4-20-2020

Inductive Constructions In Logic And Graph Theory

Davis Deaton

Belmont University, davis.deaton@pop.belmont.edu

INDUCTIVE CONSTRUCTIONS IN LOGIC AND GRAPH THEORY

Davis Deaton

A Senior Honors Thesis project submitted to the Honors Program
in partial fulfillment of the requirements for the degree

Bachelor of Mathematics

Belmont University Honors Program

2020

__________________________ Date 4/20/20

Thesis Director

__________________________ Date 4/20/2020

Committee Member

__________________________ Date 4/20/20

Committee Member

Accepted for the Honors Council and Honors Program:

__________________________ Date ___________

Dr. Bonnie Smith Whitehouse, Director
The Honors Program

INDUCTIVE CONSTRUCTIONS IN LOGIC AND GRAPH THEORY FOR THE NON-MATHEMATICIAN

DAVIS DEATON

Abstract. Just as much as mathematics is about results, mathematics is about methods. This thesis focuses on one method: induction. Induction, in short, allows building complex mathematical objects from simple ones. These mathematical objects include the foundational, like logical statements, and the abstract, like cell complexes. Non-mathematicians struggle to find a common thread throughout all of mathematics, but I present induction as such a common thread here. In particular, this thesis discusses everything from the very foundations of mathematics all the way to combinatorial manifolds. I intend to be casual and opinionated while still providing all necessary formal rigor. This way, the content can be as readable as possible while still being complete.

Contents

1. Introduction

1.1. **What Is This Document?** Most mathematics papers aim to produce a new result. Others may aim to introduce an existing result into a new context, but this is rarer. Otherwise, they are generally expository papers, explaining a field of mathematics and what it is for. This thesis is different. My aim is to introduce a structure, compact contractible combinatorial manifolds, without much regard for results about them. Instead, I focus on precisely how these objects can be constructed. I believe this results in a more authentic understanding of what mathematics is. Along the way, I build up all of the relevant mathematics required to rigorously understand each piece of the mathematical puzzle.

Mathematics texts are notoriously difficult to parse on their first read, and this document is no exception. This document differs from standard mathematical texts in a few important ways. When presenting mathematical material, I tend to prefer understandability over convention. Along the way, I make many choices which may seem strange to other mathematicians. Additionally, examples and exercises will be placed along with definitions rather than at the end of sections. I believe this aids readability. Unfortunately, this document is only a brief introduction and so falls short in terms of scope. To fully understand the content, one may need to seek out other texts for additional reading or exercises. I do my best to provide references to such material at the end of sections. Since the majority of this material is mathematical common-place and all of these particular proofs are my own, I do not provide in-text citations. Those seeking a deeper understanding should reference the books cited at the end of subsections.

The target audience of this thesis is and has always been non-mathematicians. The topic, however, has changed. Originally, this thesis intended to prove a new result about tropical mathematics, a form of combinatorial algebraic geometry. I have since realized that this topic is too complicated to serve its purpose. Much of my interest in mathematics comes from connecting algebra and geometry. Tropical geometry achieves this in a particularly beautiful way, but it relies on complicated and continuous underpinnings. Combinatorial manifolds can be presented with only discrete forms of mathematics, dramatically simplifying background material. Further, it allows me to focus on one of my favorite mathematical ideas: inductive constructions.

This text will be a challenge for the non-mathematician. It is not that this content is dumbed-down for the non-mathematician, or that it is presented in casual language. Instead, it is intended for the non-mathematician because it is complete. Or at least more complete than any other text I have seen. It will not help you complete problems. It will not help you solve puzzles. It might not even help you become a better mathematician. It will, however, spell everything out in laborious detail. You may do with it what you wish.

Another thing to note is that this document has a click-able table of contents and click-able references if you have acquired a PDF copy of this thesis.

1.2. **What is Mathematics?** Mathematics is about three things: facts, proofs, and structures.

Most people know what facts are; facts are things which are true. It is also easy to guess that a proof means a sort of formal justification. A structure is some sort of mathematical object which facts can be proved about, for example numbers or shapes. In this paper, we will be learning primarily about proof and structure while being less concerned with learning facts.

Some mathematicians care primarily about facts. Some of these mathematicians are applied mathematicians. Applied mathematicians use mathematical facts to understand our world through physics, predict the spread of diseases, design public infrastructure, or something else along these lines. It is straightforward to understand the motivation of these mathematicians and the work that they do, for these mathematicians have a direct impact on the material world. There are some non-applied mathematicians who still care primarily about facts. These mathematicians might inherently care about numbers or shapes and want to know facts about them whether or not there are material implications for these facts. Perhaps surprisingly, not all mathematicians agree on the facts. In particular, not all mathematicians agree on what it means for something to be true.

The primary distinction between what is and isn't a fact is the distinction between classical and constructive mathematics. In both camps, new facts are derived from axioms and proofs. Axioms are statements which are claimed to be true without proof. These are our base truths or our assumptions. A constructive mathematician tends to believe that a true statement is exactly one which can be proved from the axioms and that a false statement is exactly one which can be proved to contradict the axioms. A classical mathematician tends to believe that a statement is true if it can be proved from the axioms but that other statements might also be true. This difference actually boils down to one axiom: the law of the excluded middle (or one of its equivalent statements). The law of the excluded middle claims that every statement is either true or false, not both, and not somewhere in-between. Classical mathematicians accept the law of the excluded middle and use it in their proofs. A constructive mathematician does not accept the law of the excluded middle (but also does not reject it). All mathematicians agree that not every statement can either be proved or disproved, and since these are what a constructive mathematician means by true, a constructivist would not accept that every statement is either true or false.

This distinction bleeds into what mathematicians consider a proof. Some mathematicians care primarily about proofs. These mathematicians are logicians or those close by. They want to know precisely how we know what we know about mathematics. Classical mathematicians consider a proof to be a particular sort of formal justification. Constructive mathematicians consider a proof to be a construction of a particular term, a sort of certificate which verifies the truth of the claim. Consequently, constructive mathematicians are pickier about what counts as a proof. Although I tend to believe that constructive mathematics is the better of the two perspectives, it is less natural and more confusing. Thus, in this document, we will be presenting a classical perspective. However, since induction is arguably the primary mechanism of constructive mathematics, many of our methods will parallel theirs.

Further, some mathematicians care primarily about structure. These mathematicians perhaps pick a particular object to study, such as groups, manifolds, or differential equations. Or perhaps these mathematicians study connections between different objects, like representations or categories. Here, we will show that many of these structures share a fundamental commonality. Somehow, all of these different types of mathematicians are able to find enough commonality to collaborate and build upon each other's work. Together, mathematicians have created a rich and beautiful world, full of open questions. I hope to show you the smallest slice of this wonderland.

1.3. **What is Induction?** Mathematical induction refers to two related concepts: inductive constructions and induction principles. Intuitively, inductive constructions define a mathematical object by prescribing atoms, the smallest version of the construction, and recipes to create new constructions from other objects, which can be thought of as ingredients. For example, the food can make in my apartment is an inductive construction. The atoms refer to the ingredients I have on hand, and the recipes describe how to combine ingredients, which could be atoms or other recipes, into new food. The fact that the construction is inductive means that the atoms and the recipes with ingredients I can make constitute the entirety of the foods I can make. That is, I can't make anything else. For example, milk and cereal are atoms, and I can use a (simple) recipe to create cereal with milk from them. The ingredients of cereal with milk would be, unsurprisingly, cereal and milk. Or, if I can make a pie crust and a pie filling, I can use a recipe to turn them into a pie, and if I can't make a pie crust or I can't make a pie filling, then I can't make a pie. If I made a pie, ingredients would be the pie crust and the pie filling, which would each have their own ingredients.

Suppose I have made a pie, and I want to know if it is gluten free. If the pie crust and the pie filling (the only ingredients of the pie) are gluten free, then the pie is gluten free. Similarly, if all of the ingredients of the pie crust are gluten free, then the pie crust is gluten free. In general, if all the ingredients of a recipe are gluten free, the resulting recipe is gluten free. The induction principle of the food I can make in my apartment says that if all of the atoms (all of the ingredients I have in my apartment) are gluten free, then all food I can make in my apartment is gluten free.

In general, each inductive construction has an induction principle. The induction principle states that for any property, if every atom has that property, and if the ingredients having the property guarantees that the recipe does too, then every object has the property. In our case, if every atom is gluten free, and gluten free ingredients guarantee gluten free recipes, then all of the food is gluten free.

Induction principles can make mathematical structures much easier to work with. Rather than trying to understand every food I could possibly make, we only need to understand the atoms and the nature of recipes to demonstrate a property of all the foods I can make. Further, inductive constructions can be easily defined in terms of a sort of recipe book. This is arguably easier to understand than other sorts of definitions. Additionally, an induction principles comes "for free" with every inductive construction, so even in the presence of a direct definition, mathematicians may still prefer the inductive definition.

As inductive constructions are the focus of this thesis, nearly every definition I present is an inductive construction. Thus, you will get plenty of experience with them as you read this thesis.

1.4. **Summary of the Material Presented.** First, we discuss what it is like to read mathematics. This section focuses on the important components of all mathematical writing. Next, we introduce formal logic. This section will prepare you to discuss mathematics in rigorous detail. Then, we introduce set theory. Sets are the fundamental structure upon which all other classical structures are built. After set theory, we end with a discussion of different mathematical structures. In particular, we will focus on graphs, simplicial complexes, and finally compact contractible combinatorial manifolds.

2. Reading Mathematics

2.1. **Formal Languages.** Mathematics is organized so that it is "easy" to be written and read. For the most part, formal mathematics does not rely on using or interpreting pictures; people can easily make mistakes with pictures. Instead, formal mathematics uses a *formal language*.

Definition 2.1.1. A formal language is two things: a collection of valid *symbols* called the *alphabet* and a collection of valid *strings* of those symbols in the alphabet called the *words*.

Here, symbols refers to Latin letters (a, b, x, A, etc.), Greek letters (α, β, φ, Φ, etc.), punctuation ('.', ',', etc.), grouping symbols ('(', '}', etc.), and mathematical symbols like ($+$, $\forall$, $\Rightarrow$, etc.), and a string refers to a finite sequence of symbols written one after the other. Since I do not expect every reader to be familiar with mathematical notation or the Greek alphabet, I explain each new letter or symbol as they appear.

An important omission from this list is white-space characters like spaces, tabs, or newlines. Generally, these characters have no mathematical meaning and are used only for clarity. Mathematicians are free to place them wherever they wish without changing the meaning of any formal word. Although in English, the strings 'a periodic' and 'aperiodic' are distinct, they would not count as distinct formal words. Additionally, 'a periodic' counts as a single formal word, even if it is literally two English words.

In the preceding two paragraphs, I placed single quotes around the punctuation and grouping symbols as well as formal words. Generally, italics will be used to suggest that letters such as a are formal rather than being part of the prose, but often, formal words are placed in single quotes to emphasize that they are not part of the prose. For example, due to single quotes, it is especially clear that a sentence containing a formal word such as 'a.' does not actually end at the period in the formal word. It is important to note that the single quotes are *not* part of the formal word.[1]

Strictly speaking, a language is not required to provide a way to create valid words, only a collection (or rule) for determining which words are valid. However, many languages provide *production rules* for creating words from other words. Let us now describe a simple language.

Example 2.1.2. For this example, our alphabet will consist of the letters a and b. We will also have two production rules:

(1) The empty string '' is a word.
(2) If W is a word, then W'a' is a word.

The first rule tells us how to create our first word: ''. The second rule tells us how to create many more words from our first word. It says that if we have a word, we can stick an a on the end, and that's also a word. In particular, because '' is a word, so is a, for the string a is an 'a' written next to a nothing. Since a is a word, so is aa, and aaa, and $aaaa$, and so on. In fact, the words in our language are exactly the finite (possibly empty) sequences of the letter a repeated.

Languages created with production rules are an example of an *inductive construction*. That a language is inductive means that *only* those words which can be created from the rules are valid words. For example, even though b is in our alphabet, it does not appear in any of the words of the language because neither of the two production rules include the letter b.

Exercise 2.1.1. Modify the production rules in the previous example to create a language where every word contains exactly one b and arbitrarily many a's.

Example 2.1.3. A more complicated example is the language of *Dyck words*, named after the mathematician Walther von Dyck. Here, our alphabet will have the two letters '(' and ')' and the words will be those which represent valid configurations of parentheses. For example, ()()(()()) is a valid Dyck word but ()) is not, since the final parenthesis has no matching opening parenthesis.

[1]Further, punctuation serving a purpose in the prose will not be moved inside single quotes as if the formal word is quoted text.

Although this is a rule for determining which words are valid, we can also describe Dyck words with production rules:

(1) ‘’ is a Dyck word.
(2) If W is a Dyck word, (W) is a Dyck word.
(3) If V and W are Dyck words, VW is a Dyck word.

As before, the first rule says that the empty string is a Dyck word. The second rule says that an open parenthesis followed by a Dyck word followed by a closed parenthesis is a Dyck word. The third rule says that two Dyck words written next to each other are a Dyck word. It should be relatively straightforward to convince yourself that these production rules provide exactly the valid configurations of parentheses, but if not, you can take this as the definition of a valid configuration of parentheses.

Importantly, it is actually possible to define the same language with fewer production rules:

(1) ‘’ is a Dyck word.
(2) If V and W are Dyck words, $(V)W$ is a Dyck word.

It is not immediately obvious that these two sets of production rules create the same language, but they do.

Proof. First, realize that rule 2 of the second description can be achieved in the first description by applying the original rule 1 to the word V and then the original rule 2 to the new word (V) and the word W. This means that every word that can be constructed with the second description can be created with the first description.

Second, consider a Dyck word W in the sense of the first description. If W is the empty string, it is a Dyck word according to rule 1 of the first description. Otherwise, W must start with a ‘(’. In that case, find the matching ‘)’. What is inside those parentheses must be a Dyck word, and what follows it must also be a Dyck word. Thus, it can be constructed using the rule 2 of the second description. This means that every word that can be constructed with the first description can be created with the second description.

Altogether, this shows that the two sets of rules describe the same language. That is, every word of one is a word of the other. □

The preceding proof is not, technically, a formal proof. However, it does still justify the claim I made: that these description of Dyck words are the same. Thus, it is a *prose proof*, which is a proof written to be read by regular humans rather than computers or extremely picky humans. These proofs are very common in this thesis, but most of the prose proofs presented can be easily formalized. In mathematics, proofs generally end with a cute little square. This denotes that the proof is over and the desired claim has been proven.

2.2. **Variables and Substitution.** In the previous subsection, we used *variables* such as V and W to denote arbitrary words. In effect, W and V were *names* of words rather than being words themselves. That is, when I wrote "If W is a word, then W'a' is a word," I did not mean "If the formal word 'W' is a word, then so is the formal word Wa." Hopefully, this was clear because W was not even in the alphabet of the languages discussed. Instead, I meant for W to represent an arbitrary word, such as *aaaa*. For example, if I say "Let W be a Dyck word," you should imagine some actual Dyck word, like ()() taking the place of W. Then, I might say "$(W)(W)$ is a Dyck word." When interpreting that sentence, you would literally transcribe your imagined word, ()() in place of W. You would receive the sentence "(()())(()()) is a Dyck word," which is true.

Consequently, it can be important to pick a few letters outside your formal language to represent variables, but sometimes there is overlap. There are, however, common conventions to help tell the difference between formal letters and variables. Generally, certain letters will refer to certain types of mathematical objects. Previously, we used V and W for words because 'word' starts with 'w' and 'v' is close to 'w'. Similarly, we will often use n and m to denote numbers and f and g to denote functions. Famously, mathematicians also use variables x and y to denote numbers. Try to detect such patterns when they are not mentioned explicitly, for knowing them makes mathematics much easier to read and understand.

Often, particular words in a formal language will be too long to write repeatedly. In this case, we may write '$W\colon$(()())(()())' to mean that the variable W is shorthand for the word (()())(()()). This ':' may be read as "is defined to be." After this shorthand is established, one should mentally substitute every occurrence of W for what it is shorthand for. That is, one should think of $WWWW$ as (()())(()())(()())(()())(()())(()())(()())(()()). However, with this presentation, it is obvious that $WWWW$ is a Dyck word (we have a rule for writing Dyck words next to each other), while it would not be obvious that (()())(()())(()())(()())(()())(()())(()())(()()) is not missing a closed parenthesis somewhere. Thus, sometimes using shorthand can help emphasize the internal structure of complex words. To reiterate, once a shorthand is established, every occurrence of that shorthand should be mentally replaced with whatever it is shorthand for. Shorthand does not have meaning in the original language.

This means that when we write mathematics, we are often working in a meta-language. That is, when we write 'W: (()())(()())', we are writing a word in a meta-language whose words contain variables like W and punctuation like ':' as well as the original words. We will speak more about these meta-languages later.

2.3. Sequences and the Natural Numbers.

Informal Definition 2.3.1. The natural numbers, also known as the counting numbers, are the numbers 0, 1, 2, 3, etc.

We will formally describe these numbers later on. It is expected that the reader knows of the natural numbers and their arithmetic including addition, subtraction, multiplication, and division. Specific calculations will generally not be required.

Exercise 2.3.1. Create a formal language whose alphabet is the numerals $0 - 9$ and whose words are the natural numbers. Note, '01' should not be a word in your language.

The natural numbers are important because they let us represent *some amount* of things. This is important in dealing with large numbers of variables or with *sequences*.

Suppose W and V are Dyck words. Then, we established that VW is a Dyck word. If U is another Dyck word, then so is UVW. Suppose I wanted to add another Dyck word to this pattern. Which letter do I choose for my next variable? T? And what happens if I want to combine more than 26 Dyck words? It is hopefully clear that 26 Dyck words written next to each other is a Dyck word, so it would be nice to have a way to write this fact without using each individual letter of the Latin alphabet.

For a limited number of Dyck words, say 5, we can write the following:

> Suppose that W_1, W_2, W_3, W_4, and W_5 are Dyck words.
> Then, $W_1W_2W_3W_4W_5$ is a Dyck word.

Here, the small numeral placed after and below the letter is called a *subscript* or an *index*. Using indexed variables allows us to avoid using too many letters and to group similar variables with similar names. Subscripts can either be natural numbers or letters.

Further we can have a sequence of 26 Dyck words. In particular, we may write:

> Suppose that $W_1, W_2, \ldots, W_{26}$ are Dyck words. Then, $W_1W_2 \cdots W_{26}$ is a Dyck word.

Here, we have a sequence of 26 variables, but we have omitted most of them using ellipses. These ellipses should generally be read as "and so on." The first time we reference such a sequence, we generally write the first two terms and the last term. After the first time, we may choose to omit the second term, or sometimes even the last term if clarity permits. It is also possible that we write sequences of actual mathematical objects like this, for example $0, 2, 4, 6, 8, \ldots, 20$. In such cases, we write enough terms for the pattern to be clear. To be fully formal, the pattern must be specified somehow.

We can even use sequences with variable size. That is, we may write:

> Suppose that $W_1, W_2, \ldots, W_n$ are Dyck words. Then, $W_1 \cdots W_n$ is a Dyck word.

This even holds when n is 0, in which case $W_1, \ldots, W_n$ is the *empty sequence*, so $W_1 \cdots W_n$ is the empty string, which is a Dyck word. If some statement is only valid for nonempty sequences, that will generally be specified. For clarity, we may explicitly mention that we intend to include the empty sequence. In general, assume the empty sequence is included.

The number n is called the length of the sequence. We are also free to choose another letter, so that $V_1, \ldots, V_m$ denotes a sequence of length m. If we have another sequence, $W_1, \ldots, W_m$, this sequence has the same length, m. Thus, if two sequences are referenced with the same final subscript, they will have the same length. If they do not have the same final subscript, they may or may not be the same length.

Sometimes, we need to reference a particular term of a sequence. For example, suppose that we have a sequence of numbers $k_1, k_2, \ldots, k_n$ and we want to require that the third one is even. In that case, we could say k_3 is a even. Of course, this would only make sense if the length of the sequence, n, is at least 3. Similarly, if we need to reference any value in the sequence, we can write k_i. Here, it is implied that $1 \leq i \leq n$. For example, we could require that some k_i be even or that every k_i is even. An example statement is

> If $k_1, \ldots, k_n$ are natural numbers and at least one of the k_i are even, then the product $k_1 \cdot k_2 \cdots k_n$ is even.

2.4. **Syntax vs. Semantics.** So far, we have been describing mathematical *syntax*. Syntax describes which things we are allowed to write, but says nothing about what they mean. For example, in the original language we described, the strings of repeated a's had no intrinsic meaning (except perhaps as a representation of the noises you will make reading this thesis). The *semantics* of a language describes how we are supposed to interpret the words of the syntax. In general, a formal language has no meaning whatsoever.

Let us look at a more complicated example of a language.

Example 2.4.1. We will describe a new kind of language: the language of arithmetic expressions. Arithmetic expressions will have multiple different types of words. First, a (formal) number is defined by:

(1) A digit 1-9 is a number.
(2) If N is a number and d is a digit 0-9, Nd is a number.

Then, an arithmetic expression is defined by:

(1) A number is an arithmetic expression.
(2) If E_1 and E_2 are arithmetic expressions, then $E_1 + E_2$ is an arithmetic expression.
(3) If E_1 and E_2 are arithmetic expressions, then $E_1 \cdot E_2$ is an arithmetic expression.

Notice that we did not specify the alphabet of this language. In such cases, the alphabet presumed to contain all the symbols appearing in the production rules. In this case, the alphabet is the numerals 0-9 and the operations $+$ and $\cdot$.

These words encode arithmetic expressions consisting of numbers, addition, and multiplication. The numbers in this language are called formal because they are only strings of numerals not starting with a 0. That is, they are not (yet) interpreted as actual natural numbers. Expressions include things like $2+3 \cdot 6+30$. For example, the digit '1' is a number, so the word '10' is a number, so the word '10' is an expression, so '$10 + 10$' is an expression. Each of these "so"s represents a use of one of the production rules. Similarly, the digit '2' is a number, so the word '20' is a number, so the word '20' is an expression. However, as expressions, '$10+10$' is *not* equal to '20'. The language we have defined has no *semantics*. That is, the expression '$10 + 10$' does not have any meaning, at least that we have described so far. In fact, the expression '20' is not even the natural number 20. There is no preexisting meaning to the strings of a formal language.

We could give these strings meaning by defining an *interpretation*, which is to say some way of understanding what the strings represent. Unfortunately, with this definition of an expression, we probably shouldn't define semantics even if it seems like we easily could. Here is why: suppose that we define $E_1 \colon 10 + 2$ and $E_2 \colon 2 \cdot E_1$. Then, by substituting $10 + 2$ for E_1 in the definition of E_2, we find that E_2 is $2 \cdot 10 + 2$. We would interpret this value as the number 22 because we perform multiplication before addition. However, if we first interpret E_1 as 12, then we would find that E_2 is $10 \cdot 12$ which is 120. That is, it matters in which order we interpret our expressions. In other words, our semantics do not play nice with our production rules.

Fortunately, there is a simple solution. Let us create a new language of arithmetic expressions whose production rules are:

(1) A number is an arithmetic expression.
(2) If E_1 and E_2 are arithmetic expressions, then $(E_1 + E_2)$ is an arithmetic expression.
(3) If E_1 and E_2 are arithmetic expressions, then $(E_1 \cdot E_2)$ is an arithmetic expression.

where numbers refer to the same words as before. For clarity, we will refer to the previous language of arithmetic expressions as the language of simplified arithmetic expressions. These production rules are exactly the same words as before, except now there are parentheses around the second two production rules. Now, our problematic word $2 \cdot 10 + 2$ is no longer an acceptable expression. Instead, we have the word $(2 \cdot (10 + 2))$ and the word $((2 \cdot 10) + 2)$. This language has semantics that do not depend on order. That is, the semantics play nice with the production rules. In particular, we can interpret each arithmetic expression as representing a number.

Let us be annoyingly explicit about this process. To each arithmetic expression E, we define a number EI, which is read "E I" or as "E's interpretation." Try to get used to reading the notation out loud to yourself. That is, each time you see EI, try thinking "E's interpretation" until you get used to the notation. Do this for each of the pieces of notation that we define.

If E is an expression, then EI is the number that it represents. Since arithmetic expressions can only be created by 3 rules, we can define three rules to fully specify the interpretation.

(1) If E is a number n, then EI is the number n.
(2) If E is the expression $(E_1 + E_2)$ for expressions E_1 and E_2, then EI is the number $E_1I + E_2I$.
(3) If E is the expression $(E_1 \cdot E_2)$ for expressions E_1 and E_2, then EI is the number $E_1I \cdot E_2I$.

Let us work an example. Let's interpret the expression[2]

$$E\colon (2 \cdot ((1 + 1) + (3 \cdot 2))).$$

That is, we seek to find the number

$$(2 \cdot ((1 + 1) + (3 \cdot 2))I.$$

Here, E is of the form $(E_1 \cdot E_2)$ where $E_1\colon 2$ and $E_2\colon ((1 + 1) + (3 \cdot 2))$. Thus, EI is the number $E_1I \cdot E_2I$. Since E_1 is the number 2, we know that E_1I is the number 2. Determining E_2I takes more work. E_2 is

[2]Remember to read this as "E is defined to be. . . " or "E is shorthand for. . . ."

of the form $(E_3 + E_4)$ where $E_3\colon (1+1)$ and $E_4\colon (3 \cdot 2)$. Thus, $E_2 I$ is the number $E_3 I + E_4 I$. As an exercise, convince yourself the $E_3 I$ is 2 and $E_4 I$ is 6. Substituting these values, we find that $E_2 I$ is the number $2+6$ which is 8. Finally, since EI is $E_1 I \cdot E_2 I$, we can now substitute the values of 2 and 8 for $E_1 I$ and $E_2 I$ respectively to find that EI is $2 \cdot 8$ which is 16.

It is important that we are not able to say that E "is equal to" 16. As expressions, E is not equal to 16, because E is the expression

$$E\colon (2 \cdot ((1+1) + (3 \cdot 2))),$$

which is not the expression 16. Expressions are only equal when they are literally the same string of characters. Instead, we can say that EI is equal to 16, but I even recommend against that; just say that EI is 16.

This interpretation of expressions as numbers required already understanding what we mean by $+$ and $\cdot$. Although we could have rigorously defined $+$ and $\cdot$, it is not always possible to provide a rigorous definition of interpretations. In fact, since interpretations live outside the language being interpreted, some interpretation will always be intuitive rather than formal.

We mentioned earlier that when we make statements such as "If W represents the Dyck word $W\colon ()()$, then $W(W)W$ is a Dyck word," we are actually using a meta-language, a language whose words consist of the words of our original language as well as variables such as W. This is a perfect example of the use of semantics. When we say $W\colon ()()$, we mean that W should be interpreted as $()()$. In other words, we are saying that WI is the Dyck word $()()$. Thus, although I have written $()W()$ and WW differently, they are both interpreted as the Dyck word $()()()()$. Thus, we can say that their interpretations are the same. This is why I caution against making claims like "$()W()$ is equal to WW." While this is true in the language of Dyck words, it is not true in the meta-language of Dyck words and variables. In general, however, if a statement of equality is made, it should be interpreted in the smallest language possible.

2.5. **Grouping Symbols and Order of Operations.** Although it may seem like mathematicians really like to write things, mathematicians are also obsessed with shorthand that will allow them to not write things. For example, we wrote

$$E\colon (2 \cdot ((1+1) + (3 \cdot 2))).$$

The parentheses here, while necessary for having a consistent interpretation, are distracting and reduce readability. This is generally true of other grouping symbols such as [] and $\{\}$.

It would be nice, for example, to simply say

$$E\colon 2 \cdot (1 + 1 + 3 \cdot 2).$$

This is possible, if we define two types of rules: *associativity rules* and an *order of operations*. Together, these rules explain how to place parentheses in shortened expressions like our new definition of E. In our language of expressions, words such as $3 \cdot 2$ and $1 + 1$ are not valid, only $(3 \cdot 2)$ and $(1 + 1)$, but the un-parenthesized words appear in our latest definition of E. The order of operations states which parentheses we should replace first, those around $+$ or those around $\cdot$. The order of operations of arithmetic is that $\cdot$ comes before $+$. That is, in an expression like $1 + 2 \cdot 3$, we first place parentheses around the $\cdot$ sub-expression to get $1 + (2 \cdot 3)$ and then we place parentheses around the $+$ sub-expression to get $(1 + (2 \cdot 3))$, which is a legitimate expression. The associativity rules help with expressions like $1+2+3$. Should this represent the expression $((1 + 2) + 3)$ or $(1 + (2 + 3))$? Although we can agree that these are interpreted the same, these are *not* equal as expressions, so it *does* matter which one we pick in most languages. The associtivity rules of arithmetic expressions say that $+$ and $\cdot$ are *left associative*, which means that we place paretheses around the left-most $+$ and $\cdot$ first. That is, for the expression $1 + 2 + 3$, we first place the parentheses around the left-most $+$ sub-expression to get $(1 + 2) + 3$, then the next $+$ to the right to get $((1 + 2) + 3)$, which is a legitimate expression. Sometimes, parentheses are already placed around sub-expressions. In such cases, simply leave them there.

Exercise 2.5.1. Place parentheses in the following expressions according to the order of operations and associativity of arithmetic expressions.

(1) $1 \cdot 2 \cdot 3$
(2) $1 + 2 + 3 + 4$
(3) $1 + 2 \cdot 3$
(4) $1 \cdot 2 + 3$
(5) $1 + 2 \cdot 3 \cdot 4 + 5$
(6) $(1 + 2) \cdot 3 \cdot (4 + 5)$

Unfortunately, grouping symbols often get in the way of substitutions. That is, suppose $E_1\colon 1+2+3$ and $E_2\colon 2\cdot E_1$. Before, when we substituted variables for words, we were able to transcribe them directly. That is, we might that that E_2 is $2 \cdot 1 + 2 + 3$. This, however, is false. When substituting a variable for a shortened expression, *you must put the parentheses back before you substitute.* That is, even though we said that $E_1\colon 1 + 2 + 3$, E_1 is actually the expression $((1 + 2) + 3)$ according to our associativity rules. Thus, E_2 is actually $2 \cdot ((1 + 2) + 3)$ or $(2 \cdot ((1 + 2) + 3))$. This is as desired. Technically speaking, you only need to add back in the outer-most parentheses, but it is easier to claim that we must replace all the shorthand.

2.6. **Free Constructions.** For both Dyck words and arithmetic expressions, we defined two separate sets of production rules. In the case of Dyck words, both sets of production rules resulted in the same language. In the case of arithmetic expressions, the two sets of rules generated two different languages. In both cases, however, the first rules were *non-free* and the second rules were *free*. Look back at the two definitions of arithmetic expressions. There is a common form between them. Both are three rules: one taking a number and yielding an expression, and two taking two expressions and yielding an expression. These can be given by a *signature*.

Example 2.6.1. The signature of arithmetic expressions is

$$(N \to E, E^2 \to E, E^2 \to E).$$

This notation means that one rule takes a number (N) and yields ($\to$, read as "to" in this context) an expression (E), the next rules takes two expressions (E^2, read as "E two" or "E squared") and yields ($\to$) an expression (E), and the last rule takes two expressions (E^2) and yields ($\to$) an expression (E). Both production rules have this same signature. This means that they are related as languages. We can give each rule in the signature a name. This is called a *tagged signature*. For example,

$$(i\colon N \to E,\ A\colon E^2 \to E,\ M\colon E^2 \to E).$$

Here, the tags are i, A, and M. The colons are sometimes read as "from" in this context, so this reads "i from N to E, A from E two to E, M from E two to E." The *free language* on this tagged signature is given by the following production rules:

(1) If n is a number, ni is an free word.
(2) If w and v are free words, wvA is a free word.
(3) If w and v are free words, wvM is a free word.

For example $0i1iA2iM$ is a free word. The signature of these production rules is exactly the signature we used to create it. The most important property of the free language is that there is *exactly one* way of creating a word from the production rules. The second most important property of the free language is that it can be interpreted into *every* language with the same signature. Although free words are generally obtuse, these properties will reveal great insight into their nature.

Here, there are three rules for creating a free word: i for the inclusion of numbers into free words, A for the addition of two free words, and M for the multiplication of two free words. We can interpret free words into arithmetic expressions as follows:

(1) If w is of the form ni for a number n, wI is the number n treated as an arithmetic expression.
(2) If w is of the form w_1w_2A for words w_1 and w_2, then wI is the expression $(w_1I + w_2I)$. Hence, one can read the word w_1w_2A as "w_1 and w_2 added."
(3) If w is of the form w_1w_2M for words w_1 and w_2, then wI is the expression $(w_1I \cdot w_2I)$. Hence, one can read the word w_1w_2M as "w_1 and w_2 added."

As an example, the free word $0i1iA2iM$, read as "0 included and 1 included added, and 2 included multiplied," is interpreted as $((0+1)\cdot 2)$. Notice that the numbers and operations appear in the same order as in the free word.

