# MONOIDAL CATEGORIES

## A Unifying Concept in
## Mathematics, Physics, and Computing

Noson S. Yanofsky

Blurb for back of the book:
Category theory is at the center of many different fields of mathematics, physics, and computing. This book is a gentle introduction to this magical world. It starts from the basic definitions of category theory and progresses to cutting-edge research areas. Each concept and theorem is illustrated with many examples and exercises. The text also contains many mini-courses on various topics such as quantum computing, knot theory, categorical logic, quantum physics, etc. The book shows how category theory — and in particular monoidal category theory — describes and unifies various seemingly diverse fields.

Three sentence blurb:
Category theory is at the center of many different fields of mathematics, physics, and computing. Monoidal categories are structures that can mimic many different structures and processes. This text starts with a gentle introduction to the basic ideas of category theory and takes the reader through cutting-edge research areas.

Dedicated to....

*It is the harmony of the diverse parts, their symmetry, their happy balance; in a word it is all that introduces order, all that gives unity, that permits us to see clearly and to comprehend at once both the ensemble and the details.*

Henri Poincaré

Page 175 of [53]

# Preface

*... if we expand our experience into wilder and wilder regions of experience - every once in a while, we have these integrations when everything's pulled together into a unification, in which it turns out to be simpler than it looked before.*

Richard P. Feynman
Page 15 of [17]

Over the past few decades category theory has been used in many different areas of mathematics, physics, and computers. The applications of category theory have arisen in (to name just a few) quantum field theory, database theory, abstract algebra, formal language theory, quantum algebra, theoretical biology, knot theory, universal algebra, string theory, quantum computing, self-referential paradoxes, etc. This book will introduce the category theory necessary to understand large parts of these different areas.

Category theory studies categories, which are collections of structures and ways of changing those structures. Categories have been used to describe many different phenomena in mathematics and science. Our central focus will be monoidal categories, which are souped-up categories that allow one to describe even more phenomena. The theory of monoidal categories has emerged as a theory of structures and processes.

Category theory is a simple, extremely clear and concise language in which various fields of science can be discussed. It is also a unifying language. Different fields are expressed in this single language so that one can see common themes and properties. It also brings together different fields by actually establishing connections between them. Category theory is very good at showing the "big picture." Once this language is understood, one is capable of easily learning an immense amount of science, mathematics, and computing.

This book is an introduction to category theory. It begins with the basic definitions of category theory and takes the reader all the way up to cutting-edge research topics. Rather than going "down the rabbit hole" with a lot of very technical "pure" category theory, our central focus will be on examples and applications. In fact, an alternative title of this text could be *Category Theory By Example*. A major goal is to show the ubiquity of category theory and finally put an end to the silly canard that category theory is "general abstract nonsense." Another important goal is to show how many fields are related through category theory.

Within each chapter, whenever there is a definition or theorem, it is immediately followed by examples and exercises that clarify the categorical idea and makes it come alive. The fun is in the examples, and our examples are from many diverse disciplines.

This text contains fourteen self-contained mini-courses on various fields. They are short introductions to major fields such as quantum computing, self-referential paradoxes, quantum algebra, etc. These sections do not introduce new categorical ideas, rather, they use the category theory already presented to describe an entire field. The point we are making with these mini-courses is that with the language of category theory in your toolbox, you can master totally new and diverse fields with ease.

This book is different from other books on category theory. In contrast to the other books, we do not assume that the reader is already a mathematician, a physicist, or a computer scientist. Rather, this book is for anyone who wants to learn the wonders of category theory. We assume the reader is broad-minded and interested in many areas. We also assume that the reader wants to see how diverse areas are related to each other. The reader will not only learn all about category theory, she will also learn an immense amount of science, mathematics, and computers.

Another major difference between this book and other introductory category theory books is the way the book is organized. While in most books, the concept of category, functor, and natural transformation is introduced in the first few pages, here we slowly present each idea separately and in its correct time. This means that it takes us some time to get to the more interesting aspects of category theory. However, the reader will not be overwhelmed at the beginning. Structuring the book like this is pedagogically sound.

## Organization

The text commences with an introductory chapter that places category theory in historical and philosophical context. The introduction ends with a discussion of constructions on sets which will be useful for the rest of the text. Chapters 2, 3, and 4 are a simple introduction to category theory. Chapter 2 contains the basic definitions and properties of categories. Chapter 3 deals with special structures within a category. The real magic begins in Chapter 4 where we see how different categories relate to each other.

Chapters 5, 6, and 7 discuss monoidal categories. Chapter 5 describes monoidal categories, which are categories with extra structure. Chapter 6 deals with the relationships between monoidal categories. The core of the book is Chapter 7, where several variations of monoidal categories are presented with many of their properties and applications.

The final three chapters contain some advanced topics that use the various structures from Chapter 7. Many categorical ways of describing structures are explored in Chapter 8. Chapter 9 has a sampling of research areas in advanced category theory. We conclude with Chapter 10 which is a collection of more mini-courses from many different areas.

At the end of each chapter, there is a self-contained mini-course on a single topic. Every mini-course ends with several pointers to where you can learn more about the particular topic.

Appendix A contains Venn diagrams that describes the relationships of various structures used throughout the text.

Appendix B is an index of categories that appear in the text.

Appendix C is a guide to further study of category theory and its applications.

Appendix D has answers to selected exercises.

## Ancillaries

This text does not stand alone. I maintain a web page for the text at
*www.sci.brooklyn.cuny.edu/noson/mctext.*

The web page contains some answers to exercises not solved in Appendix D, links to lectures, and errata.

The reader is encouraged to send any and all corrections and suggestions to
*noson@sci.brooklyn.cuny.edu.*

Help me make this book better!

## Acknowledgment

This book would not exist without the help of many people.

Brooklyn College has been at the center of my life since I entered as a freshman in September of 1985. I am grateful for the intellectual environment that it provides. My colleagues and friends are always very helpful with a chat about an idea or editing a draft. In particular, I would like to thank David M. Arnow, Eva Cogan, James L. Cox, Scott Dexter, Lawrence Goetz, Jackie Jones, Keith Harrow, Yedidyah Langsam, Michael Mandel, Simon Parsons, Ira Rudowsky, Charles Schnabolk, Bridget Sheridan, Alexander Sverdlov, Joseph Thurm, Gerald Weiss, and Paula Whitlock. A large part of this book was written during a sabbatical. I am thankful to President Michelle J. Anderson, Dean Louise Hainline, and Chairman Yedidyah Langsam for making this possible.

This book owes much to Professor Rohit Parikh, who besides teaching me many things, taught me that “You can teach anything to anyone... You just have to teach them the prerequisites first.” How true! I am also in debt to Jamie Lennox for repeatedly reminding me that he is my target audience. I hope he is satisfied. Ralph Wojtowicz Suggested the mini-course on constructions with sets which is critical for the novice.

Rio Alvarado, Leo Caves, Ellen Cooper, Eugene Dorokhin, Vasili I. Galchin, Michael Goldenberg, Jonathan Hanon, Yigal Kamel, Deric Kwok, Moshe Lach, Armando Matos, Elina Nourmand, and Brian Porter were all very helpful in the editing process of this text.

# Contents

# Chapter 1

# Introduction

*The language of categories is affectionately known as* abstract nonsense, *so named by Norman Steenrod. This term is essentially accurate and not necessarily derogatory: categories refer to* nonsense *in the sense that they are all about the 'structure', and not about the 'meaning', of what they represent.*

Paolo Aluffi
Page 18 of [1]

We shall gently introduce you to the world of category theory. In Section 1.1, a little of the historical and philosophical context is provided. It is show how unification has always been a motivating factor at the center of category theory. Section 1.2 explains the motivation for monoidal categories. We then lay out the structure of the text in Section 1.3. Some standard notation is presented in Section 1.4. The chapter ends (Section 1.5) with a mini-course that uses constructs on sets and functions to learn many categorical ideas.

## 1.1 Categories

Category theory began with the intention of relating and unifying two different areas of study. The aim was to characterize and classify certain types of geometric objects by assigning to each of them certain types of algebraic objects as depicted in Figure 1.1. (In detail, the geometric objects are structures called topological spaces, manifolds, bundles, etc. The algebraic objects are called groups, rings, abelian groups, etc. The assignments have exotic names like homology, cohomology, homotopy and K-theory, etc.) Researchers realized that if they were going to relate geometric objects with algebraic objects they needed a language that is neither specialized to a geometric content nor an algebraic content. Only with such a general language can one discuss both fields.

Category theory was invented by Samuel Eilenberg and Saunders Mac Lane [16]. They described various souped-up collections of mathematical objects. Each collection was called a **category**. There was a collection of geometric objects and a collection of algebraic objects. They were interested in many different categories and in order to relate one category with another, they formulated the notion of a **functor** which — like a function — assigns to each entity in one category an entity in another category. They went further and formulated the notion of a **natural transformation** which is a way of relating one functor to another functor. (In a sense, a natural transformation *transfers* the results of one functor into the results of another functor.) These structures can be

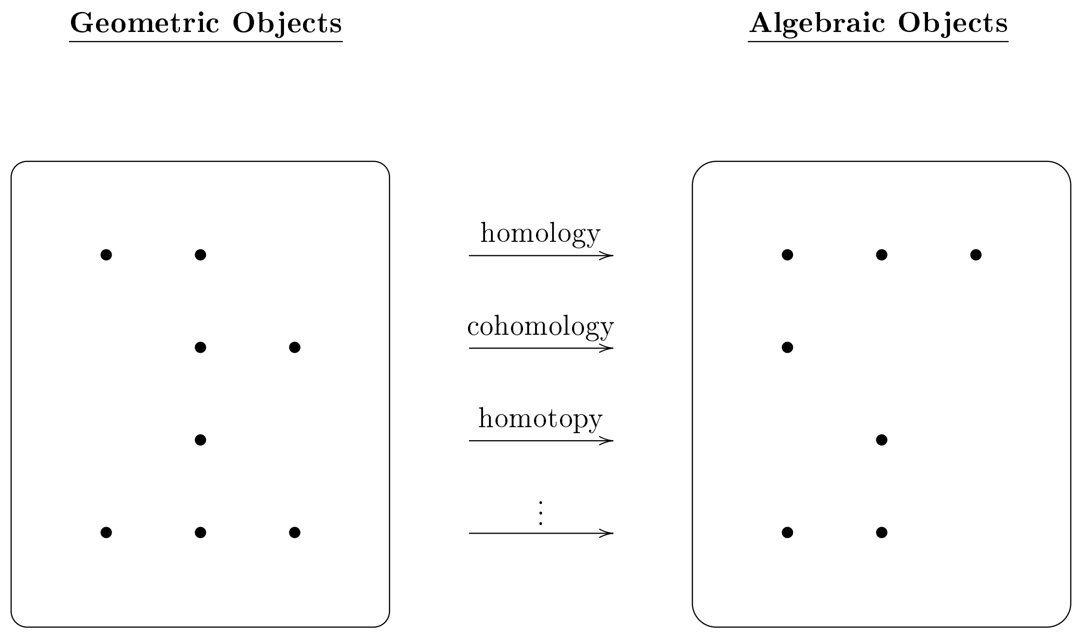
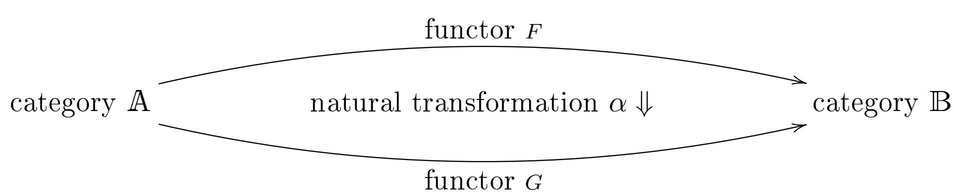



Figure 1.1: Relating geometric objects to algebraic objects.

visualized as




(1.1)

There is category $\mathbb{A}$ and category $\mathbb{B}$. Relating these categories are functor $F$ and functor $G$. And, finally, relating these functors is a natural transformation $\alpha$.

What is a category? It is not simply a collection of structures called **objects** of a particular type. Rather, a category is a collection of (i) objects and (ii) transformations or processes between the objects. We call these transformations or processes **morphisms** or **maps**. If $a$ and $b$ are objects and $f$ is a morphism from $a$ to $b$ we write it as $f\colon a \longrightarrow b$ or $a \xrightarrow{f} b$. We can visualize part of a category as Figure 1.2. The morphisms are to be thought of as ways of transforming objects. As time went on, the morphisms between objects took central stage. Category theory not only became the study of structures but also the study of transformations or processes between structures. One of the main properties of processes is that they can be combined. That is, one process followed by another process can be combined into a single process. In a category, if there is a morphism from object $a$ to object $b$ called $f$ and a morphism from object $b$ to object $c$ called $g$, then there will be an associated morphism from object $a$ to object $c$ written as $g \circ f$ and called "$g$ composed $f$," or "$g$

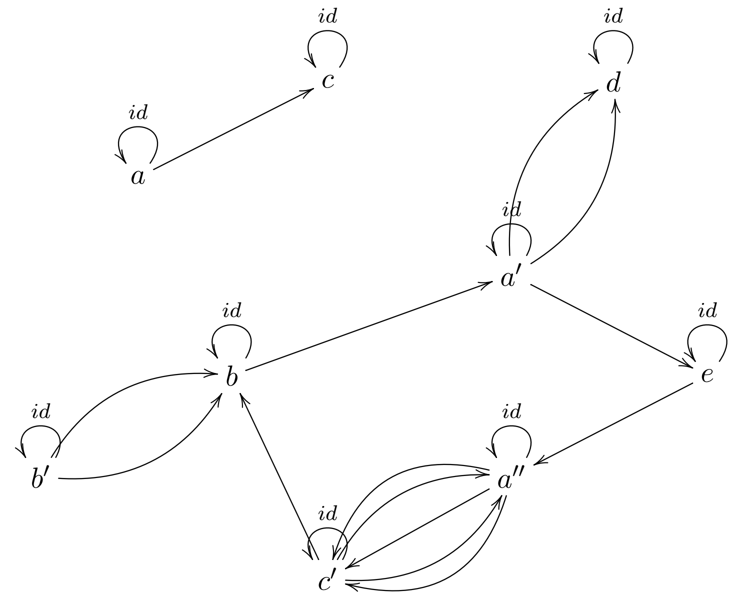
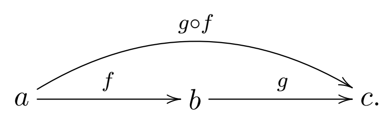



Figure 1.2: An example of part of a category.

following $f$," or "$g$ after $f$." This can be drawn as follows:




(1.2)

This is one of the defining properties of a category. We shall consider the others in a few pages.

