# CATEGORIES

ARTHUR RYMAN

Abstract. This article contains Z Notation[2] definitions for concepts related to categories. It has been type checked by *f*UZZ[3].

Contents



## 1. Introduction

The definitions in this article are primarily based on those in [1]. Wikipedia and other sources will be used as needed.

Category theory provides a useful conceptual framework for mathematics. It abstracts and generalizes many concepts and constructions that occur in other branches. For example, maps between sets, homomorphisms between groups, and linear transformations between vector spaces all form categories. The practical utility of category derives from the many examples that occur throughout mathematics. This situation presents a small dilemma for the scope of this article. Should this article treat categories as foundational or advanced?

I have taken the position that this article should be foundational and should therefore not depend on any other articles. Accordingly, the definitions presented here will not be illustrated with formal examples from other articles. For example, although this article does assert that homomorphisms between groups form a category, it does not use the formal definition of group or homomorphism. Such a formal use will be deferred to other articles. This approach allows other, more advanced, articles to reference the foundational definitions contained here without introducing circularity.

*Date*: September 29, 2022.

## 2. Categories, Functors, and Natural Transformations

2.1. **Axioms for Categories.** Mac Lane[1] defines the concepts of *metagraph*, *metacategory*, *large set*, *small set*, and others in order to avoid the well-known paradoxes of set theory. However, these concepts are unnecessary when using Z Notation which uses *simple type theory* to avoid the paradoxes. Indeed, simple type theory was conceived by Russel specifically to put set theory on a firmer foundation.

However, there is no free lunch. The price one pays when using Z Notation is to explicitly parameterize generic definitions with given sets. Specifically, a category generically depends on two given sets, one for its objects and another for its arrows.

A *graph* consists of *objects* $a, b, c, \ldots$, *arrows* $f, g, h, \ldots$, 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*.

$$
\begin{array}{|l}
\textit{Category\_Graph}[\mathsf{o}, \mathsf{a}] \\
\hline
\textit{objects} : \mathbb{P}\,\mathsf{o} \\
\textit{arrows} : \mathbb{P}\,\mathsf{a} \\
\textit{domain}, \textit{codomain} : \mathsf{a} \rightarrow\!\!\!\!\!+ \mathsf{o} \\
\hline
\textit{domain} \in \textit{arrows} \longrightarrow \textit{objects} \\
\textit{codomain} \in \textit{arrows} \longrightarrow \textit{objects} \\
\hline
\end{array}
$$

An arrow $f$ with domain $a$ and codomain $b$ is diagrammed as an arrow pointing from $a$ to $b$.

$$a \xrightarrow{f} b$$

The domain of the arrow $f$ is donated $\operatorname{dom} f$ and its codomain is denoted $\operatorname{cod} f$. The set of all arrows with domain $a$ and codomain $b$ is denoted $a \to b$.

$$
\begin{array}{|l}
\textit{Category\_Graph\_Notation}[\mathsf{o}, \mathsf{a}] \\
\hline
\textit{Category\_Graph}[\mathsf{o}, \mathsf{a}] \\
\operatorname{dom}, \operatorname{cod} : \mathsf{a} \rightarrow\!\!\!\!\!+ \mathsf{o} \\
\_ \to \_ : \mathsf{o} \times \mathsf{o} \rightarrow\!\!\!\!\!+ \mathbb{P}\,\mathsf{a} \\
\hline
\operatorname{dom} = \textit{domain} \\
\operatorname{cod} = \textit{codomain} \\
(\_ \to \_) = (\lambda\, a, b : \textit{objects} \bullet \{\, f : \textit{arrows} \mid \operatorname{dom} f = a \land \operatorname{cod} f = b \,\}) \\
\hline
\end{array}
$$

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

The identity operation maps each object to its *identity arrow*.

$$
\begin{array}{|l}
\textit{Category\_Identity}[\mathsf{o}, \mathsf{a}] \\
\hline
\textit{Category\_Graph}[\mathsf{o}, \mathsf{a}] \\
\textit{identity} : \mathsf{o} \rightarrow\!\!\!\!\!+ \mathsf{a} \\
\hline
\textit{identity} \in \textit{objects} \longrightarrow \textit{arrows} \\
\hline
\end{array}
$$

The identity arrow for the object $a$ is denoted $\mathrm{id}\, a$.

*Category_Identity_Notation*[o, a]

*Category_Identity*[o, a]
$\mathrm{id} : \mathsf{o} \pfun \mathsf{a}$

---

$\mathrm{id} = \mathit{identity}$

A pair of arrows $(g, f)$ is *composable* if the domain of $g$ is the codomain of $f$.

*Category_Composable*[o, a]

*Category_Graph_Notation*[o, a]
$\mathit{composable} : \mathsf{a} \leftrightarrow \mathsf{a}$

---

$\mathit{composable} = \{ g, f : \mathit{arrows} \mid \mathrm{dom}\, g = \mathrm{cod}\, f \}$

The following diagram shows the composable pair of arrows $(g, f)$.

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

It is an accident of history that function application is written with the function on the left of the argument. This has the affect of causing the composition of functions to be written in the opposite order of function application. One can visually eliminate this mismatch by drawing arrows that point from right to left, as in the following diagram.

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

Composition maps each composable pair of arrows $g : b \to c$ and $f : a \to b$ to some arrow $h : a \to c$.