In free words, the operations A and M are placed at the end of the two words they effect. This is called *post-fix* notation. The notation w_1+w_2, where the operation is placed in between the words it effects, is called *in-fix* notation. The advantage of post-fix notation is that allows consistent interpretations without parentheses. The disadvantage of post-fix notation is that it is hard to read.

Importantly, the process of interpreting free words as arithmetic expressions is *reversible*. For every arithmetic expression E, there is exactly one free word w such that wI is E. This is equivalent to the statement that there is one and only one way to make an expression E from the production rules. This is *not* the case for simplified arithmetic expressions. For example $0i1iA2iA$ would be interpreted as $0 + 1 + 2$, but so would $0i1i2iAA$. Consequently, the simplified arithmetic expressions are called *non-free* while the proper arithmetic expressions with parentheses are free.

Example 2.6.2. Let us also consider the two sets of production rules for Dyck words. The first one has tagged signature

$$(E\colon D^0 \to D, W\colon D^1 \to D, J\colon D^2 \to D).$$

This says that the first rule, E, takes no Dyck words (D^0) and yields a Dyck word. This is the rule that the empty string is a valid Dyck words. Then, the second rule, W, takes a Dyck word (D^1, which we can also write as simply D) and yields another Dyck word. This is the rule that wraps a Dyck word in parentheses. Finally, the third rule, J, takes two Dyck words (D^2) and yields another Dyck word. This is the rule that joins two Dyck words by writing one after the other.

The production rules for this language are straight forward:

(1) E is a free word.
(2) If w is a free word, wW is a free word.
(3) If w_1 and w_2 are free words, w_1w_2J are free words.

Like before, these free words have an interpretation into actual Dyck words. For example, $EWEWJ$ interpreted as the Dyck word "the empty string wrapped and the empty string wrapped joined" or ()().

Exercise 2.6.1. Write down the three rules for interpreting these free words as Dyck words.

The three rules for generating Dyck words are *not* free. This is because $EWEWJEWJ$ and $EWEWEWJJ$ are both interpreted as ()()().

On the other hand, the second set of production rules for Dyck words had the signature $(E\colon D^0 \to D, S\colon D^2 \to D)$. These production rules *do* turn Dyck words into a free language.

Exercise 2.6.2. Write the production rules for the free language on $(E\colon D^0 \to D, S\colon D^2 \to D)$ following the pattern before. Then, write the two rules for interpreting these free words as Dyck words. Convince yourself that this process is reversible.

Example 2.6.3. The most important free language is the one on the signature $(0\colon N^0 \to N, S\colon N \to N)$. It's production rules are:

(1) 0 is a free word.
(2) If w is a free word, wS is a free word.

The valid words are 0, $0S$, $0SS$, $0SSS$, and so on. These words can be interpreted as natural numbers with the following rule.

(1) If w is of the form 0, then wI is the number 0.
(2) If w is of the form vS for a word v, then wI is the number $vI + 1$.

For example, $0I$ is the number 0 and $0SSSSI$ is the number 0+1+1+1+1 which is 4, because there are four S's. This is why we named the rule $S : N \to N$; it stands for successor.

Exercise 2.6.3. Convince yourself that this interpretation of free words into numbers is reversible.

2.7. **Recommended Reading.** A gentler and deeper understanding of mathematical writing is presented in Devlin[2]. Devlin's goal is similar to mine, but he focuses less on rigor and more on concepts. Consequently, his book is a good companion to this thesis. He will go into topics that I do not cover, like the real numbers, but misses out on the graph theoretic topics that we will cover.

For a more complete description of languages (and their applications), I recommend Webber[11]. However, if you are new to mathematics, I recommend moving forward in this thesis before exploring formal languages further. Instead, if you progress in this thesis, and also in some of the other recommended readings, and you are still interested in learning more about languages, find this book.

3. Predicate Logic

Now that we have talked about languages at length, we are ready to introduce the language of mathematics: logic. Generally, mathematicians learn a significant quantity of advanced mathematics before taking a course in formal logic. Similarly, mathematicians learn a significant quantity of advanced mathematics before they learn about formal languages. You, if you are a non-mathematician, have the unlucky opportunity of learning languages and logic first. I have chosen this nonstandard ordering because I am choosing to focus on induction as a mathematical principle rather than a particular subject of mathematics.

Like most mathematical theories, predicate logic has two pieces: syntax and semantics. Different logics have different syntax or different semantics. Here, we will present the predicate logic of set theory. The words of predicate logic are called *propositions*, also called *logical formulae*. Semantically, propositions are supposed to represent sentences depending on some number of formal variables which are either true or false when values are plugged into the variables. Alternatively, propositions are supposed to represent the mathematical statements that can be proved or disproved.

Let us look at some informal examples of propositions. "b is reading this thesis" is a sentence which is either true or false depending on the value of b, and thus, it is a proposition. Here, b is the variable. When b is interpreted as "you", the sentence becomes true because you are reading this thesis. However, when b is interpreted as "me", the sentence becomes false because I am not reading this thesis (at least right now, probably).

A proposition can depend on multiple variables. For example, "a is the father of b" is a proposition depending on the variables a and b, and "$c + d = 1$" is a proposition depending on the variables c and d. Unsurprisingly, there are propositions depending on 3 variables such as "p_1 and p_2 went to school together at s," which depends on the variables p_1, p_2, and s.

There are also propositions which depend on *no* variables, which are called *logical sentences* or just *sentences*. For example, "you are reading this thesis" or "Davis Deaton is not the author of this thesis" are both sentences; the former being true and the latter being false. Because a proposition is a single formal word in the language of predicate logic, a sentence, a proposition depending on no variables, is also a single formal word.

Every proposition becomes a sentence when particular values are substituted for its variables. For example, while $a+b = 1$ is a proposition of 2 variables, but as soon as we agree on values for both a and b, like perhaps a is 1 and b is 2, the predicate becomes the sentence $1 + 2 = 1$.

Before we can make any sense of predicate logic, we need to agree upon a *domain of discourse*. A domain of discourse consists of two things: the mathematical objects which our variables are allowed to represent and the fundamental relationships between these objects, which are called *predicates*. The objects in the domain of discourse of this thesis, and that of most of modern mathematics, is *sets*.

Informal Definition 3.0.1. A *set* is an unordered, possibly empty, collection of objects in our domain of discourse, called *elements* counted without repetition. Sets have one fundamental relationship, written $\in$ and read as "is an element of." If x is an element of y, we write $x \in y$.

If our domain of discourse were material objects, then we could form collections such as "the collection of all people" or "the collection of all blue things" or "the collection of everything you own" and declare them to be sets. The elements of the collection of all people are individual people. These people have no inherent ordering to them; they are simply in the collection. Further, consider the collection of dog owners. This collection only contains each dog owner one time, even if they own multiple dogs; that is, sets do not keep track of repetition or how many times an element is included in the set. There should also probably be an empty collection with no elements. Using these examples, you should make sure you have an intuitive notion of sets. That is, which things are and are not sets? Because the objects in our domain of discourse are sets, the elements of sets are precisely other sets. So our sets could include an empty set, and a set containing that empty set, and a set containing both that set empty set and also the set containing that empty set.

As marked, this definition of a set is informal. Since our domain of discourse lies outside predicate logic, sets cannot be fully defined. Consequently, you can think of sets however you want. We will however, require certain properties of your conception of sets. This will be the topic of the next section.

3.1. **Propositions.** Similar to the meta-languages we discussed before, the syntax of formal logic includes variables. Generally, variables within the language of logic will be lowercase Latin letters early in the alphabet, such as a, b, c, d, possibly with subscripts (a_1, b_6, and the like). However, we will also use lowercase Latin letters, generally those towards the end of the alphabet such as x, y, z to be *meta-variables* referring to the name of an unknown variable. That is, the meta-variable x may refer to the variable a or the variable c_{14}. In all cases, the distinction will hopefully be clear. If any statement is made of the form "if x is a variable" or the like, it means that x (or whatever other symbol) is a meta-variable referring to some unknown variable such as a or b. Lowercase Greek letters such as φ, ψ, and χ will be used[3] as meta-variables representing the words of formal logic, which are propositions.

Let us delay no longer and define a proposition. This construction is complicated, so do not expect to understand it in one read. Feel free to skim it a few times, then read it again closely, or whatever works for you. Additionally, it might help to skip the definition altogether, read the following prose, then come back. Also, remember to keep track of the ways that I say to read the notation. These are designed to be helpful. I have put this definition on its own page so that you can read is as efficiently as possible.

[3] φ is the Greek letter phi said like "f-eye" or "fee," ψ is the Greek letter psi said like "sigh", and χ is the Greek letter chi said like "k-eye."

Definition 3.1.1. A *proposition* is defined by the following production rules. These production rules rely on the *free and bound variables* of a proposition, which are also defined inductively.

(1) $\bot$, read as "false," is a proposition.
(2) $(x \in y)$, read as "x is an element of y" or "x in y," is a proposition if x and y are variables.
(3) $(\varphi \wedge \psi)$, read as "φ and ψ," is a proposition if φ and ψ are propositions and no free variable of φ is bound in ψ and vice versa.
(4) $(\varphi \vee \psi)$, read as "φ or ψ," is a proposition if φ and ψ are propositions and no free variable of φ is bound in ψ and vice versa.
(5) $(\varphi \Rightarrow \psi)$, read as "$\varphi$ implies ψ" or "if φ then ψ," is a proposition if φ and ψ are propositions and no free variable of φ is bound in ψ and vice versa.
(6) $(\forall x.\, \varphi)$, read as "for all x, φ," is a proposition if φ is a proposition and x is a variable not bound in φ.
(7) $(\exists x.\, \varphi)$, read as "there exists x such that φ," is a proposition if φ is a proposition and x is a variable not bound in φ.

The free variables of a proposition are defined as follows:

(1) No variable is free in $\bot$.
(2) A variable is free in $(x \in y)$ if it is either x or y.
(3) A variable is free in $(\varphi \wedge \psi)$ if it is free in φ or ψ.
(4) A variable is free in $(\varphi \vee \psi)$ if it is free in φ or ψ.
(5) A variable is free in $(\varphi \Rightarrow \psi)$ if it is free in φ or ψ.
(6) A variable is free in $(\forall x.\, \varphi)$ if it is free in φ and is not x.
(7) A variable is free in $(\exists x.\, \varphi)$ if it is free in φ and is not x.

The bound variables of a proposition are defined as follows:

(1) No variable is bound in $\bot$.
(2) No variable is bound in $(x \in y)$.
(3) A variable is bound in $(\varphi \wedge \psi)$ if it is bound in either φ or ψ.
(4) A variable is bound in $(\varphi \vee \psi)$ if it is bound in either φ or ψ.
(5) A variable is bound in $(\varphi \Rightarrow \psi)$ if it is bound in either φ or ψ.
(6) A variable is bound in $(\forall x.\, \varphi)$ if it is bound in φ or is x.
(7) A variable is bound in $(\exists x.\, \varphi)$ if it is bound in φ or is x.

A variable is *used* in a proposition φ if it is either free or bound in φ and is *unused* otherwise. One should confirm that this definition guarantees that no variable is both free and bound in a proposition.

If you are not a mathematician (or perhaps even if you are), this definition might be the most complicated definition you have ever seen. Hopefully, we can alleviate this pain by carefully analyzing the definition. It is important to remember that we have only defined syntax and not interpretation. That is, you do not have to understand what any propositions do or mean; that will come with the semantics.

Definition 3.1.1 inductively defines a proposition and which variables in it are free and bound. It does so by defining seven recipes for creating a proposition and defining the free and bound variables of each recipe in terms of its ingredients.

Recall that a proposition is supposed to represent a statement that is either true or false about sets depending on some variables. The first recipe says that a false proposition, denoted $\perp$, is a proposition. No variable is either used or bound in $\perp$ since no variable appears in its definition.

The second recipe says that our fundamental relationship between sets, the predicate $\in$, is a proposition when it is given variables. Specifically, if x and y are variables, then $(x \in y)$ is a proposition representing "x is an element of y." The variables x and y are free in $(x \in y)$ because we are "free" to substitute actual sets for these variables.

The next three recipes provide a way of joining two propositions together with a *connective*. In order, $\wedge$ represents "and," $\vee$ represents "or," and $\Rightarrow$ represents "implies." Because this is not the section on semantics, I will not elaborate upon what these connectives mean, but you can probably guess from how they're said. The free and bound variables of these joined propositions are respectively those free or bound in the propositions joined.

The last two recipes are called the *quantifiers.* The symbol $\forall$ is the *universal quantifier* and the symbol $\exists$ is the *existential quantifier*. Quantifiers take a proposition and "consume" variables to yield another proposition. The proposition $(\forall x.\, \varphi)$ represents the sentence "for any choice of a set x, φ holds." The proposition $(\exists x.\, \varphi)$ represents the sentence "there is some choice of a set x such that φ holds." For example, such a proposition could represent "there exists an x such that x is an element of y". The variable x is considered bound in $(\exists x.\, \varphi)$ and is no longer free. This is because the sentence "there exists an x such that x is an element of y" does not need a value of x to be filled in for it to be true or false, just one for y. If we substitute a set for y, the statement has a truth value indicating whether or not y has an element. Hence, the quantifiers "consume" free variables of φ and make them bound. We use the term bound because although the proposition does not require a value for bound variables, we still should not reuse them as free variables in propositions that we join up with connectives. Strictly speaking, these justifications are semantic; however, we mandate the same behavior in the syntax.

To reiterate, the free variables of a proposition are supposed to represent the variables which need to be "plugged in" to the proposition for it to become a sentence. The bound variables of a proposition are supposed to represent the ones that are"used up" by quantification. Let us get used to these rules with some examples and exercises.

Example 3.1.2. Here, we will find free and bound variables.

(1) $((a \in b) \wedge (b \in c))$
Free: a,b,c. Bound: none.
(2) $(\forall a. ((a \in b) \Rightarrow \perp))$
Free: b. Bound: a.
(3) $(\forall a. (\exists b. (a \in b)))$
Free: none. Bound: a,b.
(4) $((\forall a. (a \in b)) \Rightarrow (\exists a. (b \in a)))$
Free: b. Bound: a.

Exercise 3.1.1. Find the free and bound variables of the following propositions according to the rules in definition 3.1.1.

(1) $((a \in b) \vee (b \in a))$
(2) $(\forall a.\ (a \in a))$
(3) $((\forall d.\ \varphi) \wedge \psi)$ if a and b are free in φ, no variable is bound in φ, a is free in ψ, and c is bound in ψ

Example 3.1.3. It is easy to come up with examples of strings which appear to be valid propositions but actually are not. Here are some examples.

(1) $((a \in b) \in c))$ is not a valid proposition because $\in$ can only connect two variables, but here, it is between $(a \in b)$ and c, the former of which is not a variable.
(2) $((a \in b) \Rightarrow c))$ is not a valid proposition because $\Rightarrow$ can only connect propositions, and c is a variable rather than a proposition.
(3) $(\exists a. (\forall a. (a \in a))$ is not a valid proposition because we cannot quantify over the same variable twice. In particular, since a is bound in $(\forall a. (a \in a))$, we cannot add $\exists a.$ to it.
(4) $((a \in a) \Rightarrow (\forall a. (a \in b)))$ is not a valid proposition because a is free on the left side of $\Rightarrow$ but is bound on the right side.

Exercise 3.1.2. Which of the following propositions are valid?

(1) $(a \in b) \Rightarrow \perp$
(2) $(\forall a. (a \in a))$
(3) $(a \in \perp)$
(4) $\perp \vee (a \in b)$
(5) $(\forall a. (\exists b. ((b \in a) \Rightarrow \perp)))$
(6) $((\exists a.(a \in b)) \Rightarrow (\forall b. b \in a))$
(7) $(\forall a. ((b \in a) \vee ((b \in a) \Rightarrow \perp)))$
(8) $(\forall a. (\forall b. (\forall c. (\exists a. ((a \in b) \wedge (b \in c))))))$
(9) $(\forall a. ((a \in a) \in a))$

Here, I will reiterate a property of all languages. Two propositions are said to be equal *only if* they are written identically. Thus, you cannot move variables around, and you cannot apply variable replacements. For example, $((\varphi \Rightarrow \psi) \Rightarrow \chi)$ is not the same proposition as $(\varphi \Rightarrow (\psi \Rightarrow \chi))$ because the parentheses differ, and $(\varphi \wedge \psi)$ is not the same proposition[4] as $(\psi \wedge \varphi)$ because the order differs. However, as before, if φ represents the proposition $(a \in a)$, then the propositions $(\varphi \vee \perp)$ and $((a \in a) \vee \perp)$ are properly the same; this variable φ is not a variable in the language of predicate logic, but instead allows us to use shorthand.

The last thing to note is that each rule for creating a proposition, except the one for $\perp$, contains some grouping symbol. This means that our language is free. That is, there is exactly one way of creating each proposition from the production rules.

As before, that means our language is hard to read. Therefore, we can choose to omit parentheses as long as we define an order of operations and associativity rules. We place parentheses around the production rules in the following order:

(1) $\in$,
(2) $\wedge$ and $\vee$ simultaneously,
(3) $\forall$ and $\exists$ simultaneously,
(4) $\Rightarrow$,

Further, since the connectives join two propositions, we must define associativity rules. We choose that $\wedge$ and $\vee$ are left-associative and that $\Rightarrow$ is right-associative. Importantly, recall that when substituting shorthand, *you must place the parentheses back in the expression you are substituting.*

It is important to remember that the syntax of predicate logic, at this point, has no meaning. This definition serves only to create propositions from $\in$ and other propositions. There is not much more that can be

[4]unless φ and ψ represent the same proposition

said about the syntax of predicate logic. The only thing left is to get practice parsing propositions. It is useful as well to practice reading these propositions in the language described in the definition of propositions. In general, you can avoid saying parentheses or other grouping symbols out loud.

Example 3.1.4.

(1) When fully parenthesized, the proposition
$$\forall a.\, a \in b \text{ is } (\forall a.\, (a \in b)).$$
(2) The proposition
$$\forall a.\, \exists b.\, a \in b \text{ is } (\forall a.\, (\exists b.\, (a \in b))).$$
(3) The proposition
$$\forall a.\, a \in b \wedge b \in c \text{ is } (\forall a.\, ((a \in b) \wedge (b \in c))),$$
which is distinct from
$$((\forall a.\, (a \in b)) \wedge (b \in c)).$$
(4) The proposition
$$a \in b \wedge b \in c \wedge c \in a \text{ is } (((a \in b) \wedge (b \in c)) \wedge (c \in a)).$$
(5) The proposition
$$a \in b \Rightarrow b \in c \Rightarrow c \in a \text{ is } ((a \in b) \Rightarrow ((b \in c) \Rightarrow (c \in a))).$$
Note the difference from the last example.
(6) The proposition
$$\forall a.\, a \in b \Rightarrow \perp \text{ is } ((\forall a.(a \in b)) \Rightarrow \perp).$$

Exercise 3.1.3. Place parentheses in the following propositions.

(1) $a \wedge b \Rightarrow \perp$
(2) $\perp \Rightarrow a \in b$
(3) $a \in b \wedge b \in c \vee b \in c \Rightarrow c \in d \Rightarrow d \in a \wedge a \in b$
(4) $\forall a.\forall b.\, a \in b \Rightarrow c \in d$
(5) $a \in b \Rightarrow \forall c.\, a \in c \Rightarrow \exists c.\, c \in c$

3.2. **Truth Values.** The syntax of predicate logic is pretty useless unless we know what it is supposed to mean. A proposition is supposed to represent a sentence that is true or false when values from the domain of discourse (so, sets) are plugged in for its free variables.

Most mathematicians believe that all formal sentences are either true or false. Even if there are sentences that we cannot prove, they are all "actually" true or false depending on the specific kind of sets we are using, whether or not we even know what kind of sets we are using.

This idea, that all formal sentences are either true or false, is called the *law of the excluded middle* because it excludes the possibility of truth values "between" true and false. The truth or falsehood of a sentence is called its *truth value*. A truth value is not a formal idea; it is an understanding that you must bring to the table. Even if you do not accept the law of the excluded middle, it is provable within set theory.

In order for a proposition to have a well-defined truth value, we must substitute values for its free variables. We do so using a *variable assignment*. A variable assignment is a sequence of distinct variables and their corresponding (possibly repeating) sets which they will be assigned. A variable assignment is written in the form $[x_1 \colon v_1, \ldots, x_n \colon v_n]$ read as "x_1 replaced with v_1, x_2 replaced with v_2, etc." where $x_1, \ldots, x_n$ is the sequence of distinct variables and $v_1, \ldots, v_n$ is a sequence (of the same length) of sets which they will be assigned. In particular, the set v_i will be assigned to the variable x_i.

If $x_1, \ldots, x_n$ contains the free variables and none of the bound variables of a proposition φ, then the truth value of φ under the variable assignment $[x_1 \colon v_1, \ldots, x_n \colon v_n]$ is denoted by $[x_1 \colon v_1, \ldots, x_n \colon v_n]\varphi$ read as "x_1 replaced with v_1, x_2 replaced with v_2, etc. in φ." It only makes sense to perform these variable assignments when the variables contain every free variable of the proposition and none of its bound variables.

As an example, if v_1 and v_2 are sets, then $[x \colon v_1, y \colon v_2](x \in y)$ represents the truth value of the sentence "v_1 is an element of v_2". We are also allowed to add other variables to our assignment which are neither free or bound in the proposition, so it would be fine to write $[x \colon v_1, y \colon v_2, z \colon v_3](x \in y)$, which would represent the same thing.

Since there are seven ways of building propositions, we must specify when each of each seven types of proposition are true or false. Strictly speaking, each mathematician is required to bring an understanding of what the seven kinds of propositions mean semantically. However, we will elaborate upon the conventional understanding, even if the discussion is technically informal.

Unsurprisingly, we will need one concept for each rule for making propositions. First, we will need a notion of false. False is supposed to represent some sort of contradiction. For the most part, you will need to provide this notion. That is, you must know what false means. Second, we need a notion of what it means for some set to be one of the elements of another set. This is part of the understanding of sets that you bring to the table. That is, when we say a set is an unordered collection, you are required to know which things are and are not elements of any collection that you imagine (or how one could theoretically know).

Next, we need interpretations of the three connectives: Let the variables p and q represent arbitrary truth values. At least for a moment, suppose that there are two truth values: T for true and F for false. We describe the semantics of the connectives using the following *truth table*.

p	q	$p \wedge q$	$p \vee q$	$p \Rightarrow q$
T	T	T	T	T
T	F	F	T	F
F	T	F	T	T
F	F	F	F	T

Truth tables summarize the truth value of compound expressions. Each column represents the truth value of a different statement, and each row represents a possible combination of truth values. There are four rows since p and q can take on 2 truth values each. You are allowed to reject the notion that all truth values are either true or false, thus rejecting the premise of this truth table, but in such cases, you will need to supply your own interpretation of $\wedge$, $\vee$, and $\Rightarrow$.

The third column represents the *conjunction* of p and q, which is true whenever both p and q are true. This represents the connective "and" and explains why $\wedge$ is read as "and." As evidenced by the table, $p \wedge q$ is true whenever both p and q are true.

The fourth column represents the *disjunction* of p and q, which is true whenever p is true or q is true or both. This represents the connective "or" and explains why $\vee$ is read as "or." Sometimes this is also called the *inclusive or* because it "includes" the possibility that both p and q are true.

The fifth column is perhaps the least intuitive. It represents the *implication* of p and q. $p \Rightarrow q$ is supposed to represent the phrases "p implies q" or "if p, then q," which is a sort of contract. For example, "if you finished your dinner then you got dessert" is a sort of contract. The statement is false if the contract was violated and is true otherwise. That is, the statement is false if you finished your dinner but did not get dessert. This is the second row in the column of the $\Rightarrow$. If you finished dinner and got dessert, the contract was not violated. This is the first row in the $\Rightarrow$ column. Further, if you did not finish your dinner, the contract was not violated *regardless* of whether or not you got dessert (the contract does not say that you did not get dessert if you did not finish dinner). These are the last two rows of the $\Rightarrow$ column. This definition of implication is generally considered weird by non-mathematicians, but this is the most powerful way to interpret implication. Thus, we have defined what the three connectives mean.

Exercise 3.2.1. Here we ask some simple questions regarding the structure of truth tables.

(1) If p is false and q is true, what is $p \Rightarrow q$?
(2) Consulting the following truth table, what is $T \veebar F$?

p	q	$p \veebar q$
T	T	F
T	F	T
F	T	T
F	F	F

(3) For arbitrary p, what is the value $p \wedge F$?
(4) For arbitrary p, what is the value $p \Rightarrow p$?
(5) For arbitrary p, what is the value $p \veebar p$?

Lastly, we must define what the two quantifiers mean. The universal quantifier $\forall x.\, \varphi$ is supposed to be interpreted as the sentence "for all values of x, φ holds." Thus, $\forall x.\, \varphi$ holds whenever the proposition φ holds for all sets x. The existential quantifier $\exists x.\, \varphi$ is supposed to be interpreted as the sentence "there is some value x such that φ holds." Thus, $\exists x.\, \varphi$ holds whenever the proposition φ holds for some set x. We will elaborate more upon these quantifiers in the full definition of the semantics of predicate logic.

Without further ado, here is how to interpret the truth value of a proposition. Again, this definition is difficult, so I have placed it on its own page for maximum readability.

Definition 3.2.1. If φ is a proposition and $[x_1\colon v_1, \ldots, x_n\colon v_n]$ is an assignment of variables not bound in φ containing the free variables of φ, then the *truth value* of $[x_1\colon v_1, \ldots]\varphi$ is inductively defined as:

(1) Suppose φ is of the form $\perp$. Then the truth value $[x_1\colon v_1, \ldots]\varphi$ is false. A notion of false is required in your interpretation.
(2) Suppose φ is of the form $(y \in z)$ for variables y and z. Since y and z are the free variables in φ, y is some x_i and z is some x_j. The truth value $[x_1\colon v_1, \ldots]\varphi$ represents $v_i \in v_j$, whether the set corresponding to y is an element of the set corresponding to z. This understanding is a required part of your understanding of sets.
(3) Suppose φ is of the form $(\psi \wedge \chi)$ for propositions ψ and χ. Then, the truth value $[x_1\colon v_1, \ldots]\varphi$ is the truth value
$$[x_1\colon v_1, \ldots]\psi \wedge [x_1\colon v_1, \ldots]\chi$$
as given by the truth table.
(4) Suppose φ is of the form $(\psi \vee \chi)$ for propositions ψ and χ. Then, the truth value $[x_1\colon v_1, \ldots]\varphi$ is the truth value
$$[x_1\colon v_1, \ldots]\psi \vee [x_1\colon v_1, \ldots]\chi$$
as given by the truth table.
(5) Suppose φ is of the form $(\psi \Rightarrow \chi)$ for propositions ψ and χ. Then, the truth value $[x_1\colon v_1, \ldots]\varphi$ is the truth value
$$[x_1\colon v_1, \ldots]\psi \Rightarrow [x_1\colon v_1, \ldots]\chi$$
as given by the truth table.
(6) Suppose that φ is of the form $(\forall y.\, \psi)$ for a proposition ψ and a variable y. Since the variables $x_1, \ldots, x_n$ are not bound in φ and y is bound in φ, the variable y does not appear in the list $x_1, \ldots, x_n$. The truth value $[x_1\colon v_1, \ldots]\varphi$ is true if for every extended variable assignment $[x_1\colon v_1, \ldots, x_n\colon v_n, y\colon u]$,
$$[x_1\colon v_1, \ldots, x_n\colon v_n, y\colon u]\psi$$
is true, and is false otherwise.
(7) Suppose that φ is of the form $(\exists y.\, \psi)$ for a proposition ψ and a variable y. Since the variables $x_1, \ldots, x_n$ are not bound in φ and y is bound in φ, the variable y does not appear in the list $x_1, \ldots, x_n$. The truth value $[x_1\colon v_1, \ldots]\varphi$ is true if there exists some extended variable assignment $[x_1\colon v_1, \ldots, x_n\colon v_n, y\colon u]$ such that
$$[x_1\colon v_1, \ldots, x_n\colon v_n, y\colon u]\psi$$
is true, and is false otherwise.

It is also pertinent to note the way that variable assignments interoperate with propositions with omitted parentheses. We choose to place variable assignment at the very beginning of the order of operations, which often means that we need to write parentheses. That is, for a variable x, a set v, and propositions φ and ψ, $[x\colon v]\varphi \wedge \psi$ is not a valid truth value because the variable assignment only applies to φ. Instead, we must write $[x\colon v](\varphi \wedge \psi)$.

While the definition of variable assignments may seem complicated, they are not so bad. Essentially, the rules just say "move the variable assignment through parentheses, duplicate them over operators, and apply them to variables." To demonstrate this, we shall work examples.

Example 3.2.2. Suppose u and v are sets.

(1) $[a\colon u, b\colon v](a \in b)$ is the value of $u \in v$.
(2) $[a\colon u, b\colon v, c\colon u](a \in b \wedge b \in c)$ is the value of $u \in v \wedge v \in u$.
(3) $[a\colon u](\forall a.\, a \in a)$ is not valid, since we many only assign free variables.
(4) $[a\colon u](\forall b.\, a \in b \Rightarrow a \in a)$ is true if for every set v, $u \in v \Rightarrow u \in u$ and is false otherwise.

Although I have not marked definition 3.2.1 as informal, I seriously considered doing so. This definition relies on the informal notions of sets, elements, implication, existence, etc. In particular, we have no way of syntactically representing a set. That is, we have not agreed upon any way of writing down a set. However, this list of notions can be given a precise interpretation inside another logic whose domain of discourse includes these logical symbols and predicates such as "is a proposition," "is a variable," and so on. Such a logic would be called a *second-order logic* because its variables are able to represent logical symbols, while ours is simply a *first-order logic*. Unfortunately, such a system would either rely on informal notions or higher-order logic, which would eventually rely on informal notions.

We are already working in a second-order system because we have made claims such as $(\varphi \wedge \bot)$ is a proposition for all propositions φ. In particular, we have been using variables such as φ, ψ, and χ to represent arbitrary first-order propositions. This indicates that we do not have a formal notion of first-order logic, but the definition we have is as close to formal as is possible.

3.3. **Shorthand.** Since we are already working in a second-order system, we might as well define useful extensions to the syntax of propositions. I am not going to mark these definitions as informal, even though we have not formalized any second-order system, because they would easily be formalizable if we were working in a second-order system.

We will start with the most difficult and most useful extension. It is called *exchange of variables* or *α conversion*[5] and (unsurprisingly) allows exchanging variables for other variables.

Definition 3.3.1. For a proposition φ and variables x and y, the proposition $[x\colon y]\varphi$, read as "x replaced with y in φ" is inductively defined by:

(1) Suppose φ is of the form $\bot$. Then, $[x\colon y]\varphi$ is the proposition $\bot$.
(2) Suppose φ is of the form $(z \in w)$ for variables z and w. Then, $[x\colon y]\varphi$ is the proposition:
    - $(z \in w)$ if neither z nor w is x.
    - $(y \in w)$ if z is x and w is not x.
    - $(z \in y)$ if z is not x and w is x.
    - $(y \in y)$ if both z and w are x.
(3) Suppose φ is of the form $(\psi \wedge \chi)$ for propositions ψ and χ. Then, $[x\colon y]\varphi$ is the proposition $([x\colon y]\psi \wedge [x\colon y]\chi)$.
(4) Suppose φ is of the form $(\psi \vee \chi)$ for propositions ψ and χ. Then, $[x\colon y]\varphi$ is the proposition $([x\colon y]\psi \vee [x\colon y]\chi)$.
(5) Suppose φ is of the form $(\psi \Rightarrow \chi)$ for propositions ψ and χ. Then, $[x\colon y]\varphi$ is the proposition $([x\colon y]\psi \Rightarrow [x\colon y]\chi)$.
(6) Suppose φ if of the form $(\forall z.\, \psi)$ for a proposition ψ and a variable z. Then, $[x\colon y]\varphi$ is:
    - $(\forall z.\, ([x\colon y]\psi))$ if the variable z is not the variable x.
    - $(\forall y.\, ([x\colon y]\psi))$ if the variable z is the variable x.
(7) Suppose φ if of the form $(\exists z.\, \psi)$ for a proposition ψ and a variable z. Then, $[x\colon y]\varphi$ is:
    - $(\exists z.\, ([x\colon y]\psi))$ if the variable z is not the variable x.
    - $(\exists y.\, ([x\colon y]\psi))$ if the variable z is the variable x.

In summary, $[x\colon y]\varphi$ is the proposition φ with every occurrence of the variable x replaced with the variable y. Importantly, this notation is only defined when the resulting proposition follows the rules for free and bound variables and is thus a valid proposition.

Take note of the many similarities between variable assignment and variable replacement. Variable assignment and variable replacement are effectively the same thing; the only difference is that variable assignment

[5]α is the lowercase Greek letter alpha.

replaces variables with sets while variable replacement replaces variables with other variables. Thus, we use similar notation for both, and we put them both at the beginning of the order of operations (variable replacement is performed before everything else). That is, $[x\colon y]\varphi \wedge \psi$ means that the variable replacement applies only to φ, while in the parenthesized expression $[x\colon y](\varphi \wedge \psi)$, it applies to both. The only other important difference is that we may replace bound variables.