Categories are related to more familiar structures called directed graphs. A directed graph is a structure that has objects (also called "vertices," "nodes," or "points") and morphisms (also called "arrows," or "directed edges") between them. One can view a category as a souped-up directed graph. Categories, like directed graphs, also have objects and morphisms, but within categories, one morphism after another can be composed. A directed graph is used to deal with various phenomena of interconnectivity. A category, with its extra structure, will deal with more sophisticated notions of interconnectivity (such as reachability.) A category can also be seen as a generalization of a group or a monoid. Group theory is the science of combining entities. A group is a set where one can combine elements to form other elements. In a category, one can combine morphisms that follow one another. This combining can be thought of as having one process follow another process. Graphs and groups are ubiquitous in modern science and mathematics. Categories — as generalizations of both structures — are even more pervasive.

Since categories are disassociated from any specific field or area, category theory received the reputation of being a language without content or "general abstract nonsense." However, it was precisely this independence from any field which gave category theory its power. By not being formulated for one particular field, it is capable of dealing with *any* field. At first, category theory was extremely successful in dealing with various fields of mathematics. As time went on, researchers realized that many branches of science which deal with structures or processes can be discussed in the language of category theory. Computer science is the study of computational processes, and hence, has taken a deep interest in category theory. More recently, category theory has been shown to be very adept at discussing structures and processes in physics. In the coming pages, we shall also see examples of structures and processes from chemistry, biology, and linguistics.

Many diverse fields are shown to be related because they are discussed in the single language of category theory. Researchers have found similar theorems and patterns in areas that were thought to be unrelated. Moreover, in the past few decades, category theory has further unified different fields by revealing amazing relationships between them. There are functors from a category in one field to a category in a totally different field that preserve properties and structures. These property preserving functors show that the two fields are similar. For example, quantum algebra is a field that uses categorical language to show how certain algebraic structures are related to geometric structures like knot theory. Another prominent example is quantum field theory, which is a branch of physics that uses functors to unite relativity and quantum theory. Quantum computing is a field that sits at the intersection of computer science, physics, and mathematics and can easily be understood using various categorical structures.

## 1.2 Monoidal Categories

In the early 1960s, Jean Bénabou and Saunders Mac Lane described categories that have more structure which we are called **monoidal categories** or **tensor categories** In these categories, one can "multiply" or "combine" objects. In symbols, within a monoidal category, object $a$ and object $b$ can be combined to form object $a \otimes b$ (read "$a$ tensor $b$"). As always in category theory, one is not only interested in combining objects but also in combining morphisms. With morphisms $f\colon a \longrightarrow a'$ and $g\colon b \longrightarrow b'$, there will be objects $a \otimes b$, $a' \otimes b'$, and there will also be a morphism $f \otimes g$ which we write as

$$\begin{array}{c} a \xrightarrow{\;f\;} a' \\ \otimes \\ b \xrightarrow[\;g\;]{} b' \end{array} \qquad \text{or} \qquad a \otimes b \xrightarrow{f \otimes g} a' \otimes b'. \tag{1.3}$$

Notice that there are two ways of combining morphisms in a monoidal category. There is $f \circ g$ and there is $f \otimes g$. In physics, the combination $f \circ g$ will correspond to performing one process after another while the combination $f \otimes g$ will correspond to performing two independent processes. In computers, the combination $f \circ g$ will correspond to sequential processes, while $f \otimes g$ will correspond to parallel processes. In mathematics, the interplay of the two combinations of morphisms will be very important.

Classical algebra is the branch of mathematics that deals with sets and operations on those sets. For sets of numbers and the addition operation, we have the rule that $x + y = y + x$ while in general, for subtraction, $x - y \neq y - x$. In the theory of monoidal categories there are rules that govern the relationship between $a \otimes b$ and $b \otimes a$. What about the relationship between $(a \otimes b) \otimes c$ and $a \otimes (b \otimes c)$? Within monoidal categories there are many more possible relationships when dealing with these combined objects. For each rule relating these operations, there will be a corresponding type of monoidal category. In Chapter 7, we will see many different types of monoidal categories. This variability allows for many phenomena to be described by monoidal categories. The area that deals with the different types of rules among operations is called "coherence theory" (i.e., how the various operations *cohere* with each other) or "higher-dimensional algebra." This area of study has become pervasive, and it is believed that higher-dimensional algebra will arise even more frequently in the science and mathematics of the coming decades.

## 1.3 The Examples and the Mini-courses

This text is centered on the examples. Our goal is to show the pervasiveness of categories, and in particular, monoidal categories. We also want to emphasize how categories can reveal the interconnectedness of various fields. We do this by introducing lots of examples from many different areas. Immediately following a definition or a theorem of category theory there are lots of examples that illustrate the idea. There are also some examples that are left to the reader as exercises. It is important to realize that although this book is chock-full of examples, we have barely scratched the surface. The literature of category theory has many more examples. We chose the examples that arise most frequently or are the easiest to understand. The reader will be directed to places in the literature where other examples are described. We are showing the beauty of category theory but only revealing the tip of the iceberg.

Most of the examples can be loosely split into three broad groupings: mathematics, physics, and computers. There will also be examples from fields like chemistry, biology, and linguistics. The problem is that the boundaries between these different areas are hazy. For example, is quantum computing part of computer science, physics or abstract mathematics? Is knot theory part of mathematics or physics? There are no firm boundaries.

Since most readers are familiar with sets and functions between sets, we usually try to first show an idea or definition in terms of sets. In later chapters, it will become apparent that sets and functions between sets are not the right context to examine certain phenomena. This is where category theory really gets interesting.

The examples are spread throughout the book. To illustrate, in Chapter 2, a category be introduced. In Chapter 3, some properties of this category will be described. This category will be related to other categories in Chapter 4. In Chapter 5 we will show that the category has a monoidal structure, and we will see how that monoidal structure relates to the monoidal structure of other categories in Chapter 6. This same category and variations of this category will be shown to have even more structure in Chapter 7. We will also see how this category arises in various mini-courses. By the time the reader finishes the book, the category will be an old friend.

Not all categories are introduced early on. In order not to overwhelm the reader in the beginning, we will introduce many categories in later chapters as well. Our aim is readability.

These examples will take the reader rather far. In mathematics, the reader will meet lots of algebra and topology. In physics we will see the basics of quantum theory as well as some more modern physical theories. In computers we will see how categories are good for describing certain models of computation and some advanced logic.

Due to space limitations and by concentrating on examples, we are going to omit some results in pure category theory. We only describe the category theory required to understand the examples. In Appendix C, we point out various places where one can learn more (pure) category theory.

Category theory is a language that can deal with many different areas of science. The really fun part of category theory is that once one has this language in their toolbox, one can easily pick up whole new branches of science. We show this flexibility with little mini-courses. At the end of every chapter is a little self-contained section that describes a whole field with the category theory already learned. Mini-courses in later chapters depend on the knowledge of earlier mini-courses. In Chapter 10, we offer several other mini-courses.

It must be noted that this is not a history book. We are not going to say who thought of some particular construction or example first. Some of the examples in this book came from other books and papers. Some examples are just known in the folklore of category theory. And some examples, we made up. The history is too complicated for us to disentangle and is of absolutely no pedagogical use to the novice. We will name some places to learn about the history of category theory in Appendix C.

This book owes a tremendous debt to previous works.

- I "cut my teeth" learning category theory from Saunders Mac Lane's *Categories for the Working Mathematician* [42]. This is *the* classic text by one of the founders of category theory. It influenced my thinking and this book in the most profound way. As the title implies, Mac Lane assumed that the reader knows large parts of mathematics before opening his book. My goal with this book is to give over the beauty of category theory as Mac Lane did but for a wider audience.
- John Baez and Michael Stay wrote a wonderful paper "Physics, Topology, Logic and Computation: A Rosetta Stone" [6] that highlights connections between many different fields. I would like to think of this book as an explanation and an expansion of that paper.
- I learned much from Christian Kassel's text book *Quantum Groups* [30]. His clarity and exactness is an inspiration.
- This book attempts to be as readable as Michael Barr and Charles Wells' textbook *Category Theory for Computing Science* [8]. Their work goes through large segments of category theory with many examples along the way. We try to do the same.

There are many potential topics that could have gone into this book. Painful choices had to be made. In the end, topics were chosen based on a desire to provide as diverse a set of examples as possible to satisfy a broad readership across disciplines, with a natural bias to those areas which the author feels more confident to address. I would like to believe that the topics chosen will be important as we march into the unfathomable future.

Finally, I would like to apologize to all my friends in the category theory community if I neglected their favorite example or did not discuss an area in which they did great work. It was not my intention to omit anyone's work.

## 1.4 Notation

In order to improve readability, for the most part, we keep to the following notation.

- Categories are in boldface: $\mathbb{A}, \mathbb{B}, \mathbb{C}, \mathbb{D}, \mathbb{Circuit}, \mathbb{Set}, \ldots$
- Objects in general categories are the first few lowercase Latin letters: $a, b, c, d, a', b', a'' \ldots$
- Morphisms in general categories are lowercase Latin letters: $f, g, h, i, j, k, f', g'', \ldots$
- Functors are capital Latin letters: $F, G, H, I, J, \ldots$
- Natural transformations are lowercase Greek letters: $\alpha, \beta, \gamma, \delta, \eta, \kappa, \ldots$
- Higher dimensional morphisms will be capital Greek letters: $\Gamma, \Delta, \Theta, \Phi, \Psi, \ldots$
- Sets of numbers are $\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}, \mathbf{C}$.
- 2-Categories are in boldface with a line above: $\overline{\mathbb{A}}, \overline{\mathbb{B}}, \overline{\mathbb{C}}, \overline{\mathbb{D}}, \overline{\mathbb{Cat}} \ldots$

- 3-Categories are in boldface with a two lines above: $\overline{\overline{\mathbb{A}}}, \overline{\overline{\mathbb{B}}}, \overline{\overline{\mathbb{C}}}, \overline{\overline{\mathbb{D}}}, \overline{\overline{2\mathbb{C}\mathbb{at}}} \ldots$
- Bicategories are in boldface with a tilde above: $\widetilde{\mathbb{A}}, \widetilde{\mathbb{B}}, \widetilde{\mathbb{C}}, \widetilde{\mathbb{D}}, \widetilde{\mathbb{C}\mathbb{at}} \ldots$

There are many different types of arrows in this book.

- Morphism, map or functor: $\longrightarrow$
- The input and output of a function or a functor: $\longmapsto$
- Inclusion in injection: $\hookrightarrow$
- Surjection or full functor: $\twoheadrightarrow$
- Natural transformation: $\Longrightarrow$

## 1.5 Mini-course: Sets and Categorical Thinking

Category theory is not just a language that is capable of describing an immense amount of science and mathematics. Rather, it is a *new and innovative way of thinking*. One of the central ideas in category theory is the following.

**Important Categorical Idea 1.5.1.** Properties and structures in a category can be described by the morphisms of the category. That is, the objects do not stand alone. One must see how the objects relate to each other with morphisms. The objects have to be seen in context of the morphisms. ◯

In order to get a feel for this, we take an in-depth look at the familiar world of sets and functions between sets. We show that many of the usual ideas and constructions about sets can be described with functions between sets. This mini-course will also be a gentle reminder of many concepts that are needed in the rest of the text. Throughout this book, we will point back to equations, diagrams, and ideas found in this section.

### Sets and Operations

A set is a collection of elements. If $S$ is a set and $x$ is an element of $S$, we write $x \in S$. If $x$ is not an element of $S$ we write $x \notin S$. We will deal with both infinite sets and finite sets. Some of the most important infinite sets of numbers are

- The natural numbers, $\mathbf{N} = \{0, 1, 2, 3, \ldots\}$.
- The integers, $\mathbf{Z} = \{\ldots, -3, -2, -1, 0, 1, 2, 3, \ldots\}$.
- The rational numbers, $\mathbf{Q} = \{\frac{m}{n} : m \text{ and } n \text{ in } \mathbf{Z} \text{ and } n \neq 0\}$.
- The real numbers, $\mathbf{R}$, that is, all numbers on the real number line.
- The complex numbers, $\mathbf{C} = \{a + bi : a \text{ and } b \text{ in } \mathbf{R}\}$.

The most interesting property about a finite set is the number of elements in the set. For every set $S$, we write $|S|$ to denote the number of elements in $S$.

We begin by discussing several operations on sets. Let $S$ and $T$ be sets. If $s$ is in $S$ and $t$ is in $T$ we write an **ordered pair** of the elements as $(s, t)$. The set of all ordered pairs is called the **Cartesian product** of $S$ and $T$

$$S \times T = \{(s, t) \colon s \in S, t \in T\}. \tag{1.4}$$

**Example 1.5.2.** If *Pants* = {black, blue1, blue2, gray} is the set of pants that you own, and *Shirts* = {white, blue, orange} is the set of shirts that you own, then the set of *Outfits* is

$$Pants \times Shirts = \left\{ \begin{array}{c} \text{(black, white), (black, blue), (black, orange),} \\ \text{(blue1, white), (blue1, blue), (blue1, orange),} \\ \text{(blue2, white), (blue2, blue), (blue2, orange),} \\ \text{(gray, white), (gray, blue), (gray, orange)} \end{array} \right\} \tag{1.5}$$

(True scholarly category theorists do not care if their clothes fail to match!) □

**Technical Point 1.5.3.** One must realize that the most important aspect of an ordered pair is its order. In contrast, sets are just collections, and as such, do not have a preferred order. The set $\{s, t\}$ is considered to be the same set as $\{t, s\}$. In contrast, the pair $(s, t)$ is not considered the same as $(t, s)$. Hence, we cannot simply use two curly brackets to describe ordered pairs. There are other ways of describing an ordered pair of elements from $S$ and $T$. For example, we could write them as $\langle s, t\rangle$ or $\{s, t, \{s\}\}$ (where we collect the elements but indicate the first element by putting it into a set by itself), or even $\{s, t, \{t\}\}$ (where we indicate the second element by putting it into a set by itself). There is nothing special about the notation $(s, t)$. (We will see this again in the beginning of Section 3.1) ♡

If there are $m$ elements in $S$ and $n$ elements in $T$ then there are $mn$ elements in $S \times T$. In symbols we write this as

$$|S \times T| = |S| \cdot |T|. \tag{1.6}$$

**Exercise 1.5.4.** How many elements are in *Pants*? How many elements are in *Shirts*? How many elements in *Outfits*? Show that the above formula works.

**Solution:** 4, 3, $12 = 4 \cdot 3$.

■

We can generalize the notion of ordered pairs to **ordered triples**, **ordered 4-tuples**, **ordered 5-tuples**, etc. If there are $n$ sets, $S_1, S_2, \ldots, S_n$, then an **ordered n-tuple** is written as $(s_1, s_2, \ldots, s_n)$ where $s_i$ is in $S_i$. The set of all $n$-tuples is $S_1 \times S_2 \times \cdots \times S_n$. The number of $n$-tuples follows a generalization of Equation (1.6):

$$|S_1 \times S_2 \times \cdots \times S_n| = |S_1| \cdot |S_2| \cdots |S_n|. \tag{1.7}$$

**Exercise 1.5.5.** In addition to pants and shirts, an outfit might consist of a hat, socks, and shoes. How many outfits are there if there are $m$ hats, $n$ pairs of socks, $p$ pairs of shoes, $q$ pants and $r$ shirts?

