf{t}]$ | _____ |
| $Set_Data[\mathbf{t}]$ | |
| $objects = SET[\mathbf{t}]$ | |

| | |
|---------------------------------|-------|
| $Set_Arrows[\mathbf{t}]$ | _____ |
| $Set_Data[\mathbf{t}]$ | |
| $arrows = FUNCTION[\mathbf{t}]$ | |

The domain and codomain of the arrows are defined in the obvious way.

| | |
|--|-------|
| $Set_Graph[\mathbf{t}]$ | _____ |
| $Set_Objects[\mathbf{t}]$ | |
| $Set_Arrows[\mathbf{t}]$ | |
| $domain = (\lambda f : arrows \bullet f.domain)$ | |
| $codomain = (\lambda f : arrows \bullet f.codomain)$ | |

Composition of arrows is composition of functions.

| | |
|---|-------|
| $Set_Composition[t]$ | _____ |
| $Set_Graph[t]$ | |
| $composition = \{ Function_Composition[t, t, t] \bullet (g, f) \mapsto h \}$ | |
| $composable = \text{dom } composition$ | |

Identify arrows are identity functions.

| | |
|---|-------|
| $Set_Identity[t]$ | _____ |
| $Set_Graph[t]$ | |
| $identity = \{ Function_Identity[t] \bullet S \mapsto id \}$ | |

Let $Set[t]$ be the structure with objects, arrows, composition, and identity as defined above.

| | |
|-----------------------|-------|
| $Set[t]$ | _____ |
| $Set_Composition[t]$ | |
| $Set_Identity[t]$ | |

Theorem 1. *Sets and functions with composition and identity as defined above form a category.*

$$\forall Set[T] \bullet \\ Category[SET[T], FUNCTION[T]]$$

TODO: Sketch the proof of this theorem. Break up the proof into proofs for objects, arrows, composition, and identity. State all the proof obligations. It may be better to first define the category of relations and then specialize that into functions. Define injections, surjections, bijections, partial functions, total functions.

REFERENCES

- [1] Saunders Mac Lane. *Categories for the Working Mathematician*. Second Edition. Springer, 1998.
- [2] Bertrand Russell. *The Principles of Mathematics*. Cambridge University Press, 1903. URL: <https://people.umass.edu/klement/pom/pom-portrait.pdf>.
- [3] J. M. Spivey. *The Z Notation*. Second Edition. Prentice Hall International, 1992. URL: <https://spivey.oriel.ox.ac.uk/wiki/files/zrm/zrm.pdf>.
- [4] Mike Spivey. *The fuzz Manual*. Second Edition. The Spivey Partnership, 2000. URL: <https://spivey.oriel.ox.ac.uk/wiki/images-corner/c/cc/Fuzzman.pdf>.

Email address, Arthur Ryman: arthur.ryman@gmail.com