Let us consider the proposition $[a\colon c](a \in b)$, read as "a replaced with c in $a \in b$." This is supposed to replace the a in $a \in b$ with c, yielding the proposition $c \in b$. As another example, $[a\colon c](\forall a.\, a \in b)$ represents the proposition $\forall c.\, c \in b$. Here, we have replaced the bound variable a with the variable c. However, $[b\colon a](\forall b.\, \forall a.\, a \in b)$, which represents $\forall a.\, \forall a.\, a \in a$, which is not a valid proposition because it quantifies a twice, is not a valid variable substitution. One solution to avoid this problem and others like it is to avoid using bound variables to replace other variables. This is sometimes permissible though. For example, $[b\colon a](\forall a.\, a \in c \Rightarrow \forall b.\, b \in c)$ represents the proposition $(\forall a.\, a \in c \Rightarrow \forall a.\, a \in c)$, which is valid.

From these examples, we can discover the rules for the free and bound variables of $[x\colon y]\varphi$. In particular, a variable is free or bound in $[x\colon y]\varphi$ if it is a variable free or bound in φ and is not the variable x (so it is not involved in the replacement) or if it is the variable y and x is free or bound in φ. As a consequence x is always unused in $[x\colon y]\varphi$ unless x and y are the same variable (in which case the replacement does nothing anyway). As a final note, since the variable replacement syntax is not part of predicate logic, $[a\colon b](a \in c)$ is the same proposition as $b \in c$ even though they are written differently.

Next, we will present what is perhaps the easiest piece of additional syntax. Note that we have a proposition $\perp$ that represents a false statement. It is natural to want a proposition for a *true* statement as well.

Definition 3.3.2. The proposition $\top$, read as "true," is shorthand for the proposition $\perp \Rightarrow \perp$. That is,

$$\top\colon (\perp \Rightarrow \perp).$$

Thus, true ($\top$) is shorthand for false implies false. I'm not really sure why the philosophers have been looking so long for a definition of truth when they could simply use this one. We agreed earlier that $F \Rightarrow F$ is T as a truth value, so then this proposition $\top$ must be a true proposition. In fact, if we did not agree that true and false were the only truth values, we could use this as the definition of true.

Similarly, if φ is a proposition, it is natural to want a proposition representing that φ is false.

Definition 3.3.3. For a proposition φ, the *negation* of φ, denoted as $\varphi\neg$ and read as "φ not," is shorthand for the proposition $\varphi \Rightarrow \bot$. That is,

$$\varphi\neg : (\varphi \Rightarrow \bot).$$

We agreed earlier that $p \Rightarrow F$ is true when p is false and is false when p is true. Thus, $\varphi\neg$, that is $\varphi \Rightarrow \bot$, must be the opposite or negation of φ. We choose to place $\neg$ just after $\in$ in the order of operations so that $\varphi \wedge \psi\neg$ is $\varphi \wedge (\psi \Rightarrow \bot)$ rather than $(\varphi \wedge \psi) \Rightarrow \bot$. With this definition, true is not false. That is, $\top$ and $\bot\neg$ are the same proposition.

We will also define special shorthand for the negation of $\in$. We will not use this shorthand for awhile, but keep it in your mind.

Definition 3.3.4. For variables x and y, the proposition $x \notin y$, read as "x is not an element of y" or "x not in y," is shorthand for the proposition $(x \in y)\neg$. That is,

$$x \notin y \colon (x \in y)\neg.$$

We will also define one additional connective for joining propositions.

Definition 3.3.5. For propositions φ and ψ, the *equivalence* of φ and φ, denoted as $\varphi \Leftrightarrow \psi$ read as "φ if and only if ψ," and sometimes spelled "φ iff ψ," is shorthand for the proposition $(\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$. That is,

$$\varphi \Leftrightarrow \psi \colon (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi).$$

Since $\varphi \Rightarrow \psi$ is the proposition meaning that the truth of φ guarantees the truth of ψ and $\psi \Rightarrow \varphi$ is the proposition meaning that the truth of ψ guarantees the truth of φ, $\varphi \Leftrightarrow \psi \colon (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$ is the proposition meaning that φ and ψ guarantee the truth of each other. That is, they are only true or false together. In that sense, if $\varphi \Leftrightarrow \psi$ is true, then φ and ψ are essentially the same statement. While they may not be equal, if $\psi \Leftrightarrow \psi$, φ and ψ are called *equivalent*.

We place $\Leftrightarrow$ after $\Rightarrow$ in the order of operations, so that $\varphi \Rightarrow \psi \Leftrightarrow \psi \Rightarrow \chi$ is $(\varphi \Rightarrow \psi) \Leftrightarrow (\psi \Rightarrow \chi)$ and $\forall x.\, \varphi \Leftrightarrow \psi$ is $(\forall x.\, \varphi) \Leftrightarrow \psi$. We do not, however, provide associativity rules for $\Leftrightarrow$. This is because we define some special shorthand for these cases.

Definition 3.3.6. For a sequence of at least two propositions $\varphi_1, \ldots, \varphi_n$, we define

$$\varphi_1 \Leftrightarrow \varphi_2 \Leftrightarrow \cdots \Leftrightarrow \varphi_n \colon \varphi_1 \Leftrightarrow \varphi_2 \wedge \varphi_2 \Leftrightarrow \varphi_3 \wedge \cdots \wedge \varphi_{n-1} \Leftrightarrow \varphi_n.$$

In particular, $\varphi \Leftrightarrow \psi \Leftrightarrow \chi$ is the proposition $\varphi \Leftrightarrow \psi \wedge \psi \Leftrightarrow \chi$. Intuitively, this says that φ, ψ, and χ each guarantee the truth of each other.

Next, we define some notation with sequence of variables so that we have to write less.

Definition 3.3.7. For a proposition φ and a sequence of variables $x_1, \ldots, x_n$, we define

$$\forall x_1, \ldots, x_n.\, \varphi \colon \forall x_1.\forall x_2. \cdots \forall x_n.\, \varphi.$$

If n is 0, $\forall x_1, \ldots, x_n$ should be interpreted as just φ.

Definition 3.3.8. For a proposition φ and a sequence of variables $x_1, \ldots, x_n$, we define

$$\exists x_1, \ldots, x_n.\, \varphi \colon \exists x_1.\exists x_2. \cdots \exists x_n.\, \varphi.$$

If n is 0, $\exists x_1, \ldots, x_n$ should be interpreted as just φ.

This shorthand allows us to avoid writing a bunch of quantifiers over and over again, which makes reading easier. That is, instead of $\forall x.\forall y.\forall z.$, we can just write $\forall x, y, z.$, which has the same meaning.

Finally, we also define shorthand for combined variable replacement.

Definition 3.3.9. For a proposition φ and two sequences of variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of the same length, we define

$$[x_1 \colon y_1,\ x_2 \colon y_2, \ldots,\ x_n \colon y_n]\varphi \colon [x_1 \colon y_1][x_2 \colon y_2] \cdots [x_n \colon y_n]\varphi$$

whenever the right side is a valid proposition. If n is 0, $[]\varphi$ should be interpreted as just φ.

With this notation, we can perform multiple substitutions at the "same time." That is, we can write $[a \colon c, b \colon c](a \in b)$ to get $(c \in c)$ and can even pull off a "swap" with $[c \colon b, b \colon a, a \colon c](a \in b)$ to get $(b \in a)$.

3.4. **Formal Proofs.** The last part of formal logic that we will need is the notion of a formal proof. A formal proof is a mathematical object which has assumptions, which are propositions, and conclusions, which are also propositions, combined with some sort of justification. The idea behind a formal proof is that it is a demonstration that if the assumptions are true, then so is the conclusion.

Formal proofs are not common in mathematics. As mentioned, we will primarily rely on prose proofs. Consequently, we will be a bit more "fast and loose" in this section than in others, and will not prove every stated result. Further, we are not going to spend the time to develop a definition of formal proof. Those interested in the details should consult the readings at the end of the section.

As is standard in this thesis, we will build formal proofs inductively in terms of fundamental proofs: *rules* and *axioms*. These rules and axioms may be valid or invalid depending on the semantics of logic that you choose. This thesis will only be concern with logics with these rules and axioms.

Rule 3.1. The rule of *implication elimination* states that if φ and ψ are propositions, then φ and $\varphi \Rightarrow \psi$ prove ψ. This is denoted as

$$\frac{\varphi \quad \varphi \Rightarrow \psi}{\psi} \Rightarrow\text{elim.}$$

which can be read as "φ and $\varphi \Rightarrow \psi$ prove ψ by $\Rightarrow$ elimination."

This rule, and all other rules and axioms, are assumed to hold only when all propositions involved are valid. In this case, $\varphi \Rightarrow \psi$ must be a proposition according to the rules for free and bound variables.

This is our first example of the structure of a proof. The propositions above the horizontal line are called the *assumptions*. Here, we have two assumptions, φ and $\varphi \Rightarrow \psi$. The proposition under the line is called the *conclusion*. Here, our conclusion is ψ. The writing to the right of the line is called the *rule*. This rule explains why the proof is valid. Here, our rule is implication elimination.

A proposition φ is not something that can be true or false; only when it is assigned variables does it become true or false. When we have a rule like this, it is assumed to hold of *all* variable assignments. Further, the *same* variable assignment must be applied to each of the assumptions and conclusions. As a consequence, any variable bound in any of the assumptions or conclusion of a proof cannot be free in any of the other assumptions or the conclusion.

Since any rule must be valid under any variable assignment, let us investigate $\Rightarrow$ elimination using a truth table. Let p and q stand for the truth values of φ and ψ respectively under a variable assignment $[x_1\colon v_1, \ldots]$.

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	F	T
F	F	T

Intuitively, the rule of implication elimination says that in every row our assumptions (here, p and $p \Rightarrow q$) are true, so is our conclusion (here, q). The first row is the only row where both p and $p \Rightarrow q$ are true, and q is also true in this row. This means that our rule is *sound* in respect to our semantics. If we chose different semantics (meaning our truth table might look different), then this rule might not be justified. All of our rules will be sound according to our semantics.

We can use this rule to construct larger proofs. We will do so in order to prove a *theorem*. A theorem is simply an important fact that is true. Theorems are generally accompanied by proofs, whether formal or prose.

Theorem 3.4.1. If φ, ψ, and χ are propositions, then φ, ψ, and $\varphi \Rightarrow \psi \Rightarrow \chi$ prove χ. That is,

$$\frac{\varphi \quad \psi \quad \varphi \Rightarrow \psi \Rightarrow \chi}{\chi}\,3.4.1 \quad .$$

Before we present the proof, recall that $\varphi \Rightarrow \psi \Rightarrow \chi$ is $(\varphi \Rightarrow (\psi \Rightarrow \chi))$. Also, notice that rather than a rule, we have placed our theorem number here. This is done instead of a name when the theorem is not important enough to name.

Proof.

$$\frac{\psi \qquad \dfrac{\varphi \quad \varphi \Rightarrow \psi \Rightarrow \chi}{\psi \Rightarrow \chi}\Rightarrow\text{elim.}}{\chi}\Rightarrow\text{elim.}$$

$\square$

This is our first non-trivial formal proof. Let us investigate its structure. First, there are three propositions without lines above them: φ, ψ, and $\varphi \Rightarrow \psi \Rightarrow \chi$. These are the assumptions, which match the assumptions of the theorem. Next, there is exactly one proposition with no line under it: χ. This is our conclusion, which matches the conclusion of the theorem. Each horizontal line represents a rule. Let us take note of two special cases of implication elimination:

$$\frac{\varphi \quad \varphi \Rightarrow \psi \Rightarrow \chi}{\psi \Rightarrow \chi}\Rightarrow\text{elim} \quad \text{and} \quad \frac{\psi \quad \psi \Rightarrow \chi}{\chi}\Rightarrow\text{elim} \quad .$$

The left proof is implication elimination where φ is the proposition φ and ψ is the proposition $\psi \Rightarrow \chi$. One should confirm that this is exactly the same as the definition of the rule with each occurrence of ψ replaced with $\psi \Rightarrow \chi$. Similarly, the right proof is implication elimination where φ is replaced with ψ and ψ is replaced with χ. Notice that the left proof has a conclusion of $\psi \Rightarrow \chi$ and the right proof has an assumption of $\psi \Rightarrow \chi$. We can *glue* these two proofs together along this shared copy of $\psi \Rightarrow \chi$ to get a larger proof. This is exactly what we did to construct the proof of our theorem.

Let us now prove a similar theorem.

Theorem 3.4.2. If φ, ψ, and χ are propositions, then φ, $\varphi \Rightarrow \psi$, and $\varphi \Rightarrow \psi \Rightarrow \chi$ prove χ. That is,

$$\frac{\varphi \quad \varphi \Rightarrow \psi \quad \varphi \Rightarrow \psi \Rightarrow \chi}{\chi}\,3.4.2\;.$$

Proof.

$$\dfrac{\dfrac{\varphi \quad \varphi \Rightarrow \psi}{\psi}\Rightarrow\text{elim.} \qquad \dfrac{\varphi \quad \varphi \Rightarrow \psi \Rightarrow \chi}{\psi \Rightarrow \chi}\Rightarrow\text{elim.}}{\chi}\Rightarrow\text{elim.}\;.$$

□

Notice that in this proof, we appear to have four assumptions: φ, $\varphi \Rightarrow \psi$, φ, and $\varphi \Rightarrow \psi \Rightarrow \chi$. Fortunately, this is okay, for φ is repeated in this list. This is still a valid proof, even though we have used the assumption of φ twice. Similarly, we do not have to use every assumption. For example, the same proof as that for theorem 3.4.2 proves the following theorem.

Theorem 3.4.3. For propositions φ, ψ, and χ,

$$\frac{\varphi \quad \varphi \Rightarrow \psi \quad \varphi \Rightarrow \psi \Rightarrow \chi \quad \chi \Rightarrow \psi}{\chi}\;.$$

This is an example of a theorem that is so unimportant, we do not even write a rule next to it. Generally, this is done when the theorem exists only for pedagogical purposes, rather than as a mathematical tool.

Although our proof of theorem 3.4.2 did not use the new assumption $\chi \Rightarrow \psi$, we did not need to use the assumption to prove our conclusion. Thus our proof of theorem 3.4.2 is still sufficient for this theorem. Logic where each assumption must be used exactly once is called *linear logic*. Such logic is interesting, but will not be covered here.

Next, we will speak of axioms. Axioms are rules with no assumptions. We will have one or more axioms for each way of building propositions, except for $\in$, whose axioms will be given in the section on set theory.

Axiom 3.2. For a proposition φ,

$$\frac{}{\bot \Rightarrow \varphi}\,\bot\;.$$

Since this is the only axiom for $\bot$, we just call this axiom $\bot$.

Let us investigate this axiom using a truth table. Let p denote the truth value of φ (under a variable assignment).

p	$F \Rightarrow p$
T	T
F	T

Notice that the last column is entirely true. This means that the conclusion $\perp \Rightarrow \varphi$ holds regardless of the truth value of φ. If the axiom has this property, then it is called sound in respect to our semantics. All of our axioms will be sound in respect to our semantics.

This axiom yields an easy theorem.

Theorem 3.4.4. The *explosion principle* says that a contradiction implies any proposition. That is, for a proposition φ,

$$\dfrac{\perp}{\varphi}\,\text{explos.}\quad .$$

Proof.

$$\dfrac{\perp \qquad \dfrac{}{\perp \Rightarrow \varphi}\,\perp}{\varphi}\,\Rightarrow\text{elim.}$$

□

This theorem states that any conclusion follows from a falsehood. In other words, as soon as we accept one contradiction, our entire logical universe "explodes" into everything being true. If all propositions are true, mathematics becomes uninteresting and useless. Thus, we avoid accepting contradictions.

There is an even easier theorem that this axiom provides.

Theorem 3.4.5. True is always true. That is,

$$\dfrac{}{\top}\,\top\quad .$$

Proof. Recall that $\top$ is defined as $\perp \Rightarrow \perp$.

$$\dfrac{}{\perp \Rightarrow \perp}\,\perp\quad .$$

□

Next, we will present the three axioms for $\land$.

Axiom 3.3. The axiom of *left $\land$ elimination* states that for propositions φ and ψ,

$$\dfrac{}{\varphi \land \psi \Rightarrow \varphi}\,\text{L}\land\text{elim.}\quad .$$

Axiom 3.4. The axiom of *right $\land$ elimination* states that for propositions φ and ψ,

$$\dfrac{}{\varphi \land \psi \Rightarrow \psi}\,\text{R}\land\text{elim.}\quad .$$

Axiom 3.5. The axiom of *$\land$ introduction* states that for propositions φ and ψ,

$$\dfrac{}{\varphi \Rightarrow \psi \Rightarrow \varphi \land \psi}\,\land\text{intro.}\quad .$$

Left $\land$ elimination, in essence, states that if $\varphi \land \psi$ is true, so is φ. That is, if φ and ψ are true, then φ is true. Let us verify this axiom with a truth table. Let p represent the truth value of φ and q represent the truth value of ψ.

p	q	$p \land q$	$p \land q \Rightarrow p$
T	T	T	T
T	F	F	T
F	T	F	T
F	F	F	T

Notice that the last column is all true. This demonstrates that left $\land$ elimination is sound.

Right $\land$ elimination states the same for ψ. $\land$ introduction states that the truth of φ guarantees that the truth of ψ guarantees the truth of $\varphi \land \psi$.

Exercise 3.4.1. Verify the soundness of right $\land$ elimination and $\land$ introduction using a truth table.

One can interpret $\land$ introduction as stating that φ and ψ together guarantee $\varphi \land \psi$. Let us show this by constructing a new proof.

Theorem 3.4.6. For propositions φ and ψ, φ and ψ prove $\varphi \land \psi$. That is,

$$\dfrac{\varphi \quad \psi}{\varphi \land \psi}\,3.4.6\quad .$$

Proof. We can produce two special version of $\Rightarrow$ elimination,

$$\dfrac{\psi \quad \psi \Rightarrow \varphi \land \psi}{\varphi \land \psi}\,\Rightarrow\text{elim.} \quad \text{and} \quad \dfrac{\varphi \quad \varphi \Rightarrow \psi \Rightarrow \varphi \land \psi}{\psi \Rightarrow \varphi \land \psi}\,\Rightarrow\text{elim.}\quad .$$

Notice that our two proofs both include $\psi \Rightarrow \varphi \wedge \psi$. We can glue our proofs along $\psi \Rightarrow \varphi \wedge \psi$ to get

$$\dfrac{\psi \qquad \dfrac{\varphi \quad \varphi \Rightarrow \psi \Rightarrow \varphi \wedge \psi}{\psi \Rightarrow \varphi \wedge \psi}\Rightarrow\text{elim.}}{\varphi \wedge \psi}\Rightarrow\text{elim.} \quad .$$

This proof has the conclusion we seek, $\varphi \wedge \psi$, and has three assumptions: φ, ψ, and $\varphi \Rightarrow \psi \Rightarrow \varphi \wedge \psi$. This last assumption is undesired. Fortunately, this is the conclusion of the axiom of $\wedge$ introduction. Thus, we can create a complete proof that φ and ψ prove $\varphi \wedge \psi$.

$$\dfrac{\psi \qquad \dfrac{\varphi \quad \dfrac{}{\varphi \Rightarrow \psi \Rightarrow \varphi \wedge \psi}\wedge\text{intro.}}{\psi \Rightarrow \varphi \wedge \psi}\Rightarrow\text{elim.}}{\varphi \wedge \psi}\Rightarrow\text{elim.} \quad .$$

$\square$

Exercise 3.4.2. Construct a formal proof that $\varphi \wedge \psi$ proves φ.

Altogether, these three axioms of $\wedge$ essentially specify its entire behavior. In particular, they state that φ and ψ prove the combined proposition $\varphi \wedge \psi$, and that the combined proposition $\varphi \wedge \psi$ proves both φ and ψ. Regardless of whether or not you believe in the semantics of truth values, these axioms should still be valid in whatever semantics you choose.

An important consequence is that $\wedge$ must always be *symmetric*, which is to say that $\varphi \wedge \psi$ and $\psi \wedge \varphi$ are effectively the same proposition.

Theorem 3.4.7. For propositions φ and ψ,

$$\dfrac{\varphi \wedge \psi}{\psi \wedge \varphi}\wedge\text{sym.} \quad .$$

Proof.

$$\dfrac{\dfrac{\varphi \wedge \psi \quad \dfrac{}{\varphi \wedge \psi \Rightarrow \psi}\text{L}\wedge\text{elim.}}{\psi}\Rightarrow\text{elim.} \qquad \dfrac{\varphi \wedge \psi \quad \dfrac{}{\varphi \wedge \psi \Rightarrow \varphi}\text{R}\wedge\text{elim.}}{\varphi}\Rightarrow\text{elim.}}{\psi \wedge \varphi}3.4.6$$

$\square$

Next, we will define the three axioms of $\lor$.

Axiom 3.6. The axiom of *left* $\lor$ *introduction* states that for propositions φ and ψ,

$$\cfrac{}{\varphi \Rightarrow \varphi \lor \psi}\,\text{L}\lor\text{intro.}$$

Axiom 3.7. The axiom of *right* $\lor$ *introduction* states that for propositions φ and ψ,

$$\cfrac{}{\psi \Rightarrow \varphi \lor \psi}\,\text{R}\lor\text{intro.}$$

Axiom 3.8. The axiom of $\lor$ *elimination* states that for propositions φ, ψ, and χ,

$$\cfrac{}{(\varphi \Rightarrow \chi) \Rightarrow (\psi \Rightarrow \chi) \Rightarrow \varphi \lor \psi \Rightarrow \chi}\,\lor\text{elim.}$$

The first two axioms are straightforward; they claim that φ and ψ both individually prove $\varphi \lor \psi$. That is, if φ is true, so is $\varphi \lor \psi$, and the same with ψ. The third axiom is less trivial. And further, why should that be called elimination? This axiom can be interpreted as follows:

> Suppose you know that $\varphi \Rightarrow \chi$ and $\psi \Rightarrow \chi$, but you don't know φ and you don't know ψ. This axiom states that it is enough to know $\varphi \lor \psi$ to conclude χ.

This allows "eliminating" an occurrence of $\lor$ for an occurrence of χ. We can present this as a theorem.

Theorem 3.4.8. For propositions φ, ψ, and χ,

$$\cfrac{\varphi \Rightarrow \chi \quad \psi \Rightarrow \chi \quad \varphi \lor \psi}{\chi}\,3.4.8$$

Proof.

$$\cfrac{\varphi \lor \psi \qquad \cfrac{\psi \Rightarrow \chi \qquad \cfrac{\varphi \Rightarrow \chi \qquad \cfrac{}{(\varphi \Rightarrow \chi) \Rightarrow (\psi \Rightarrow \chi) \Rightarrow \varphi \lor \psi \Rightarrow \chi}\,\lor\text{elim.}}{(\psi \Rightarrow \chi) \Rightarrow \varphi \lor \psi \Rightarrow \chi}\,\Rightarrow\text{elim.}}{\varphi \lor \psi \Rightarrow \chi}\,\Rightarrow\text{elim.}}{\chi}\,\Rightarrow\text{elim.}$$

$\square$

Exercise 3.4.3. Verify these axioms with truth tables.

Theorem 3.4.9. For propositions φ and ψ,

$$\cfrac{\varphi \lor \psi}{\psi \lor \varphi}\,\lor\text{sym.}$$

Exercise 3.4.4. Prove this theorem.

Next, there are two axioms for $\Rightarrow$.

Axiom 3.9. The axiom of $\Rightarrow$ *introduction* states that for propositions φ and ψ,

$$\frac{}{\varphi \Rightarrow \psi \Rightarrow \varphi}\Rightarrow\text{intro}.$$

In effect, this axiom asserts that if φ is true, then $\psi \Rightarrow \varphi$ is true. Intuitively, this means that the truth of φ guarantees that the truth of ψ guarantees the truth of φ. This is because if we are guaranteed that φ is true, we do not even need ψ to conclude that φ is true. Let us investigate the claim using a truth table. Let p and q be the truth values of φ and ψ respectively.

p	q	$q \Rightarrow p$	$p \Rightarrow q \Rightarrow p$
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

This last column is all true, and so this axiom is sound.

Now, we present the final axiom of implication.

Axiom 3.10. The axiom of *transitivity* states that for propositions φ, ψ, and χ,

$$\frac{}{(\varphi \Rightarrow \psi \Rightarrow \chi) \Rightarrow (\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \chi)}\text{trans}.$$

The truth table for this axiom is much more complicated, much too massive for me to ask you to make. In fact, it is so massive that I will take advantage of the only landscape page in this thesis.

p	q	r	$q \Rightarrow r$	$p \Rightarrow q \Rightarrow r$	$p \Rightarrow q$	$p \Rightarrow r$	$(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$	$(p \Rightarrow q \Rightarrow r) \Rightarrow (p \Rightarrow q) \Rightarrow (p \Rightarrow r)$
T	T	T	T	T	T	T	T	T
T	T	F	F	F	T	F	F	T
T	F	T	T	T	F	T	T	T
T	F	F	T	T	F	F	T	T
F	T	T	T	T	T	T	T	T
F	T	F	F	T	T	T	T	T
F	F	T	T	T	T	T	T	T
F	F	F	T	T	T	T	T	T

Here, p, q, and r represent the truth values of φ, ψ, and χ respectively. As before, the last column is all true, representing that this statement is true regardless of the truth of p, q, or r, and thus that our axiom is sound.

This table has eight rows because we have three fundamental truth values: p, q, and r. Each truth value has two possible options, so we need eight rows to account for every combination of their truth values.

Let us understand this truth table by examining a random row, say the second. This is the case where p is true, q is true, and r is false. The fourth column says that $q \Rightarrow r$ is false, which is correct because true (q) does not imply false (r). Since $q \Rightarrow r$ is false but p is true, the fifth column says that $p \Rightarrow q \Rightarrow r$ is false too. The sixth and seventh columns are the values for $p \Rightarrow q$ and $p \Rightarrow r$, the former of which is true and the latter of which is false. Thus, $(p \Rightarrow q) \Rightarrow (p \Rightarrow r)$ is false, since it is true implies false. This is recorded in the eighth column. Finally, the tenth column is marked as true because false $(p \Rightarrow q \Rightarrow r)$ implies false $((p \Rightarrow q) \Rightarrow (p \Rightarrow r))$.

Again, this justification relies on the semantics of truth values, but it says that those semantics will agree with these semantics. There is another intuition we can use instead. The claim of the axiom is that $(\varphi \Rightarrow \psi \Rightarrow \chi) \Rightarrow (\varphi \Rightarrow \psi) \Rightarrow (\varphi \Rightarrow \chi)$. We could read this in English as "that φ guarantees that ψ guarantees χ implies that if φ implies ψ, then φ also implies χ." Hopefully reading this sentence to yourself seven or eight times will convince you of the axiom.

These two axioms of $\Rightarrow$ are incredibly important. Notice that so far, we only have one rule, and all of our axioms are defined in terms of $\Rightarrow$. A consequence is the following theorem.

Theorem 3.4.10. The *deduction theorem* states that for a sequence of propositions $\varphi_1, \psi_2, \ldots, \varphi_n$ and another proposition ψ,

$$\frac{\varphi_1 \quad \varphi_2 \quad \cdots \quad \varphi_n}{\psi} \text{ if and only if } \frac{}{\varphi_n \Rightarrow \varphi_{n-1} \Rightarrow \cdots \Rightarrow \varphi_1 \Rightarrow \psi}.$$

This is to say, if one can prove ψ from the assumption $\varphi_1 \ldots \varphi_n$, then one can prove $\varphi_n \Rightarrow \cdots \Rightarrow \varphi_1 \Rightarrow \psi$ from no assumptions, and vice versa.

In short, one can "move" the assumptions of a theorem into the conclusion using the $\Rightarrow$ connective, and one can also move them back if desired. This is one of the few theorems that I will not present a proof of. The proof requires using a definition of a formal proof, which we have not yet presented. If you wish to see a proof of this theorem, it will be presented in any decent textbook on formal logic.

This theorem is important because it is often useful to prove a statement one way, but use the other way. For example, we will present this list of alternate forms of preexisting rules, axioms, and theorems.

- $\dfrac{\varphi \wedge \psi}{\varphi}$ L$\wedge$elim.
- $\dfrac{\varphi \wedge \psi}{\psi}$ R$\wedge$elim.
- $\dfrac{\varphi \Rightarrow \psi \Rightarrow \chi \quad \varphi \wedge \psi}{\chi}$ $\wedge$elim.
- $\dfrac{\varphi \quad \psi}{\varphi \wedge \psi}$ $\wedge$intro.
- $\dfrac{\varphi}{\varphi \vee \psi}$ L$\vee$intro.
- $\dfrac{\psi}{\varphi \vee \psi}$ R$\vee$intro.
- $\dfrac{\varphi \Rightarrow \chi \quad \psi \Rightarrow \chi \quad \varphi \vee \psi}{\chi}$ $\vee$elim.
- $\dfrac{\varphi}{\psi \Rightarrow \varphi}$ $\Rightarrow$intro.
- $\dfrac{\varphi \quad \varphi \Rightarrow \psi \quad \varphi \Rightarrow \psi \Rightarrow \chi}{\chi}$ trans.

Exercise 3.4.5. Use a truth table to convince yourself of the deduction theorem for small values of n, maybe 1 and 2.

Exercise 3.4.6. Try to use something intrinsic about the structure of proofs to convince yourself of the deduction theorem. This is difficult, so do not be concerned if you get stuck.

The deduction theorem also gives us the following result.

Theorem 3.4.11. The *identity theorem* states that for a proposition φ,

$$\frac{}{\varphi \Rightarrow \varphi}\,\text{id.}\quad .$$

This states that φ guarantees the truth of itself.

Proof. Although we have not discussed it, there is a straightforward proof of the for the claim

$$\frac{\varphi}{\varphi}\,\text{id.}\quad .$$

This is just the proof

$$\varphi.$$

This may not look like a proof, but it has one proposition with no lines above it (φ) and one proposition with no line below it (φ). Thus, this is a proof with assumption φ and conclusion φ. Applying the deduction theorem, we can move a copy of φ from an assumption to a conclusion to find that

$$\frac{}{\varphi \Rightarrow \varphi}\,.$$

□

We will present one last important theorem of implication.

Theorem 3.4.12. For a propositions, φ, ψ, and χ,

$$\frac{\varphi \Rightarrow \psi \quad \psi \Rightarrow \chi}{\varphi \Rightarrow \chi}\,\Rightarrow\!\text{trans.}\quad .$$

Intuitively, this states that if the truth of φ guarantees the truth of ψ, and the truth of ψ guarantees the truth of χ, then the truth of φ also guarantees the truth of χ.

Proof. Using the deduction theorem, we can instead prove

$$\frac{\varphi \quad \varphi \Rightarrow \psi \quad \psi \Rightarrow \chi}{\chi}\quad .$$

This proof is not difficult.

$$\dfrac{\dfrac{\varphi \quad \varphi \Rightarrow \psi}{\psi}\Rightarrow\text{elim.} \qquad \psi \Rightarrow \chi}{\chi}\Rightarrow\text{elim.}$$

.

□

Next, we describe the behavior of our two quantifiers. Each one will have a rule and an axiom. First, we begin with the universal quantifier.

Rule 3.11. The rule of $\forall$ *introduction* states that for propositions φ and ψ, a variable x not free in φ, and a variable y not free in φ and unused in ψ.

$$\dfrac{\varphi \Rightarrow \psi}{\varphi \Rightarrow \forall y.\, [x\colon y]\psi}\,\forall\text{intro.}$$

.

In effect, this rule says that if we have knowledge of $\varphi \Rightarrow \psi$ (which means that we have such knowledge for all possible variable assignments) and φ does not depend on x (in effect, the value of x has no bearing on the truth of φ), then we also have knowledge of $\varphi \Rightarrow \forall x.\, \psi$. In a simple case, suppose that φ has a single free variable a and ψ has two free variables a and b. Suppose also that we know $\varphi \Rightarrow \psi$ for all variable assignments. That is, we know $[a\colon u, b\colon v]\varphi \Rightarrow [a\colon u, b\colon v]\psi$ for all sets u and v. This is the same thing as saying $[a\colon u]\varphi \Rightarrow [a\colon u, b\colon v]\psi$ because φ does not depend on the value of b. So for any sequence of sets $v_1, \ldots$, we know that $[a\colon u]\varphi \Rightarrow [a\colon u, b\colon v_1]\psi$, $[a\colon u]\varphi \Rightarrow [a\colon u, b\colon v_2]\psi$, $[a\colon u]\varphi \Rightarrow [a\colon u, b\colon v_3]\psi$, and so on. Thus, since this implication holds for *all* values of b, it must also be that $[a\colon u]\varphi \Rightarrow [a\colon u](\forall b.\, \psi)$.