**Solution:** $m \cdot n \cdot p \cdot q \ \cdot r$.

■

Another operation performed on sets is the union. Let $S$ and $T$ be sets. The **union** of $S$ and $T$ is the set $S \bigcup T$ which contains those elements that are in $S$ or in $T$.

$$S \bigcup T = \{x \colon x \in S \text{ or } x \in T\}. \tag{1.8}$$

It is important to notice that if there is some element that is in both $S$ and in $T$ then it will occur only once in $S \bigcup T$. This is because when dealing with a set, repetition does not matter. The set $\{a, b, c, b\}$ is considered the same set as $\{a, b, c\}$.

A related operation is the disjoint union. Given sets $S$ and $T$, one forms the **disjoint union** $S \coprod T$ which contains the elements from $S$ and $T$ but considers elements that are in both sets as different elements. One way this is done is by tagging every element with extra information that says which set it comes from. This way an element that is both in $S$ and in $T$ would be considered two different elements. For example, if $S = \{a, b, c, x, y\}$ and $T = \{q, w, b, x, e, r\}$, then

$$S \coprod T = \{(a, 0), (b, 0), (c, 0), (x, 0), (y, 0), (q, 1), (w, 1), (b, 1), (x, 1), (e, 1), (r, 1)\} \tag{1.9}$$

where the elements of $S$ are tagged with a 0 and the elements of $T$ are tagged with a 1. In general for sets $S$ and $T$, we have

$$S \coprod T = (S \times \{0\}) \bigcup (T \times \{1\}) \tag{1.10}$$

The formula for the number of elements in the disjoint union is $|S \coprod T| = |S| + |T|$.

**Exercise 1.5.6.** When does the union of two sets have the same number of elements as the disjoint union of those same sets?

**Solution:** When the two sets have nothing in common, i.e., when the intersection of the two sets is empty.

■

### Functions

**Important Categorical Idea 1.5.7.** In category theory, whenever we have a notion (for example, a set), the immediate next task is to consider how these notions relate to each other (for example, functions are ways for sets to relate to each other.) As we have stated, category theory is not about "things," but about how "things" relate to other "things." Relations between objects are usually described by morphisms or functions between the objects.

Following this rule, once we describe the morphisms between the objects we must immediately ask what is between the morphisms. Usually the answer will be other morphisms. The computer scientist might protest that this recursive procedure is going to get us into an infinite loop. It will! We will see this in Section 9.5 when we describe infinite levels of morphisms in higher-dimensional category theory. Such structures are a consequence of this important idea that is at the center of category theory. ○

The central idea of this mini-course is the notion of functions between sets and how they determine properties of sets.

**Definition 1.5.8.** Let $S$ and $T$ be sets. A **function**, $f$, from $S$ to $T$, written $f: S \longrightarrow T$ is a rule that assigns to every element of $S$ an element of $T$. The value of $f$ on the element $s$ is written as $f(s)$. If $f(s) = t$ we write $s \mapsto t$.

It is important to understand the difference between the symbol $\longrightarrow$ and the symbol $\mapsto$. The symbol $\longrightarrow$ goes between two sets. It describes a function from one set to another. In contrast, the symbol $\mapsto$ goes from an element in the first set to an element in the second set. It describes how the function is defined.

**Example 1.5.9.** For every set $S$, there is an **identity function** $id_S: S \longrightarrow S$ which takes every element to itself. In symbols it is defined as $id_S(s) = s$ or $s \mapsto s$. □

Functions can be used as a way of describing or choosing elements of a set. Consider a one-element set, $\{*\}$. (There are many one-element sets such as $\{a\}$, $\{b\}$ , $\{\text{Bill}\}$, etc.) For a set $S$, a function $f\colon \{*\} \longrightarrow S$ picks out one element of $S$. The single element $*$ goes to the selected element $s$ in $S$. In symbols, $f(*) = s$ or $* \mapsto s$.

**Example 1.5.10.** Let $S$ be the set {Jack, Jill, Joane, June, Joe, John}. The element Joe in $S$ can be described as a function $f\colon \{*\} \longrightarrow S$ where $f(*) = \text{Joe}$. We might want to distinguish this function by calling it $f_{\text{Joe}}\colon \{*\} \longrightarrow S$. There will be other functions like $f_{\text{Jill}}\colon \{*\} \longrightarrow S$ where $f_{\text{Jill}}(*) = \text{Jill}$. For this set of six elements, there are six different functions from $\{*\}$ to $S$. □

If we were interested in choosing two elements of $S$, we can look at functions from a two element set to $S$. So $f\colon \{0,1\} \longrightarrow S$ will choose two elements of $S$. The first element is $f(0)$ and the second element is $f(1)$. If $f(0) \neq f(1)$ then $f$ will choose two *different* elements of $S$. Every such function chooses two elements of $S$. Functions from $\{a, b, c\}$ to $S$ will choose three elements of $S$. This can go on: if we wanted to choose $n$ elements of a set, we would look at functions of the form

$$\{1, 2, \ldots, n\} \longrightarrow S. \tag{1.11}$$

**Definition 1.5.11.** A set $T$ is a **subset** of $S$ if every element of $T$ is an element of $S$. We write this as $T \subseteq S$. If $T$ is a subset of $S$ but not equal to $S$, we call $T$ a **proper subset** and write $T \subsetneq S$ or $T \subset S$. If $T$ is a subset of $S$, there is an **inclusion function** that takes every element of $T$ to its corresponding element of $S$ which is written as $inc\colon T \hookrightarrow S$. There is a special set that has no elements. It is called the **empty set** and is denoted $\emptyset$. The empty set is a subset of every set. Furthermore, for every set $S$, there is a unique function from the empty set to $S$. This function is denoted $!\colon \emptyset \longrightarrow S$.

**Definition 1.5.12.** For a set $S$, the set of all subsets of $S$ is called the **powerset** of $S$ and is denoted $\mathcal{P}(S)$.

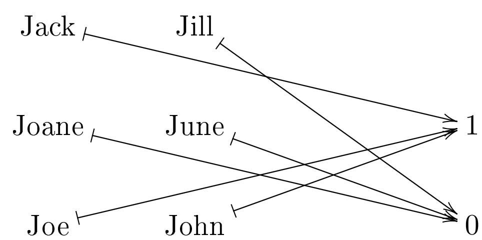
**Example 1.5.13.** For the set $\{a\}$, the powerset is $\mathcal{P}(\{a\}) = \{\emptyset, \{a\}\}$. The powerset of a two element set $\{a, b\}$ is $\mathcal{P}(\{a, b\}) = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. The powerset of a three element set $\{a, b, c\}$ is $\mathcal{P}(\{a, b, c\}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$. Whenever we add an element to a set, we double the number of elements in the powerset. There is the following rule: if $S$ has $n$ elements, then $\mathcal{P}(S)$ has $2^n$ elements. In symbols, $|S| = n$ implies $|\mathcal{P}(S)| = 2^n$. We can also write this as $|\mathcal{P}(S)| = 2^{|S|}$. □

Functions can be used to describe subsets. For any set $S$ and subset $T \subseteq S$ there is an associated **characteristic function** $\chi_T\colon S \longrightarrow \{0,1\}$ which assigns each element $s$ of set $S$ to 1 if $s$ is in $T$, and to 0 if $s$ is not in $T$, i.e.,

$$\chi_T(s) = \begin{cases} 1 & : s \in T \\ 0 & : s \notin T. \end{cases} \tag{1.12}$$

(The Greek letter $\chi$ is pronounced 'chi" and should remind you of the first syllable of "characteristic".) The function $\chi_T$ tells which elements of $S$ are in $T$ and which elements are not in $T$.

**Example 1.5.14.** Let $S$ be the set {Jack, Jill, Joane, June, Joe, John}. Consider the subset $T = \{\text{Jack, Joe, John}\}$ of $S$ that contains all the boys in $S$. This subset can be described by the function $\chi_T \colon S \longrightarrow \{0,1\}$ which can be visualized as




(1.13)

□

**Exercise 1.5.15.** For the sets of numbers, we know that $\mathbf{N} \subsetneq \mathbf{Z} \subsetneq \mathbf{Q} \subsetneq \mathbf{R} \subsetneq \mathbf{C}$. Give the characteristic function for each of these proper subsets.

**Solution:** Let us just focus on the subset $\mathbf{Q} \subsetneq \mathbf{R}$. The characteristic function $\chi_{\mathbf{Q}} \colon \mathbf{R} \longrightarrow \{0,1\}$ is defined as follows:

$$\chi_{\mathbf{Q}}(r) = \begin{cases} 1 & : r \text{ is rational} \\ 0 & : r \text{ is not rational.} \end{cases} \tag{1.14}$$

The others are done similarly.

■

A characteristic function assigns the elements of $S$ to one of two possible values. There might be a need to assign one of many values to every element of $S$. For example, a function $S \longrightarrow \{a,b,c,d\}$ assigns to every element of $S$ one of these letters which can stand for different ideas. In general, a function

$$S \longrightarrow \{1,2,\ldots,n\} \tag{1.15}$$

assigns every element of $S$ one of $n$ numbers. We can also assign to every element of $S$ an element of $[0,1]$, the real interval between 0 and 1. This can correspond to assigning a probability to every element.

**Example 1.5.16.** In school, every student usually has an associated grade point average (GPA). This is written as a function $\mathit{Students} \longrightarrow [0,4]$. □

If $S$ is a set, then there is a function called the **diagonal function** $\Delta \colon S \longrightarrow S \times S$ which takes every element to an ordered pair of the same element. In symbols, for $s$ in $S$ we have

$$\Delta(s) = (s,s). \tag{1.16}$$

If $f \colon S \longrightarrow S'$ and $g \colon T \longrightarrow T'$ are functions then there exists a function $f \times g \colon S \times T \longrightarrow S' \times T'$ that takes an ordered pair of elements and applies $f$ to the first element and $g$ to the second. In symbols, the function is defined for elements $s$ of $S$ and $t$ of $T$ as

$$(f \times g)((s,t)) = (f(s), g(t)) \in S' \times T'. \tag{1.17}$$

In a sense, this process is a parallel process. The function $f$ processes $s$ while the function $g$ processes $t$.

**Exercise 1.5.17.** Let $f \colon \mathbf{N} \longrightarrow \mathbf{R}$ be defined by $f(n) = \sqrt{n}$ and $g \colon \mathbf{R} \longrightarrow \mathbf{Z}$ be the ceiling function denoted as $g(r) = \ulcorner r \urcorner$. What is $(f \times g)((5, -5.1))$

**Solution:** $(\sqrt{5}, -5)$.

■

**Definition 1.5.18.** There are some special types of functions. We say $f\colon S \longrightarrow T$ is

- **one-to-one** or **injective** if different elements in $S$ go to different elements in $T$. That is, for all $s$ and $s'$ in $S$, if $s \neq s'$ then $f(s) \neq f(s')$. Another way to say this is that if $f(s) = f(s')$, then it must be that $s = s'$. This means that if the function takes elements to the same output, the elements must have started off equal.
- **onto** or **surjective** if for every element $t$ in $T$, there is an $s$ in $S$ such that $f(s) = t$.
- an **isomorphism** or a **one-to-one correspondence** or a **bijection** if $f$ is one-to-one and onto. That is, for every element $s$ of $S$ there is a unique element $t$ of $T$ so that $f(s) = t$ and for every element $t$ of $T$ there is a unique element $s$ of $S$ so that $f(s) = t$.

**Definition 1.5.19.** Sets $S$ and $T$ are **isomorphic** if there is an isomorphism between them. We write this as $S \cong T$.

Isomorphism of sets is not the same idea as equality of sets. Consider a simple example: the set $\{x\}$ and the set $\{y\}$ — although these two sets have exactly the same properties — they are not equal. They are only isomorphic.

**Exercise 1.5.20.** Explain why two finite sets that have the same number of elements are isomorphic. ■

**Exercise 1.5.21.** Show that the Cartesian plane, $\mathbf{R} \times \mathbf{R}$, is isomorphic to plain of complex numbers, $\mathbf{C}$.

**Solution:** The isomorphism will take a pair of real number $(r_1, r_2) \in \mathbf{R} \times \mathbf{R}$ to the complex number $r_1 + ir_2 \in \mathbf{C}$ where $i = \sqrt{-1}$.

■

One of the central ideas about sets is that given sets $S$ and $T$ we can form a set which consists of all functions from $S$ to $T$. We denote this **set of functions** or **function set** as $Hom(S, T)$ or $T^S$.

**Exercise 1.5.22.** Write down the set of all the functions from the set $\{a, b, c\}$ to the set $\{0, 1\}$.

**Solution:** Each of the following lines is a function.

| | | |
|---|---|---|
| $f(a) = 0$ | $f(b) = 0$ | $f(c) = 0$ |
| $f(a) = 0$ | $f(b) = 0$ | $f(c) = 1$ |
| $f(a) = 0$ | $f(b) = 1$ | $f(c) = 0$ |
| $f(a) = 0$ | $f(b) = 1$ | $f(c) = 1$ |
| $f(a) = 1$ | $f(b) = 0$ | $f(c) = 0$ |
| $f(a) = 1$ | $f(b) = 0$ | $f(c) = 1$ |
| $f(a) = 1$ | $f(b) = 1$ | $f(c) = 0$ |
| $f(a) = 1$ | $f(b) = 1$ | $f(c) = 1$ |

■

We saw that every element in a set $S$ can be described as a function $\{*\} \longrightarrow S$. This correspondence between elements of $S$ and functions from $\{*\}$ to $S$ shows that

$$S \cong S^{\{*\}} = Hom(\{*\}, S). \tag{1.18}$$

Using characteristic functions, we saw that there is a correspondence between subsets of $S$ and functions from $S$ to $\{0, 1\}$. This correspondence can be stated as

$$\mathcal{P}(S) \cong \{0, 1\}^S = Hom(S, \{0, 1\}). \tag{1.19}$$

We will denote the set $\{0, 1\}$ as 2 and then write this as

$$\mathcal{P}(S) \cong 2^S = Hom(S, 2). \tag{1.20}$$

**Example 1.5.23.** Consider the simple binary addition operation $+ : \mathbf{N} \times \mathbf{N} \longrightarrow \mathbf{N}$. Let us write this function with its inputs clearly marked as follows

$$(\ \ ) + (\ \ ) : \mathbf{N} \times \mathbf{N} \longrightarrow \mathbf{N}. \tag{1.21}$$

Now consider the function $(\ \ ) + 5 : \mathbf{N} \longrightarrow \mathbf{N}$. This is a function with only one input. We could also make another function of one variable $(\ \ ) + 7 : \mathbf{N} \longrightarrow \mathbf{N}$. In fact we can do this for any natural number. We can define a function that inputs a natural number and outputs a function from natural numbers to natural numbers. That is, there is a function $\Phi : \mathbf{N} \longrightarrow Hom(\mathbf{N}, \mathbf{N})$ which is defined as $\Phi(n) = (\ \ ) + n$. It is easy to see that the assignment described by the $(\ \ ) + (\ \ )$ function is the same as the assignment described by the $\Phi$ function.