*Category_Composition*[o, a]

*Category_Composable*[o, a]
$\mathit{composition} : \mathsf{a} \times \mathsf{a} \pfun \mathsf{a}$

---

$\mathit{composition} \in \mathit{composable} \longrightarrow \mathit{arrows}$

$\forall f, g : \mathit{arrows} \mid$
$\quad (g, f) \in \mathit{composable} \bullet$
$\quad\quad \mathit{composition}(g, f) \in \mathrm{dom}\, f \to \mathrm{cod}\, g$

The composition $h : a \to c$ of arrows $g : b \to c$ and $f : a \to b$ is diagrammed as a directed graph whose vertices are labelled by the objects $a, b, c$ and whose edges are labelled by the arrows $f, g, h$.

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[dr, "h"'] & b \arrow[d, "g"] \\ & c \end{tikzcd}$$

In such a diagram, 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 be *commutative*.

If arrows $g, f$ are composable then their composition is denoted $g \circ f$.

$$\begin{tikzcd} & b \arrow[dr, "g"] & \\ a \arrow[ur, "f"] \arrow[rr, "g \circ f"'] & & c \end{tikzcd}$$

*Category_Composition_Notation*[o, a]

*Category_Composition*[o, a]

$\_ \circ \_ : \mathsf{a} \times \mathsf{a} \rightarrow\!\!\!\!\!\!+ \mathsf{a}$

---

$(\_ \circ \_) = \mathit{composition}$

The composition and identity operations satisfy *associativity* and the *unit law.*

*Category_Associativity*[o, a]

*Category_Composition_Notation*[o, a]

---

$\forall\, a, b, c, d : \mathit{objects};\ f, g, k : \mathit{arrows} \mid$
$\quad f \in a \rightarrow b\ \land$
$\quad g \in b \rightarrow c\ \land$
$\quad k \in c \rightarrow d\ \bullet$
$\qquad k \circ (g \circ f) = (k \circ g) \circ f$

The following commutative diagram illustrates associativity.

$$\begin{tikzcd} a \arrow[d, "f"'] \arrow[rr, "k \circ (g \circ f) = (k \circ g) \circ f"] \arrow[drr, "g \circ f" near start] & & d \\ b \arrow[urr, "k \circ g"' near end] \arrow[rr, "g"'] & & c \arrow[u, "k"'] \end{tikzcd}$$

*Category_UnitLaw*[o, a]

*Category_Identity_Notation*[o, a]

*Category_Composition_Notation*[o, a]

---

$\forall\, a, b, c : \mathit{objects};\ f, g : \mathit{arrows} \mid$
$\quad f \in a \rightarrow b\ \land$
$\quad g \in b \rightarrow c\ \bullet$
$\qquad \mathrm{id}\ b \circ f = f\ \land$
$\qquad g \circ \mathrm{id}\ b = g$

The following commutative diagram illustrates the unit law.

$$\begin{tikzcd} a \arrow[r, "f"] \arrow[dr, "f"'] & b \arrow[d, "\mathrm{id}\ b"] \arrow[dr, "g"] & \\ & b \arrow[r, "g"'] & c \end{tikzcd}$$

*Category*[o, a]

*Category_Associativity*[o, a]

*Category_UnitLaw*[o, a]

A given category is fully determined by its set of objects, its set of arrows, and its composition operation. In fact, given the composition operation we can cover the set of arrows since ever arrow is the composition of itself and an identity arrow. Furthermore, the set of identity arrows is isomorphic to the set of objects. Therefore given two categories that have the same composition operation, there is a canonical isomorphism between their sets of objects, defined by the domain (or codomain) of the identity arrows. However, it is more explicit to give the objects and arrows in addition to the composition operation. Let $Cat[\mathsf{o}, \mathsf{a}]$ denote the set of all categories.

$$Cat[\mathsf{o}, \mathsf{a}] == \{\ Category[\mathsf{o}, \mathsf{a}] \bullet (objects, arrows, composition)\ \}$$

For example, the set of all subsets of a given set $\mathsf{u}$ form the objects of a category whose arrows are partial functions $\mathsf{u} \nrightarrow \mathsf{u}$ and whose composition operation is the usual composition of composable functions.

$Sets\_and\_Functions[\mathsf{u}]$

$sets : \mathbb{P}(\mathbb{P}\,\mathsf{u})$
$functions : \mathbb{P}(\mathsf{u} \nrightarrow \mathsf{u})$
$composition : (\mathsf{u} \nrightarrow \mathsf{u}) \times (\mathsf{u} \nrightarrow \mathsf{u}) \nrightarrow (\mathsf{u} \nrightarrow \mathsf{u})$

---

$sets = \mathbb{P}\,\mathsf{u}$

$functions = \mathsf{u} \nrightarrow \mathsf{u}$

$composition = (\lambda\, g, f : \mathsf{u} \nrightarrow \mathsf{u} \mid \mathrm{dom}\, g = \mathrm{ran}\, f \bullet g \circ f)$

**Example.** *Sets and functions form a category.*

$$\forall\, Sets\_and\_Functions[\mathsf{U}] \bullet$$
$$(sets, functions, composition) \in Cat[\mathbb{P}\,\mathsf{U}, \mathsf{U} \nrightarrow \mathsf{U}]$$

## References

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

[2] J. M. Spivey. *The Z Notation.* Second Edition. Prentice Hall International, 1992. URL: `https://spivey.oriel.ox.ac.uk/wiki/files/zrm/zrm.pdf`.

[3] 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`