In our rule, the variable x could be (and likely is) free in our assumption $(\varphi \Rightarrow \psi)$. Thus, we cannot bind the variable x in our conclusion (for then we could not apply the same variable assignments to the assumption and conclusion). Therefore, we must change it out for another variable, y. This variable should not be free in φ or ψ so that it is not free in the assumptions and so that $\varphi \Rightarrow \forall y.\, [x\colon y]\psi$ is a valid proposition. This variable should also not be bound in ψ so that it may be used for quantification.

Axiom 3.12. The axiom of $\forall$ *elimination* states that for a proposition φ and variables x and y not bound in φ,

$$\dfrac{}{\forall y.\, [x\colon y]\varphi \Rightarrow \varphi}\,\forall\text{elim.}$$

.

In effect, this axiom states that if we have knowledge of $\forall x.\, \varphi$, then we also have knowledge of φ for every value of x. We require x to not be bound in φ so that $\forall x.\,\varphi$ is a valid proposition, and then switch out the variable x with a variable y so that $\forall y.\,[x\colon y]\varphi \Rightarrow \varphi$ is a valid proposition. This variable must not be bound so that this substitution is valid.

We can use these axioms to prove one of the most important theorems of quantifiers.

Theorem 3.4.13. The $\forall$ *conversion theorem* states that for a proposition φ, a variable x not bound in φ, and a variable y unused in φ,

$$\dfrac{\forall x.\,\varphi}{\forall y.\,[x\colon y]\varphi}\,\forall\text{conv.}\quad .$$

Proof. Let z be a variable unused in φ. We will need this variable for an intermediate substitution. Since x is not bound in φ and z is unused in φ, both x and z are both not bound in $[x\colon z]\psi$. This allows us to apply the $\forall$ elimination axiom,

$$\dfrac{}{\forall x.\,[z\colon x][x\colon z]\varphi \to [x\colon z]\varphi}\,\forall\text{elim.}\quad .$$

Note that $[z\colon x][x\colon z]\varphi$ is just φ. Further, y is not used in $[x\colon z]\varphi$, so we can use the $\forall$ introduction rule

$$\dfrac{\top \Rightarrow [x\colon z]\varphi}{\top \Rightarrow \forall y.\,[z\colon y][x\colon z]\varphi}\,\forall\text{intro.}\quad .$$

Here, note that $[z\colon y][x\colon z]\varphi$ is just $[x\colon y]\varphi$. With this in mind, we can construct our proof.

$$\dfrac{\dfrac{}{\top}\,\top \qquad \dfrac{\dfrac{\dfrac{\forall x.\,\varphi \qquad \dfrac{}{\forall x.\,[z\colon x][x\colon z]\varphi \Rightarrow [x\colon z]\varphi}\,\forall\text{elim.}}{[x\colon z]\varphi}\,\Rightarrow\text{elim.}}{\top \Rightarrow [x\colon z]\varphi}\,\Rightarrow\text{intro.}}{\top \Rightarrow \forall y.\,[z\colon y][x\colon z]\varphi}\,\forall\text{intro.}}{\forall y.\,[x\colon y]\varphi}\,\Rightarrow\text{elim.}$$

$\square$

This says that we are allowed to switch out a quantified variable for an unused variable. Intuitively, this makes sense because it should not matter what the name of our quantified variable is.

The rule and axiom for the existential quantifier are similar.

Rule 3.13. The rule of $\exists$ *elimination* states that for propositions φ and ψ, a variable x not free in ψ, and a variable y not free in ψ and unused in φ.

$$\frac{\varphi \Rightarrow \psi}{\exists y.\, [x\colon y]\varphi \Rightarrow \psi}\,\exists\text{elim}.$$

This rule says that if we have knowledge of $\varphi \Rightarrow \psi$ (for all variable assignments) and ψ does not depend on x, then the existence of one assignment $[x\colon v]$ such that φ also guarantees ψ. In a simple case, suppose that φ has free variables a and b and φ has one free variables a. Suppose also that we know $\varphi \Rightarrow \psi$ for all variable assignments. That is, we know $[a\colon u, b\colon v]\varphi \Rightarrow [a\colon u, b\colon v]\psi$ for all sets u and v. This is the same thing as saying $[a\colon u, b\colon v]\varphi \Rightarrow [a\colon u]\psi$ because ψ does not depend on the value of b. So for any sequence of sets $v_1, \ldots$, we know that $[a\colon u, b\colon v_1]\varphi \Rightarrow [a\colon u]\psi$, and so on. Thus, since this implication holds for *all* values of b, it must also be that $[a\colon u](\exists b.\, \varphi) \Rightarrow [a\colon u]\psi$.

In our rule, the variable x could be (and likely is) free in our assumption ($\varphi \Rightarrow \psi$). Thus, we cannot bind the variable x in our conclusion (for then we could not apply the same variable assignments to the assumption and conclusion). Therefore, we must change it out for another variable, y. This variable should not be free in φ or ψ so that it is not free in the assumptions and so that $\exists y.\, [x\colon y]\varphi \Rightarrow \psi$ is a valid proposition. This variable should also not be bound in φ so that it may be used for quantification.

Axiom 3.14. The axiom of $\exists$ *introduction* states that for a proposition φ and variables x and y not bound in φ,

$$\frac{}{\varphi \Rightarrow \exists y.\, [x\colon y]\varphi}\,\exists\text{intro}.$$

In effect, this axiom states that knowledge of φ yields knowledge of $\exists x.\, \varphi$. We require x to not be bound in φ so that $\exists x.\, \varphi$ is a valid proposition, then switch out the variable x with a variable y so that $\varphi \Rightarrow \exists y.\, [x\colon y]\varphi$ is a valid proposition. This variable must not be bound so that this substitution is valid.

There is a conversion theorem for $\exists$ just like the one for $\forall$.

Theorem 3.4.14. The $\exists$ *conversion theorem* states that for a proposition φ and variables x, y not bound and unused respectively in φ,

$$\frac{\exists x.\, \varphi}{\exists y.\, [x\colon y]\varphi}\,\exists\text{conv}.$$

Exercise 3.4.7. Prove this theorem.

There is actually a much stronger version of the conversion theorem.

Theorem 3.4.15. The *conversion theorem* states that for a proposition φ, a variable x not free in φ, and a variable y unused in φ,

$$\frac{\varphi}{[x\colon y]\varphi}\,\text{conv.}\quad .$$

and that for propositions $\varphi_1, \ldots, \varphi_n$ and ψ, a variable x, and a variable y unused in the $\varphi_1, \ldots, \varphi_n$ and in ψ,

$$\frac{\varphi_1 \quad \cdots \quad \varphi_n}{\psi} \quad \text{if and only if} \quad \frac{[x\colon y]\varphi_1 \quad \cdots \quad [x\colon y]\varphi_n}{[x\colon y]\psi} \quad .$$

The first part of this theorem states that if φ is a proposition and x is not a free variable of φ, then we are free to replace the variable x with any unused variable y without affecting the truth value under any variable assignment. The second part of this theorem states that any proof of a proposition does not depend on which variables in particular are chosen in the proposition. That is, we can exchange a variable in a proof for an unused one and still get a valid proof.

The proof of the first part of the theorem requires a *proof by induction* over propositions. Such proofs are long, so we will only do one such proof in the entire document. This will be the proof of the substitution theorem. Thus, we will not formally prove the first part of the conversion theorem. If you wish to prove it yourself, first observe the form of the substitution theorem in the next section, and repeat a similar structure for this proof. The intuition, however, is simple: If the variable x is used in φ, since it is not free, it must be quantified somewhere. Apply the particular conversion theorem required for that quantifier, and you are good to go.

The proof of the second part of the theorem requires the full definition of a proof. Since we do not have a full definition of a proof, we leave a proof of the conversion theorem to other texts. However, again, the intuition is simple: None of our rules or axioms depended on the specific variables chosen. Thus, if you have a proof

$$\frac{\varphi_1 \quad \cdots \quad \varphi_n}{\psi} \quad ,$$

simply exchange the variables at each step, leaving the individual steps alone. This will result in a proof

$$\frac{[x\colon y]\varphi_1 \quad \cdots \quad [x\colon y]\varphi_n}{[x\colon y]\psi} \quad .$$

Finally, I would like to discuss some properties of the equivalence connective, $\Leftrightarrow$.

Theorem 3.4.16. Equivalence is *symmetric*. That is, for propositions φ and ψ,

$$\frac{\varphi \Leftrightarrow \psi}{\psi \Leftrightarrow \varphi}\,\Leftrightarrow\text{sym.}$$

Intuitively, this states that if φ and ψ have the same truth value, then ψ and φ have the same truth value. In essence, this follows from the symmetry of $\wedge$.

Exercise 3.4.8. Recalling the definition of $\Leftrightarrow$, create a proof of this theorem.

Theorem 3.4.17. Equivalence is *reflexive*. That is, for a proposition φ,

$$\frac{}{\varphi \Leftrightarrow \varphi}\,\Leftrightarrow\text{refl.}$$

Intuitively, this states that φ has the same truth value as itself.

Proof.

$$\frac{\dfrac{}{\varphi \Rightarrow \varphi}\,\text{id.}}{\varphi \Leftrightarrow \varphi}\,\wedge\text{intro}$$

□

Notice that in this proof, the $\wedge$ introduction only had one assumption. This is because $\varphi \Leftrightarrow \varphi$ is $(\varphi \Rightarrow \varphi) \wedge (\varphi \Rightarrow \varphi)$. Thus, our one proof of $\varphi \Rightarrow \varphi$ provides both of the required assumptions.

Theorem 3.4.18. Equivalence is *transitive*. That is, for propositions φ, ψ, and χ,

$$\frac{\varphi \Leftrightarrow \psi \quad \psi \Leftrightarrow \chi}{\varphi \Leftrightarrow \chi}\,\Leftrightarrow\text{trans.}$$

Intuitively, this states that if φ and ψ have the same truth value, and ψ and χ have the same truth value, then φ and χ have the same truth value.

Proof.

$$\frac{\dfrac{\dfrac{\varphi \Leftrightarrow \psi}{\varphi \Rightarrow \psi}\,\text{L}\wedge\text{elim.} \quad \dfrac{\psi \Leftrightarrow \chi}{\psi \Rightarrow \chi}\,\text{L}\wedge\text{elim.}}{\varphi \Rightarrow \chi}\,\Rightarrow\text{trans.} \qquad \dfrac{\dfrac{\psi \Leftrightarrow \chi}{\chi \Rightarrow \psi}\,\text{R}\wedge\text{elim.} \quad \dfrac{\varphi \Leftrightarrow \chi}{\psi \Rightarrow \varphi}\,\text{R}\wedge\text{elim.}}{\chi \Rightarrow \varphi}\,\Rightarrow\text{trans.}}{\varphi \Leftrightarrow \chi}\,\wedge\text{intro.}$$

□

Formal proofs are not particularly important in the remainder of the thesis, so do not worry if you find these rules or axioms difficult. This content is presented here for two primary purposes: so that you can see what axioms look like and so that you can say you've seen a formal proof.

3.5. **Recommended Reading.** A much lighter introduction to the concepts of logic and proof are presented in Devlin[2]. A more complete introduction can be found in Magnus[9] (which is freely available online). He covers all of this content, albeit with a slightly different perspective. Additionally, I think he falls short in his discussion of proofs. A complete discussion of proof theory can be found in Buss[1]. In particular, Buss presents a short proof of the deduction theorem early on.

Another detailed discussion of both logic and proofs can be found in Mints[10]. This book focuses on intuitionistic logic, which is the subject I most regret not including. I highly recommend this text to the dedicated reader, but I am not sure how approachable it is to a beginner.

4. Set Theory

Now that we have covered a significant portion of logic, we can begin discussing set theory. While predicate logic describes how mathematicians are allowed to talk, set theory describes what mathematicians are talking about. Set theory is the conventionally accepted foundation for mathematics, although there are important alternatives such as type theory. The particular instance of set theory we will use is essentially the same as ZFC, an axiomatization of set theory named after mathematicians Ernst Zermelo and Abraham Fraenkel (the 'Z' and the 'F'). The 'C' is for the axiom of choice, which we will discuss soon.

As defined informally earlier, a set is an unordered collection of other sets, called elements, counted without repetition. These sets have one predicate, $\in$, representing if one set is an element of another. Practically speaking, the elements of a set can be anything you want, but strictly speaking, they are other sets. Discreetly, this implies that we are able to encode anything we want as a set.

Because set theory is the foundation of mathematics, we cannot formally define a set. Thus, you can come up with any notion of a set that you want, and I'll use that one, as long as it follow the axioms I present in this section. The axioms in the last section described the behavior of all types of propositions excluding $\in$. Now, we will work with this type of proposition in detail.

4.1. **Equality.** I have said that you must provide a notion of a set (an unordered collection), but I have not said that you must provide a notion of when two collections are "the same." I will provide this notion for you.

Definition 4.1.1. For variables x and y, define the proposition $x = y$, read as "x equals y," as shorthand for $\forall z. (x \in z \Leftrightarrow y \in z) \wedge (z \in x \Leftrightarrow z \in y)$ where z is any other variable. That is,

$$x = y \colon \forall z. ((x \in z \Leftrightarrow y \in z) \wedge (z \in x \Leftrightarrow z \in y)).$$

Generally, the variable z will be chosen not to conflict with the variables in neighboring expressions. If we have a sequence of variables $x_1, \ldots, x_n$, then the proposition

$$x_1 = x_2 = \cdots = x_n$$

is shorthand for the proposition

$$(x_1 = x_2) \wedge (x_2 = x_3) \wedge \cdots \wedge (x_{n-1} = x_n).$$

We also define $x \neq y$, read as "x is not equal to y," by

$$x \neq y \colon (x = y)\neg.$$

This operator will appear alongside $\in$ in the order of operations.

That is, two sets are equal when they both have the same elements and are elements of the same sets. In other words, two sets are equal when they cannot be distinguished by use of our only predicate $\in$.

Intuitively, this is a good notion for equality of sets. However, I would like to do better. Consider the following informal sets: the collection of people with your full name born to parents of your full name and the collection of people currently reading this thesis. I would imagine that these collections both have exactly one element, that element being you. I would like to suppose that these are the same collections.

This is not a necessity of a theory of sets. For example, the collections could reasonably differ because of how they are defined. That is, you are an element of the first set because of your name and your parents' names, but you are an element of the second set because of your current activity. These could reasonably be two different collections. This is similar to how a white bag with two marbles is a different bag of marbles from a brown bag with the same two marbles. As one final example, it is conceivable that the set of even natural numbers and the set of natural numbers whose last digit is a 0, 2, 4, 6, or 8 could be different sets.

However, I would like to stipulate that a set is defined entirely by its elements. That is, if two sets have the same elements, they are the same set. In other words, the "packaging" does not matter. This is expressed by the following axiom.

Axiom 4.1. The axiom of *extensionality* states that

$$\frac{}{\forall a, b.\, (\forall c.\, (c \in a \Leftrightarrow c \in b) \Rightarrow \forall c.\, (a \in c \Leftrightarrow b \in c))}\,\text{ext.}$$

One should confirm that this axiom is a valid proposition and that it has no free variables. Notice something special about this axiom: nowhere did we have to state "for a proposition φ" or "for a variable x." Since propositions and variables are part of the language of propositions, each of the prior axioms and rules have actually been *axiom schemas*. This is the first axiom we have seen which has no second-order dependence. Because this distinction is surface level, we will refer to both axioms and axiom schemas as axioms. Also, we should note that we could have specified this as an axiom schema by letting the variables a, b, and c be arbitrary, but we can also just use the conversion theorems to replace the variables in this proper axiom.

Literally, the axiom of extensionality states that if two sets have the same elements, then they are elements of the same sets. Given our definition of equality of sets, this axiom states that two sets are equal whenever they have the same elements. That is, we cannot use predicate logic to distinguish between sets with the same elements. We can state this in the form of a theorem.

Theorem 4.1.2.

$$\frac{}{\forall a, b.\, (\forall c.\, (c \in a \Leftrightarrow c \in b) \Rightarrow a = b)}\,\text{ext.}$$

Exercise 4.1.1. Use the corresponding theorems for $\Leftrightarrow$ to prove the following theorems of equality.

Theorem 4.1.3. For variables x and y,

$$\frac{x = y}{y = x}\,{=}\text{sym.}$$

Theorem 4.1.4. For a variable x,

$$\frac{}{x = x}\,{=}\text{refl.}$$

Theorem 4.1.5. For variables x, y, and z,

$$\frac{x = y \quad y = z}{x = z}\,{=}\text{trans.}$$

Now, I would like to present our first example of a *proof by induction*. A proof by induction can be used to prove theorems about any inductive construction. This will be the last proof in this thesis which uses the language of formal proofs. Since inductive constructions are built using recipes, a proof by induction proves the theorem for each recipe. In general, however, this is too hard to do without assistance. That is, these recipes have ingredients, and we might need to know something about the ingredients to conclude something about the entire recipe. This is codified by an *induction principle*. We will present proofs by induction formally in a later section. For now, just try to observe this one.

Theorem 4.1.6. For a proposition φ and variables x and y not bound in φ,

$$\frac{}{x = y \Rightarrow \varphi \Rightarrow [x\colon y]\varphi}\,\text{sub.}$$

Proof. Using the deduction theorem, we can instead prove

$$\frac{x = y \quad \varphi}{[x\colon y]\varphi}\,\text{sub.}$$

We proceed by induction on φ. In particular, this proof will depend on the value of $[x\colon y]\varphi$, so our induction cases will be identical to those in the definition of variable replacement.

(1) Suppose φ is of the form $\bot$. Then, we are seeking to prove $[x\colon y]\varphi\colon \bot$ from $\varphi\colon \bot$. Here, we use the explosion principle.

$$\frac{\bot}{\bot}\,\text{explos.}$$

(2) Suppose φ is of the form $z \in w$ for variables z and w. Then, we have the following four sub-cases:
   - If neither z nor w is x, we are seeking to prove $[x\colon y]\varphi\colon z \in w$ from $\varphi\colon z \in w$. Here, we use the identity proof.

     $$\frac{z \in w}{z \in w}\,\text{id.}$$

     We could have used this same proof for the last case, but the explosion principle is more fun.

- If z is x and w is not x, we are seeking to prove $[x\colon y]\varphi\colon y \in w$ from $\varphi\colon x \in w$. This is the first interesting case, for now φ and $[x\colon y]\varphi$ actually differ. Here, we must actually use knowledge of $x = y$.

$$\dfrac{x \in w \qquad \dfrac{\dfrac{\dfrac{x = y}{(x \in w \Leftrightarrow y \in w) \wedge (w \in x \Leftrightarrow w \in y)}\,\forall\text{elim.}}{x \in w \Leftrightarrow y \in w}\,\text{L}\wedge\text{elim.}}{x \in w \Rightarrow y \in w}\,\text{L}\wedge\text{elim.}}{y \in w}\,\Rightarrow\text{elim.}$$

- If z is not x and w is x, we are seeking to prove $[x\colon y]\varphi\colon z \in y$ from $\varphi\colon z \in x$. This proof is almost identical to the last one.

  Exercise 4.1.2. Mimic the form of the last proof to prove this case.

- If both z and w are x, we are seeking to prove $[x\colon y]\varphi\colon y \in y$ from $\varphi\colon x \in x$. This proof is essentially a combination of the two prior proofs. Fortunately, this proof is as complicated as it gets.

$$\dfrac{\dfrac{x \in x \qquad \dfrac{\dfrac{\dfrac{x = y}{(x \in x \Leftrightarrow y \in x) \wedge (x \in x \Leftrightarrow x \in y)}\,\forall\text{elim.}}{x \in x \Leftrightarrow x \in y}\,\text{R}\wedge\text{elim.}}{x \in x \Rightarrow x \in y}\,\text{L}\wedge\text{elim.}}{x \in y}\,\Rightarrow\text{elim.} \qquad \dfrac{\dfrac{\dfrac{x = y}{(x \in y \Leftrightarrow y \in y) \wedge (y \in x \Leftrightarrow y \in y)}\,\forall\text{elim.}}{x \in y \Leftrightarrow y \in y}\,\text{L}\wedge\text{elim.}}{x \in y \Rightarrow y \in y}\,\text{L}\wedge\text{elim}}{y \in y}\,\Rightarrow\text{elim.}$$

(3) Suppose φ is of the form $\psi_1 \wedge \psi_2$ for propositions ψ_1 and ψ_2. Then, we seek to prove $[x\colon y]\varphi\colon [x\colon y]\psi_1 \wedge [x\colon y]\psi_2$. In general, we have no way of proving $[x\colon y]\varphi$. However, let us assume two *inductive hypotheses* which state that we have already performed this proof on our ingredients ψ_1 and ψ_2. That is, suppose we know

$$\dfrac{x = y \quad \psi_1}{[x\colon y]\psi_1}\,\text{IH1} \quad \text{and} \quad \dfrac{x = y \quad \psi_2}{[x\colon y]\psi_2}\,\text{IH2}.$$

We can present a proof in terms of these inductive hypotheses.

$$\dfrac{\dfrac{x = y \qquad \dfrac{\psi_1 \wedge \psi_2}{\psi_1}\,\text{L}\wedge\text{elim.}}{[x\colon y]\varphi_1}\,\text{IH1} \qquad \dfrac{x = y \qquad \dfrac{\psi_1 \wedge \psi_2}{\psi_2}\,\text{R}\wedge\text{elim.}}{[x\colon y]\psi_2}\,\text{IH2}}{[x\colon y]\psi_1 \wedge [x\colon y]\psi_2}\,\wedge\text{intro.}$$

(4) Suppose φ is of the form $\psi_1 \vee \psi_2$ for propositions ψ_1 and ψ_2. Then, we seek to prove $[x\colon y]\varphi\colon [x\colon y]\psi_1 \vee [x\colon y]\psi_2$. This time, we will assume the inductive hypotheses in a different form:

$$\dfrac{x = y}{\psi_1 \Rightarrow [x\colon y]\psi_1}\,\text{IH1} \quad \text{and} \quad \dfrac{x = y}{\psi_2 \Rightarrow [x\colon y]\psi_2}\,\text{IH2}.$$

This is justified by the deduction theorem. Then, we can present our proof.

$$\dfrac{\psi_1 \vee \psi_2 \qquad \dfrac{x = y}{\psi_1 \Rightarrow [x\colon y]\psi_1}\,\text{IH1} \qquad \dfrac{x = y}{\psi_2 \Rightarrow [x\colon y]\psi_2}\,\text{IH2}}{[x\colon y]\psi_1 \vee [x\colon y]\psi_2}\,\vee\text{elim.}$$

(5) Suppose φ is of the form $\psi_1 \Rightarrow \psi_2$ for propositions ψ_1 and ψ_2.

Exercise 4.1.3. Mimic the form of the previous two proofs to come up with inductive hypotheses and a proof that $x = y$ and φ prove $[x\colon y]\varphi$ in this case.

(6) Suppose φ if of the form $\forall z.\,\psi$ for a proposition ψ and a variable z. Since we supposed that x and y were not bound in φ, z must be distinct both x and y. Thus, we are seeking to prove $[x\colon y]\varphi\colon \forall z.\,[x\colon y]\psi$ from $\varphi\colon \forall z.\,\psi$. Let w be an unused variable. As our inductive hypothesis, suppose

$$\dfrac{x = y \quad \psi}{[x\colon y]\psi}\,\text{IH}.$$

By the conversion theorem, this is the same as supposing

$$\dfrac{x = y \quad \psi[z\colon w]}{[z\colon w, x\colon y]\psi}\,\text{IH}.$$

Notice that variable substitutions like $[w\colon z, z\colon w]\psi\colon \psi$ do not change propositions, but instead are there to match the forms of our rules and axioms. Now, we can present our proof.

$$\dfrac{\dfrac{}{\top}\,\top \qquad \dfrac{\dfrac{\dfrac{x = y \qquad \dfrac{\forall z.\,\psi \qquad \dfrac{}{\forall z.\,[w\colon z, z\colon w]\psi \Rightarrow [z\colon w]\psi}\,\forall\text{elim.}}{[z\colon w]\psi}\,{\Rightarrow}\text{elim.}}{[z\colon w, x\colon y]\psi}\,\text{IH}}{\top \Rightarrow [z\colon w, x\colon y]\psi}\,{\Rightarrow}\text{intro.}}{\top \Rightarrow \forall z.\,[w\colon z, z\colon w, x\colon y]\psi}\,\forall\text{intro.}}{\forall z.\,[x\colon y]\psi}\,{\Rightarrow}\text{elim.}$$

(7) Suppose φ if of the form $\exists z.\, \psi$ for a proposition ψ and a variable z.

Exercise 4.1.4. Mimic the form of the previous proof to develop an inductive hypothesis and a proof for this case.

$\square$

Again, I will summarize what this proof accomplished. We wished to show that for all propositions φ,

$$\frac{x = y \quad \varphi}{[x\colon y]\varphi}\,\text{sub.}$$

There are seven recipes for a proposition, so we created a proof for each recipe. For some of these recipes, we did not need an inductive hypothesis. That is, we did not need to suppose anything about our ingredients to develop the proof. These were the first two cases. Each other case required an inductive hypothesis. For example, the proposition $\psi_1 \wedge \psi_2$ has ingredients ψ_1 and ψ_2. These were each constructed using a recipe as well, and same with their ingredients. Thus, a proof of the theorem consists of proofs for some cases which do not involve inductive hypotheses, and then demonstrations of how to stitch proofs for ingredients into proofs for recipes. This is what occurred in each of the other cases. Even if the proof of the substitution theorem seems elusive, this point should be clear: equality of sets means that the sets may be substituted for each other in any proposition.

The notion of equality allows us to define another quantifier: the *uniqueness* quantifier.

Definition 4.1.7. For a proposition φ and a variable x not bound in φ, we define the proposition $\exists! x.\, \varphi$, read as "there exists a *unique* x such that φ," as

$$\exists! x.\, \varphi\colon (\exists x.\, \varphi) \wedge (\forall y. \forall z.\, [x\colon y]\varphi \wedge [x\colon z] \Rightarrow y = z)$$

where y and z are distinct unused variables of φ. Further, for a proposition φ and a nonempty sequence of distinct variables $x_1, \ldots, x_n$ not bound in φ, we define

$$\begin{aligned} \exists! x_1, \ldots, x_n.\, \varphi\colon & (\exists x_1, \ldots, x_n.\, \varphi) \wedge \\ & (\forall y_1, \ldots, y_n. \forall z_1, \ldots, z_n. \\ & [x_1\colon y_1, \ldots, x_n\colon y_n]\varphi \wedge [x_1\colon z_1, \ldots, x_n\colon z_n]\varphi \Rightarrow \\ & y_1 = z_1 \wedge \cdots \wedge y_n = z_n) \end{aligned}$$

where $y_1, \ldots, y_n$ and $z_1, \ldots, z_n$ are distinct unused variables of φ. If n is 0, $\exists! x_1, \ldots, x_n.\, \varphi$ should be interpreted as just φ.

While this definition may seem complicated, it really is not so bad. The proposition $\exists! x.\, \varphi$ is that there exists some x such that φ holds, and for any sets y and z, if $[x\colon y]\varphi$ and $[x\colon z]$, then $y = z$. That is to say, if there are multiple sets x such that φ, then they are equal (and so are actually the same set). The definition with sequences says that there exists a sequence $x_1, \ldots, x_n$ such that φ, and any pair of sequences satisfying φ are (pairwise) equal.

4.2. **Single-Valued Functions.** If there is one largest flaw of this thesis so far, it is that I have not told you how to write down a set. We can write down propositions and proofs, and we have proven properties of sets, and even described an axiom of them, but yet, we have not written down a set. This is because, formally speaking, it is very difficult to define how to write down a set.

Definition 4.2.1. A *single-valued propositional function* or just *function* for short, is a proposition Φ[6] in the free variables $x_1, \ldots, x_n, y$ such that

$$\forall x_1, \ldots, x_n.\, \exists! y.\, \Phi.$$

The variables $x_1, \ldots, x_n$ are called the inputs and the variable y is called the output. That is, a proposition is a function when for any choice of inputs, there is exactly one choice of output satisfying the proposition. In other words, a function is some sort of process or algorithm, determined by a proposition, which transforms every input into some output. If Φ is a function with inputs $x_1, \ldots, x_n$ and an output y, we write

$$\Phi\colon x_1, \ldots, x_n \mapsto y,$$

read as "Φ maps $x_1, \ldots, x_n$ to y." Importantly, this use of the : is *not* meaning "defined to be." Instead, this stands for the proposition that defines a function. A function with no inputs, so a proposition Φ such that $\exists! y.\, \Phi$ is called a *propositional set* or simply a *constant.* Such propositions choose a unique set from our domain of discourse. We will revisit the definition of a function soon.

Propositional sets are, of course, not sets; they are propositions. But, propositional sets allow us to use the syntax of propositions to represent unique sets. This is very powerful. Unfortunately, we still can't actually create an example of a propositional set. Regardless, we will do it anyway.

[6] Φ is uppercase φ. We will use this as well as Ψ (uppercase ψ) for propositional functions.

Theorem 4.2.2. The *empty set* theorem states that

$$\exists a.\, \forall b.\, a \notin b.$$

Intuitively, this states that there exists a set with no elements. We are not yet able to prove this theorem, so we will just accept it for now. Since two such sets a would have the same elements (that is, would both have no elements), the axiom of extensionality yields that this set is unique. This unique set is a special constant.

Definition 4.2.3. The *empty set*, denoted by $\emptyset$, is defined to be the constant

$$\emptyset \colon \forall y.\, y \notin x$$

where x and y are variables chosen based on context.

By the empty set theorem, this is a propositional set. The importance of this definition is that the empty set, $\emptyset$, is the set with no elements.

Of course, how we have defined it, $\emptyset$ is actually a proposition, but there is a unique set which makes it true, a unique "solution" so to speak. Constants such as this, as well as functions, are not useful until we can use them like actual sets.

Definition 4.2.4. Suppose φ and ψ are propositions and x is a variable. Then, the propositional variable assignment $[x\colon \psi]\varphi$ is shorthand for

$$[x\colon \psi]\varphi \colon \forall x.\, (\psi \Rightarrow \varphi).$$

This is only defined when the right side is a proposition. It is sufficient to assume that no free variable of φ is bound in ψ and vice versa and that x is not bound in ψ or φ. This definition technically makes sense for all propositions, but it is most meaningful when ψ is a function.

For example, let φ denote the proposition $\exists b.\, b \in a$. This is a proposition representing that a has an element (that it is not empty). Then, the proposition $[a\colon \emptyset]\varphi$ should be a proposition representing that the empty set has an element (which it does not, by nature of being empty). This proposition can be fully expressed as

$$\forall a.\, (\forall b.\, b \notin a \Rightarrow \exists b.\, b \in a).$$

We can interpret this as "for every set a which is the empty set, a has an element." Notice that the variables a and b before the $\Rightarrow$, matching the definition of $\emptyset$, are chosen based on context. Indeed, if $\emptyset$ is replacing a variable x, this is the correct choice of x in the definition of $\emptyset$, and y should be some other bound or unused variable.

We are also allowed to "evaluate" these assignments when they are for syntactic sets. For example,

$$\exists b.\, b \in \emptyset$$

is a perfectly valid way of writing the previous proposition. In this case, an arbitrary unused variable is chosen for x, like a. Using the rules for quantifiers, it is not difficult to prove that this statement is false.

As another example, let us consider a constant Φ in the free variable a and the proposition $[b\colon [a\colon b]\Phi, c\colon [a\colon c]\Phi](b = c)$ (which represents something like $\Phi = \Phi$). Notice that we must replace the free variable of Φ for b and c so that this propositional variable assignment quantifies over the correct variables. This is the proposition

$$\forall b.\, ([a\colon b]\Phi \Rightarrow (\forall c.\, ([a\colon c]\Phi \Rightarrow b = c)).$$

We can pull the quantifiers to the front to find that this is equivalent to the proposition

$$\forall b. \forall c.\, ([a\colon b]\Phi \Rightarrow [a\colon c]\Phi \Rightarrow b = c)$$

which is equivalent to

$$\forall b. \forall c.\, ([a\colon b]\Phi \wedge [a\colon c]\Phi \Rightarrow a = b).$$

Since Φ is a constant, we know that $\exists! a.\, \Phi$. Recall the definition of the uniqueness quantifier, and compare it to the last proposition we wrote. This is the latter half of the $\wedge$ in the definition of $\exists!$, thus, this proposition is true. Therefore, for a constant Φ, $\Phi = \Phi$, which is as we would expect. Most of the time, our constants will be defined in terms of arbitrary variables like $\emptyset$ was. Thus, we can often write expressions closer to $[b\colon \emptyset, c\colon \emptyset](b \in c)$.

Next, consider the proposition $a = \emptyset$, which could be expanded as $\forall b.\, (\forall c.\, c \notin b \Rightarrow a = b)$. Using extensionality, this is equivalent to $\forall c.\ c \notin a$, or in other words that a is empty. In general, if Φ is a constant in the variable x, then $y = \Phi$ is equivalent to $[x\colon y]\Phi$.