Notice that what we said about $+$ really applies to every function with two inputs. If $f : T \times U \longrightarrow S$ is a function from $T \times U$, then for every $u \in U$ there is a function $f(\ \ , u) : T \longrightarrow S$. This shows that there is a function $f' : U \longrightarrow Hom(T, S)$. It is easy to see that the assignment described by any function $f$ is the same as the assignment described by the function $f'$. □

This example brings to light the following important theorem about sets.

**Theorem 1.5.24.** For sets $S$, $T$ and $U$ there is an isomorphism

$$Hom(T \times U, S) \cong Hom(U, Hom(T, S)) \qquad \text{or} \qquad S^{T\times U} \cong (S^T)^U \tag{1.22}$$

★

---

**Proof.** To show that these two sets are isomorphic, consider $f\colon T \times U \longrightarrow S$. From this function let us define an $f'\colon U \longrightarrow Hom(T, S)$. For a $u$ in $U$ we have the function $f'(u)\colon T \longrightarrow S$ which is defined as follows: for $t$ in $T$, set $f'(u)(t) = f(t, u)$. This function has the same information as $f$. It is not hard to go from $f'$ back to $f$. ♣

Let us count how many functions there are between two finite sets. Consider $S$ with $|S| = m$ and $T$ with $|T| = n$ Consider a function $f\colon S \longrightarrow T$. For each element $s$ in $S$ there are $n$ possible values of $f(s)$ in $T$. For two elements in $S$ there are $n \cdot n$ possibilities of choices in $T$. In total, there are $n \cdot n \cdot \cdots \cdot n$ ($m$ times) possible maps. So

$$|Hom(S, T)| = |T^S| = n^m = |T|^{|S|}. \tag{1.23}$$

For three sets $S$, $T$, and $U$, we have

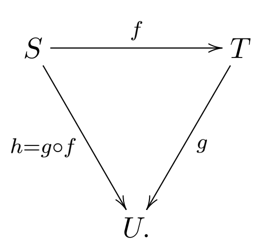
$$\begin{aligned} |Hom(S \times T, U)| = |U^{S\times T}| = |U|^{|S\times T|} = |U|^{|S|\cdot|T|} = (|U|^{|S|})^{|T|} \\ = |Hom(S, U)|^{|T|} = |Hom(T, Hom(S, U))|. \end{aligned} \tag{1.24}$$

We see that the rule about exponentials usually learned in elementary school, $m^{(n\cdot p)} = (m^n)^p$ is expanded to a rule about sets and functions.

## Operations on Functions

There are many times that we are going to take two functions and perform an operation to get a third function. Three such operations are composition, extension and lifting. Remarkably, many ideas about functions can be understood as operations in one of these three forms.

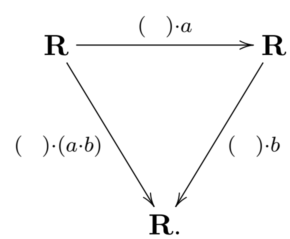
The simplest operation is **composition**. If there is a function $f\colon S \longrightarrow T$ and a function $g\colon T \longrightarrow U$, then the composition of them is a function $h = g \circ f\colon S \longrightarrow U$ that is defined on an $s$ in $S$ as $h(s) = g(f(s))$. We write these functions as




(1.25)

We say that $f$ and $g$ are **factors** of $h$ or $h$ factors through $T$. This diagram is a called a **commutative diagram**. It means that if you start with any element $s$ in $S$ and you apply the functions to go from $S$ to $U$, you will get to the same resulting element in $U$. In detail, for all $s$, we have that $g(f(s)) = h(s)$. There will be many such diagrams in the coming pages. In all the cases, start from any set and follow all the paths of composible functions to another set, and you will get the same element. Throughout this text, unless otherwise stated, all diagrams are commutative.

**Example 1.5.25.** Consider the set of real numbers, $\mathbf{R}$. Let $a$ and $b$ be any real numbers. Consider the function $(\ \ )\cdot a\colon \mathbf{R} \longrightarrow \mathbf{R}$ that takes any real number and multiplies it by $a$. There is also a function $(\ \ )\cdot b\colon \mathbf{R} \longrightarrow \mathbf{R}$ that takes any real number and multiplies it by $b$. The composition of these two functions is the function $(\ \ )\cdot(a\cdot b)\colon \mathbf{R} \longrightarrow \mathbf{R}$ that takes any real number and multiplies it by $a \cdot b$ as in the following commutative diagram




(1.26)

One can make similar compositions with other arithmetic operations. □

**Exercise 1.5.26.** Show that function composition is associative. That is, let $f\colon S \longrightarrow T$, $g\colon T \longrightarrow U$ and $h\colon U \longrightarrow V$, show that $h \circ (g \circ f) = (h \circ g) \circ f$.

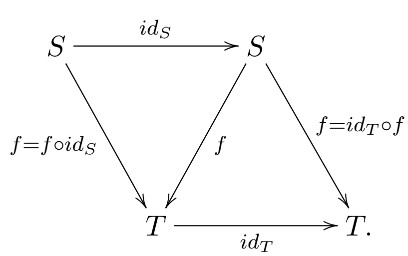
**Solution:** The fact that function composition is associative will be used many times throughout this text. There are two ways of associating the functions: $h \circ (g \circ f)$ and $(h \circ g) \circ f$. We claim that these two function are the same. Both functions on input $s$ of $S$ have the value $h(g(f(s)))$.

■

When dealing with the identity function, the input is the same as the output. This has an interesting consequence when dealing with composition. If you compose a function with an identity map, then you do not change the function. In detail, for $f\colon S \longrightarrow T$, $id_S\colon S \longrightarrow S$, and $id_T\colon T \longrightarrow T$, we have

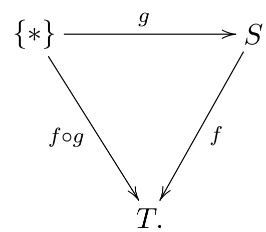
$$f \circ id_S = f \qquad \text{and} \qquad id_T \circ f = f. \tag{1.27}$$

We can see these equations as the following commutative diagram:




(1.28)

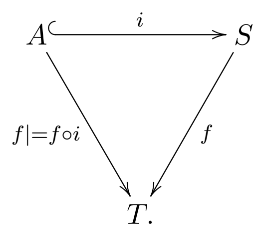
**Example 1.5.27.** Evaluation of a function can be seen as composition. Let $f\colon S \longrightarrow T$ be a function and let an element be described by the function $g\colon \{*\} \longrightarrow S$. Then the value of $f$ on the element that $g$ chooses is the element that $f \circ g$ chooses as in




(1.29)

$f \circ g \colon \{*\} \longrightarrow T$ picks out the output of $f$. If $g$ performs the assignment $* \mapsto s$, then $f \circ g$ performs the assignment $* \mapsto f(s)$. □

**Example 1.5.28.** If $f \colon S \longrightarrow T$ and $A$ is a subset of $S$ with inclusion function $i \colon A \hookrightarrow S$, then the restriction of $f$ to $A$ is the function $f| \colon A \longrightarrow T$ which is given as the composition



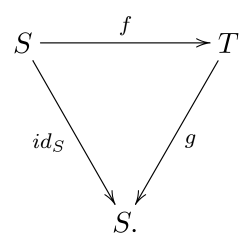

(1.30)

□

---

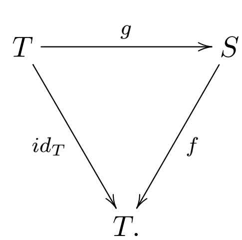
**Theorem 1.5.29.** The three properties of functions that we saw in Definition 1.5.18 can be described with function composition. $f \colon S \longrightarrow T$ is

- **one-to-one** if and only if there exists a $g \colon T \longrightarrow S$ such that $g \circ f = id_S$.




(1.31)

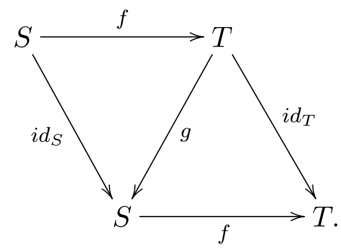
  (Proof: The existence of a $g$ implies $f$ is one-to-one. If $f(s) = f(s')$ then apply $g$ to both sides of the equation and get $g(f(s)) = g(f(s'))$. But $g \circ f = id_S$ implies $s = s'$.
  If $f$ is one-to one, then there exists a $g$. Let $g(t)$ be the unique $s$ such that $f(s) = t$. If $t$ is not an output of $f$ then it does not matter what value you give to $g(t)$.)
- **onto** if and only if there exists a $g \colon T \longrightarrow S$ such that $f \circ g = id_T$.




(1.32)

(Proof: The existence of a $g$ implies $f$ is onto. The function $f$ is onto because for any $t$, $g(t) = s$ for some $s \in S$ and $f(s) = f(g(t)) = id_T(t) = t$.
Onto implies the existence of $g$. Let $g(t)$ equal any $s$ such that $f(s) = t$. There must be one because $f$ is onto. )

- **one-to-one correspondence** if and only if there exists a $g\colon T \longrightarrow S$ such that $g \circ f = id_S$ and $f \circ g = id_T$. Or putting the previous two triangles together, we have

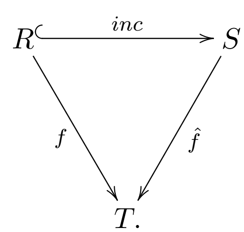



(1.33)

★

---

A second operation of functions is an **extension**. In detail, if $f\colon R \longrightarrow T$ is a function and $R$ is a subset of $S$ with the inclusion function $inc\colon R \hookrightarrow S$, then an extension of $f$ along $inc$ is a function $\hat{f}\colon S \longrightarrow T$ such that the following commutes



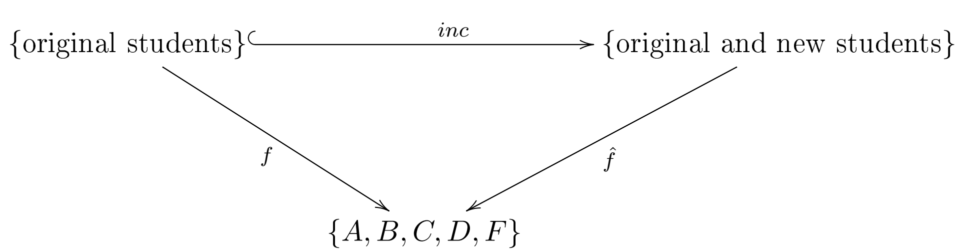

(1.34)

In English, $\hat{f}$ extends $f$ to a different (larger) domain.

**Example 1.5.30.** As a simple example, consider $R$ to be a set of students and

$$f\colon R \longrightarrow \{A, B, C, D, F\} \tag{1.35}$$

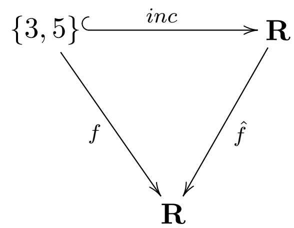
assigns every student a grade. If some new students came into the class, the teacher would have to extend $f$ to give grades to all the students (including the new ones) as $\hat{f}\colon S \longrightarrow \{A, B, C, D, F\}$. We want $\hat{f}$ to assign the same grades as $f$ did for any of the original students. This is clear with the following commutative diagram:




(1.36)

□

**Example 1.5.31.** Let $\{3,5\}$ be a set of two real numbers. There is an obvious inclusion of the two real numbers into the set of all numbers $inc\colon \{3,5\} \hookrightarrow \mathbf{R}$. Let $f\colon \{3,5\} \longrightarrow \mathbf{R}$ be any function that picks two values. Then, there exists a function $\hat{f}\colon \mathbf{R} \longrightarrow \mathbf{R}$ that extends $f$.




(1.37)

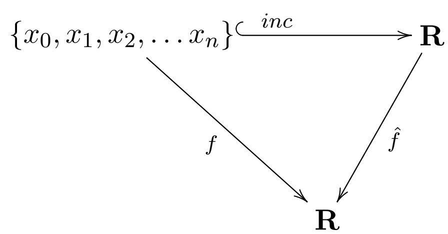
This is just the simple idea that given any two points on the plane, there is a straight line that connects them. In detail

$$\hat{f} = mx + b = \frac{\Delta y}{\Delta x}x + b = \frac{f(5)-f(3)}{5-3}x + \frac{5f(3)-3f(5)}{5-3} = \frac{f(5)-f(3)}{2}x + \frac{5f(3)-3f(5)}{2}. \quad (1.38)$$

□

Thinking of a straight line as a function, the previous example of an extension can be ... extended...

**Example 1.5.32.** Let $\{x_0, x_1, x_2, \ldots, x_n\}$ be a set of $n+1$ different real numbers and let $inc\colon \{x_0, x_1, x_2, \ldots, x_n\} \hookrightarrow \mathbf{R}$ be the inclusion function. Every $f\colon \{x_0, x_1, x_2, \ldots, x_n\} \longrightarrow \mathbf{R}$ has an extension along $inc$ called $\hat{f}\colon \mathbf{R} \longrightarrow \mathbf{R}$ which is a polynomial function of degree at most $n$.

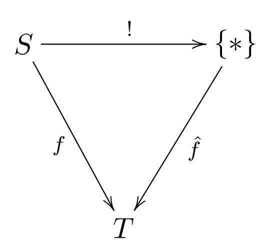



(1.39)

The function $\hat{f}$ is called the "Lagrange interpolating polynomial" of the points described by $f$. (We will not use this in the text.) □

While extensions are usually about inclusion functions, we can also use the setup of an extension for functions that are not inclusion functions.

**Example 1.5.33.** A function $f\colon S \longrightarrow T$ is a **constant function** if it outputs the same value for any input. That means there is some $t_0 \in T$ such that for all $s \in S$ we have $f(s) = t_0$. We can describe a constant function by using the same notation of an extension but forget about the inclusion function. In detail, $f$ is a constant function if there exists an extension $\hat{f}\colon \{*\} \longrightarrow T$ of $f$ as in the diagram




(1.40)

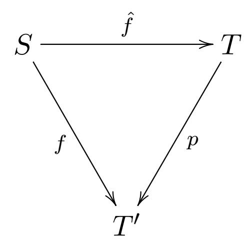
where the function $!: S \longrightarrow \{*\}$ is the unique function that always outputs $*$, the only element it can output. Another way of saying this, is that $f$ is a constant function if it can be written as a function that factors through some function from $\{*\}$. □

**Exercise 1.5.34.** Show that if $id_S : S \longrightarrow S$ can be extended along the function $f: S \longrightarrow T$, then $f$ is a one-to-one function.

**Solution:** This is essentially the content of Diagram (1.31).

■

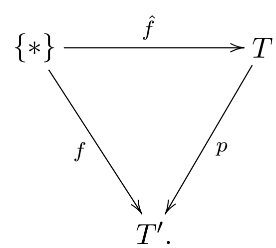
The third operation of functions is a **lifting**. Consider an onto function $p: T \longrightarrow T'$. Let $f: S \longrightarrow T'$ be any function. A lifting of $f$ along $p$ is a function $\hat{f}: S \longrightarrow T$ that makes the following triangle commute.




(1.41)

Here is the simplest example of lifting.