It is reasonable to ask whether every set is a propositional set. In other words, for an actual set v, is it possible to find a constant Φ such that $[x\colon v]\Phi$ is true, which is that Φ represents the set v? The answer is generally considered to be "no", but technically speaking, it is up to you.

Finally, we consider a way to combine functions.

Theorem 4.2.5. Suppose Φ is a function with inputs $x_1, \ldots, x_n$ and output y and Ψ is a function with inputs $y, y_1, \ldots, y_m$ and output z and that no free variable of Φ is bound in Ψ and vice versa. Then, $[y\colon \Phi]\Psi$ is a function with inputs $x_1, \ldots, x_n, y_1, \ldots, y_m$ and output z.

Proof. The proposition $[y\colon \Phi]\Psi$ may be expanded as

$$\forall y. (\Phi \Rightarrow \Psi).$$

We seek to show that this is a function, which is that

$$\forall x_1, \ldots, x_n, y_1, \ldots, y_m. \exists! z. \forall y. (\Phi \Rightarrow \Psi).$$

We know that $\forall y_1, \ldots, y_n. \forall y. \exists! z. \Psi$ and that $\forall x_1, \ldots, x_n. \exists! y. \Phi$.

Consider any choice of inputs $x_1, \ldots, x_n, y_1, \ldots, y_m$. Then, we have a unique set y satisfying Φ. Thus, pick the unique set z satisfying Ψ with this value of y. This set z satisfies $\forall y. (\Phi \Rightarrow \Psi)$ because whenever Φ is true (that is, whenever y is the value we selected), Ψ is true (since we picked z based on our value of y). This is also the only z we could have chosen or else we would have two z's satisfying Ψ for our particular value of y. Thus, $[y\colon \Phi]\Psi$ is a function. $\square$

4.3. **Pairs and Unions.** Now that we have an example of a (propositional) set, $\emptyset$, and we know how to write down certain sets in terms of propositions, we might seek methods to find other sets. For example, we might expect there to be a set containing the empty set, and we might expect there to be a set containing the empty set and the set containing the empty set. We will provide these to create these sets and others through additional axioms of set theory.

The first such axiom is arguably the most obvious axiom of set theory.

Axiom 4.2. The axiom of *pairing* states that

$$\frac{}{\forall a, b. \exists c. a \in c \land b \in c}\,\text{pair}.$$

This axiom states that for every pair of sets, there is a set containing both of them. That is, we may take two sets and find another set containing both of them. Metaphorically, if we have two bags, we can put them both into a larger bag. Using the axiom of extensionality, we can convert this into the following form:

$$\forall a, b. \exists c. \forall d. (d = a \lor d = b \Rightarrow d \in c).$$

That is, for all sets a and b, there is a set c such that if any set d is equal to a or b, then d is in c. Intuitively, this is the same thing.

Unfortunately, this axiom alone is not enough to create a set whose elements are *exactly* a and b. That is, the set c may contain extra elements. In other words, we are unable to specify that we want c to contain *only* a and b. This is remedied by the next axiom.

Axiom 4.3. The axiom of *specification* states that for each sequence of distinct variables $x, y, z, w_1, \ldots, w_n$ and every proposition φ such that each of the free variables and none of the bound variables of φ are in the sequence $x, z, w_1, \ldots, w_n$ and such that y is unused in φ,

$$\forall w_1, \ldots, w_n. \forall z. \exists y. \forall x. (x \in y \Leftrightarrow x \in z \wedge \varphi)$$

This axiom states that for each set z, we can form a set y whose elements x are exactly those elements x of z for which the proposition φ holds. In other words, if we have a set z, we can specify that we only which to talk about the elements x which satisfy a proposition φ.

We can take the axiom of pairing and the axiom of specification and combine them to create a set whose elements are *exactly* the sets a and b. In particular, let φ be the proposition $d = a \vee d = b$ and let c be a set such that $d = a \vee d = b \Rightarrow d \in c$ as given by the axiom of pairing. If we take this set c to be the z in the axiom of specification, we get a set e such that $d \in e \Leftrightarrow d \in c \wedge (d = a \vee d = b)$. Since $(d = a \vee d = b) \Rightarrow d \in c$, the proposition $d \in c \wedge (d = a \vee d = b)$ is equivalent to $d = a \vee d = b$. Thus, e is a set such that $d \in e \Leftrightarrow d = a \vee d = b$. Such a set e must be unique (because we know its elements).

This leads us to the following definition.

Definition 4.3.1. For variables x and y, define the function $\{x, y\}$, read as "the set containing x and y," as

$$\forall w. (w \in z \Leftrightarrow w = x \vee w = y)$$

where the variables z and w are chosen from context. The inputs are x and y, and the output is z. For a single variable x, define the function $\{x\}$, read as "the set containing x," as

$$\forall w. (w \in z \Leftrightarrow w = x)$$

where z and w are again chosen from context. The input is x and the output is z.

Because $w = x \vee w = x$ is equivalent to $w = x$, $\{x\}$ and $\{x, x\}$ are equivalent propositions for all variables x. Using the axioms of pairing and specification, we know that

$$\forall a, b. \exists! c. \forall d. (d \in c \Leftrightarrow d = a \vee d = b).$$

That is, $\{a, b\}$ is a function with inputs a and b.

Due to our notation for replacement, it makes perfect sense to write $[a\colon\emptyset]\{a\}$ or simply $\{\emptyset\}$, which is a proposition in some free variable (say, x) determining if that variable's only element if the empty set, or in other words if it is the set $\{\emptyset\}$. That is, $\{\emptyset\}$ is a constant, and more generally $\{\Phi\}$ is a constant for every constant Φ. Even better, $\{\Phi, \Psi\}$ is function for all functions Φ and Ψ, and its inputs are the inputs of Φ and Ψ. In particular, $\{\emptyset, \{\emptyset\}\}$ is a constant, as is $\{\emptyset, \{\emptyset, \{\emptyset\}\}\}$.

We are now well on our way to creating very large sets. Unfortunately, we don't yet have a way to create a set with three or more elements. For this, we need another axiom.

Axiom 4.4. The axiom of *union* states that

$$\overline{\forall a.\, \exists b.\, \forall c, d.\, (c \in d \wedge d \in x \Rightarrow c \in b)}\bigcup.$$

Intuitively, this axioms states that for every set a (whose elements are also sets), there is a set b containing every element c of every element d of a. This set b, in a sense, joins all the elements of x, hence the term union. Metaphorically, this means that we can take a bag of bags and dump all the smaller bags into one single bag. Using the axiom of specification, we can create a syntactic set whose elements are exactly those elements c of elements d of x.

This provides a convenient definition.

Definition 4.3.2. For a variable x, define the function $x\bigcup$, read as "x unioned" or *x's union* by

$$x\bigcup\colon \forall z.\, z \in w \Leftrightarrow (\exists y.\, z \in y \wedge y \in x)$$

where the variables y, z, w are chosen based on context. The input is x and the output is w. For variables x and y, we also define the special shorthand $x \cup y$ as

$$x \cup y\colon \{x, y\}\bigcup.$$

Just like with ordered pairs, $x\bigcup$ becomes a constant (or function) when x is replaced by a constant (or function). For example, we can take

$$\{\{\emptyset, \{\emptyset\}\}, \{\{\{\emptyset\}\}\}\}\bigcup.$$

This set is difficult to read, but it is the pairing of $\{\emptyset, \{\emptyset\}\}$ and $\{\{\{\emptyset\}\}\}$ unioned. Consequently we could also write this set as

$$\{\emptyset, \{\emptyset\}\} \cup \{\{\{\emptyset\}\}\}.$$

This set has exactly three elements: $\emptyset$, $\{\emptyset\}$, and $\{\{\emptyset\}\}$. Thus, we have created a set with three elements.

We can create sets with as many elements as we want.

Definition 4.3.3. For a sequence of variables $x_1, \ldots, x_n$ define the function $\{x_1, \ldots, x_n\}$, read as "the set containing x_1, x_2, …, and x_n," as

$$\{x_1, \ldots, x_n\}\colon \forall z.\, z \in y \Leftrightarrow (z = x_1 \vee z = x_2 \vee \cdots \vee z = x_n)$$

where the variables y and z are chosen from context. The inputs are $x_1, \ldots, x_n$ and the output is y. In the special case of $n = 0$, the proposition $z = x_1 \vee \cdots \vee z = x_n$ should be interpreted as the proposition $\perp$.

Theorem 4.3.4. For a sequence of variables $x_1, \ldots, x_n$ and variables y and z not appearing in the list,

$$\forall x_1, \ldots, x_n.\, \exists! y.\, \forall z.\, (z \in y \Leftrightarrow z = x_1 \vee \cdots \vee z = x_n).$$

In other words, the proposition $\{x_1, \ldots, x_n\}$ is a function with inputs $x_1, \ldots, x_n$ and output z.

Proof. Here, we will see another (informal) proof by induction, although a much simpler one. In this proof, we will perform induction over the length of the list of variables.

The case where n is zero has already been proven in our discussion of the empty set, and the cases where n is 1 or 2 have been proven in our discussion of pairing.

Recall our initial discussion of the natural numbers. We noted that every natural number is either 0 or is the successor of a natural number. This is an inductive construction, just like propositions are, except it only has two recipes. We have already provided a proof for the first recipe. The second recipe takes a single ingredient, a natural number, and produces its successor.

Thus, here is our key argument: Suppose that

$$\forall x_1, \ldots, x_{n-1}.\, \exists! y.\, \{x_1, \ldots, x_{n-1}\}$$

which is that $\{x_1, \ldots, x_n\}$ is a function. Then, apply our pairing argument to create the function

$$\{\{x_1, \ldots, x_{n-1}\}, \{x_n\}\}.$$

Thus,

$$\{\{x_1, \ldots, x_{n-1}\}, \{x_n\}\} \bigcup$$

is function, and this is the set whose elements are exactly $x_1, \ldots, x_n$, so therefore $\{x_1, \ldots, x_n\}$ is a function, which is the desired result. $\square$

Exercise 4.3.1. Even with the very limited number of ways we can build sets, we now have many redundant ways of writing the same syntactic sets. Prove or otherwise convince yourself of the following equations for variables x and y:

(1) $\emptyset = \{\}$ (using the definition of $\{x_1, \ldots, x_n\}$ when $n = 0$)
(2) $\{x, y\} = \{y, x\}$ (thus, the order of the elements in a set is irrelevant)
(3) $\{x, y, y\} = \{x, y\}$ (thus, repetition of elements in a set is irrelevant)
(4) $\{\{x\}\}\bigcup = x$
(5) $x \cup y = y \cup x$

4.4. **Subsets and Set Builder Notation.** Recall a particular pattern of the last subsection: An axiom guaranteed the existence of some larger set containing elements we wanted, but we wanted a smaller set containing exactly those elements. We constructed these smaller sets using the axiom of specification. In particular, we were constructing *subsets*.

Definition 4.4.1. For variables x and y, define the proposition $x \subseteq y$, read as "x is a *subset* of y," as

$$x \subseteq y \colon \forall z. \, (z \in x \Rightarrow z \in y)$$

where z is some other variable. If $x \subseteq y$, we say that x is a *subset* of y.

Intuitively, this definition says that a set x is a subset of a set y if every element z of x is also an element z of y. If we have a set x, it may be useful to study certain subsets of x, in particular those which satisfy some proposition. For example, if x is the set of natural numbers and φ is a proposition representing "is even," then we might wish to use x and φ to create the set of even natural numbers.

We can achieve this with new notation for the axiom of specification.

Definition 4.4.2. For a proposition φ and variables x and y not bound in φ, define the function $\{x \in y \mid \varphi\}$, read as "the set of all x in y such that φ," as

$$\{x \in y \mid \varphi\} \colon \forall x. \, (x \in z \Leftrightarrow x \in y \wedge \varphi)$$

where z is a variable chosen from context. The inputs are y and the free variables of φ excluding x, and the output is z. This is call *set builder notation.*

This proposition represents that z is exactly the set of all elements x in y such that the proposition φ holds. These sets are guaranteed to exist by the axiom of specification. That is, this proposition is a function representing the special subset of y whose elements all satisfy φ. Like before, due to our replacement syntax, we can replace any of these variables, or the proposition φ, with particular examples.

Set builder notation has a variety of nice properties. For example,

$$\{x \in y \mid \varphi\} \subseteq y,$$

which says that the set of all the elements of y which satisfy φ is a subset of y. Further,

$$\forall x. (\varphi \Rightarrow \psi) \Rightarrow \{x \in y \mid \varphi\} \subseteq \{x \in y \mid \psi\},$$

which says that if $\varphi \Rightarrow \psi$, that is if every x satisfying φ also satisfies ψ, then the relevant subsets of y have the corresponding subset relationship.

Example 4.4.3. Let us explore this notation with some examples.

(1) $\{x \in \emptyset \mid \varphi\} = \emptyset$ for any proposition φ.
(2) $\forall y. \{x \in y \mid \bot\} = \emptyset$.
(3) $\forall y. \{x \in y \mid \top\} = y$.
(4) $\{x \in \{\emptyset\} \mid x = \emptyset\} = \{\emptyset\}$.
(5) $\{x \in y \mid x \in y\} = y$.
(6) $\{x \in \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \mid \emptyset \in x\} = \{\{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$. Here, the value of x is allowed to range over the values, $\emptyset$, $\{\emptyset\}$, and $\{\emptyset, \{\emptyset\}\}$, and we select only those sets x which contain the empty set.

The axiom of specification allows for creating a construction similar to that of unions.

Definition 4.4.4. For a set x, define the proposition $x \bigcap$, read as "x's intersection," as

$$x \bigcap : \forall y. (y \in z \Leftrightarrow \forall w. (w \in x \Rightarrow y \in w)),$$

where the variables y, z, and w are chosen from context. Although this is *not* a function, one should think of x as the input and z as the output. For two variables x and y, we define the function $x \cap y$, read as "x intersect y," as

$$x \cap y \colon \{x, y\} \bigcap .$$

This is actually a function with inputs x and y and output z.

The proposition $x \bigcap$ is that y is a set whose elements are exactly those sets z which are elements of every element w of x. Suppose a is a non-empty set and b is an element of a. Then, the axiom of specification guarantees that $a \bigcap$ is a unique set because

$$a \bigcap = \{c \in b \mid \forall d. (d \in x \Rightarrow c \in d)\}.$$

However, if a is empty, we cannot use set builder notation in this way, because we have no choice for the element b. This is the only obstruction to $x \bigcap$ being a function. That is,

$$\forall x.\, x = \emptyset \vee \exists! y.\, x \bigcap .$$

We might wonder if $\emptyset \bigcap$ truly does not exist, or if it exists but is just not guaranteed by the axiom of specification. In fact, it truly does not exist.

Theorem 4.4.5. There is no set $\emptyset \bigcap$. That is,

$$(\exists x.\, (y \in x \Leftrightarrow \forall z.(z \in \emptyset \Rightarrow z \in y)))\neg.$$

Proof. Let us consider what the constant $\emptyset \bigcap$ would be. It would be the set of all sets who are elements of every element of the empty set. This is the set of all sets because every set is an element of the (no) elements of the empty set. This is the same reasoning as why $\perp \Rightarrow \varphi$ for any proposition φ.

But, this set of all sets cannot exist. Let us suppose that $\emptyset \bigcap$ is a propositional set (namely, let us assume it exists). Let Φ be the (propositional) set $\{a \in \emptyset \bigcap \mid a \notin a\}$. This is guaranteed to exist by the axiom of specification. Intuitively, this is the set of all sets a which are not elements of themselves.

Let us suppose that $\Phi \in \Phi$. By the definition, this means that Φ satisfies the proposition we used for specification, which is to say that $[a\colon \Phi](a \notin a)$ or simply $\Phi \notin \Phi$ holds.

But, let us suppose that $\Phi \notin \Phi$ holds. It still must be that $\Phi \in \emptyset \bigcap$, so we must have that Φ does not satisfy our predicate of specification, which is to say that $[a\colon \Phi](a \notin a)\neg$ or simply $(\Phi \notin \Phi)\neg$.

In both of these cases, we assumed something and concluded the opposite. This is reveals that we have reached a contradiction, and thus $\emptyset \bigcap$ does not exist.

This is the famous Russel's paradox, which forbids the existence of a set of all sets. This is also essentially the same argument as the barber's paradox. Consider a village with a barber who shaves everyone who does not shave themself. Does this barber shave himself or not? In mathematics, we would like to forbid questions which cannot have answers, such as this one. Thus, we declare that this village, and the set of all sets, are invalid (or just not sets). Some people interpret this as that the set of all sets is "too big" to be a set. $\square$

Example 4.4.6. Let us look at some of the properties of $\bigcap$ now that we have considered some of its abstract behavior. For variables x and y, these propositions are true.

(1) $x \cap x = x$.
(2) $x \cap y = y \cap x$.
(3) $x \cap y \subseteq x$.
(4) $x \subseteq y \Rightarrow (x \cap y = x)$.
(5) $\forall y.\, (y \in x \Rightarrow x \bigcap \subseteq y)$.

Example 4.4.7. We can also perform some calculations using $\bigcap$.

(1) $\{\emptyset, \{\emptyset\}\} \bigcap = \emptyset$ because $\emptyset$ is in the set on the left and it has no elements.
(2) $\{\{\emptyset\}, \{\emptyset, \{\emptyset\}\}\} \bigcap = \{\emptyset\}$ since $\emptyset$ is the only element of both sets in the set on the left.

Although we have managed to describe many properties of and about subsets, there is also a simple axiom related to subsets.

Axiom 4.5. The *axiom of power set* states that

$$\frac{}{\forall a. \exists b. \forall c.\, (c \subseteq a \Rightarrow c \in b)}\,\text{pwr.}.$$

Intuitively, this states that for every set a, there is a set b which contains every subset c of a. Like before, we can use the axiom of specification to create a set whose elements are exactly the subsets of a.

Definition 4.4.8. For a variable x, define the function $x\mathcal{P}$, read as "x's power set" or just "xP," as

$$x\mathcal{P}\colon \forall y.\, (y \in z \Leftrightarrow y \subseteq x)$$

where the variables y and z are chosen from context. The input is x and the output is z.

The axioms of power set and specification mean that this is actually a function. Intuitively, the set $a\mathcal{P}$ is the set whose elements are the subsets of a.

Every element of the empty set (of which there are none) is an element of every other set. Thus, $\forall a.\, \emptyset \subseteq a$. As a consequence, $\forall a.\, \emptyset \in a\mathcal{P}$. Thus, we know that the power set of any set has an element, and thus is not the empty-set. Further, $\forall a, b.\, (a \in b \Rightarrow \{a\} \in b\mathcal{P})$.

Combining the power set with other operations on sets can make complicated statements. For example, $a\mathcal{P} \bigcup = a$ and $a\mathcal{P} \bigcap = \emptyset$, which are relatively simple.

4.5. **Infinity and Induction.** We have now defined enough material to introduce our first set. Recall that we have not proved the empty set theorem, so we do not technically know that there even is a set. We will resolve this with the next axiom.

First, we must define a new function.

Definition 4.5.1. For a variable x, define a function xS, read as "x's successor" or just "xS," as

$$xS\colon x \cup \{x\}.$$

We will use this to turn the natural numbers into sets, and then create a set of these natural numbers.

We will encode the number 0 as $\emptyset$. That is, although 0 is a number, which is not traditionally a set, we will pick that the set $\emptyset$ refers to the number 0. This is perhaps the most obvious choice for how to encode 0 as a set; 0 is nothingness and $\emptyset$ is the nothingness set. Then consider the set $0S$ which is $0 \cup \{0\}$. This is the set which we want to call 1, the successor of 0. This yields the equation $1 = \{0\}$, which has one element. Then, we can consider the set $0SS$ also known as $1S$ also known as 2. This is the set, $1 \cup \{1\}$ which is the set $\{0\} \cup \{1\}$ which is the set $\{0, 1\}$. Take a stab at what $0SSS$ aka 3 is. Spoiler: it's $\{0, 1, 2\}$.

We would like to collect all of these elements into a set.

Axiom 4.6. The axiom of *infinity* states that

$$\frac{}{\exists a.\, (\exists b.\, (b \in a \wedge \forall c.\, c \notin b)) \wedge (\forall b.\, (b \in a \Rightarrow bS \in a))}\ \text{infty.}$$

This axiom says that there exists a set a such that there is an element b in a that is the empty set and for every element b of a, the successor of b is also an element of a. This is called the axiom of infinity because it asserts the existence of an infinite set, although it is a particularly special infinite set.

This provides us with many useful properties. For example, let us now prove the empty set theorem.

Proof. Consider the set a guaranteed by the axiom of infinity and form the set $\{b \in a \mid \bot\}$. This set has no elements, and so this set is the empty set. Therefore, the empty set exists. $\square$

Now that we have proven this result, all of the other sets we built using the empty set are also valid sets.

The set a guaranteed by the axiom of infinity contains all of the natural numbers, but it might contain some additional values. We would like to remove those other elements from the set a using the axiom of specification. This is more difficult than the previous uses of the axiom of specification, but it is possible nonetheless. First, let us define some shorthand.

Definition 4.5.2. For a variable x, let xI, read as "x is *inductive*" or just "xI," be the proposition

$$\emptyset \in x \wedge \forall y. (y \in x \Rightarrow yS \in x)$$

where y is some other variable.

With this notation, the axiom of infinity is that

$$\exists a. aI.$$

This proposition aI is intended to mean that the set a contains each of the natural numbers.

Intuitively, the natural numbers are exactly those elements which are contained in every set containing the natural numbers, that is in every inductive set. Thus, we create the following definition.

Definition 4.5.3. Define the proposition $\mathbb{N}$, read as "the set of natural numbers," "the natural numbers," or just "N," by

$$\mathbb{N}\colon n \in x \Leftrightarrow \forall y. (yI \Rightarrow n \in y)$$

where the variables x and y are chosen based on context.

We would like this to be a constant in the free variable x. That is, we would like this to serve as a definition of the set of natural numbers.

Theorem 4.5.4. $\mathbb{N}$ is a constant.

Proof. That is, we want to find a set satisfying the definition of the natural numbers as described above, or equivalently we would like to find a set b whose elements are exactly those elements n of every set c such that cI (such that c contains all the natural numbers). To see that this set exists, we can take the set a guaranteed by the axiom of infinity and create the set $\{n \in a \mid \forall c. (cI \Rightarrow n \in c)\}$. The elements of this set are the elements n of a such that $\forall c. (cI \Rightarrow n \in c)$. But, since a satisfies aI, any set n such that $\forall c. (cI \Rightarrow n \in c)$ is also an element of a. Thus, these elements are exactly those sets n such that $\forall c. (cI \Rightarrow n \in c)$. Any two sets with this property would necessarily have the same elements, and thus this set is unique. Thus, $\mathbb{N}$ is a propositional set. $\square$

Now that we have the set of natural numbers, we can present the most important proof about this set.

Theorem 4.5.5.

$$n \in \mathbb{N} \Leftrightarrow n = 0 \vee \exists m.\, m \in \mathbb{N} \wedge n = mS.$$

That is, n is a natural number if and only if it is 0 or is the successor of a natural number.

Proof. We will break this proof into two pieces. First we will suppose that $n = 0 \vee \exists m.\, (m \in \mathbb{N} \wedge n = mS)$ and show that $n \in \mathbb{N}$. Since cI requires that $0 \in c$, 0 is in every set c such that cI, and therefore $0 \in \mathbb{N}$. Thus, if $n = 0$, $n \in \mathbb{N}$. If $n = mS$ for some $m \in \mathbb{N}$, then m must be in every set c such that cI. From the definition of cI, if $m \in c$, then $mS \in c$ and thus $n \in c$. Thus, n is in every set c such that cI. Thus, $n \in \mathbb{N}$. This completes the first part of the proof: if $n = 0 \vee \exists m.\, m \in \mathbb{N} \wedge n = mS$, then $n \in \mathbb{N}$.

Next, we suppose that $n \in \mathbb{N}$ and show that $n = 0 \vee \exists m.\, m \in \mathbb{N} \wedge n = mS$. This direction is more difficult. Suppose that $n \in \mathbb{N}$ but not $n = 0 \vee \exists m.\, m \in \mathbb{N} \wedge n = mS$. In particular, we suppose that n is not 0 and that n is not the successor of any $m \in \mathbb{N}$. Choose a particular set c such that cI and that there is not any set $m \in c$ such that $n = mS$. This is guaranteed to exist because we know not all sets c such that cI contain m (or else m would be in $\mathbb{N}$). Then, form the set $\{m \in c \mid m \neq n\}$. That is, form a set that is c with n removed. It is still that $\{m \in c \mid m \neq n\}I$ because the element we removed is neither 0 nor mS for some $m \in c$. Therefore, we have found a set c satisfying cI which does not contain n. Thus not all such sets contain n, and thus $n \notin \mathbb{N}$. This contradicts our assumption that $n \in \mathbb{N}$. Therefore, our later supposition that not $n = 0 \vee \exists m.\, m \in \mathbb{N} \wedge n = mS$ must have been a contradiction. Consequently, it must be the case that for every $n \in \mathbb{N}$, $n = 0 \vee \exists m.\, m \in n \wedge n = mS$. This completes the proof. $\square$

This confirms our earlier intuition of the natural numbers as an inductive construction. A natural number can be constructed by two recipes: 0 and S. A consequence is the following theorem.

Theorem 4.5.6. The *induction principle* of the natural numbers states that for any proposition φ in a free variable x and an unused variable n,

$$\frac{[x\colon 0]\varphi \quad \forall n.\,(n \in \mathbb{N} \wedge \varphi[x\colon n]\varphi \Rightarrow [x\colon nS]\varphi)}{\forall n.\,(n \in \mathbb{N} \Rightarrow [x\colon n]\varphi)}\,\mathbb{N}\text{ind}.$$

That is, in order to prove a proposition φ for all natural numbers, it suffices to prove φ holds for 0 and that if some natural number n satisfies φ, then nS does too.

Let us restate this theorem intuitively. Suppose that φ is true of 0 and that if φ is true of n, then it is true of nS. Then, since φ is true of 0, it is true of 1. Then, since φ is true of 1, it is true of 2. Then, since φ is true of 2, it is true of 3. Since this argument could be repeated without stopping, φ must be true of all natural numbers. However, this is not a proof; asserting that we can repeat this argument forever to prove all natural numbers is just asserting that induction works on the natural numbers. Unfortunately, we cannot assert our claim in order to prove it. Instead, we present the following proof.

Proof. Suppose φ is a proposition such that

$$[x\colon 0]\varphi \text{ and } \forall n.\,(n \in \mathbb{N} \wedge [x\colon n]\varphi \Rightarrow [x\colon nS]\varphi).$$

Consider the set $\{x \in \mathbb{N} \mid \varphi\}$. We will show that $\{x \in \mathbb{N} \mid \varphi\}$ is an inductive set. Since $0 \in \mathbb{N}$ and $[x\colon 0]\varphi$, we know that $0 \in \{x \in \mathbb{N} \mid \varphi\}$. Further, suppose that $n \in \{x \in \mathbb{N} \mid \varphi\}$ which is to say that $n \in \mathbb{N}$ and $[x\colon n]\varphi$. By our assumption, $[x\colon nS]\varphi$, and thus $nS \in \{x \in \mathbb{N} \mid \varphi\}$. Consequently, we have shown that $\{x \in \mathbb{N} \mid \varphi\}$ is an inductive set. Therefore, this set contains every natural number, so $\mathbb{N} \subseteq \{x \in \mathbb{N} \mid \varphi\}$. By the definition of set builder notation, it is also that $\{x \in \mathbb{N} \mid \varphi\} \subseteq \mathbb{N}$ and thus that these sets are equal. In particular, every natural number satisfies φ, meaning that $\forall n.\,[x\colon n]\varphi$. This completes the proof. $\square$

In essence, the natural number induction principle is a form of infinite argumentation. We are able to make such an infinite argument because the axiom of infinity gives us access to an infinite set. From this single induction principle, we are able to prove induction principles for the elements of many other sets. However, we cannot use the natural number induction principle to create an induction principle that works for sets themselves because, as we have shown, there is no set of all sets. However, mathematicians, in particular constructive mathematicians, have such success working with induction principles that we assert an induction principle for sets.

Axiom 4.7. The axiom of *set induction* states that for every proposition φ, z a free variable of φ, and variables x and y unused in φ,

$$\dfrac{}{\forall x.\, (\forall y.\, (y \in x \Rightarrow [z\colon y]\varphi) \Rightarrow [z\colon x]\varphi) \Rightarrow \forall z.\, \varphi}\in\text{ind.}\,.$$

We can restate this as

$$\dfrac{\forall x.\, (\forall y.\ (y \in x \Rightarrow [z\colon y]\varphi) \Rightarrow [z\colon x]\varphi)}{\forall z.\, \varphi}\in\text{ind.}\,.$$

That is, if a proposition being true of all elements y of a set x implies that the proposition is true of x, then the proposition is true of all sets.

Traditionally, this axiom is replaced with the axiom of *foundation* or the axiom of *regularity*, which are both equivalent to set induction, but given that this thesis is about induction, I have chosen this formulation.

Let us compare this axiom to the natural number induction principle. The first difference is that the natural number induction principle has two assumptions while the set induction principle has only one. This is because the natural number induction principle treats the smallest natural number, 0, separately. The set induction principle does not need to do this. This is because, supposing the assumption of the set induction principle of some proposition φ, $\forall y.(y \in \emptyset \Rightarrow [z\colon y]\varphi)$ since φ holds of each of the (no) elements of the empty set (that is, $F \Rightarrow \varphi$ is true), so therefore $[z\colon \emptyset]\varphi$ which is that φ holds of the empty set, the smallest set.

The next difference is that the "ingredients" of a set x are its elements y, rather than the ingredients of the natural number n being the m such that $n = mS$ or nothing if $n = 0$. Intuitively, this seems right, as a set consists entirely and only of its elements. Thus, we are asserting that all sets are "built" from elements which are built from their elements, and so on, until the empty set is reached. That is, all of our sets are built from the empty set and the methods we have for building sets and thus cannot be "too big" like the set of all sets would be. Of course, they are still allowed to be quite big, like power set of the natural numbers.

We could state the natural number induction principle in the same way by creating an "ingredients" function which sends a natural number to a set of ingredients: the empty set in the case of 0 and the set n in the case of nS. Then, the natural number induction principle as that if φ's truth for the ingredients of n guarantees φ's truth for n, then φ holds for all natural numbers. We prefer the stated formulation of the natural number induction principle because it is easier to write down.

Let us use set induction to prove a simple theorem.

Theorem 4.5.7.

$$\forall a.\, a \notin a.$$

That is, no set is an element of itself.

Proof. Consider some set b such that $b \in b$. Then, there is some set $c \in b$ such that $c \in c$ (in particular, the set c is the set b). Therefore, suppose that no element c in b is an element of itself. Then, it must be that b is not an element of itself either. That is, $\forall b.\, (\forall c.\, (c \in b \Rightarrow c \notin c)) \Rightarrow b \notin b)$. Therefore, by the set induction principle, $\forall a.\, a \notin a$. □

This is, in a sense, the second part of the resolution to Russel's paradox; it says that it is just true that $a \notin a$, so we cannot create the set of all a such that $a \notin a$, as this would be the set of all sets, which would be an element of the set of all sets (since it would be a set), which would be a contradiction, since the set of all sets cannot contain itself.

It is still possible to do most of mathematics without the axiom of set induction. For example, it is still possible to develop most of calculus. However, mathematicians keep it around to prove some of the more "exotic" properties of sets. For our purposes, this will be the least-used axiom, even though this thesis is about induction. For almost all cases, natural number induction is enough.

4.6. **Pairs and Small Propositions.** Now that we have briefly delved into the realm of the infinite, we can return to some simple constructions regarding sets.

Earlier, we were able to construct a pair $\{a, b\}$. This has the property that $\{a, b\} = \{b, a\}$. Thus, this sort of pair is *unordered*. It is also reasonable to desire a set representing the *ordered* pair of a and b. Further, we had unordered triples $\{a, b, c\}$ and we could also make quadruples, and so on. We wish to do the same thing for ordered collections.

Definition 4.6.1. For variables $x_1, \ldots, x_n$, define $(x_1, x_2, \ldots, x_n)$, read as "the sequence $x_1, \ldots, x_n$" or just "$x_1, \ldots, x_n$," by

$$()\colon \emptyset \text{ and } (x_1, \ldots, x_n)\colon \{\{x_1\}, \{x_1, (x_2, \ldots, x_n)\}\}.$$

A set of the form $(x_1, \ldots, x_n)$ is called a sequence of length n or an n-tuple. The value x_i is called the i-th component of $(x_1, \ldots, x_n)$.

This definition is somewhat hard to unpack, so let's go through it. First, $() = \emptyset$. Then, $(x) = \{\{x\}, \{x, ()\}\}$. In particular, $(\emptyset) = \{\{\emptyset\}\}$. Then, $(x, y) = \{\{x\}, \{x, (y)\}\}$. In particular, $(\emptyset, \emptyset) = \{\{\emptyset\}, \{\emptyset, \{\{\emptyset\}\}\}\}$. For a last example, $(x, y, z) = \{\{x\}, \{x, (y, z)\}\}$.

If we have some set x which we know to be a sequence, it is important to be able to access the sets that appear in the sequence. We do this with a variety of functions.

Definition 4.6.2. For a variable x, define the proposition xH, read as "x's head," "x head," or just "xH", by

$$xH\colon x\bigcap\bigcup.$$

and define the proposition xT, read as "x's tail," "x tail," or just "xT," by

$$xT\colon \left\{y\in x\bigcup \;\middle|\; x\bigcup\neq x\bigcap\Rightarrow y\notin x\bigcap\right\}\bigcup$$

where y is some other variable.

Suppose that $x=\{\{h\},\{h,t\}\}$. This looks like the definition of $(x_1,\ldots,x_n)$ with x_1 replaced by h and $(x_2,\ldots,x_n)$ replaced by t. These names are intentional: the variable h is supposed to represent the "head" of the sequence, that is to say the first component, and the variable t is supposed to represent the "tail" of the sequence, that is to say the rest of the sequence.

Let us consider the set xH. First, $x\bigcap$ is the intersection of all the sets in h, which is $\{h\}$ since h is the only element shared between $\{h\}$ and $\{h,t\}$. Then, $\{h\}\bigcup$ is just all the elements of h, that is, h. So $xH=x\bigcap\bigcup=h$. This is good news. It means that the head of a sequence is actually the first component. If instead x were the empty sequence, then $\emptyset H$ would not exist (since $\emptyset\bigcap$ does not exist). Thus, the empty sequence has no first component, as is expected.

Let us then consider the set xT This set is substantially more complicated. xT is a subset of $x\bigcup=\{h,t\}$. In order to find the tail, t, we would want to remove the element h. However, if $h=t$, then removing the element h would also remove the element t. Thus, we wish to remove h whenever it is not t. Recall that $x\bigcap=\{h\}$. This is equal to $\{h,t\}$ whenever $h=t$ and is not equal to $\{h,t\}$ whenever $h\neq t$. Thus, if $x\bigcup=x\bigcap$, we need to remove the element h, which is the unique element of $x\bigcap$. Thus, we take the set of all $y\in x\bigcup$ such that if $x\bigcup\neq x\bigcap$, then $y\notin x\bigcap$. This is the definition of xT. In particular, we have that $xT=t$, which is the desired behavior. In the case of the empty set, $\emptyset T=\emptyset$. Perhaps we should expect that the empty sequence has no tail, but the empty sequence having an empty tail is at least acceptable.

The sets $\{\{h_1\},\{h_1,t_1\}\}$ and $\{\{h_2\},\{h_2,t_2\}\}$ are equal if and only if $h_1=h_2$ and $t_1=t_2$. As an extension, two sequences $x=(x_1,\ldots,x_n)$ and $y=(y_1,\ldots,y_m)$ are equal if and only if they are the same length (so $m = n$), $xH = yH$, and $xT = yT$. In particular, this means that they are equal if and only if $x_i = y_i$ for each i.

Further, a set x is a sequence whenever x is empty or xT is a sequence and $x = \{\{xH\}, \{xH, xT\}\}$. Although it is nice to have a method to get the head and tail of a sequence, it is just as important to be able to get at arbitrary values in a sequence. Thus, we create the following definition.

Definition 4.6.3. For a variable x, define the *projection functions* $x\pi_n$,[7] read as "the n-th projection of x," by

$$x\pi_1 \colon xH, \ x\pi_2 \colon xTH, \ x\pi_3 \colon xTTH, \text{ etc..}$$

That is, π_2 is the head of the tail, and π_3 is the head of the tail of the tail, and so on. In particular, $(x_1, \dots, x_n)\pi_i = x_i$, and so we can access the elements of our sequence using the π_i functions.

Although the language of sequences may seem inconsequential, it is actually extremely important in set theory. $x = \emptyset$ is a proposition. For any set y, the axiom of specification allows us to create the set $\{x \in y \mid x = \emptyset\}$, that is the set of all element of y such that $x = \emptyset$. However, we also know that there is a set of all empty sets: $\{\emptyset\}$. On the other hand, we know that $x \notin x$ is a proposition, and the axiom of specification allows us to form the set $\{x \in y \mid x \notin x\}$ (which is just y by the axiom of regularity), but we *cannot* form the set of all sets which do not contain themselves.

Thus, there is a difference between the propositions $x = \emptyset$ and $x \notin x$: we can create a set of all solutions to the former, but the solutions to the latter are so "large" as to not constitute a set. With this in mind, we create the following extension to set builder notation.

Definition 4.6.4. For a proposition φ and a variable x, define the proposition $\{x \mid \varphi\}$, read as "the set of all x such that φ," by

$$\{x \mid \varphi\} \colon \forall x.(x \in y \Leftrightarrow \varphi)$$

where the variable y is chosen from context. Although this is not a function, one should interpret the free variables of φ excluding x as inputs and y as the output. Intuitively, if such a y exists, it should be interpreted as the set of all sets x satisfying φ.

This does not exist for all propositions φ. In particular, we know that $\{x \mid x = \emptyset\}$ exists but that $\{x \mid x \notin x\}$ does not exist. However, we know that $\{x \in y \mid \varphi\} = \{x \mid x \in y \wedge \varphi\}$ always exists by the axiom of specification.

[7] π is the lowercase Greek letter π. This use of π has no relation to the circle constant $3.14\ldots$, instead just referring to projection.

Let us define one further extension of the set builder notation.

Definition 4.6.5. For propositions φ and ψ and variables $x_1, \ldots, x_n$ and y, define the proposition $\{\psi\colon x_1, \ldots, x_n \mapsto y \mid \varphi\}$, read as "the set of all ψ such that φ," by

$$\{\psi\colon x_1, \ldots, x_n \mapsto y \mid \varphi\}\colon \{y \mid \exists x_1, \ldots, x_n.\, \psi \wedge \varphi\}.$$

Intuitively, ψ should be thought of as a function mapping sets $x_1, \ldots, x_n$ to sets y, and the set $\{\psi\colon x_1, \ldots, x_n \mapsto y \mid \varphi\}$ should be interested as the set of all outputs y corresponding to some inputs $x_1, \ldots, x_n$ which satisfy φ. In most cases, the variables $x_1, \ldots, x_n \mapsto y$ are known from context and so are omitted.

This definition may look obtuse; however, it is fairly natural to use in practice. For example, $\{nS \mid n \in \mathbb{N}\}$. Here, we could have specified that n is the output and the free variable of nS is the output, but since nS is a function, this is clear from context. This is the set of all successors of natural numbers. Thus, $\{nS \mid n \in \mathbb{N}\} = \{n \in \mathbb{N} \mid n \neq 0\}$. As another example $\{(n, nS) \mid n \in \mathbb{N}\}$ is the set $\{(0,1), (1,2), (2,3), \ldots\}$.

With this, we can create an extremely important definition.

Definition 4.6.6. Suppose that φ is a proposition in the free variables $x_1, \ldots, x_n$. φ is said to be *small* if $\{(x_1, \ldots, x_n) \mid \varphi\}$ exists and is called *large* otherwise. Alternatively, φ is small if

$$\exists y.\, \forall z.\, (z \in y \Leftrightarrow \exists x_1, \ldots, x_n.\, z = (x_1, \ldots, x_n) \wedge \varphi)$$

where y and z are unused variables. That is, a proposition is small if the truth of the proposition is equivalent to membership in some set y.

Thus, $x = \emptyset$ is small and $x \notin x$ is large. Intuitively, a set is small whenever its solutions are few enough to be placed in a set. That is, whether or not $\{(x_1, \ldots, x_n) \mid \varphi\}$ is a set is a matter of size, rather than something else. This leads us to the following conclusion. Suppose that Φ is a function $\Phi\colon x \mapsto y$. Then, for some set z, $\{\Phi\colon x \mapsto y \mid x \in z\} = \{\Phi \mid x \in z\}$ must also be a set. This is because, since z is a set and Φ maps each x to exactly one y, the set of all y's must be no larger than the set z. We codify this in an axiom.

Axiom 4.8. The axiom of *replacement* states that for a proposition φ in the variables $w_1, \dots, w_n, x, y, z$, and v a variable unused in φ,

$$\frac{}{\forall w_1, \dots, w_n. \forall z. (\forall x. (x \in z \Rightarrow \exists! y. \varphi) \Rightarrow \exists v. \forall y. (y \in v \Leftrightarrow \exists x. x \in z \wedge \varphi))} \text{ repl.}.$$

Intuitively, this states that if φ is a proposition such that all values x in a set z are each assigned to exactly one value of y, then there is some set v containing all such values y. As described, this is some sort of smallness preservation axiom.

This is the longest and most complicated axiom we have presented so far. However, its intuition is quite natural. Like the axiom of set induction, this axiom is not needed for more of mathematics. However, we keep it around so that using functions is easier.

4.7. **Multi-valued Functions.** For completeness, we will also describe multi-valued functions. Here, we present the full definition of a propositional function.

Definition 4.7.1. A proposition Φ in the free variables $x_1, \dots, x_n$ and $y_1, \dots, y_m$ is a (propositional) function if

$$\forall x_1, \dots, x_n. \exists! y_1, \dots, y_m. \Phi.$$

The variables $x_1, \dots, x_n$ are called the inputs and the variables $y_1, \dots, y_n$ are called the outputs. If Φ has one output, it is called *single-valued*. If Φ has no inputs, it is called a *constant*. If Φ is constant and single-valued, it is called a *propositional set*. If Φ is a function in the inputs $x_1, \dots, x_n$ and the outputs $y_1, \dots, y_n$, we write

$$\Phi \colon x_1, \dots, x_n \mapsto y_1, \dots, y_m,$$

read as "Φ maps $x_1, \dots, x_n$ to $y_1, \dots, y_m$." Importantly, this use of : does *not* mean "is defined to be."

This is a slight expansion of functions to allow multiple outputs as well as multiple inputs. Like before, a function $\Phi \colon x_1, \dots, x_n \mapsto y_1, \dots, y_m$ should be thought of as a unique *sequence* of sets $y_1, \dots, y_m$ for each choice of sets $x_1, \dots, x_n$.

Since we now have multiple outputs, we need to expand our propositional replacement syntax to support these new functions.

Definition 4.7.2. Suppose φ and ψ are propositions and $x_1, \dots, x_n$ are variables. Then, the propositional variable assignment $[x_1, \dots, x_n \colon \psi]\varphi$ is shorthand for

$$[x_1, \dots, x_n \colon \psi]\varphi \colon \forall x_1, \dots, x_n. (\psi \Rightarrow \varphi).$$

This is only defined when the right side is a proposition. It is sufficient to assume that no free variable of φ is bound in ψ and vice versa and that the x_i are not bound in ψ or φ. This definition technically makes sense for all propositions, but it is most meaningful when ψ is a function.

In particular, if Ψ is a function $\Psi : x_1, \ldots, x_n \mapsto y_1, \ldots, y_m$ and Φ is a function $\Phi\colon z_1, \ldots, z_k \mapsto w_1, \ldots, w_l$ (presumably such that there is overlap between the $y_1, \ldots, y_m$ and the $z_1, \ldots, z_k$), then $[y_1, \ldots, y_m\colon \Psi]\Phi$ is a function whose inputs are the $x_1, \ldots, x_n$ and those $z_1, \ldots, z_k$ which are not in $y_1, \ldots, y_m$ and whose outputs are the $w_1 \ldots, w_l$.

Lastly, we present a modification of set builder notation to support multivalued functions.

Definition 4.7.3. For proposition φ and ψ and free variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ define the proposition $\{\psi\colon x_1, \ldots, x_n \mapsto y_1, \ldots, y_m \mid \varphi\}$, read as "the set of all ψ such that φ," by

$$\{\psi\colon x_1, \ldots, x_n \mapsto y_1, \ldots, y_m \mid \varphi\}\colon \{(y_1, \ldots, y_m) \mid \exists x_1, \ldots, x_n.\, \psi \wedge \varphi\}.$$

In most cases, the variables $x_1, \ldots, x_n \mapsto y_1, \ldots, y_m$ are clear from context, and so are omitted. This makes most sense when ψ is a function. Intuitively, this is the set of all sequences $(y_1, \ldots, y_m)$ of outputs corresponding to inputs $x_1, \ldots, x_m$ such that φ holds. Technically, if $m = 1$, this conflicts with the old set builder notation. In particular, this notation creates a set of all 1-element tuples (y) rather than a set of plain old sets y. Thus, context will determine whether (y) or y is meant (but it will almost always be y).

4.8. **Internal Functions and Finiteness.** The axiom of replacement hints at something important: a proposition that is almost a function except that it only makes inputs x to outputs y if the inputs are in a set z. Further, since the inputs come from the set z, the outputs are also in some set (this is the axiom of replacement). Consider a proposition Φ such that $\forall a.\, (a \in c \Rightarrow \exists! b.\, \Phi)$. For every value $a \in c$, there is exactly one pair (a, b) such that Φ. Thus, $\{(a, b) \mid \Phi\}$ should have the same size as c and thus should be a set. To demonstrate this, we will need a way to build sequences where each component comes from a fixed set.

Definition 4.8.1. For a sequence of variables $x_1, \ldots, x_n$, define the proposition $\langle x_1, \ldots, x_n\rangle$, read as "the product of $x_1, \ldots, x_n$," by

$$\langle\rangle\colon \{()\}$$

and

$$\begin{aligned}\langle x_1, \ldots, x_n\rangle\colon \{z \in (x_1 \cup \langle x_2, \ldots, x_n\rangle)\mathcal{P}\mathcal{P} \mid \exists h, t.&\\ z = \{\{h\}, \{h, t\}\} \wedge h \in x_1 \wedge t \in \langle x_2, \ldots, x_n\rangle\}.&\end{aligned}$$

These are guaranteed to exist by the axioms of union, power set, and specification. This definition states that $\langle\rangle$ is the set containing the (unique) empty sequence and that $\langle x_1, \ldots, x_n\rangle$ is the set containing sequences whose head is in x_1 and whose tail is in $\langle x_2, \ldots, x_n\rangle$. Thus, $\langle x_1, \ldots, x_n\rangle$ is the set of all sequences of length n whose i-th component is in x_i.

Now, consider a proposition Φ such that $\forall a.\,(a \in c \Rightarrow \exists! b.\,\Phi)$. By the axiom of replacement, $\{\Phi\colon a \mapsto b \mid a \in c\}$ is a set. Call this set d. Thus, $\{(a,b) \mid a \in c \wedge \Phi\} = \{(a,b) \in \langle c, d\rangle \mid \Phi\}$ which then exists by the axiom of specification. If it is also the case that $\Phi \Rightarrow a \in c$, then $\{(a,b) \mid \Phi\}$ is this same set, and so thus Φ is small. This leads to the following definition.

Definition 4.8.2. For variables f, x, and y, define the proposition $f\colon x \to y$, read as "f is a function from x to y," by

$$(f\colon x \to y)\colon f \subseteq \langle x, y\rangle \wedge \forall u.\,(u \in x \Rightarrow \exists! v.\,(u, v) \in f)$$

where u and v are other variables. If $f\colon x \to y$, f is said to be a function with *domain* x and *codomain* y. Like before, the : in $f\colon x \to y$ is not "is defined to be." Also, notice the difference between $\Phi\colon x \mapsto y$, which says that x and y are free variables of Φ and $f\colon x \to y$ which says that x and y are sets which the inputs and outputs come from.

It is easy to extend this to multiple inputs and outputs. For variables f, $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$, define the proposition $f\colon x_1, \ldots, x_n \to y_1, \ldots, y_m$ by

$$\begin{aligned}(f\colon x_1, \ldots, x_n \to y_1, \ldots, y_m)\colon f \subseteq \langle x_1, \ldots, x_n, y_1, \ldots, y_m\rangle \wedge&\\ \forall u_1, \ldots, u_n.\,(u_1 \in x_1 \wedge \cdots u_n \in x_n \Rightarrow&\\ \exists! v_1, \ldots, v_m.\,(u_1, \ldots, u_n, v_1, \ldots, v_m) \in f)&\end{aligned}$$

where $u_1, \ldots, u_n$ and $v_1, \ldots, v_m$ are other variables. If $f\colon x_1, \ldots, x_n \to y_1, \ldots, y_m$ then f is said to be a function with domains $x_1, \ldots, x_n$ and codomains $y_1, \ldots, y_m$.

If $f\colon x_1, \ldots, x_n \to y_1, \ldots, y_m$, then for variables $u_1, \ldots, u_n$, define the proposition $[u_1, \ldots, u_n]f$, read as "the value of $u_1, \ldots, u_n$ under f" or just "$u_1, \ldots, u_n\ f$," by

$$[u_1, \ldots, u_n]f\colon (u_1, \ldots, u_n, v_1, \ldots, v_m) \in f$$

where the variables $v_1, \ldots, v_m$ are chosen from context. This is called *evaluation*. The set, or sequence of sets, corresponds to the unique values $v_1, \ldots, v_m$ of $y_1, \ldots, y_m$ which correspond to the inputs $u_1, \ldots, u_n$ under the function f. Values in $x_1, \ldots, x_n$ are chosen for $u_1, \ldots, u_n$, then

$[u_1, \ldots, u_n]f$ becomes a constant. If f has a single codomain, this is a propositional set.

Since $[u_1, \ldots, u_n]f$ acts like a propositional function, it can be used like one in any context where a propositional function could be used. If f has a single domain, one is free to write vf rather than $[v]f$.

This definition is rather complicated, but its intuition is simple. Thus, let us look at an example.

Example 4.8.3. Consider the set $s = \{(n, nS) \mid n \in \mathbb{N}\}$. Since $nS \in \mathbb{N}$ if $n \in \mathbb{N}$, $s \subseteq \langle\mathbb{N}, \mathbb{N}\rangle$. Further, for each $n \in \mathbb{N}$, there is a unique $m \in \mathbb{N}$ such that $(n, m) \in s$. In particular, this value of m is nS. Thus, s is an internal function. In particular, $s : \mathbb{N} \to \mathbb{N}$. This set s, in effect, encodes the function nS over the natural numbers as a set rather than a proposition. With our evaluation syntax, ns is the unique m such that $(n, m) \in s$ and so is nS. That is, $\forall n \in \mathbb{N}, ns = nS$. We could repeat this procedure for any propositional function.

We could also consider a special set $\mathrm{id}_{\mathbb{N}} = \{(n, n) \mid n \in \mathbb{N}\}$ called the *identity function* on $\mathbb{N}$. As the name implies, $\mathrm{id}_{\mathbb{N}} : \mathbb{N} \to \mathbb{N}$. In this case, for $n \in \mathbb{N}$, $n\,\mathrm{id}_{\mathbb{N}} = n$.

Let us form the set $t = \{(n, nS, nSS) \mid n \in \mathbb{N}\}$. In this case, $t \subseteq \langle\mathbb{N}, \mathbb{N}, \mathbb{N}\rangle$. If we want this to be a function, it could either be that $t\colon \mathbb{N}, \mathbb{N} \to \mathbb{N}$ or $t\colon \mathbb{N} \to \mathbb{N}, \mathbb{N}$. However, since there is no m such that $(0, 0, m) \in t$, we can conclude that t is not a function with two inputs. It is, however a function with two outputs, because for each $n \in \mathbb{N}$, there is exactly one pair m_1, m_2 such that $(n, m_1, m_2) \in t$.

As a non-example, let us consider the set $p = \{(n, nS) \mid n \in \mathbb{N}\} \cup \{(nS, n) \mid n \in \mathbb{N}\}$. Here, it is still that $p \subseteq \langle\mathbb{N}, \mathbb{N}\rangle$, so it is conceivable that $p\colon \mathbb{N} \to \mathbb{N}$. We can realize that $(1, 0)$ and $(1, 2)$ are both elements of p, and thus there is not a unique m such that $(1, m) \in p$, and thus p is not a function.

It is a property of every function $f\colon x \to y$ that $f = \{(v, [v]f) \mid v \in x\}$. The most important feature of functions is that they can be composed. Recall that a function is an assignment of inputs from a set x to outputs in a set y. If there is another function which takes inputs from y to outputs in some set z, we could conceivably use an input from x to produce a value of y using the first function, then use this value and the second function to produce a value of z. This will make every element of x to an element of z, and thus should be a function.

Definition 4.8.4. For functions $f\colon x \to y$ and $g\colon y \to z$ for some variables x, y, and z, define the *composite* of f and g, written as $f\,;g$ and read as "f then g," by

$$f\,;g\colon \{(u, [[u]f]g) \mid u \in x\},$$

where u is another variable. With this definition the composite is a function $f\,;g\colon x \to z$. If $f\colon x \to y$ and $g\colon y \to z$ for some variables x, y, and z, then f and g are called *composable*.

Exercise 4.8.1. Composition of functions has a variety of nice properties. Prove or otherwise convince yourself of the following:

(1) $(f\,;g)\,;h = f\,;(g\,;h)$. This justifies writing $f\,;g\,;h$ to refer to either.
(2) $[u](f;g) = [[u]f]g$.

Now is as good of a time as any to discuss how this notation differs from the standard mathematical convention. First, most authors use () rather than [] for functions, but I find [] to be better, as I do not use it as a grouping symbol, it not the same as the notation for sequences, and it looks like variable assignment. Also, most authors put the inputs after functions, rather than before. The convention where inputs come first is called *backward notation* because it is backwards compared to the standard convention. I have chosen backwards notation so that the composition operator takes an $f\colon x \to y$ and a $g\colon y \to z$ and produces $f\,;g\colon x \to z$. If the standard convention is used, the composition operator instead takes a function $f\colon y \to z$ and a function $g\colon x \to y$ and produces $f \circ g\colon x \to z$. This is so that $(f \circ g)[u] = f[g[u]]$ when the input is written after. I think this is more confusing, because the second function to appear (g) is applied first. With the backwards notation, in $f\,;g$, f is applied, and then g is applied. That is, $[u](f\,;g) = [[u]f]g$. My choice of convention also applied to operators like $\neg$ and $\bigcup$ which generally precede their arguments, rather than come after like we have written. I am strongly of the opinion that this backwards notation is superior. There is less extraneous notation to both read and write, and further, the meaning is clearer in terms of sequentiality.

There is a special function that exists for every set. In fact, it is one we have already mentioned.

Definition 4.8.5. For a variable x, define the *identity function on x*, written as id_x and read as "id x," as

$$\mathrm{id}_x : \{(u, u) \mid u \in x\}.$$

With this notation, $\mathrm{id}_x : x \to x$.

The identity functions satisfy the important property that for a function $f : x \to y$, $\mathrm{id}_x \,;\, f = f = f \,;\, \mathrm{id}_y$.

There are some special kinds of functions that will be useful to us.

Definition 4.8.6. A function $f : x \to y$ is called *injective* when

$$\forall u_1, u_2.\, (u_1 \in x \wedge u_2 \in x \wedge [u_1]f = [u_2]f \Rightarrow u_1 = u_2)$$

where u_1 and u_2 are other variables. That is, a function is injective when, if any two inputs with equal outputs are equal, or in other words that no distinct inputs are sent to the same output. That $f : x \to y$ is an injective function, also called an *injection*, is a proposition in the free variables f, x, and y.

Definition 4.8.7. A function $f : x \to y$ is called *surjective* when

$$\forall v.\, (v \in y \Rightarrow \exists u.\, u \in x \wedge [u]f = v)$$

where u and v are other variables. That is, a function is surjective when every possible output value in the codomain is actually the output corresponding to some input in the domain. That $f : x \to y$ is a surjective function, also called a *surjection*, is a proposition in the free variables f, x, and y.

These two properties can be combined into a single term.

Definition 4.8.8. A function $f : x \to y$ is called *bijective* when it is both injective and surjective. A bijective function may also be called a bijection. Alternatively a bijection $f : x \to y$ is a function such that

$$\forall v.\, (v \in y \Rightarrow \exists ! u.\, (u, v) \in f)$$

where u and v are other variables. That is, a function is bijective whenever each output in the codomain is mapped back to a unique input in the domain. If there exists a bijection $f : x \to y$, then x and y are called bijective. That x and y are bijective is a proposition in the free variables x and y.

There is an alternative characterization of a bijection which is perhaps more important than the one above.

Theorem 4.8.9. A function $f\colon x \to y$ is a bijection if and only if there is a function $g\colon y \to x$ such that $f\,;g = \mathrm{id}_x$ and $g\,;f = \mathrm{id}_y$. Further, this function g is unique.

Proof. First, suppose that $f\colon x \to y$ is a bijection. We wish to show that there is a function $g\colon y \to x$ such that $f\,;g = \mathrm{id}_x$ and $g\,;f = \mathrm{id}_y$. Let $g\colon \{([u]f, u) \mid u \in x\}$. This is a function because each $v \in y$ has a unique $u \in x$ such that $[u]f = v$. Further, $[[u]f]g = u$ for $u \in x$. In fact, this shows that $f\,;g = \mathrm{id}_x$ (because $f\,;g$ is simply pairs (u, u)).

We also wish to show that $g\,;f = \mathrm{id}_y$. To accomplish this, take some $v \in y$. Then, take the unique $u \in x$ such that $[u]f = v$. This is guaranteed to exist since f is a bijection. Then, $[v](g\,;f) = [[v]g]f = [[[u]f]g]f = [u]f = v$. That is, $[v](g\,;f) = v$ for $v \in y$, and thus $g\,;f = \mathrm{id}_y$. This is what we desired to prove.

Next, suppose that $f\colon x \to y$ is a function and there exists a $g\colon y \to x$ such that $f\,;g = \mathrm{id}_x$ and $g\,;f = \mathrm{id}_y$. Consider some $v \in y$. Let $u\colon [v]g$. Then, $u \in x$. Further, $[u]f = [[v]g]f = v$ since $g\,;f = \mathrm{id}_y$. Thus, every $v \in y$ corresponds to some $u \in x$ such that $[u]f = v$. If this u is unique, the f is bijective. Thus, suppose that u_1 and u_2 are elements of x such that $[u_1]f = [u_2]f = v$. Then, $[[u_1]f]g = [[u_2]f]g$ and so $u_1 = u_2$ since $f\,;g = \mathrm{id}_x$. Thus, this value of u is unique, and so f is bijective.

As a final note, this g is unique. In particular, suppose that $g_1\,;f = \mathrm{id}_y$ and $f\,;g_2 = \mathrm{id}_x$. Then $g_1 = g_1\,;id_x = g_2\,;f\,;g_2 = \mathrm{id}_y\,;g_2 = g_2$. □

Definition 4.8.10. If f is a bijection $f\colon x \to y$, then let f^{-1}, read as "f inverse," be the unique function $f^{-1}\colon y \to x$ such that $f\,;f^{-1} = \mathrm{id}_x$ and $f^{-1}\,;f = \mathrm{id}_y$.

If a bijection f is thought of as an algorithm or procedure for constructing elements of y from element of x, then f^{-1} is the procedure which undoes this procedure.

Bijections are particularly important in set theory because they represent that two sets have the same "size." In many areas of mathematics, we can only wish to work with sets with a finite number of elements. We know of some finite sets already. The prototypical example is the natural numbers: the set corresponding to the natural number n has n things in it. In fact, we would like to consider a set finite if and only if it has a natural number of things in it. Thus, we present the following definition.

Definition 4.8.11. A set x is called *finite* if there exists an $n \in \mathbb{N}$ and a bijection $f\colon x \to n$. In particular, we say that is has *cardinality* n.

An obvious theorem is that every subset of a finite set is a finite set. That is, if you have a finite set and remove some elements, then you are still left with a finite set.

Proof. We will use the induction principle of the natural numbers to show that every subset of a finite set with cardinality n is finite.

First, we will show that every subset of a set with cardinality 0 is finite. Thus, suppose a is a set with cardinality 0. That is, a is bijective with the empty set. Since every element of a corresponds to an object of the empty set, and there are no elements of the empty set, a must have no elements and be empty too. Thus, if b is a subset of a, then b must be a and so b is bijective to 0 so is finite. That is, every subset of a set with cardinality 0 is finite (and has cardinality 0).

Next, suppose that every subset of a set with cardinality n is finite for $n \in \mathbb{N}$. We will show that every subset of a set with cardinality nS is finite. Thus, suppose a is a set with cardinality nS and suppose b is a subset of a. We will show that b is finite. Let $f : a \to nS$ be a bijection, as promised by the fact that a has cardinality n. If $b = a$, then f is a bijection from $f\colon b \to n$ and so b is finite.

If $b \neq a$, then there must be some element of a missing from b. That is, there must be an element $c \in a$ such that $c \notin b$. Let d be the unique element of a such that $[d]f = n$ or that $(d, n) \in f$. This is guaranteed to exist since f is a bijection. Now, we will make a new bijection $g : a \to nS$ using these elements. Let g denote the set $\{(e, [e]f) \mid e \in a \wedge e \notin \{c, d\}\} \cup \{(c, n)(d, [c]f)\}$. That is, g is the set of all input-output pairs in f such that the input is not either of out elements c or d along with the ordered pairs (c, n) and $(d, [c]f)$, which are the inputs c and d with their outputs exchanged. It is still the case that each input $e \in a$ appears in exactly one ordered pair, and thus g is a function. Further, every output in nS is corresponds to exactly one input, either its original input if it is not n or $[c]f$, and c or d respectively if it is n or $[c]f$. Thus, g is a bijection $g\colon a \to nS$. The important property of g is that $[c]f = n$. That is, the element of a missing from b is the "last" element of a.

We will make one final bijection $h = \{(e, [e]g) \mid e \in a \wedge e \neq c\}$. This is the set of input-output pairs in h with the pair (c, n) removed. This is a bijection $h\colon \{e \in a \mid e \neq c\} \to n$. This is because each element e of a other than c gets mapped into n by g because $[c]f = n$ is the only element of nS that is not in n (since $nS = n \cup \{n\}$). Since n is the only output whose pair was removed, every element of n still corresponds to a unique input in a which is not c. Thus, $h\colon \{e \in a \mid e \neq c\} \to n$ is a bijection. That is, $\{e \in a \mid e \neq c\}$ has cardinality n. That is, if we remove one element of a finite set, its cardinality goes down by 1.

Importantly, since $c \notin b$, $b \subseteq \{e \in a \mid e \neq c\}$, thus b is a subset of a set of cardinality n and thus b is finite by our inductive hypothesis. Thus, we have shown that if every subset of a set of cardinality n is finite, then every subset of a set of cardinality nS is finite. Using the induction principle of the natural numbers, this proves that a subset of any set with any natural number cardinality is finite. This is exactly that every subset of a finite set is finite. □

Not all sets are finite. This is guaranteed by the axiom of infinity. In particular, $\mathbb{N}$ cannot be finite because for each $n \in \mathbb{N}$, it contains all the elements of n (which are the smaller natural numbers) as well as n itself. Without the axiom of infinity, we do not know that there is an infinite set, but we also do not know that all sets are finite.

4.9. **Choice and the Law of the Excluded Middle.** We will now present the final axiom of set theory. This final axiom is what distinguishes ZF from ZFC. This is also the most controversial axiom of set theory.

Axiom 4.9. The axiom of *choice* states that

$$\frac{}{\forall a.\Big(\emptyset \notin a \Rightarrow \exists f.\Big(f\colon a \to a\bigcup\Big) \wedge \forall b.(b \in a \Rightarrow [b]f \in b)\Big)}\,\text{choice}\;.$$

This axiom states that for every set a, if a does not contain the empty set, then there exists a *choice function* f which takes, as inputs, the elements of a, and yields, as outputs, the elements of the elements of a with the property that each set $b \in a$ is mapped to an element of itself. That is, such that for $b \in a$, $[b]f \in b$. In effect, this function f "chooses" from each b an element of b. Since each element of a is not empty, this axiom asserts that we can choose an element of each of them.

This axiom is controversial because it does not explain *how* one is supposed to pick an element of each set $b \in a$. If the truth of a proposition is interpreted to mean "we can prove it,", then the axiom of choice would be interpreted as saying that we can *construct* each choice function (as that is what a proof would be). The axiom of choice is an axiom, and it is not provable from the other axioms of set theory. Further, there are versions of set theory in which the axiom of choice is false.

Most modern mathematicians, however, embrace the axiom of choice. The intuition is obvious, and there are some particularly nice results that do not hold in the absence of choice. Additionally, choice can simplify some otherwise complicated proofs. We will not have much use for the axiom of choice, but we present it here for completeness's sake.

As mentioned in the section on formal logic, the axiom of choice can be used to prove the law of the excluded middle, that is that every proposition is either true or false. One does not have to accept the law of the excluded middle, and in fact many don't, but all who do not accept the law of the excluded middle also cannot accept the axiom of choice.

Theorem 4.9.1. The law of the excluded middle states that for a proposition φ,

$$\dfrac{}{\neg\varphi \vee \varphi}\,\text{l.e.m.}\ .$$

This is also called Diaconescu's theorem.