**Example 1.5.35.** Consider the following diagram:




(1.42)

Here a function $f: \{*\} \longrightarrow T'$ picks out an element of $T'$. A lifting of $f$ is $\hat{f}: \{*\} \longrightarrow T$ which picks out an element of $T$ that $p$ maps onto the element that $f$ chose. In other words, if $f$ picked out $t_0 \in T'$ then $\hat{f}$ will pick out an element in $T$ that will map onto $t_0$. There might be many liftings. The set of all possible elements that a lifting can pick out is denoted $p^{-1}(t_0) \subseteq T$ which is called the preimage of $p$. □

**Example 1.5.36.** Here is a cute example of a lifting from the world of politics. Let $T$ be the set of 320 million American citizens and let $T'$ be the set of 50 states. The function $p$ takes every citizen to the state they live in. Let $S$ be a set of 3 elements such as $\{a, b, c\}$. The function $f: S \longrightarrow T'$ chooses 3 states. A lifting of $f$ along $p$ is a function $\hat{f}: S \longrightarrow T$ which will choose three citizens, one from each of the states $f$ chose. There are obviously many such liftings. □

**Example 1.5.37.** Let us build on the last example. Let $T$, $T'$ and $p$ be as in the last example. Let $S$ be the set $\{a, b, c\} \times T'$, i.e., pairs of letters and states. The $f: S \longrightarrow T'$ function is defined as follows: $f(b, \text{New Jersey}) = \text{New Jersey}$ i.e., $f$ takes a letter and a state and outputs the same state. Notice that for each state, there are three elements in $S$ that go to that state. For example

$$f(a, \text{New Jersey}) = f(b, \text{New Jersey}) = f(c, \text{New Jersey}) = \text{New Jersey}$$

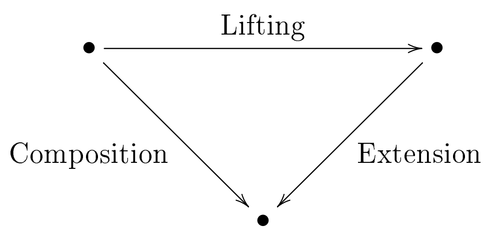
A lifting of $f$ along $p$ is a function $\hat{f}\colon S \longrightarrow T$ which will choose three citizens from each state. There are many such liftings. □

**Exercise 1.5.38.** Show that if $id_T\colon T \longrightarrow T$ can be lifted along the function $f\colon S \longrightarrow T$, then $f$ is an onto function.

**Solution:** This is essentially the content of Diagram (1.32).

■

One can see these three operations — composition, extension and lifting — as three sides of a triangle:




(1.43)

Each side uses the other two sides as the input to the operation. Composition will be used on every page of this text. We will see (especially in Section 9.4) that the extension and lifting operations are very important in many contexts besides sets and functions.

### Equivalence Relations

We are not only interested in how a set is related to other sets. Sometimes the elements of a set are related to each other in interesting ways.

**Definition 1.5.39.** Let $S$ be a set. A **relation** on $S$ is a subset $R$ of the set $S \times S$. The ordered pair $(s_1, s_2)$ is in $R$ means $s_1$ and $s_2$ are related.

**Example 1.5.40.** Let $S$ be the set of citizens of the United States. Consider the following relations on this set.

- $R_1$ consists of those $(s, t)$ where $s$ and $t$ are cousins.
- $R_2$ consists of those $(s, t)$ where $s$ is the same age or older than $t$.
- $R_3$ consists of those $(s, t)$ where $s$ and $t$ live in the same state.
- $R_4$ consists of those $(s, t)$ where $s$ and $t$ belong to the same political party.

□

The following three properties will characterize the notion of "sameness." When do two elements in a set share some property that makes them the "same"?

**Definition 1.5.41.** The relation $R \subseteq S \times S$ on a set $S$ is

- **reflexive** if every element is related to itself: for all $s$ in $S$, $(s, s)$ is in $R$.
- **symmetric** if one element is related to another, then the other is related to the first: for all

$s$ and $t$ in $S$, if $(s,t)$ is in $R$, then $(t,s)$ is in $R$.

- **transitive** if $s$ is related to $t$ and $t$ is related to $u$, then $s$ is related to $u$: for all $s$, $t$ and $u$ in $S$, if $(s,t)$ is in $R$ and $(t,u)$ is in $R$, then $(s,u)$ is in $R$.

**Example 1.5.42.** Let us look which properties are satisfied from the relations of Example 1.5.40.

- The cousin relation $R_1$ is not reflexive (no one is their own cousin); it is symmetric; but it is not transitive ($x$ can be a cousin to $y$ through $y$'s mother's side and $y$ can be a cousin to $z$ from $y$'s father's side. In this case $x$ will, in general, not be cousins to $z$.)
- The older relation $R_2$ is reflexive (everyone is the same age as themselves); not symmetric (if $x$ is older than $y$ then $y$ is not older than or the same age as $x$) and it is transitive.
- The state relation $R_3$ is reflexive, symmetric and transitive.
- The political party relation $R_4$ is reflexive, symmetric and transitive.

□

Many times a set of elements can be split up ("partitioned") into different subsets where each subset will have all the elements with a particular property. For example, the set of cars can be split up by color. So there will be the subset of blue cars, the subset of red cars, the subset of green cars, etc. The collection of all such subsets will form a set itself. Formally this can be said as follows.

**Definition 1.5.43.** A relation on a set is an **equivalence relation** if it is reflexive, symmetric, and transitive. We write such relations as $\sim \subseteq S \times S$ and write $r \sim s$ for $(r,s) \in \sim$. With an equivalence relation on the set $S$, we can describe subsets of $S$ called **equivalence classes**. If $s$ is an element of $S$, then the equivalence class of $s$ is the set of all elements that are related to it:

$$[s] = \{r \in S : r \sim s\} \tag{1.44}$$

That is, $[s]$ is the set of all elements that are "the same" as $s$. For a given set $S$ and an equivalence relation $\sim$ on $S$ we form a **quotient set** denoted $S/\sim$. The objects of $S/\sim$ are all the equivalence classes of elements in $S$. There is an obvious **quotient function** from $S$ to $S/\sim$ that takes $s$ to $[s]$.

**Example 1.5.44.** Let us examine the equivalence classes for the equivalence relations of Example 1.5.40.

- Each equivalence class for the relation $R_3$ consists of all the residents of a particular state. The quotient set contains the 50 equivalence classes corresponding to the 50 States (we are ignoring abnormalities like Guam and Washington D.C.). The quotient function takes every citizen to the state in which they reside.
- Each equivalence class for the relation $R_4$ consists of all the people in a particular political party. The quotient set consists of a set whose elements correspond to political parties. The quotient function takes every citizen to the political party to which they belong (we are ignoring independents.)

□

## Graphs

Directed graphs are a common structure (based on sets) that have applications everywhere. They also have many similarities to categories.

**Definition 1.5.45.** A directed **graph** $G = (V(G), A(G), src_G, trg_G)$ is

- a set of vertices, $V(G)$, and
- a set of arrows, $A(G)$.

Furthermore

- every arrow has a source: there is a function $src_G \colon A(G) \longrightarrow V(G)$, and
- every arrow has a target: there is a function $trg_G \colon A(G) \longrightarrow V(G)$.

If $f$ is an element of $A(G)$ with $src_G(f) = x$ and $trg_G(f) = y$, we draw this arrow as

$$x \xrightarrow{f} y \ . \tag{1.45}$$

An example of a graph is Figure 1.3.

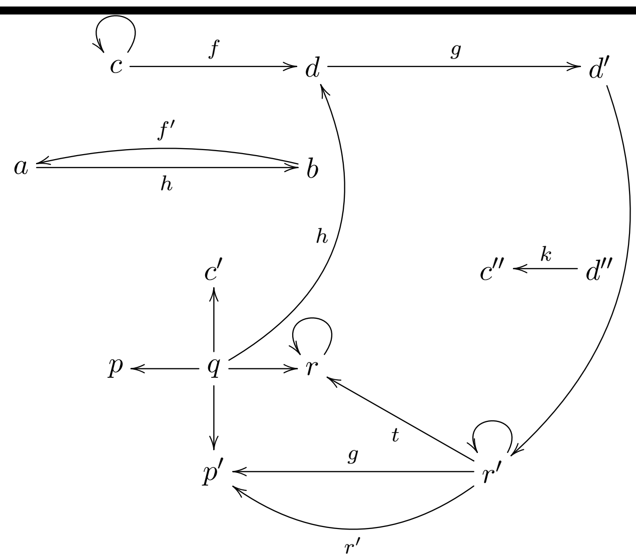



Figure 1.3: An example of a directed graph.

**Example 1.5.46.** Graphs are everywhere.

- A street map can be thought of as a directed graph where the vertices are street corners and there is an arrow from one corner to the other if there is a one-way street between them.

When there is a two-way street we might write it like this

$$* \rightleftarrows * \qquad \text{or} \qquad * \longrightarrow\!\!\!\!\!\!\longleftarrow *. \tag{1.46}$$

Such an arrow is called a **symmetric edge**.

- A Feynman diagram is a type of souped-up directed graph where the arrows correspond to particles traveling in space and time. The vertices correspond to interactions of the particles. The direction of the arrows are typically going up because they are going forward in time. There are also arrows going down meaning they are going backward in time. There will be more about this on page 37.
- An electrical circuit can be viewed as a directed graph. The vertices are the branching points and the resistors, batteries, capacitors, diodes, etc. The arrows describe the direction of the flow of electricity.
- Computer networks can be seen as a graph where the vertices are computers and there is an arrow from one computer to another if there is a way for the first computer to communicate with the second.
- The billions of web pages in the World Wide Web form a graph. The vertices are the web pages and there is an arrow if there is a link from one web page to another.
- Facebook can be seen as a graph. Every personal Facebook account is a vertex, and there are arrows between two Facebook accounts if they are friends. Notice that if $x$ is friends with $y$ then $y$ must be friends with $x$. So all the arrows are symmetric edges and the graph is called symmetric.
- All the people on Earth form a graph. The vertices are the people. There is an arrow from $x$ to $y$ if $x$ knows $y$. (We are not being specific as to what it means to "know" someone.) There is an idea called "six degrees of separation," which says that in this graph, you never need to traverse more than six arrows to get from any person to any other person. We are all connected!
- The set of all sets and functions form a giant graph. In detail, the verticies are all possible sets. The arrows are functions from a set to another set.

□

A graph homomorphism is a way of mapping one graph to another. This will be similar to what happens when we talk about mapping one category to another category. Basically the vertices map to the vertices and the arrows map to the arrows but we insist that they match up well. In detail:

**Definition 1.5.47.** Let $G = (V(G), A(G), src_G, trg_G)$ and $G' = (V(G'), A(G'), src_{G'}, trg_{G'})$ be graphs. A **graph homomorphism** $H\colon G \longrightarrow G'$ consists of

- A function that assigns vertices, $H_V\colon V(G) \longrightarrow V(G')$
- A function that assigns arrows, $H_A\colon A(G) \longrightarrow A(G')$

These two maps must respect the source and target of each arrow. That means:

- For all $f$ in $A(G)$, $H_V(src_G(f)) = src_{G'}(H_A(f))$.
- For all $f$ in $A(G)$, $H_V(trg_G(f)) = trg_{G'}(H_A(f))$.

Saying that these axioms are satisfied is the same as saying that the following two squares commute:

$$
\begin{array}{ccc}
A(G) & \xrightarrow{H_A} & A(G') \\
\downarrow{\scriptstyle src_G} & & \downarrow{\scriptstyle src_{G'}} \\
V(G) & \xrightarrow[H_V]{} & V(G')
\end{array}
\qquad
\begin{array}{ccc}
A(G) & \xrightarrow{H_A} & A(G') \\
\downarrow{\scriptstyle trg_G} & & \downarrow{\scriptstyle trg_{G'}} \\
V(G) & \xrightarrow[H_V]{} & V(G').
\end{array}
\tag{1.47}
$$

Another way to understand these requirements is to see what the maps $H_V$ and $H_A$ do to a single arrow $f$ (that is $f \mapsto H_A(f)$):

$$
\begin{array}{ccc}
\underline{\textbf{Graph G}} & & \underline{\textbf{Graph G}'} \\
 & & \\
src_G(f) & \xmapsto{H_V} & H_V(src_G(f)) = src_{G'}(H_A(f)) \\
{\scriptstyle f}\downarrow & \xmapsto{H_A} & \downarrow{\scriptstyle H_A(f)} \\
trg_G(f) & \xmapsto{H_V} & H_V(trg_G(f)) = trg_{G'}(H_A(f))
\end{array}
\tag{1.48}
$$

Just as we can determine many properties of sets by examining functions, we can also determine many properties of graphs by examining graph homomorphisms.

**Example 1.5.48.**

- A vertex of a graph $G$ can be described by a graph homomorphism from the one-vertex graph $(*)$ as follows $H : * \longrightarrow G$.
- A directed edge of a graph can be determined by a graph homomorphism from the graph $* \longrightarrow *$ to $G$.
- A triangle in a graph $G$ can be determined by a graph homomorphism from the graph

$$
\begin{array}{ccc}
* & \longrightarrow & * \\
 & \searrow \quad \swarrow & \\
 & * &
\end{array}
\tag{1.49}
$$

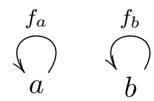
  to the graph $G$.
- A path of length $n$ in a graph $G$ can be determined by a graph homomorphism from the "snake" graph

$$
* \longrightarrow * \longrightarrow * \longrightarrow \cdots \longrightarrow * \tag{1.50}
$$

  of length $n$ to $G$.

□

**Exercise 1.5.49.** Prove that a graph $G$ is weakly connected (for any two vertices $x$ and $y$, there is either a path from $x$ to $y$ or a path from $y$ to $x$) if there does *not* exist a graph homomorphism that is surjective on vertices from $G$ to the graph




(1.51)

**Solution:** If $G$ is not weakly connected, then we will be able to split the graph into two parts with no arrow from one part to the other part. There is then a function from the graph to this two-vertex graph where one part of the graph has value $a$ and all the arrows of that part to the single arrow $f_a$ and the nodes of the other part have values in $b$ and all the other arrows to $f_b$. If the graph is weakly connected, then this partitioning would not be possible.

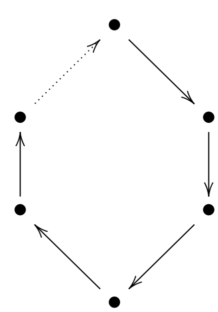
■

**Exercise 1.5.50.** Use graph homomorphisms to determine different types of paths in a graph.

- How do you describe a simple path in a graph (a simple path is a path that does not have repeated vertices)?
- What about a cycle of length $n$ (a cycle is a path that starts and ends at the same vertex)?
- Do the same for a simple cycle of length $n$ (a simple cycle is a cycle in which the only repeating vertex is the starting point which is the ending point).

**Solution:**

- One-to-one graph homomorphisms from the "snake" graph, Diagram (1.50), to any graph will correspond to simple paths.
- graph homomorphisms from "ring" graphs of the form



  will describe such cycles.
- One-to-one graph homomorphisms from the "ring" graphs will correspond to simple cycles.