Proof. Consider a proposition φ and a variable x unused in φ. Using the axiom of pairing and our construction of the natural numbers, we can create the set $B\colon \{0,1\}$. We can then create the sets

$$u\colon \{x \in B \mid x = 0 \vee \varphi\}$$

and

$$v\colon \{x \in B \mid x = 1 \vee \varphi\}.$$

Using the law of the excluded middle, u is the set $\{0\}$ if φ is false and is $\{0,1\}$ if φ is true, and v is the set $\{1\}$ if φ is false and $\{0,1\}$ if φ is true. But, we cannot assert this, since we are trying to prove the law of the excluded middle. Importantly, if φ is true, then $u = v$.

Using the axiom of a choice, there is a function $f : \{u,v\} \to B$ such that $[u]f \in u \wedge [v]f \in v$. From the definitions of u and v, we know that

$$([u]f = 0 \vee \varphi) \wedge ([v]f = 1 \vee \varphi)$$

and so that either $[u]f = 0$ and $[v]f = 1$ or φ. In particular, $[u]f \neq [v]f \vee \varphi$. Since $\varphi \Rightarrow u = v$, it is also that $\varphi \Rightarrow [u]f = [v]f$. Therefore, if $[u]f \neq [v]f$, then $\varphi\neg$.[8]

So, we have that $[u]f \neq [v]f \vee \varphi$, and if $[u]f \neq [v]f$, then $\varphi\neg$, so we must have that $\varphi\neg \vee \varphi$. □

Thus, even if you decided not to believe that there are only two truth values, this can now be seen as a result of the axioms of set theory.

[8]This is because $[u]f \neq [v]f$ is that $[u]f = [v]f \Rightarrow \bot$, so thus $\varphi \Rightarrow \bot$ by transitivity which is $\varphi\neg$.

4.10. **Summary.** Let us consider what we have done in this section. We started with a notion of a set as an unordered collection of other sets, counted without repetition. We said that such a notion cannot be purely formal, as some notion must be informal. Therefore, we had no way to agree upon exactly what a set is. We reconciled these potential differences by agreeing upon nine rules, really axioms, for how sets behave.

The axiom of extensionality described, in effect, a property of sets; that a set is fully determined by its elements. The majority of other axioms allowed us to *construct* other set from sets we already have. Specification let us craft certain subsets, pairing let us create pairs of sets, union let us create even larger sets, power set let us create sets of subsets, replacement let us apply functions to sets to get other sets, and infinity let us actually create a set (namely the set of natural numbers). The axiom of set induction allowed us to prove properties of sets with an induction principle. Rather than asserting the existence of new sets, this, somewhat, set a restriction on how large sets are allowed to be.

Finally, the axiom of choice allowed us to create choice functions. Except it actually didn't: it only asserted that they exist. The other axioms, excluding set induction and extensionality, each allowed us to define new propositional sets, that is show that certain propositions have a unique solution. Choice did not allow us to write down arbitrary choice functions as propositional sets. This is what separates the axiom of choice from the other axioms of set theory.

4.11. **Recommended Reading.** Although set theory is the foundation for most of modern mathematics, it is rarely covered in rigorous detail. I recommend Jech[8] for an in-depth description of most topics in set theory. An easier read which still covers the basic topics of set theory is Epp's[4] discrete mathematics textbook. This book covers much more than set theory, and does not cover set theory in rigorous detail, but it has plenty of exercises and examples which might be useful in comparison to the purely formal description I have presented. There are also plenty of PDFs of set theory books floating around on the internet.

5. Graph Theory and Complexes

The character of this document will now change slightly. Up until this point, we have been speaking *about* mathematics, then *about* logic, then *about* set theory. Now we will be talking about discrete geometry, but we will be talking *in* (the language of) set theory. Consequently, instead of saying "for some variable x" or "for some proposition φ," we will be saying "for some set x" or "for some function f." We already started speaking this way towards the end of the section on set theory. This becomes useful when the propositions described (i.e. f is a bijection from x to y) become cumbersome to notate. The ability to speak in the language of set theory is the entire reason why we spent so long talking about set theory.

The purpose of this section is not to present important facts about shapes, or provide a better understanding of shapes, or anything like that. Instead, this section is supposed to explain how shapes can be represented as mathematical objects, thus allowing shapes (pictures) to be reasoned about with symbols.

In this section, we will tend to pretend that certain things which are not sets are, in fact, sets. For example, we have shown that natural numbers can be regarded as sets, but we have not shown, for example, that people are sets. This is also somewhat of an absurd claim, for I am relatively certain that the identity of a person is not fully specified by their elements. However, since we have natural numbers, and also sequence of natural numbers, we can make English words in set theory. All we have to do is assign a number to each letter (there are pre-existing ways to do this, like ASCII or Unicode). Then, perhaps a person is fully specified by their name, social security number, birth date, etc. Then, a set of people, is a set of this information. Or perhaps a Twitter user is fully specified by their user ID, and so a set of Twitter uses is a set of natural numbers indicating their IDs.

This will lead to a distinction between objects which we mean to be sets and objects which are only sets because everything is a set. Sets that are intended to be sets will generally be notated with capital letters (A, B, X), and sets which are intended primarily as elements will be notated with lowercase letters (a, v, x).

Finally, this section will be the first one to use visualizations. This is because the section is about geometry, and geometry is about pictures. The previous sections might well have benefited from visualizations too, however I did not think that justified the expansion of the length.

5.1. **Relations.** In the section on small propositions, we discussed how propositions may be encoded as subsets of product sets. Later, we acknowledged that a special case of 2-variable propositions, single-value functions, were particularly important. In fact, it is the case that 2-variable propositions themselves are particularly important.

Definition 5.1.1. A *relation* between sets X and Y is a set $R \subseteq \langle X, Y \rangle$. If $x \in X$ and $y \in Y$, we write xRy, read as "x is related to y," for the proposition $(x, y) \in R$ and $x \not R y$, read as "x is not related to y," for the proposition $(x, y) \notin R$.

Intuitively, a relation R between X and Y is a set of pairs of (x, y) which are "related" in some way.

For example, if X is the set of people and Y is the set of books, then R could be the set of all pairs (x, y) such that the person x has read the book y. Then, the proposition xRy would be interpreted as x has read y. As another example, X and Y could both be the set of Twitter uses, and R could be the set of all (x, y) such that x follows y.

X and Y could both be the set of natural numbers excluding 0 (that is the set $\mathbb{N}_+ = \{n \in \mathbb{N} \mid n \neq 0\}$ and the relation R could be the set d of all (a, b) such that a *divides* b in the sense of conventional arithmetic. We can draw this relation.

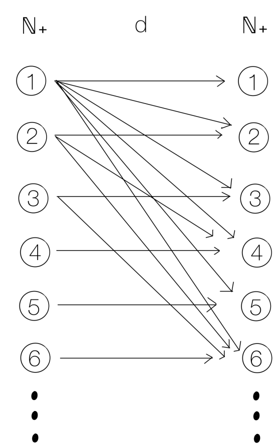

FIGURE 1. The divisibility relation on $\mathbb{N}$.

This is generally how we draw relations: the set X on one side, the set Y on the other, and an arrow from the element of X to the elements of Y when they are related. With this description, functions have a particularly nice form.

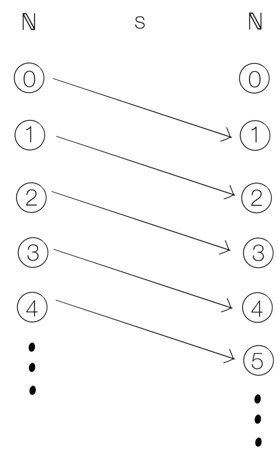

Figure 2. The successor relation on $\mathbb{N}$, which is a function.

In particular, a relation is a function whenever each input has exactly one arrow coming from it.

A bijection is even nicer.

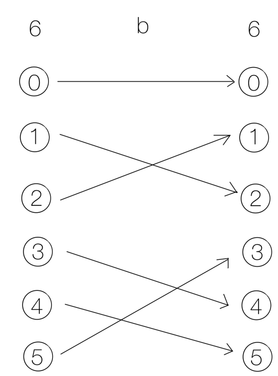

Figure 3. A bijection on the set $6 = \{0, 1, 2, 3, 4, 5\}$.

In particular, a function is a bijection whenever each output has exactly one arrow going to it.

In each of the cases we just listed, the relations we defined have been between a set and itself. These relations have a special name.

Definition 5.1.2. A *directed graph* G is a pair (V, E) where V is a set and E is a relation from V to itself V. The set V is called the set of *vertices* and the set E is called the set of *edges*. If uEv for u and v in V, then u and v are called *adjacent*.

Graphs are generally drawn differently from relations. In particular, since the edge relation is always from the vertices to the vertices, we only draw the vertex set once. Let us redraw the previous bijection as a graph.

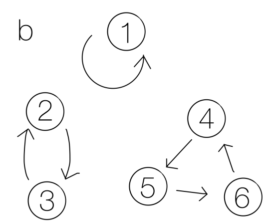

FIGURE 4. The previous bijection drawn as a graph.

Directed graphs are important both in pure mathematics and in applications. For example, the connections between websites is represented as a directed graph (the website x links to the website y) and Google uses this graph to determine how relevant or popular a website is.

5.2. **Graphs and Complexes.** Graphs are often used to represent shapes. In such cases, the direction of edges does not really matter. Thus, we use our unordered pairs $\{u, v\}$ rather than our ordered pairs (u, v).

Definition 5.2.1. An (undirected) graph G is a pair (V, E) where V is a set and E is a set of 2-element subsets of V (that is, $E \subseteq \{\{u, v\} \mid u \in V \wedge v \in V \wedge u \neq v\}$). If $\{u, v\} \in E$, we write uEv and say that u and v are adjacent.

Because $\{u, v\} = \{v, u\}$, uEv if and only if vEu. Thus, graphs are undirected. We draw (undirected) graphs much in the same way that we draw directed graphs. Graphs are good at representing certain types of "shapes." In particular, there is a graph I_6 representing an interval of length 6 and a graph C_5 representing a cycle of length 5.

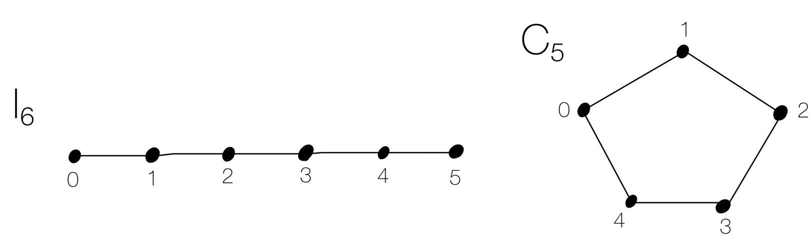

FIGURE 5. The interval graph I_6 and the cycle graph C_5.

Importantly, graphs are not pictures. Instead, they are pairs of sets. However, the pictures that we draw fully determine the sets. For example $I_6 = (\{0, 1, 2, 3, 4, 5, 6\}, \{\{0, 1\}, \{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 5\}, \{5, 6\}\})$.

Exercise 5.2.1. Write down C_5 as a pair of sets.

Not all graphs can be drawn with only straight, non-overlapping edges. This is perfectly okay.

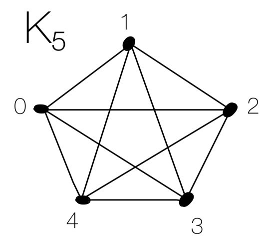

FIGURE 6. The complete graph K_5. This graph cannot be drawn on paper without overlap.

Further, the same graph can drawn in multiple ways, and drawings that look like the same may be different graphs if the labels are different.

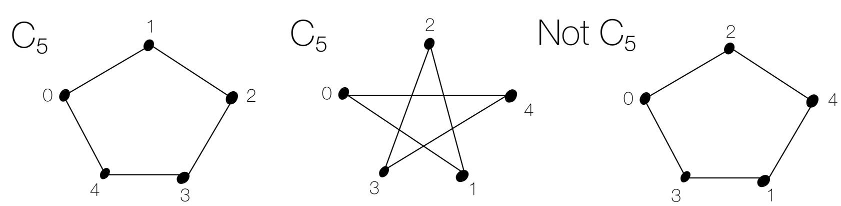

FIGURE 7. Two ways of drawing the graph C_5 and a different graph that looks like C_5 but is not due to the names of the vertices.

It is clear from the drawings that C_5 represents a pentagon. Suppose that I asked you to imagine an ant walking on C_5. What would you imagine? I would guess that you would imagine the ant walking on the border of the pentagon. However, if I asked you to imagine an ant walking on this shape,

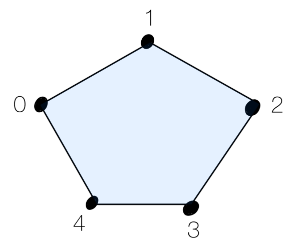

FIGURE 8. A pentagon.

I would guess that the ant would be walking on the pentagon itself. I bring attention to this scenario to point out that a graph can only represent low-dimensional (in particular one dimensional) data. That is, we cannot make a two dimensional shape using graphs. For this, we need a new notion.

First, recognize that many two dimensional surfaces can be represented using only triangles. For example, the surface of a cube can be constructed with two triangles for each side. Further, it is a common technique for video games and animation software to represent models by *triangulating* their surface. In the same way that graphs can represent many one dimensional shapes with only simple ones (edges), we will make two dimensional shapes using only simple ones (triangles).

However, if we want to make a three dimensional shape (one that is “filled-in”), we cannot use triangles. The next jump is to create a tetrahedron. This is, in effect the simplest three dimensional shape.

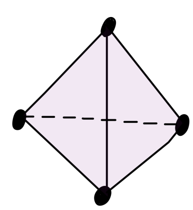

FIGURE 9. A tetrahedron. This should be imagined as “filled in” in three dimensions.

It is not too big of a leap to claim that many three dimensional shapes can be built from tetrahedrons.

Mathematics does not care how we draw things. Mathematics does not care that our world is (basically) three dimensional. So, mathematics will let us build n-dimensional shapes from n-dimensional *simplices*.

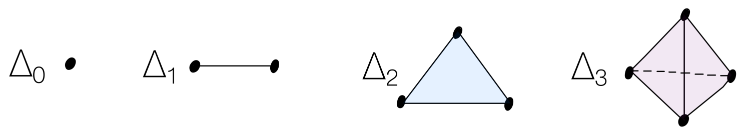

FIGURE 10. The first few simplices.

We cannot draw the higher dimensional simplices very easily, but they still exist mathematically. The important thing to recognize about simplices is that an n dimensional simplex has $n+1$ vertices and is "filled in" in all n dimensions.

This leads us to the following definition.

Definition 5.2.2. A (abstract) *simplicial complex* S is a pair (V, C) where V is a set and C is a set of finite subsets of V such that if $c \in C$ and $d \subseteq c$, then $d \in C$, $\{v\} \in C$ for each $v \in V$, and $\emptyset \in C$. The set V is called the vertices and the set C is called the simplices or *cells*. If $c \in C$ and $d \subseteq c$, then d is called a *face* of c. The cells of cardinality nS are called n dimensional cells or just n cells. Simplicial complexes are often called complexes when the context is clear.

In the case of a graph G, $\{u, v\} \in E$ meant that there was an edge (a one dimensional simplex) between u and v. For a simplicial complex, if $\{v_0, \dots, v_n\} \in C$, we interpret that as meaning that there is an n dimensional simplex with the vertices $v_0, \dots, v_n$. For example, if there is $\{u, v, w\} \in C$, then we understand that there is a triangle with vertices u, v, and w. The condition that if $c \in C$ and $d \subseteq C$ implies that $d \in C$ means that if there is a simplex in the complex S, then each face is also a simplex in the complex. For example, if $\{v_0, v_1, v_2, v_3\}$, then there is a tetrahedron in the complex, which has four triangular faces, and so there are also four triangles $\{v_1, v_2, v_3\}$, $\{v_0, v_2, v_3\}$, $\{v_0, v_1, v_3\}$, and $\{v_0, v_1, v_2\}$ in the complex. The condition that each $\{v\} \in C$ for $v \in V$ is that each vertex of the complex is a zero dimensional simplex (a point) in the graph. The condition that $\emptyset \in C$ is that each complex S contains the unique "-1 dimensional" simplex. This is more of a mathematical convenience rather than anything important. In fact, some mathematicians requires that the empty set is *not* in C, but that results in a different set of complications.

Let us draw some simplicial complexes.

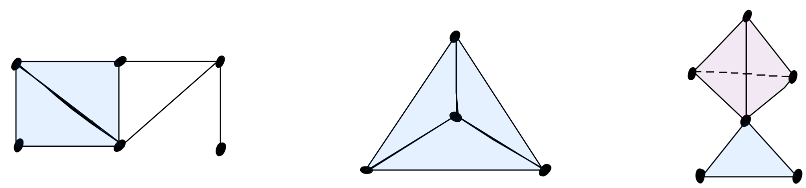

Figure 11. Three examples of simplicial complexes. The blue regions should be regarded as two dimensional and the purple regions as three dimensional.

Although I have not labeled these vertices, one should understand that these points are distinct elements of some set of vertices. As seen, these allow us to create two and three dimensional shapes as well as ones even bigger. Importantly, complexes and graphs do not have to be *connected*. Thus, we can form complexes such as this

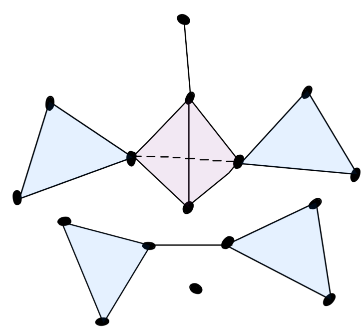

Figure 12. A disconnected simplicial complex.

which consist of separate disconnected pieces. Again, this is a *single* complex.

With the definitions we now have, graphs are technically not simplicial complexes. This is because the edge sets of graphs contain only 2-element subsets of the vertices, and now those subsets of the 2-element subsets. This we present the following refinement to the definition of a graph.

Definition 5.2.3. A graph G is a simplicial complex (V, E) such that every element of E has cardinality 0, 1, or 2.

5.3. **Homomorphisms.** Having a notion of complexes is not particularly useful unless we understand how complexes relate to each other. For example, there are two "copies" of C_5 present in this complex.

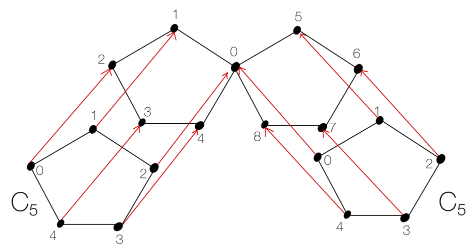

FIGURE 13. A graph representing two C_5s glued together and two ways of mapping a C_5 into the gluing, denoted with red arrows.

As another example, it is possible to "wrap" I_6 around C_5 in the following way.

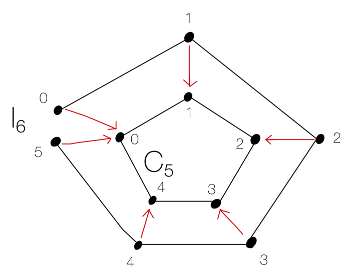

FIGURE 14. A mapping of I_6 into C_5 by "wrapping" I_6 around C_5.

These relationships can be encoded by functions between the vertex sets of complexes satisfying some sort of coherence condition. We need to assert a coherence condition so that we forbid cases like this one.

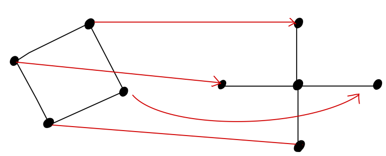

FIGURE 15. A mapping of a square into a cross which does not map edges of the square to edges of the cross.

This leads to the following definition.

Definition 5.3.1. A *homomorphism* f from a complex $S = (V_S, C_S)$ to a complex $T = (V_T, C_T)$ is a function $f\colon V_S \to V_T$ such that if $c \in C_S$, then $\{[v]f \mid v \in c\} \in C_T$. That is, a homomorphism between complexes is a function between the vertex sets which maps cells to cells.

In effect, a homomorphism is a way of mapping or embedding a complex S into a complex T.

Thus, C_5 being mapped into the loops is a homomorphism, and I_6 being wrapped around C_5 is a homomorphism, but the mapping of the square into the cross is not a homomorphism between adjacent vertices in the square are not sent to adjacent vertices in the cross.

In the two examples of homomorphisms we have seen thus far, the two dimensional cells of our complex (the edges) get mapped to edges. However, this is not always necessary. For example, we can "collapse" a vertex of C_4 to get a copy of C_3, or we can collapse a filled in triangle onto an edge. You can think of this as "pinching" the left edge of the triangle down to a point, bringing the rest of the triangle with it.

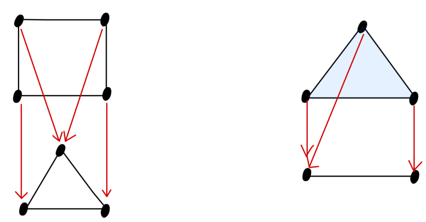

FIGURE 16. Two valid homomorphisms, denoted with red arrows.

We cannot, however, send a filled in triangle to a hollow triangle.

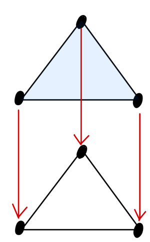

FIGURE 17. A vertex mapping which is not a homomorphism. The triangular cell is not mapped to a triangular cell.

Let us prove an extremely basic result of homomorphisms.

Theorem 5.3.2. If $S = (V, C)$ is a complex, then define $\mathrm{id}_S : S \to S$ to be the homomorphism $\mathrm{id}_S = \mathrm{id}_V$.[9] This is a homomorphism.

Proof. A function is a homomorphism if it maps cells to cells. In particular, we need to show that if $c \in C$, then $\{[v]\,\mathrm{id}_S \mid v \in c\} \in C$. However, since id_S is just id_V, id_S does nothing to the vertices of S. That is, $\{[v]\,\mathrm{id}_S \mid v \in c\} = \{v \mid v \in c\} = c$. Thus, it is the case that $\{[v]\,\mathrm{id}_S \mid v \in c\} \in C$, and so id_S is a homomorphism. $\square$

This is not particularly exciting. It says that the obvious way of embedding a complex into itself is in fact a homomorphism of complexes. This does mean, however, that we have not messed up our definition of a homomorphism too badly.

A more important result is that if S can be embedded into T and T can be embedded into U, then S can also be embedded into U. In particular, we wish to prove the following theorem.

Theorem 5.3.3. If $f: S \to T$ and $g: T \to U$ are homomorphisms for complexes $S = (V_S, C_S)$, $T = (V_T, C_T)$, and $U = (V_U, C_S)$, then $f\,;g: S \to U$ is a homomorphism.

Proof. In order to show that $f\,;g$ is a homomorphism, we need to show that if $c \in C_S$, then $\{[u](f\,;g) \mid u \in c\} \in C_S$. First, let d: $\{[u]f \mid u \in c\}$. Then, $d \in C_T$ since f is a homomorphism. Further, since $d \in C_T$, then $\{[v]g \mid v \in d\} \in C_U$ since g is a homomorphism. The key observation is that $\{[u](f\,;g) \mid u \in c\} = \{[v]g \mid \exists .u \in c \mid v = [u]f\} = \{[v]g \mid v \in d\}$. And thus, $\{[u](f\,;g) \mid u \in c\} \in C_U$. Therefore, $f\,;g$ is a homomorphism. $\square$

There is a certian kind of homomorphism that is of particular importance in studying complexes.

Definition 5.3.4. A homomorphism $f: S \to T$ of complexes is called an *isomorphism* of complexes if there exists a homomorphism $f^{-1}: T \to S$ such that $f\,; f^{-1} = \mathrm{id}_S$ and $f^{-1}\,; f = \mathrm{id}_T$. In particular, f is an isomorphism whenever it is a bijection and its inverse is also a homomorphism. If there is an isomorphism between complexes S and T, S and T are called *isomorphic* and we write $S \simeq T$.

[9]Here id_S mean the identity homomorphism of the graph, and is a homomorphism $S \to S$. This means that it is a function $V \to V$, and thus it makes sense to say that it does nothing to vertices, that it is id_V. However, id_S is not intended to be the identity function on the set $S = (V, C) = \{\{V\}, \{V, (C)\}\}$.

The notion of isomorphisms explains why two different complexes can have identical drawings without being the same. Because of the labeling, they were not equal, but were instead isomorphic.

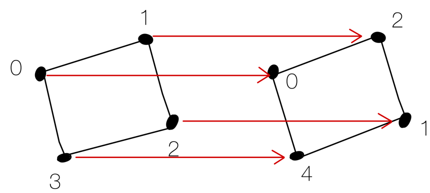

FIGURE 18. An isomorphism between two square shaped graphs.

Not all bijective homomorphisms are isomorphisms. Consider this mapping shown below.

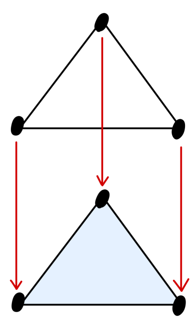

FIGURE 19. A bijective homomorphism which is not an isomorphism.

While this mapping is a bijective homomorphism, the inverse (that is, the diagram with the arrows flipped) is not a homomorphism as we discussed earlier.

There is an analog to injective functions, too.

Definition 5.3.5. A homomorphism $f\colon S \to T$ is a *monomorphism* if the underlying function $f\colon V_S \to V_T$ is injective. That is, if no two distinct vertices in S are mapped to the same vertex in T.

In effect, there is a monomorphism $f\colon S \to T$ whenever there is a complete copy of S inside of T. For example, the mappings of C_5 into the two loops were monomorphisms, but the wrapping of I_6 around C_5 was not a monomorphism.

5.4. **Inductive Definitions.** Now that we have built up a significant amount material relating to simplicial complexes, we will conclude this thesis by creating a few simple constructions of special kinds of simplicial complexes both with and without induction, to classify their differences.

The first construction we will consider is complete graphs.

Definition 5.4.1. A graph $G = (V, E)$ is said to be *complete* whenever $u \in V$ and $v \in V$, uEv. That is, a graph is complete if it has every possible edge.

Definition 5.4.2. The *standard* complete graph K_n, for a natural number n is the complete graph whose vertex set is the set n (recalling that n is the set $\{0, 1, \ldots, n-1\}$).

Here, I have drawn the first few complete graphs.

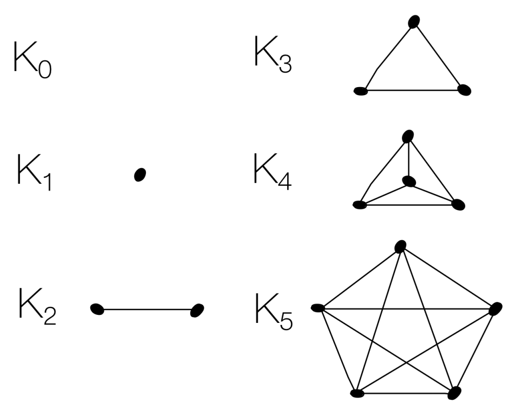

FIGURE 20. The first few complete graphs.

Although this definition of K_n is relatively straightforward, it is not the only one we could have chosen. In particular, notice the following observation: K_0 has no vertices. K_{nS} is K_n with the extra vertex n inserted and connected to all the previous vertices.

We can use this to define K_n as an inductive construction.

Definition 5.4.3. K_n is defined to be the pair (V_{K_n}, C_{K_n}) where V_{K_n} is defined by

$$V_{K_0} \colon \emptyset$$
$$V_{K_{nS}} \colon V_{K_n} \cup \{n\}$$

and C_{K_n} is defined by

$$C_{K_0} \colon \{\emptyset\}$$
$$C_{K_{nS}} \colon C_{K_n} \cup \{\{n, v\} \mid v \in V_{K_{nS}}\}.$$

This definition says that K_0 has no vertices, the vertices of K_1 are $\{0\}$, the vertices of K_2 are $\{0, 1\}$, the vertices of K_3 are $\{0, 1, 2\}$, etc. This is exactly what we expect it to be.

The definition also says that K_0 has no edges (but it still does have the empty cell so that it is a complex), K_1 has the cells of K_0 and a cell from 0 to every other vertex so that its cells are $\{\emptyset, \{0\}\}$, K_2 has the cells of K_1 and a new cell from 1 to every other vertex, and the pattern repeats. Assuming that this definition is valid, it is relatively straightforward that this creates the same graph as our original definition.

Let us first prove that this kind of definition is valid.

Theorem 5.4.4. The *recursion theorem* states that for a set X, an element $x \in X$, and a function $F \colon \mathbb{N}, X \to X$, there is a unique function $f \colon \mathbb{N} \to X$ such that $[0]f = x$ and $[nS]f = [n, [n]f]F$.

Before we prove this theorem, let us use it to prove that our definition of V_{K_n} is valid. Rather than constructing a sequence of sets V_{K_n}, let us think of V_{K_n} as a function f such that $[n]f = V_{K_n}$. Notice that each of our sets $[n]f$ are sets of natural numbers. Thus, let the set X be $\mathbb{N}\mathcal{P}$. Then, notice that our definition of $[nS]f$ is $[n]f \cup \{n\}$. Create a function $F \colon \mathbb{N}, X \to X$ given by $[n, x]F = x \cup \{n\}$. This takes a natural number and a set of natural numbers to a set of natural numbers, so this is a valid definition. Then, we have declared that $[nS]f = [n, [n]f]F$. Further, let x be $\emptyset$. Then, our definition of $[0]f$ is that $[0]f = \emptyset$. In summary, we have used a set X, and element $x \in X$, and a function $F \colon \mathbb{N}, X \to X$ and claimed that $f \colon \mathbb{N} \to X$ is a function such that $[nS]f = [n, [n]f]F$. The recursion theorem guarantees that this function f is unique, and thus that our definition is valid.

Let us now prove the recursion theorem.

Proof. Suppose we have a set X, an element $x \in X$ and a function $F\colon \mathbb{N}, X \to X$. We wish to show that there is a unique function $f\colon \mathbb{N} \to X$ such that $[0]f = x$ and $[nS]f = [n, [n]f]F$.

Consider the set

$$A\colon \{U \in \langle \mathbb{N}, X\rangle \mathcal{P} \mid (0, x) \in U \wedge ((n, y) \in U \Rightarrow (nS, [n, y]F) \in U)\}.$$

This is guaranteed to be a set by the axioms of power set and specification. That is, A is the set of all relations U from $\mathbb{N}$ to X such that $0Ux$ and if nUy then $(nS)U[n, y]F$. Note the similarities between these relations and the desired function f (f has these properties where y takes the place of $[n]f$). This set A is necessarily non-empty, because $\langle \mathbb{N}, X\rangle \in A$ (it contains every pair, rather than just the ones prescribed). Therefore, this set has an intersection.

It is necessarily the case that $A \bigcap \in A$ because every set $U \in A$ has all the elements required to be elements of A, and thus so will their intersection. Let us use the induction principle of the natural numbers to show that $\forall n. (n \in \mathbb{N} \Rightarrow \exists! y. (n, y) \in A \bigcap)$, which is that $A \bigcap$ is a function. First, we prove this for $n = 0$. $\exists y. (0, y) \in A$ because $(0, x) \in U$ for each $U \in A$, and thus $(0, x) \in A \bigcap$. Further, this is unique, for if there is a pair $(0, y) \in A \bigcap$ other than $(0, x)$, then $\{z \in A \bigcap \mid z \neq (0, y)\}$ is still an element of A (because $(0, y)$ could not be required as 0 is not nS for any n). Thus, $(0, y)$ couldn't be in $A \bigcap$ because it is not in every element of A.

Next, suppose that

$$\exists! y. (n, y) \in A \bigcap.$$

We wish to show that

$$\exists! z. (nS, z) \in A \bigcap.$$

This is a similar argument. Let y be the prescribed value. Since this is in every U in A, $(nS, [n, y]F)$ is in every U in A and is therefore also in $A \bigcap$. Thus, let z be $[n, y]F$. If there is some other value w such that $(nS, w) \in A \bigcap$, then it could be removed from $A \bigcap$ while maintaining membership in A. This is because, since there will be no $(n, y_w) \in A \bigcap$ such that $[n, y_w]F = w$, the only such y_w is y which satisfies $[n, y]F = z$. Thus, this value (nS, w) is not in every element of A and thus is not in $A \bigcap$. Therefore, the pair (n, z) is unique.

Thus, we have shown that $\forall n. (n \in \mathbb{N} \Rightarrow \exists! y. (n, y) \in A \bigcap)$ and thus that $A \bigcap$ is a function. Denote this function by f. Then, f satisfies the properties prescribed, because $[0]f = x$ and if $[n]f = y$, then $[nS]f = [n, y]F$ which is that $[nS]f = [n, [n]f]F$.

By induction, this f is unique, for if g satisfies the required properties, the $[0]f = [0]g = x$ and $[n]f = [n]g \Rightarrow [nS]f = [n, [n]f]F = [n, [n]g]F = [nS]g$, so that $[n]f = [n]g$ for all $n \in \mathbb{N}$ by induction. Thus, we have proven the desired claim. $\square$

The recursion theorem is extremely powerful. It allows us to create simple definitions for necessarily infinite structures.