■

**Exercise 1.5.51.** Show that the composition of graph homomorphisms is a graph homomorphism. Show that the composition is associative.

**Solution:** Let $H\colon G \longrightarrow G'$ and $H'\colon G' \longrightarrow G''$ be graph homomorphisms. Then $H' \circ H : G \longrightarrow G''$. The fact that $H' \circ H$ preserves the sources of arrows amounts to the commutativity of the following diagram

$$\begin{array}{ccccc} A(G) & \xrightarrow{H_A} & A(G') & \xrightarrow{H'_A} & A(G'') \\ \Big\downarrow{\scriptstyle src_G} & & \Big\downarrow{\scriptstyle src_{G'}} & & \Big\downarrow{\scriptstyle src_{G''}} \\ V(G) & \xrightarrow[H_V]{} & V(G') & \xrightarrow[H'_V]{} & V(G''). \end{array} \tag{1.52}$$

which is assured because each square commutes. A similar argument must be made to show that $H' \circ H$ preserves targets. The proof of the associativity of the composition of graph homomorphisms comes from the fact that they are functions and is similar to the solution to Exercise 1.5.26.

■

**Exercise 1.5.52.** Define the **identity graph homomorphism**, $I_G$, for any graph $G$. Show that if $H\colon G \longrightarrow G'$ is a graph homomorphism then $H \circ I_G = H$ and $I_{G'} \circ H = H$.

**Solution:** The identity graph homomorphism is defined as $I_{G,A}(x) = x$ and $I_{G,V}(f) = f$. This graph homomorphism preserves the sources and targets of the arrows for trivial reasons. The fact that it acts like a unit to composition is because it is essentially two identity functions.

■

## Groups

Another important structure that is based on sets and related to categories is a group. It is nice to see the definition of a group from a function perspective.

First a discussion of operations. We all know what we mean by operations on numbers. If you take numbers $x$ and $y$, you can perform the addition operation and get $x+y$. You can also perform other operations like $x - y$ or $y \cdot x$. All of these are examples of binary operations. Operations are really just functions. For a given set, $S$, a **binary operation** is a function $f\colon S \times S \longrightarrow S$. For two elements $s$ and $s'$, we write the value of $f$ as $f(s, s')$. A **unary operation** is a function that takes one element of $S$ and outputs one element of $S$, i.e., $f\colon S \longrightarrow S$. An example of a unary operation is the inverse operation that takes $x$ and returns $x^{-1}$. There are also **trinary operations** $f\colon S \times S \times S \longrightarrow S$ and $n$**-ary operations** for all natural numbers $n$

$$f\colon \underbrace{S \times S \times \cdots \times S}_{n \text{ times}} \longrightarrow S. \tag{1.53}$$

If $n = 0$ then we write the 0-ary product as the set with one element $\{*\}$ and a 0**-ary operation** is written as $f\colon \{*\} \longrightarrow S$ which basically picks out an element of $S$. Such an operation describes an element that does not change, i.e., a constant.

Let us put this all together and give the formal definition of a group.

**Definition 1.5.53.** A **group** $(G, \star, e, (\ )^{-1})$ is a set $G$ with the following operations:

- A multiplication operation: a binary operation $\star\colon G \times G \longrightarrow G$.
- An identity: there is a special element $e$ in $G$ called the identity of the group, i.e, a 0-ary operation $u : \{*\} \longrightarrow G$ where $u(*) = e$.
- An inverse operation: a unary operation $(\ )^{-1}\colon G \longrightarrow G$

These operations satisfy the following axioms:

- The multiplication is associative: for all $x$, $y$ and $z$, we have $(x \star y) \star z = x \star (y \star z)$.
- The identity acts like a unit of the multiplication (like when you multiply a number with 1, the result does not change, i.e., $1 \times n = n$ hence 1 is a "unit"): for all $x$, $x \star e = x = e \star x$.
- Multiplying an element with its inverse gives the identity: For all $x$ in $G$, $x \star x^{-1} = e = x^{-1} \star x$

**Example 1.5.54.** Here are some examples of groups.

- The additive integers: $(\mathbf{Z}, +, 0, -)$. Addition and subtraction are the usual operations.
- The additive real numbers: $(\mathbf{R}, +, 0, -)$. Addition and subtraction are the usual operations.
- The multiplicative positive reals: $(\mathbf{R}^+, \cdot, 1, (\ )^{-1})$ where $\mathbf{R}^+$ are the positive real numbers, $\cdot$ is multiplication, and the function $(\ )^{-1}$ takes $r$ to $\frac{1}{r}$.
- Clock arithmetic: $(\{0, 1, 2, 3, \ldots, 11\}, +, 0, -)$ where addition and subtraction is going around the clock. 0 is the unit because when you add 0 to any number you get back to the original number. Notice that 11 did not play an important role and that this would be true for any non-negative integer.
- The **trivial group**: $(\{0\}, +, 0, -)$. This is the world's smallest group. It has only one element and the operations work as expected.

□

Parts of the three axioms of a group can be seen as commutative diagrams in Figure 1.4. Around each of the commutative diagrams are maps showing the values of the functions on elements.

**Exercise 1.5.55.** The second diagram in Figure 1.4 shows the $x = x \star e$ axiom. Give a commutative diagram for the $x = e \star x$ axiom.

**Solution:** It is essentially the same diagram but change the $G \times \{*\}$ to $\{*\} \times G$.

■

**Exercise 1.5.56.** The third diagram in Figure 1.4 shows the $e = x \star x^{-1}$ axiom. Give the commutative diagram for the $e = x^{-1} \star x$ axiom.

**Solution:** It is essentially the same diagram with the $id \times (\ )^{-1}$ map switched to $(\ )^{-1} \times id$.

■

**Important Categorical Idea 1.5.57.** Many times, even when we have a nice, clear definition or description of a mathematical structure in terms of elements, we still desire a description in terms of morphisms. The reason why such a description is important is that

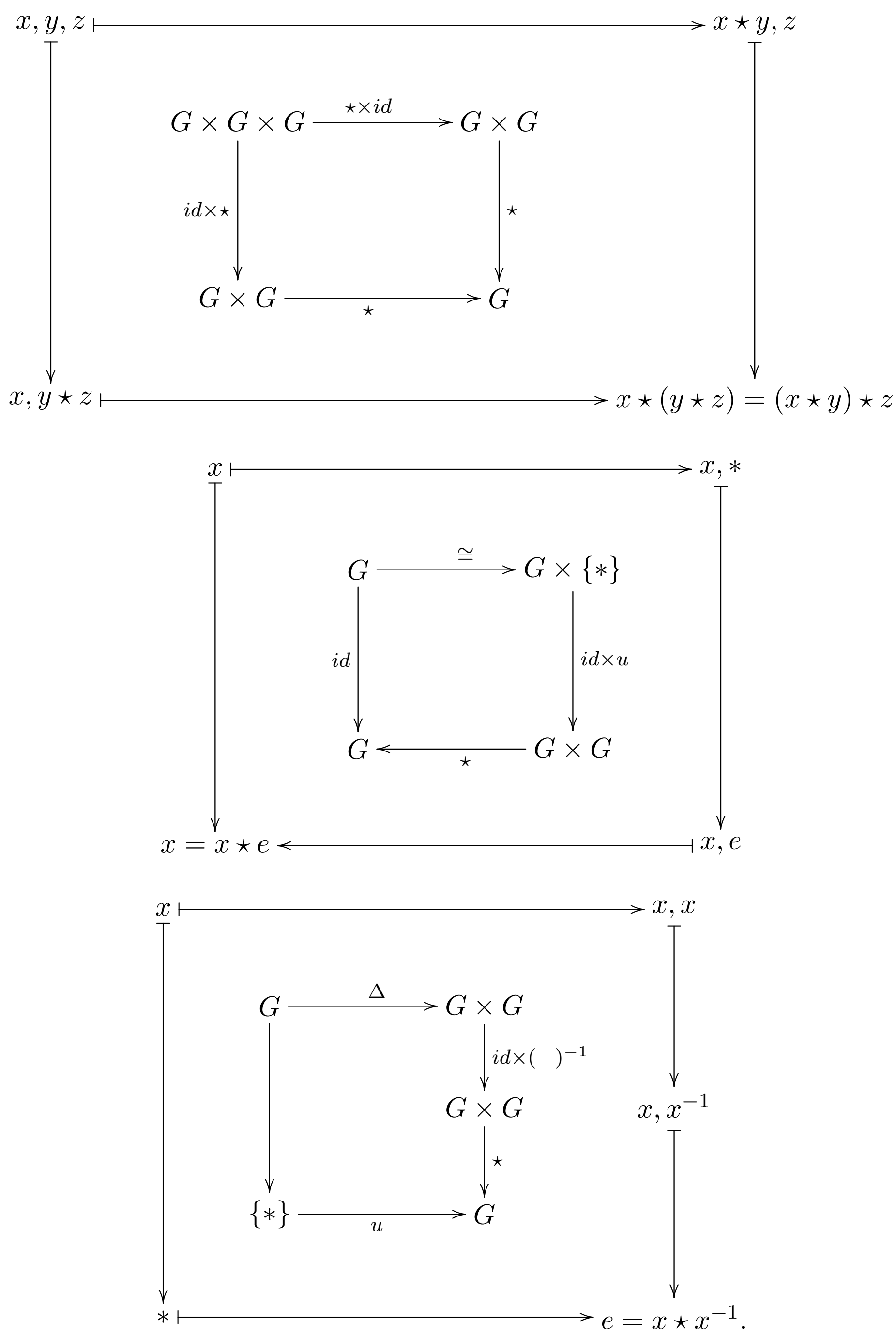



Figure 1.4: Commutative diagrams of the axioms of a group

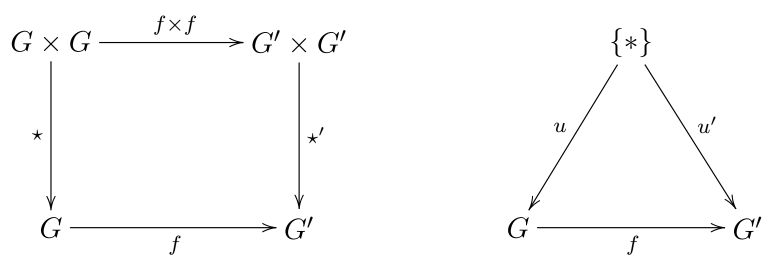
once we have it, we can use the morphism description in many different categories. Whereas a description in terms of elements is good only in one context, a description in terms of morphisms can be used in many different categories and contexts. For example, we will see this definition of a group arise in other contexts besides sets and functions. ○

Just as a function is a way of mapping one set to another, and a graph homomorphism is a way of mapping one graph to another, a group homomorphism is a way of mapping one group to another.

**Definition 1.5.58.** Let $(G, \star, e, (\ )^{-1})$ and $(G', \star', e', (\ )'^{-1})$ be groups. A **group homomorphism** $f\colon (G, \star, e, (\ )^{-1}) \longrightarrow (G', \star', e', (\ )'^{-1})$ is a function $f\colon G \longrightarrow G'$ that satisfies the following two axioms

- The function respects the group operation: for all $x, y \in G$, $f(x \star y) = f(x) \star' f(y)$
- The function respects the unit: $f(e) = e'$

We can write these two requirements as the following two commutative diagrams.




(1.54)

**Technical Point 1.5.59.** We did not insist that $f$ respect inverses. Do not worry about it. It is true without saying it because it is a consequence of the other two axioms. First notice that in any group, $x^{-1}$ is the unique inverse of the element $x$. To see this, imagine that $x$ has two inverses $y$ and $y'$. Consider the following sequence of equalities

$$\begin{aligned}
y &= y \star e && \text{unit axiom} && (1.55)\\
&= y \star (x \star y') && y' \text{ is the inverse of } x && (1.56)\\
&= (y \star x) \star y' && \text{associativity axiom} && (1.57)\\
&= e \star y' && y \text{ is the inverse of } x && (1.58)\\
&= y' && \text{unit axiom} && (1.59)
\end{aligned}$$

This shows that $y = y'$. Now let us use this fact to show that inverses are preserved by group homomorphisms. First consider

$$e' = f(e) = f(x \star x^{-1}) = f(x) \star' f(x^{-1}). \tag{1.60}$$

This shows that the inverse of $f(x)$ can not only be written as $f(x)^{-1}$ but also as $f(x^{-1})$. But since inverses are unique, we proved $f(x)^{-1} = f(x^{-1})$. ♡

**Example 1.5.60.** Here are some examples of group homomorphisms.

- There is always a unique group homomorphism from any group to the trivial group where every element of the group goes to 0 of the trivial group.
- There is always a unique group homomorphism from the trivial group to any group in which the 0 of the trivial group goes to the identity of the group.
- There is an inclusion group homomorphism $inc\colon \mathbf{Z} \longrightarrow \mathbf{R}$.
- There is a group homomorphism $\mathbf{Z} \longrightarrow \{0,1,2,3,\ldots,11\}$ that takes every whole number $x$ and sends it to the remainder when $x$ is divided by 12.
- Let $b$ be some positive real number called the "base". There is a exponential function

$$b^{(\ \ )}\colon (\mathbf{R},+,0,-) \longrightarrow (\mathbf{R}^+,\cdot,1,(\ \ )^{-1}) \tag{1.61}$$

  that takes a real number $r$ and sends it to $b^r$. The two requirements to be a group homomorphism turn out to mean that $b^{r+r'} = b^r \cdot b^{r'}$ ($b^{(\ \ )}$ takes addition to multiplication) and $b^0 = 1$.
- There is a logarithm function that is the inverse of the exponential function.

$$Log_b\colon (\mathbf{R}^+,\cdot,1,(\ \ )^{-1}) \longrightarrow (\mathbf{R},+,0,-). \tag{1.62}$$

  The function $Log_b$ takes a positive real number $r$ to $Log_b(r)$. The requirements to be a group homomorphism are the well-known facts that $Log_b(r \cdot r') = Log_b(r) + Log_b(r')$ ($Log_b$ takes multiplication to addition) and $Log_b(1) = 0$.

□

**Exercise 1.5.61.** Show that the composition of group homomorphisms is a group homomorphism. Show that the composition is associative.

**Solution:** Let $f\colon G \longrightarrow G'$ and $f'\colon G' \longrightarrow G''$ be group homomorphisms. Then $(f' \circ f)(x) = f'(f(x))$. To show that it preserves the group operations:

$$(f' \circ f)(x \star x') = f'(f(x \star x')) = f'(f(x) \star' f(x')) = f'(f(x)) \star'' f'(f(x')) = (f' \circ f)(x) \star'' (f' \circ f)(x')$$

and

$$(f' \circ f)(e) = f'(f(e)) = f'(e') = e''.$$

■

**Exercise 1.5.62.** Define the **identity group homomorphism**, $id_G$, for every group $(G,\star,0,(\ \ )^{-1})$. Show that if $f\colon (G,\star,e,(\ \ )^{-1}) \longrightarrow (G',\star',e',(\ \ )'^{-1})$ is a group homomorphism then $f \circ id_G = f$ and $id_{G'} \circ f = f$.

**Solution:** The identity trivially respects the group operations. The last part follows from the fact that group homomorphisms are simply functions (that satisfy certain properties.)

■

## Guide to Further Study

Most of the material found in this section can be found in any discrete mathematics or finite mathematics textbook e.g. [57, 56]. They can also be found in many pre-calculus textbooks.

The importance of looking at functions between sets is central to all of category theory. This is also stressed by two books coauthored by one of the leaders of category theory, F. William Lawvere. With Robert Rosebrugh he wrote *Sets for Mathematics* [38] and with Stephen H. Schanuel he wrote a textbook titled *Conceptual Mathematics* [40].

The novice can find basic group theory in any introduction to modern algebra or abstract algebra, e.g., [1, 18].

This idea that most of the operations on functions can be seen as compositions, extensions and liftings (as in Diagram (1.43)) was taken from [63] where much of category theory is built out of these operations. We will see more of extensions and liftings in Section 9.4.

However, if you really want to learn more about categorical thinking, roll up your sleeves and keep on reading the rest of this book!

# Chapter 2

# Categories

*A good stack of examples, as large as possible, is indispensable for a thorough understanding of any concept, and when I want to learn something new, I make it my first job to build one.*
Paul Halmos Page 63 of *I Want to be a Mathematician: An Automathography*

With the ideas of set and functions in hand, we move on to the world of categories. Section 2.1 begins with a formal definition of categories. It then proceeds to form a giant stack of examples of categories from all over. In Section 2.2 we discuss some simple properties of morphisms in categories. We elaborate on some simple categories related to a categories in Section 2.3. The chapter ends with Section 2.4 which is a mini-course that teaches the basics of linear algebra. The study of linear algebra is essentially an in-depth exploration of the category of vector spaces.

## 2.1 Basic Definitions and Examples

Before formally defining a category, let us summarize what we saw in Section 1.5 concerning sets and functions. The collection of sets and functions form a category. By carefully examining this collection, we will see what is needed in the definition of a category.

**Example 2.1.1.** Consider the collection of all sets. There are functions between sets. If $f$ is a function from set $S$ to set $T$, then we write it as $f\colon S \longrightarrow T$ and we call $S$ the **domain** of $f$ and $T$ the **codomain** of $f$. Certain functions can be composed: for $f\colon S \longrightarrow T$ and $g\colon T \longrightarrow U$, there exists a function $g \circ f\colon S \longrightarrow U$ which is defined for $s$ in $S$ as $(g \circ f)(s) = g(f(s))$. This composition operation is associative, which means that for $f\colon S \longrightarrow T$, $g\colon T \longrightarrow U$ and $h\colon U \longrightarrow V$, both ways of associating the functions $h \circ (g \circ f)$ and $(h \circ g) \circ f$ are equal to the function described as follows

$$s \mapsto f(s) \mapsto g(f(s)) \mapsto h(g(f(s))). \tag{2.1}$$

That is, $h \circ (g \circ f) = (h \circ g) \circ f$ and on $s$ of $S$ this function gives the value $h(g(f(s)))$. For every set $S$, there is a function $id_S\colon S \longrightarrow S$ which is called the **identity function** and is defined for $s$ in $S$ as $id_S(s) = s$. These identity functions have the following properties: for all $f\colon S \longrightarrow T$, it is true that $f \circ id_S = f$ and $id_T \circ f = f$. The collection of sets and functions form a category called $\mathbb{Set}$. This category is easy to understand, and we use it to hone our ideas about many structures of category theory. □

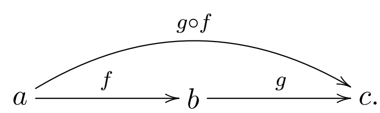
Now for the formal definition of a category.

**Definition 2.1.2.** A **category** $\mathbb{A}$ is a collection of objects, $Ob(\mathbb{A})$, and a collection of morphisms, $Mor(\mathbb{A})$, which has the following structure:

- Every morphism has an object associated to it called its domain: there is a function $Dom_{\mathbb{A}} \colon Mor(\mathbb{A}) \longrightarrow Ob(\mathbb{A})$.
- Every morphism has an object associated to it called its codomain: there is a function $Cod_{\mathbb{A}} \colon Mor(\mathbb{A}) \longrightarrow Ob(\mathbb{A})$. We write

$$f \colon a \longrightarrow b \qquad \text{or} \qquad a \xrightarrow{\quad f \quad} b \tag{2.2}$$

  for the fact that $Dom_{\mathbb{A}}(f) = a$ and $Cod_{\mathbb{A}}(f) = b$.
- Adjoining morphisms can be composed: if $f \colon a \longrightarrow b$ and $g \colon b \longrightarrow c$, then there is an associated morphism $g \circ f : a \longrightarrow c$. We can describe these morphisms as




(2.3)

- Every object has an identity morphism: there is a function $Ident_{\mathbb{A}} \colon Ob(\mathbb{A}) \longrightarrow Mor(\mathbb{A})$.We denote the identity of $a$ as $id_a : a \longrightarrow a$ or


(2.4)

This structure must satisfy the following two axioms:

- Composition is associative: given $f \colon a \longrightarrow b$, $g \colon b \longrightarrow c$ and $h \colon c \longrightarrow d$, the two ways of composing these maps are equal:

$$h \circ (g \circ f) = (h \circ g) \circ f \colon a \longrightarrow d. \tag{2.5}$$

- Composition with the identity does not change the morphism: for any $f \colon a \longrightarrow b$ the composition with $id_a$ is $f$, i.e., $f \circ id_a = f$ and composition with $id_b$ is also $f$, i.e., $id_b \circ f = f$.

### Basic Examples

**Example 2.1.3.** Let us mention three examples of categories that we already saw in this text. Although we did not call them categories, the examples and exercises showed that they have the structure of a category.

- Sets and functions between them form the category $\mathbb{Set}$.
- Directed graphs and graph homomorphisms give us $\mathbb{Graph}$.
- Groups and group homomorphisms make up $\mathbb{Group}$.

□

The definition of a category is a "mouthful" that has many parts to it. There are several important comments concerning this definition.

- We called the elements of $Mor(\mathbb{A})$ the **morphisms** of the category. We will also interchangeably use the words **maps** and **arrows**.
- $Ob(\mathbb{A})$ and $Mor(\mathbb{A})$ are called "collections" rather than the more set theoretical "set" or "class". The reason for this is because we do not want to get bogged down in the language of set theory. If you know the language of set theory, then realize that sometimes our objects and morphisms will be sets and sometimes proper classes. Often we will not specify which and just use the word "collection."
- It is important to notice that if we have morphisms $f: a \longrightarrow b$ and $g: b \longrightarrow c$ then we write the composition as $g \circ f$ rather than $f \circ g$. We do this because in many categories the morphisms will be types of functions. When we apply the composition of functions, it looks like $g(f(\ \ ))$ which is notationally closer to $g \circ f$ than $f \circ g$. As we get more and more used to the language we will write $gf$ rather than $g \circ f$.
- Another way of seeing the definition of a category is to discuss collections of morphisms. For objects $a$ and $b$ in category $\mathbb{A}$, there is a collection of all the morphisms from $a$ to $b$ which we write $Hom_{\mathbb{A}}(a, b)$. The word $Hom$ comes from the word "homomorphism" which is a vestige of the algebraic and topological background of category theory. We call these collections **Hom sets** even though the collections might not be a set. Composition in the category in terms of the Hom sets becomes the operation

$$\circ\colon Hom_{\mathbb{A}}(b, c) \times Hom_{\mathbb{A}}(a, b) \longrightarrow Hom_{\mathbb{A}}(a, c) \tag{2.6}$$

$$(g, f) \longmapsto g \circ f.$$

  We refer to mappings between Hom sets as functions even though the Hom sets might be a proper class. The fact that every element $a$ of $\mathbb{A}$ has an identity element means that there is a special morphism in $Hom_{\mathbb{A}}(a, a)$ that satisfies the properties stated. As time goes on, and it is obvious what category we mean, we will drop the subscript and write $Hom(a, b)$.

The categories $\mathbb{Set}$, $\mathbb{Graph}$ and $\mathbb{Group}$ each have collections of objects and morphisms that are infinite. Let us look at some examples of categories with finite collections of objects and morphisms.

**Example 2.1.4.** These finite categories are depicted in Figure 2.1. In detail,

- $\mathbf{0}$, the empty category, has no objects and no morphisms. All the axioms of being a category are trivially true.
- $\mathbf{1}$ has one object and the single identity morphism on that object.
- $\mathbf{2}$ has two objects and three morphisms. Two of the morphisms are identity morphisms on the two objects and the third morphism goes from one object to the other.
- $\mathbf{2}_\circ$ has the two objects and the two identity maps but does not have the non-identity morphism.
- $\mathbf{2}_I$ is like $\mathbf{2}$ but there are two non-identity morphisms and their combinations are the identity morphisms. In total, it has two objects and four morphisms.

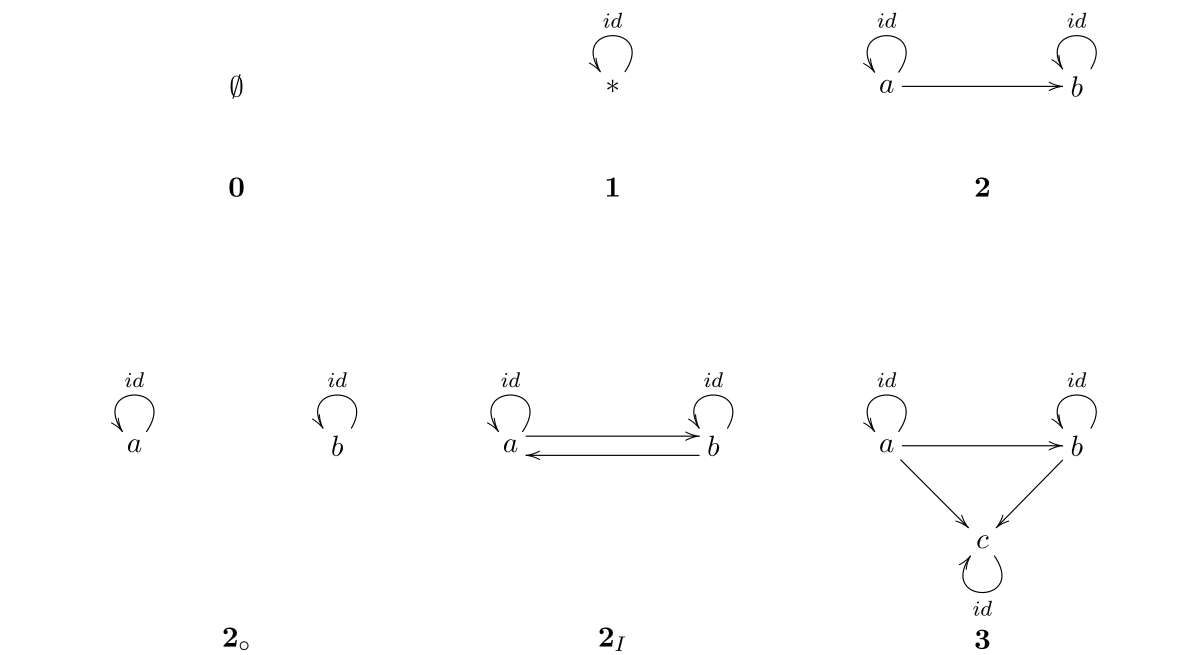
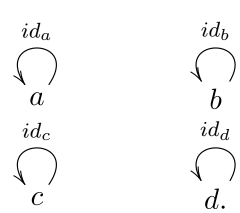



Figure 2.1: Several finite categories.

- **3** is a category with three objects, three identity morphisms, and three non-identity morphisms. The non-identity morphisms form a commutative triangle.

□

**Example 2.1.5.** Not only does the collection of all sets form a category, but each individual set can also be thought of as having the structure of a category. Let $S$ be a set (it can be finite, infinite, or even a proper class, if you understand that jargon.) We form the category $d(S)$ where the objects are the elements of $S$ and the only morphisms are identity morphisms. The composition operation can only compose identity maps with themselves. We call a category with only identity morphisms a **discrete category**. For example the set $S = \{a, b, c, d\}$ becomes the category:




(2.7)

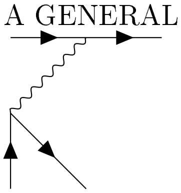
□

Let us go through more examples of categories.

## An Example from Physics

**Example 2.1.6.** An example of a category that will warm the heart of every physicist is $\mathbb{Feynman}$, the category of Feynman diagrams. A Feynman diagram is a way of visualizing the interactions of particles in physics. The directed lines in the diagram correspond to particles and the vertices correspond to interactions of particles. A simple example of a Feynman diagram is

A GENERAL PICTURE OF A FEYNMAN DIAGRAM

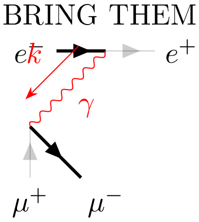


We do not have to get into the nitty-gritty details of Feynman diagrams to understand them as examples of categories. Feynman diagrams are drawn in two-dimensions. The vertical dimension corresponds to time while the horizontal dimension corresponds to space. It is conventional that we think of time as progressing up the page. Let us describe the categorical structure. The objects of the category are going to be finite sequences of signed particles. The particles can be electrons, quarks, protons, etc. By a signed particle, we mean a particle with a + (corresponds to going forward in time) or a − (corresponding to going backwards in time). An example of an object is $(p+, \epsilon-, q-, \nu+)$. We do not have to know what these particles actually are. The morphisms are the central focus. A morphism from one sequence of signed particles to another is a Feynman diagram that starts with the first sequence and ends with the second. Think of a morphism as a possible way of going from one sequence to another. There are obviously many morphisms between any two objects. Composition will be given by bringing two Feynman diagrams together one on top of the other. So the two Feynman diagrams

LEFT AND RIGHT FEYNMAN DIGRAMS

can be composed as follows

BRING THEM TOGETHER.




Composition will be associative because all we are doing is bringing three diagrams together. The identity morphism will simply be a sequence of straight lines.

PIC OF STRAIGHT LINES ... SOME GOING UP AND SOME GOING DOWN. □

**Technical Point 2.1.7.** There is one slight problem with this example. Consider the identity morphism of some sequence of signed particles. They are small lines. Nevertheless when we compose some Feynman diagram with those small lines we do not get the *exact* same Feynman diagram. We get the original Feynman diagram with small arrows added.

PICTURE HERE WITH COMPOSITION

Let us just notice this problem now. There will be similar problems with other categories and will deal more with these problems in Technical Point 2.1.60. ♡

There are two major differences between $\mathbb{Feynman}$ and the other infinite categories that we already saw. (i) Whereas in $\mathbb{Set}, \mathbb{Graph}$, and $\mathbb{Group}$ the objects are the main protagonist, in $\mathbb{Feynman}$ the morphisms are the main protagonist. The objects of the category are just used for bookkeeping to keep track of where the morphisms come and go. The Feynman diagrams are the morphisms not the objects. (ii) The morphisms in $\mathbb{Feynman}$ do not correspond to functions like we saw in the other categories. Rather they are types of graphs. We will deal with both types of categories in the future.

### Example from Computers

Here is an example of a category for someone who appreciates computer science.

**Example 2.1.8.** The category of computable functions, $\mathbb{CompFunc}$, is central for computer science. A function is **computable** if there exists a computer program that can tell a computer how to execute the function. That means, there is a computer program (written in some programming language) and if $f(x) = y$ then when $x$ is entered into the computer as input, the program will output $y$. Computable functions take certain forms of data as input and return certain forms of data as output. The kind of input and output is called a **type**. A type is a class of data. Computers deal with types like $Nat$ (natural numbers), $Int$ (integers), $Real$, $Bool$ (Boolean), $String$, etc. The objects of $\mathbb{CompFunc}$ are sequences (or products) of types. For example, $Int \times Bool \times Bool \times Real$. Given two sequences of types, a morphism of this category will be a computable function from the first sequence of types to the second sequence of types. A typical computable function might look like

$$f\colon Int \times String \times Bool \longrightarrow Bool \times Real \times Real \times Nat. \tag{2.8}$$

Composition of two computable functions is easily seen to be a computable function (the program for the first program can be "composed" or "tagged onto" the program for the second function to form a program for the composition function.) Just like functions, composition of computable functions are associative. For every list of types there exists a (useless) computable function that accepts data of the appropriate type and outputs the same data without changing it. Such functions serve as the identity morphisms in this category. Composition with the identity functions does not change the function. (Notice that the name of this category comes from the morphisms, not the objects of the category.) □

### Example from Logic

**Example 2.1.9.** The category $\mathbb{Prop}$ is about propositional logic. The objects of the category are propositional statements which are statements that are either true or false. Statements will be combined with logical operations like "and" (or "conjunction") $\wedge$, "or" (or "disjunction") $\vee$, "implication" $\Rightarrow$, "biconditional" (or"bi-implication") $\Leftrightarrow$, and "negation" (or "not") $\neg$. There is a single morphism from proposition $P$ to proposition $Q$ if and only if $P$ logically implies $Q$ (sometimes the word "entails" is used instead of "implies".) For example, there will be arrows $P \wedge Q \longrightarrow P$, $P \wedge Q \longrightarrow Q$, and $P \longrightarrow P \vee Q$. The composition in the category exists because if $P$ implies $Q$, i.e., $(P \longrightarrow Q)$ and $Q$ implies $R$, i.e., $(Q \longrightarrow R)$, then it is obvious that $P$ implies $R$, i.e., $(P \longrightarrow R)$. Associativity follows from the fact that there is at most one morphism between any two objects. The identities in the category come from the fact that for every propositional statement $P$, it is tautologically true that $P$ implies $P$ ($P \longrightarrow P$). This category is different than the other infinite categories that we have already seen because between any two objects there is either a single morphism or there is no morphism at all. The fact that there is at most one morphism between objects means that for any objects $P$ and $Q$ the set $Hom_{\mathbb{Prop}}(P, Q)$ is either the empty set or a one-object set. □

### Examples from Mathematics

Let us move on to many more examples. If at any point while going through these examples you get lost, skip to the next one. Return to the skipped examples later.

Since category theory started off as a branch of mathematics, there are many examples found in mathematics. We begin with algebraic structures. These are sets with operations that satisfy certain axioms. The morphisms between these algebraic structures are usually set functions that respect the operations.

We already saw the definition of a group. A group is just one type of algebraic structure. Here is a list of some of the algebraic structures we will meet in this book. One can see the way these different algebraic structures are related to each other with the Venn Diagram A.1 in Appendix A. There are concrete examples of such structures in Example 2.1.12.

**Definition 2.1.10.**

- A **magma** $(M, \star)$ is a set $M$ with a binary operation (an operation with two inputs) $\star \colon M \times M \longrightarrow M$. This operation is called the "multiplication" of the magma. It is not assumed that this operation satisfies any axiom.
- A **semigroup** $(M, \star)$ is a magma whose binary operation is associative, i.e., for all $x$, $y$ and $z$ in $M$, $x \star (y \star z) = (x \star y) \star z$.
- A **monoid** $(M, \star, e)$ is a semigroup whose binary operation has an identity element, that is, there exists an element $e$ such that for all $x$ in $M$, $x \star e = x = e \star x$.
- A **commutative monoid** $(M, \star, e)$ is a monoid whose binary operation is commutative. That is, the multiplication satisfies the axiom that for all $x$ and $y$ in $M$, $x \star y = y \star x$.
- A **group** $(M, \star, e, -)$ is a monoid that has an inverse operation. That is, there is a function $-(\ \ ) \colon M \longrightarrow M$ such that for all $x$ in $M$, $x \star -x = e = -x \star x$.
- A**commutative group** or an **abelian group** $(M, \star, e, -)$ is a group whose binary operation is commutative. That is, the multiplication satisfies the axiom that for all $x$ and $y$ in $M$, $x \star y = y \star x$. Another way to think about it is as a commutative monoid with an inverse operation.
- A **ring** $(M, \star, e, -, \odot, u)$ is an abelian group with another binary associative operation $\odot \colon M \times M \longrightarrow M$ and another identity element $u$ (i.e., $(M, \odot, u)$ forms a monoid) and for which the new operation distributes over the old operation. That means that for all $x$, $y$ and $z$ in $M$

$$x \odot (y \star z) = (x \odot y) \star (x \odot z) \qquad \text{and} \qquad (y \star z) \odot x = (y \odot x) \star (z \odot x). \tag{2.9}$$

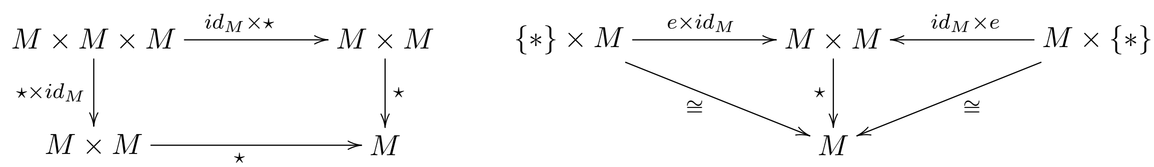
- A **field** $(M, \star, e, -, \odot, u, (\ \ )^{-1})$ is a ring with a partial inverse for the second binary operation. This means that there is an operation $(\ \ )^{-1} : M \longrightarrow M$ which is defined for all $x$ in $M$ except the identity element $e$. The inverse operation satisfies the axiom: for all $x \neq e, x \odot x^{-1} = u = x^{-1} \odot x$.

Monoids will play a major role in this text and it pays to spell out all the details of their definition and to state them with commutative diagrams.

**Definition 2.1.11.** An **monoid** is a triple $(M, \star, e)$ where

- $M$ is a set,
- a multiplication, i.e., a set function $\star\colon M \times M \longrightarrow M$,
- a unit $e \in M$, i.e., a set function that picks out $e$ in $M$ $\upsilon\colon \{*\} \longrightarrow M$.

These ingredients must satisfy the following requirements: $\star$ is associative and $e$ must behave like a unit, i.e. the commutativity of the following two diagrams




(2.10)

**Example 2.1.12.** Figure 2.2 on page 41 has many examples of algebraic structures using the major number systems, $\mathbf{N}, \mathbf{Z}, \mathbf{Q}, \mathbf{R}, \mathbf{C}$, and the major operations, $+, -, \times, (\ )^{-1}$. We will also include examples from positive rational numbers $\mathbf{Q}^+$ and positive real numbers $\mathbf{R}^+$.

There are a few comments about these examples

- Notice that all of our examples of magmas are the same as our examples of semigroups. That is because we are dealing with the usual number operations which are associative. We will see sets and operations that are magmas but not semigroups. (For the reader who is a member of the Illuminati, the octonions, is an example of a number system that is not associative and hence not a semigroup.)
- Notice that the examples for abelian groups are the same as groups. That is because with these number systems, all the operations are commutative. (Examples of structures that are groups but not abelian groups are quaternions and the group of 2 by 2 invertable matrices.)
- Throughout this text we will sometime have operations which are similar to addition. In that case we will write the unit of the operation as 0 and the inverse of that operation as $-(\ )$.We will also have operations which are similar to multiplication. In that case we will write the unit as 1 and the inverse as $(\ )^{-1}$. Things really get hairy when we are talking about rings which have both types of operations.

□

Other common examples of algebraic structures are:

**Example 2.1.13.**

- The set $\mathbf{Z}$ with $-(\ )$ is only a magma because subtraction is not associative.

Magmas:

| | | | | |
|---|---|---|---|---|
| $(\mathbf{N},+)$ | $(\mathbf{Z},+)$ | $(\mathbf{Q},+)$ | $(\mathbf{R},+)$ | $(\mathbf{C},+)$ |
| $(\mathbf{N},\cdot)$ | $(\mathbf{Z},\cdot)$ | $(\mathbf{Q},\cdot)$ | $(\mathbf{R},\cdot)$ | $(\mathbf{C},\cdot)$ |

Semigroups:

| | | | | |
|---|---|---|---|---|
| $(\mathbf{N},+)$ | $(\mathbf{Z},+)$ | $(\mathbf{Q},+)$ | $(\mathbf{R},+)$ | $(\mathbf{C},+)$ |
| $(\mathbf{N},\cdot)$ | $(\mathbf{Z},\cdot)$ | $(\mathbf{Q},\cdot)$ | $(\mathbf{R},\cdot)$ | $(\mathbf{C},\cdot)$ |

Monoids:

| | | | | |
|---|---|---|---|---|
| $(\mathbf{N},+,0)$ | $(\mathbf{Z},+,0)$ | $(\mathbf{Q},+,0)$ | $(\mathbf{R},+,0)$ | $(\mathbf{C},+,0)$ |
| $(\mathbf{N},\cdot,1)$ | $(\mathbf{Z},\cdot,1)$ | $(\mathbf{Q},\cdot,1)$ | $(\mathbf{R},\cdot,1)$ | $(\mathbf{C},\cdot,1)$ |

Groups:

| | | | | |
|---|---|---|---|---|
| | $(\mathbf{Z},+,0,-)$ | $(\mathbf{Q},+,0,-)$ | $(\mathbf{R},+,0,-)$ | $(\mathbf{C},+,0,-)$ |
| | | $(\mathbf{Q}^+,\cdot,1,(\ )^{-1})$ | $(\mathbf{R}^+,\cdot,1,(\ )^{-1})$ | |

Abelian gp.:

| | | | | |
|---|---|---|---|---|
| | $(\mathbf{Z},+,0,-)$ | $(\mathbf{Q},+,0,-)$ | $(\mathbf{R},+,0,-)$ | $(\mathbf{C},+,0,-)$ |
| | | $(\mathbf{Q}^+,\cdot,1,(\ )^{-1})$ | $(\mathbf{R}^+,\cdot,1,(\ )^{-1})$ | |

Rings:

| | | | | |
|---|---|---|---|---|
| | $(\mathbf{Z},+,0,-,\cdot,1)$ | $(\mathbf{Q},+,0,-,\cdot,1)$ | $(\mathbf{R},+,0,-,\cdot,1)$ | $(\mathbf{C},+,0,-,\cdot,1)$ |

Fields:

| | | | | |
|---|---|---|---|---|
| | | $(\mathbf{Q},+,0,-,\cdot,1,()^{-1})$ | $(\mathbf{R},+,0,-,\cdot,1,()^{-1})$ | $(\mathbf{C},+,0,-,\cdot,1,()^{-1})$. |

Figure 2.2: Number systems as examples of algebraic structures.

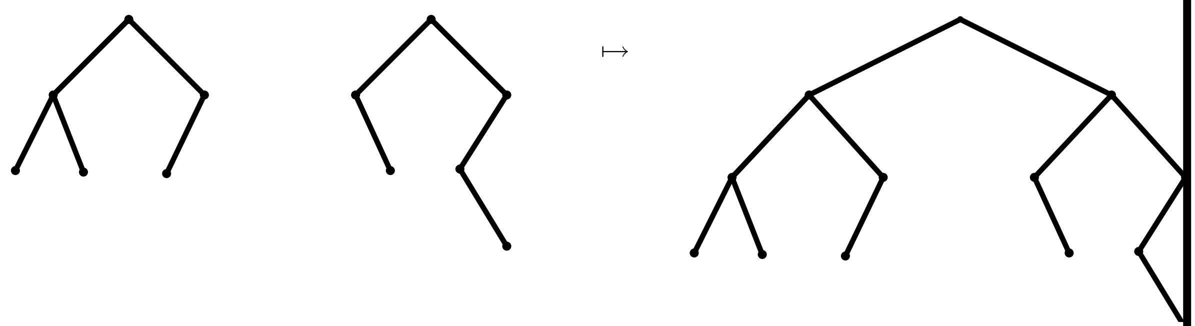


Figure 2.3: An operation on binary trees.

- The set $\mathbf{Q}$ with $(\ \ )^{-1}$ is not even a magma because when $b = 0 \in \mathbf{Q}$ the operation is not defined.
- Take the set of rooted binary trees. These are trees where each node has 0, 1 or 2 children and there is a distinguished root in every tree. This set has a binary operation that takes two trees and connects each root to another root. This operation is described in Figure 2.3. The operation is not associative. This set and operation forms a magma.
- Let $\Sigma$ be a finite set, thought of as an alphabet of symbols. The set of all nonempty strings with symbols in $\Sigma$ is denoted by $\Sigma^+$. This set has a concatenation operation $\bullet$ which is associative. The set and operation forms a semigroup $(\Sigma^+, \bullet)$.
- If we look at all strings in $\Sigma$ including the empty word, then the set $\Sigma^*$ is formed. The empty word, $\emptyset$ acts like a unit to the concatenation operation. That is, any word concatenated with the empty word is itself. This forms a monoid $(\Sigma^*, \bullet, \emptyset)$.

□

An example of a monoid that will arise over and over again is the following.

**Example 2.1.14.** Let $\mathbb{A}$ be any category and $a$ be any object in $\mathbb{A}$, then consider all the morphisms that start and end at $a$, i.e. $Hom_{\mathbb{A}}(a, a)$. Such a morphism that starts and ends at the same object is called an "endomorphism." We write this collection as $End(a)$. This collection forms a monoid. Given $f\colon a \longrightarrow a$ and $g\colon a \longrightarrow a$ we can multiply them as $f \circ g$. The multiplication is ossociative because the composition in $\mathbb{A}$ is associative. The unit is the identity morphism $id_a$ because for all $f$, we have $f \circ id_a = f = id_a \circ f$. Putting this all together we have the monoid $(End(a), \circ, id_a)$ which we call the **endomorphism monoid** of $a$ in $\mathbb{A}$. □

Following Important Categorical Idea 1.5.7, we must discuss morphisms between each of these algebraic structures. For each of these types of algebraic structures there is a notion of a **homomorphism** from one structure to another structure of the same type. If $M$ and $M'$ are of the same type of structure, then a homomorphism $f\colon M \longrightarrow M'$ is a set function from $M$ to $M'$ that "respects" or "preserves" all the operations. For example, if $(M, \star, e, -, \odot, u)$ and $(M', \star', e', -', \odot', u')$ are both