Let us show how the definition of C_{K_n} is of the form of the recursion theorem. Each such C_{K_n} is a set of sets of natural numbers, so let X be $\mathbb{N}\mathcal{P}\mathcal{P}$. Define a function $F\colon \mathbb{N}, X \to X$ given by $[n, x]F = x \cup \{\{n, v\} \mid v \in V_{K_{nS}}\}$. Since both sets in this union of sets of sets of natural numbers, F is a valid function. Then, let x be $\{\emptyset\}$. Using the recursion theorem, we get a function $f\colon \mathbb{N} \to \mathbb{N}\mathcal{P}\mathcal{P}$ satisfying the definition of C_{K_n} so that we can simply set $C_{K_n} = [n]f$.

Let us create another useful definition.

Definition 5.4.5. Define the standard n simplex $\Delta_n\colon (V_{\Delta_n}, C_{\Delta_n})$ by[10]

$$V_{\Delta_0}\colon \{0\}$$
$$V_{\Delta_{nS}}\colon V_{\Delta_n} \cup \{nS\}$$

and

$$C_{\Delta_0}\colon \{\emptyset, \{0\}\}$$
$$C_{\Delta_{nS}}\colon C_{\Delta_n} \cup \{c \cup \{nS\} \mid c \in C_{\Delta_n}\}.$$

This definition states that Δ_0 is a complex representing a point, and Δ_{nS} is a complex appending a new point, and extending every previous cell by this new point.

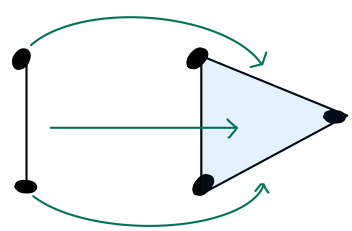

FIGURE 21. Adding a new point to Δ_1 to get Δ_2. The green arrows take a cell of Δ_1 to its extension in Δ_2. The extension of the empty cell is the new vertex.

[10]Δ is the uppercase Greek letter delta. It is used here because it is shaped like a triangle, and so is Δ_2.

Because of the recursion theorem, this definition is valid. There is actually a much simpler characterization of Δ_n though.

Theorem 5.4.6. For each $n \in \mathbb{N}$, $\Delta_n = \{nS, nS\mathcal{P}\}$.

Proof. Since the Δ_n are defined inductively, we can prove this theorem using the induction principle of the natural numbers.

For $n = 0$, this theorem is trivial, for $1 = \{0\}$ thus $V_{\Delta_0} = 1$, and $1\mathcal{P} = \{\emptyset, \{0\}\}$, and thus $C_{\Delta_0} = nS\mathcal{P}$.

Now, assume that $\Delta_n = (nS, nS\mathcal{P})$. We wish to show that $\Delta_{nS} = (nSS, nSS\mathcal{P})$. Since $V_{\Delta_n} = nS$, we know that $V_{\Delta_{nS}} = nS \cup \{nS\} = nSS$. We know that $C_{\Delta_n} = nS\mathcal{P}$. We wish to show that $C_{\Delta_{nS}} = C_{\Delta_n} \cup \{c \cup \{nS\} \mid c \in C_{\Delta_n}\} = nS\mathcal{P} \cup \{c \cup \{nS\} \mid c \in nS\mathcal{P}\} = nSS\mathcal{P}$. It is clear that $nS\mathcal{P} \cup \{c \cup \{nS\} \mid c \in nS\mathcal{P}\} \subseteq nSS\mathcal{P}$, since every element of $nS\mathcal{P}$ is in $nSS\mathcal{P}$, and every element of every element of the other set is either in nS or is nS and so is in nSS. Thus, we need to show that $nSS\mathcal{P} \subseteq nS\mathcal{P} \cup \{c \cup \{nS\} \mid c \in nS\mathcal{P}\}$ To do this, suppose we have some $d \in nSS\mathcal{P}$. If $nS \notin d$, then $d \in nS\mathcal{P}$, and so $d \in nS\mathcal{P} \cup \{c \cup \{nS\} \mid c \in nS\mathcal{P}\}$. Otherwise, $d = \{c \in d \mid c \neq nS\} \cup \{nS\}$. In this case, $d \in \{c \cup \{nS\} \mid c \in nS\mathcal{P}\}$. Thus, for any $d \in nSS\mathcal{P}$, $d \in nS\mathcal{P} \cup \{c \cup \{nS\} \mid c \in nS\mathcal{P}\}$. Thus an element of one set is an element of the other, and so they are the same sets. Namely, $C_{\Delta_{nS}} = nSS\mathcal{P}$. Therefore, we have shown that $\Delta_{nS} = (nSS, nSS\mathcal{P})$.

By the induction principle of the natural numbers, we have shown that $\forall n.\ (n \in \mathbb{N} \Rightarrow \Delta_n = (nS, nS\mathcal{P}))$. □

There is not much to be gained from using the induction definition of the simplices rather than this one. Instead, this was to present another example of how to use induction definitions.

Exercise 5.4.1. Use the definition of K_n to prove that

$$K_n = (n, \{\emptyset\} \cup \{(u, v) \mid u \in \mathbb{N} \wedge v \in \mathbb{N}\}.$$

Definition 5.4.7. The path graph I_n is defined as the complex

$$I_n \colon (nS, \{\emptyset\} \cup \{\{k\} \mid k \in nS\} \cup \{\{k, kS\} \mid k \in n\}).$$

That is, the graph I_n has nS vertices and and edge from k to kS for each k. This means that I_n is just a line with n pieces.

Exercise 5.4.2. Each path graph I_{nS} is the graph I_n with an extra vertex and edge appended. Codify this intuition into an inductive definition.

5.5. **Combinatorial Manifolds.** The real reason that we introduced the recursion theorem is so that we may define a more complicated structure of an *compact contractible combinatorial manifold.* Let us work with each one of those words one at a time.

All the geometry we have seen so far has been discrete: that is, everything has been built out of finite pieces. This is not how most of geometry is done. For example, a circle does not consist of a finite number of edges, in any meaningful way, even though it seems similar to graphs like C_5. These more general shapes are studied in the area of mathematics called *topology.* Topology, however, is complicated. Proving results from the axioms of set theory, even simple results like the existence of the circle, require much more work than we have done in this thesis. In fact, the topic change from tropical geometry to discrete geometry was in order to avoid discussing topology. Regardless, we will still discuss the intuitions of topology.

The best behaved kind of shape in topology is a *manifold.* A manifold is, intuitively, a shape that is locally flat. For example, the flat Earth conspiracy has arisen because a sphere is a manifold. A proportionally small piece of a sphere appears to be flat. This is not all bad, however, for it allows us to study round things like a sphere with mathematics developed for flat shapes. For example, high school physics does not generally concern itself with the fact that throwing a ball 20ft actually causes the force of gravity that the ball experience to change slightly in direction. This is because the change is so small because the ball moves along such a small portion of the Earth.

The discrete versions of manifolds are called *combinatorial manifolds.* The basic intuition for a two dimensional combinatorial manifold is that it is a piece of paper. Except, imagine that, for some reason, you need a bigger piece of paper. Since you don't have a bigger piece of paper, you glue two pieces of paper together along an edge. This is still, basically, a piece of paper. If you need more paper, you're allowed to glue more paper along its edges. But suppose that you accidentally glue the papers into a tube. This is okay, for at every intersection of the papers, the tube still acts basically like a piece of paper. Conceivably, you could even glue the tubes together into a doughnut shape. However, if you ever glued three pieces of paper together along one edge, well, that wouldn't look like a piece of paper at that intersection. Further, if you glued two pieces of paper at their corners, then that wouldn't look like a piece of paper at the intersection either. Thus, we create the following definitions.

Definition 5.5.1. Suppose S is a simplicial complex. S is called n *dimensional* if it contains an n and no larger cells. S is called *pure* n *dimensional* if every cell is a face of an n cell.

These let us classify how "filled in" a complex is.

Informal Definition 5.5.2. S is called an n *dimensional (combinatorial) manifold* if it is pure n dimensional and the space around every cell "looks like" a piece of n-dimensional space. The *boundary* of a manifold S, denoted by ∂S is the complex whose cells are all of the $n-1$ cells of S which appear in exactly one n cell, and each of the subcells of those cells. There is a natural inclusion homomorphism $\iota_S\colon \partial S \to S$ for each manifold S.[11]

In effect, a complex is a manifold if we never glued three pieces of paper together or glued two at their corners. Let us consider examples of these definitions. First, every non-empty graph is 0 or 1 dimensional. It is 1 dimensional if it contains an edge. It is pure 1 dimensional if every vertex is included in an edge. Here is an example of a two dimensional complex.

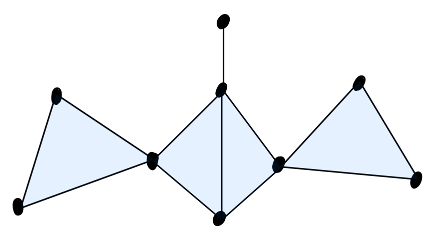

FIGURE 22. A two dimensional complex.

This is two dimensional because it contains at least one triangle and no higher dimensional cells. It is not pure, however, because of the stray edge hanging off. However, this next complex is pure two dimensional.

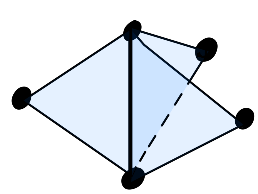

FIGURE 23. A pure two dimensional complex.

[11]ι is the lowercase Greek letter iota.

This complex, however, is not a manifold. This is because the central edge is shared by three triangles. A complex can also fail to be a manifold if triangles are glued together at vertices. The impure two dimensional complex would not be a manifold, even with the edge removed, for this reason. Finally, we present a two dimensional manifold.

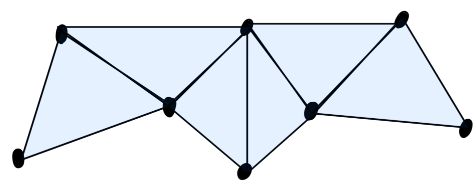

FIGURE 24. A two dimensional manifold.

As a particularly important example, each of the complexes Δ_n are n dimensional manifolds. Although we have only presented an informal definition of a manifold, we will not need the precise definition to present our final construction.

The definition of a compact contractible manifold relies heavily on on the notion of *gluing*, so we need to nail down this notion. However, it is complicated, so we're going to work through it before we present the full definition.

Complexes can be glued together two at a time. Thus, suppose S and T are the following complexes.

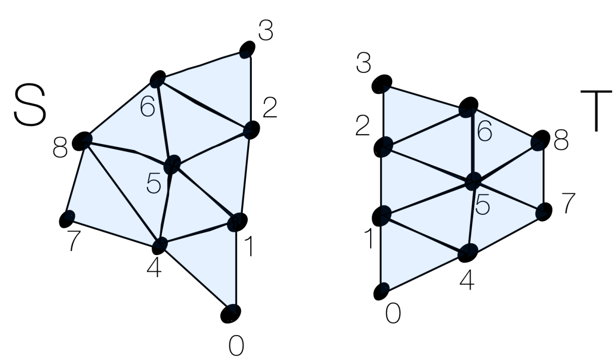

FIGURE 25. The complexes S and T.

We need to express how these complexes are going to be glued. Thus, suppose that I is the following complex.

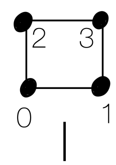

FIGURE 26. The complex I.

We will define two homomorphisms $f_S\colon S \to I$ and $f_G\colon S \to I$ representing how the complex I should be represented in S and T respectively.

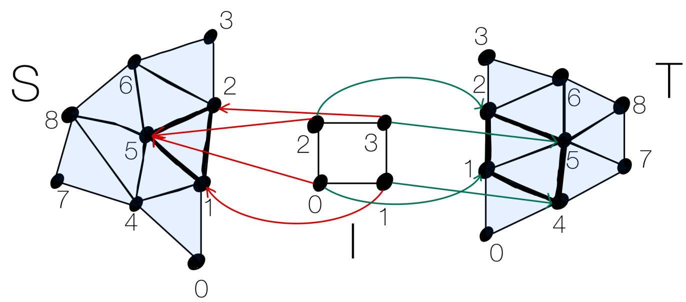

FIGURE 27. Homomorphisms from I into S and T, denoted with red and green arrows respectively.

Since 1 maps to 5 and 2 in S and T respectively, we mean to glue these vertices together. Also that the vertex 0 and 1 in I are both mapped to the vertex 5 in S. Thus, both 1 and 2 in T will get glued to 5 in S. In fact, we will glue the bold square onto the bold triangle, squishing the edge of the triangle in the process.

Before we learn how to identify vertices of a complex, we need a way to join complexes together which avoids the difficulties of set theory. In particular, we want a single complex which is S and T sitting next to each other. One guess might be a union, which sticks sets together. However, if we make a complex whose vertex set is $V_S \cup V_T$, then the 0 in S and the 0 in T would get regarded as the same point, which we do not want.

Instead, we define a complex U such that

$$V_U\colon \{(s,1) \mid s \in V_S\} \cup \{(t,2) \mid t \in V_T\}$$
$$C_U\colon \{\{(s,1) \mid s \in c\} \mid c \in C_S\} \cup \{\{(t,2) \mid t \in c\} \mid c \in C_T\}.$$

as well as inclusion homomorphisms $i_S\colon S \to U$ and $i_T\colon T \to U$ given by

$$[s]i_S = (s,1) \text{ for } s \in V_S$$
$$[t]i_T = (t,1) \text{ for } t \in V_T.$$

That is, U is the complexes S and T next to each other, where the vertices of S get decorated with an 1 and the vertices of T get decorated with a 2 so that they can be distinguished. I has two mappings into U, one from S and one from T.

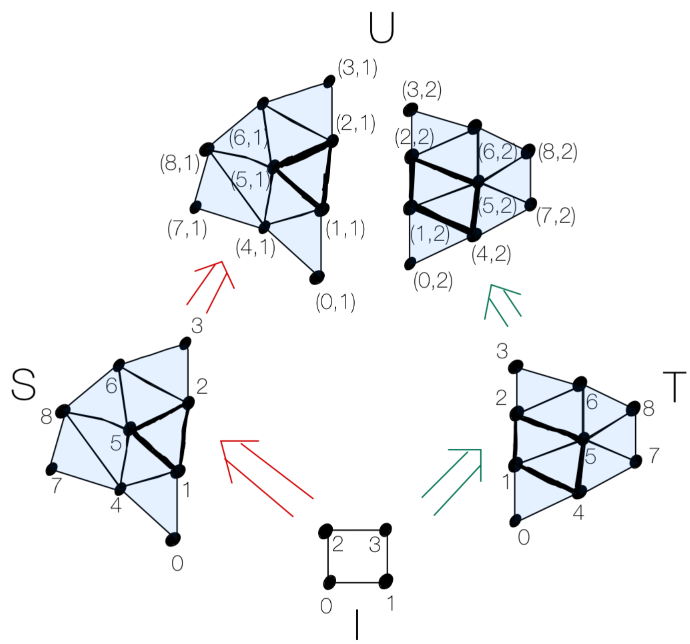

FIGURE 28. The complex U and its relationship to S, T, and I.

In order to figure out which vertices of U actually get glued together, we define a relation, R between V_U and itself.

$$R\colon \{(u,v) \mid \exists i.\, i \in V_I \wedge [i](f_S\,;i_S) = u \wedge [i](f_T\,;i_T) = v\}.$$

In particular, uRv whenever there is a vertex i of I such that i gets sent to u by embedding into S and then U and i gets sent to v by embedding into T and then U. That is, if uRv, the u and v are supposed to be glued together. But suppose uRv and wRv. In that case, u gets glued to v and so does w, so thus u and w are also glued together. For some vertex u of U, we need a way to find the set of all vertices of U that are glued to u. Thus, we define a function $[\ ]_R \colon V_U \to V_U\mathcal{P}$. For each u in V_U, the value of this function is denoted as $[u]_R$ and is called the *equivalence class of u under R* or the *gluing class of u*. This function is defined such that for any vertex u of U,

$$[u]_R = \{c \in V_U\mathcal{P} \mid u \in c \wedge \forall v, w.\, (vRw \Rightarrow v \in c \Leftrightarrow w \in c)\} \bigcap.$$

This intersection is guaranteed to exist because V_U satisfies the required conditions. Further, $u \in [u]_R$ and if vertices v and w are glued together (vRw) and one is in the gluing class, then so is the other. If two vertices get mapped to the same gluing class, then they are supposed to be glued to each other. For example, $[(5,1)]_R = \{(5,1),(2,2),(2,1)\}$, $[(6,2)]_R = \{(2,1),(6,2)\}$, and $[(8,1)]_R = \{(8,1)\}$.

As a result, we can finally form the vertex set of our gluing. In particular, we set

$$V_G \colon \{[u]_R \mid u \in V_U\}.$$

That is, the vertices are the set of all gluing classes. Then, finally, we define the cells of the complex.

$$C_G \colon \{\{[u]_R \mid u \in c\} \mid c \in C_U\}.$$

That is to say, there is a cell in the gluing complex only if that cell "was there before the gluing." This yields our final glued complex.

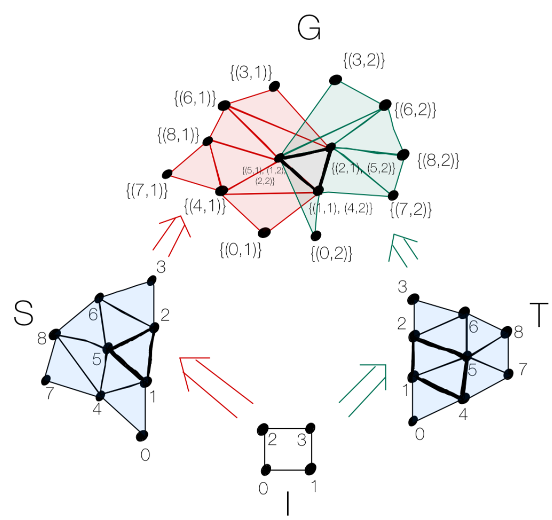

FIGURE 29. The glued complex G and its relationship to S, T, and I.

One should think of the green bits of the complex as "sticking out" of the page. Although we have glued the square to the triangle, we have not glued the rest of the paper around it. As a final note, there are homomorphisms $g_S\colon S \to G$ and $g_T\colon T \to G$ given by $[s]g_S = [[s](f_S\,;i_S)]$ and $[t]g_T = [[t](f_T\,;i_T)]$ which explain how S and T are mapped into their gluing. Now, we present the definition of gluing.

Definition 5.5.3. Suppose that S, T, and I are complexes and that $f_S\colon I \to S$ and $f_T\colon I \to T$ are homomorphisms. We define the *gluing* of S and T along I as the complex $[f_S, f_T]G$ with vertex set

$$V_G\colon \{[u]_R \mid u \in V_U\}$$

where U is the such that

$$V_U\colon \{(s,1) \mid s \in V_S\} \cup \{(t,2) \mid t \in V_T\}$$
$$C_U\colon \{\{(s,1) \mid s \in c\} \mid c \in C_S\} \cup \{\{(t,2) \mid t \in c\} \mid c \in C_T\}$$

and $[\ ]_R\colon U \to U\mathcal{P}$ is the function defined for $u \in V_U$ as

$$[u]_R = \{c \in V_U\mathcal{P} \mid u \in c \wedge \forall v, w.\,(vRw \Rightarrow v \in c \Leftrightarrow w \in c)\}\bigcap$$

where R is the relation on V_U given by

$$R\colon \{(u,v) \mid \exists i.\, i \in V_I \wedge [i](f_S\,;i_S) = u \wedge [i](f_T\,;i_T) = v\}$$

where $i_S\colon S \to U$ and $i_T\colon T \to U$ are homomorphisms for $s \in V_S$ and $t \in V_T$ given by $[s]i_S = (s,1)$ and $[t]i_T = (t,1)$. We also define gluing homomorphisms $g_S\colon S \to G$ and $g_T\colon T \to G$ given by $[s]g_S = [[s](f_S\,;i_S)]$ and $[t]g_T = [[t](f_T\,;i_T)]$.

Like many of the other definitions in mathematics, this definition is quite complicated, but it expresses a natural intuition of gluing complexes together. This definition can, in fact, be simplified significantly using more advanced forms of mathematics. This structure is actually called a *pushout* which can be defined in terms of a *coproduct* and a *coequalizer*, which it itself a special kind of *quotient*. Rather than present is these notions, I have decided to directly present the definition of gluing because gluing is the only notion needed to present the definition of a compact contractible manifold. We will now present this notion with an inductive construction.

Definition 5.5.4. An n dimensional compact contractible (combinatorial) manifold (CCCM) is defined as follows:

(1) Δ_n is an n dimensional CCCM.
(2) The gluing of two nS dimensional CCCMs S and T along homomorpisms $f_S\,;\iota_S\colon I \to S$ and $f_T\,;\iota_T\colon I \to T$ where $f_S\colon I \to \partial S$ and $f_T\colon I \to \partial T$ are monomorphisms and I an n dimensional CCCM is an nS dimensional CCCM.

At this point, we have *completed the circle* or whatever the idiom is. This inductive definition is different from the ones defining K_n and Δ_n. In those two, we were defining a sequences of complexes. In this definition, we define a sequence of *sets* of complexes. That is, for each n, we define a collection of the n dimensional CCCMs. Not only are these defined in terms of $n-1$ dimensional CCCMs but also in terms of other n dimensional CCCMs. This is reminiscent of the definition of a proposition, where propositions were defined in terms of themselves. In fact, the inductive definition of a CCCM is both of these techniques combined, since there was not a sort of proposition for each n.

Before we justify this definition, let us describe what it means. First, since 0 is not nS for any n, only the first rule allows us to create 0 dimensional CCCMs. The definition says that Δ_0 is the only 0 dimensional CCCM.

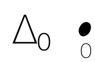

FIGURE 30. The only 0 dimensional CCCM.

Next, we know that Δ_1 is a 1 dimensional CCCM.

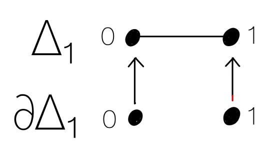

FIGURE 31. The 1 dimensional CCCM Δ_1 and its boundary.

The boundary of Δ_1 is two points. This is important for the definition of gluing. Let us glue Δ_1 to itself. For this, we need a 0 dimensional CCCM, so we take Δ_0, and homomorphisms $f_S\colon \Delta_0 \to \partial\Delta_1$ and $f_T\colon \Delta_0 \to \partial\Delta_1$. Here, we let f_S pick out the vertex 1 and f_T pick out the vertex 0. Then, we can perform the gluing to get the complex

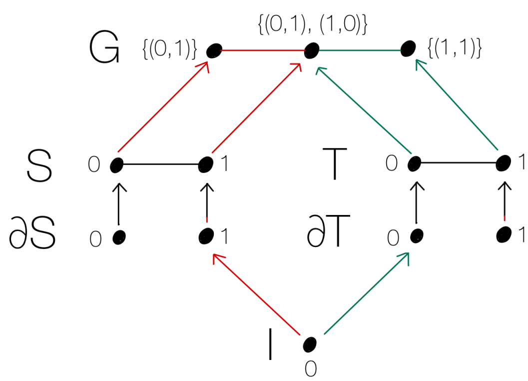

FIGURE 32. Two intervals Δ_1 glued together into a larger interval.

Since the only 0 dimensional CCCM is the point, a 1 dimensional CCCM is the segment Δ_1 or a gluing of two segments at an end point. That is, each 1 dimensional CCCM is isomorphic to an interval I_{nS}.

Now let us consider 2 dimensional CCCMs. The first rule means that Δ_2 is a 2 dimensional CCCM. The boundary of Δ_2 is an empty triangle. Using the second rule, we can glue two Δ_2s along a Δ_1 inside their boundary.

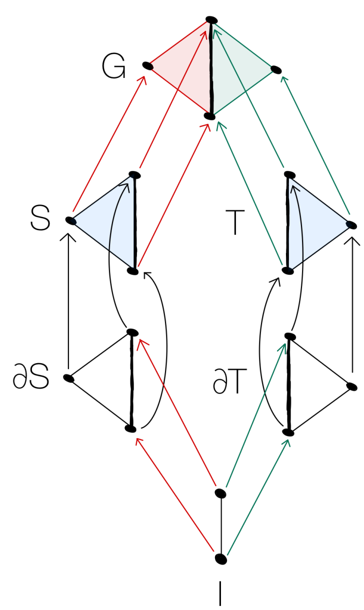

FIGURE 33. Two triangles glued together to get a square.

We could glue another triangle onto this square.

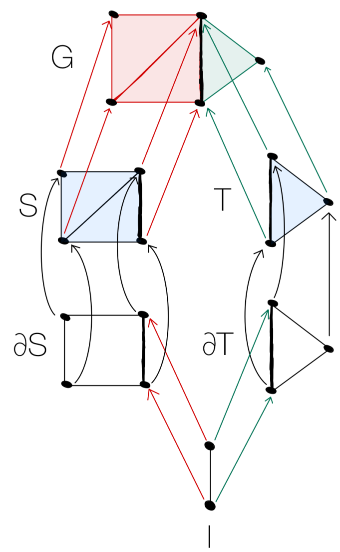

FIGURE 34. A square and a triangle glued to get a pentagon.

We can also glue complexes along longer intervals.

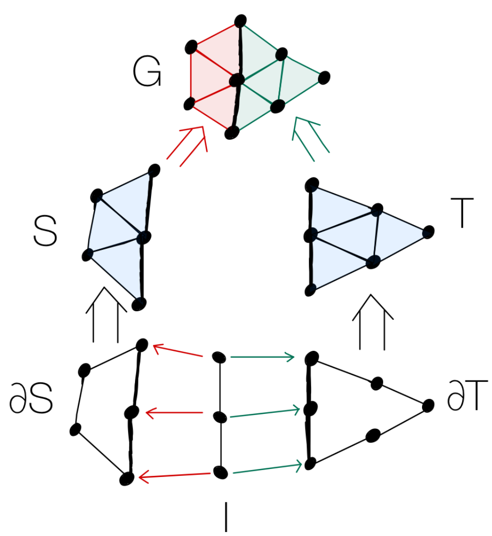

FIGURE 35. A gluing along a longer interval.

These, and complexes like it, are the CCCMs.

We requires that the maps we glue along be monomorphisms (and from lower dimensional CCCMs) to present getting "holes" in the complex. For example, we do not allow the following complex

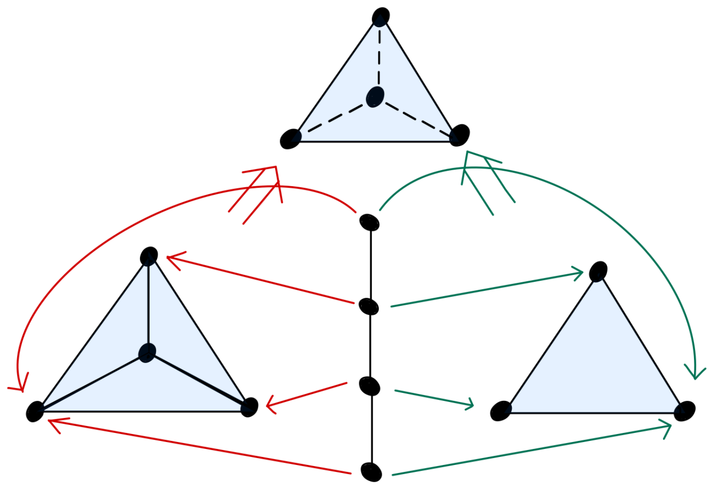

FIGURE 36. A gluing that does not result in a CCCM since the maps from I are not monomorphisms.

which is "hollow". Since there were no "filled in" three dimensional cells in the complexes glued, the inside of this shape is hollow. This is the origin of the term *contractible*. It refers to the fact that the complex is just some "thickened" point rather than some exotic shape. The term *compact* refers to the fact that the complex is finite, and so must be relatively small. Each of these complexes are finite because the Δ_n are and a gluing of finite complexes is still finite. Although the definition of a manifold was informal, this definition is fully formal. Since 1 dimensional CCCMs are intervals, and 2 dimensional CCCMs can only be glued along such intervals of their boundary, we have effectively guaranteed that our "pieces of paper" are only glued along their edges.

Finally, we will discuss 3 dimensional CCCMs. Δ_3 is the simplest 3 dimensional CCCM. The other three dimensional CCCMs are gluings of such complexes along 2 dimensional CCCMs. For example, we may glue a tetrahedron to a square pyramid along a triangular face.

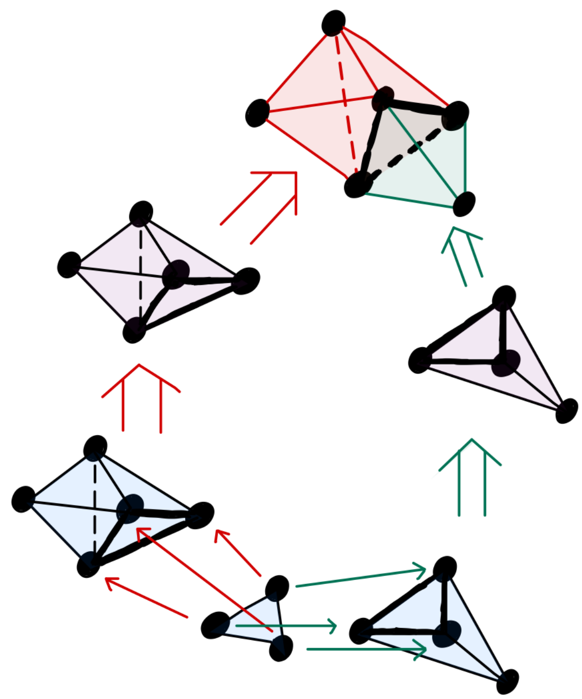

FIGURE 37. A gluing of 3 dimensional CCCMs.

This process repeats for higher dimensional CCCMs, but those are not as simple to visualize.

Let us now show that this definition is valid. This inductive construction is not a simple application of the recursion theorem, but we will explain how to formalize it here. Let M_n denote the set of all n dimensional CCCMs.

First, it is easy to show that the set of 0 dimensional CCCMs are specified: it is the set $M_0 \colon \{\Delta_0\}$. Now, suppose that M_n has been well-defined. Let us use this to build M_{nS}. This is itself an inductive construction, because it depends on itself. Thus, we will build M_{nS} in stages, which we will denote as M^k_{nS} for a natural number k. First, let $M^0_{nS} \colon \{\Delta_{nS}\}$. Then, define a function F such that $[n, M]F$ is the set of all gluings of of complexes in M by monomorphisms from complexes in M_n into their boundaries, as described in the definition of CCCMs, unioned with M. These sets are all guaranteed to exists due to the existence of function sets, specification, and replacement. Then, M^k_{nS} is defined by the recursion theorem such that M^0_{nS} is the set containing the simplex and M^{kS}_{nS} is the set of complexes in M^k_{nS} as well as gluings of such complexes by monomorphic boundary maps from M_n. In that sense, M^k_{nS} is a sequence of stages of gluings, where at each step more and more complicated gluings are allowed. The complexes at stage k are those complexes from the previous stages and gluings of the complexes from the previous stages. Then, the set $M_{nS} = \{M^k_{nS} \mid k \in \mathbb{N}\} \bigcup$ which exists by the axioms of replacement and union. That is, M_{nS} is the set of complexes created at any stage M^k_{nS}.

Importantly, this has created a function G which takes a set of n dimensional complexes and makes a set of nS dimensional complexes. In particular, let $[n, M]G$ be the set defined above with M used as the set of n dimensional complexes. Then, we can use the recursion theorem again to get a function M_n such that $M_0 = \{\Delta_0\}$ and M_{nS} is the set containing Δ_{nS} and all gluings from the complexes in M_n. Therefore, this definition uniquely determines the set of n dimensional complexes for each n. Here, we had to use the recursion theorem twice, because M_n was defined both in terms of itself and in terms of the lower dimensional complexes. This reveals the true power of the recursion theorem: Even though the recursion theorem is "one dimensional" (the natural numbers are in a "line"), we can use the recursion theorem multiple times to get structures which are "two dimensional" or even structures with higher dimension.

Although we have defined the CCCMs, we have not described what they can be used for. This is, for the most part, out of the scope of this thesis. The purpose of this thesis was to show how complex geometric structures can be defined using induction, and we have now completed this goal. However, these complexes were not invented for no reason.

As far as I am aware, I am the one who invented this definition of CCCMs. The terms compact and contractible are terms which originate from topology. The "true" definition of a CCCM is a complex that, when recognized as a topological space, is topologically a compact and contractible manifold. However, topology is hard. Rather than using topology to discuss these cell complexes, which are discrete, using the continuous methods of topology, this inductive definition allows CCCMs to be described using only discrete methods (in "only" 100 pages or so).

CCCMs are the building block for a theory of homotopy of graphs that I am developing. Homotopy theory, to an extent, studies holes in a shape. Truly, it studies the failure of certain shapes embedded inside other shapes to be *contracted*. For example, if a circle is drawn on a piece of paper, you could imagine shrinking the circle until it becomes a point, having it remain on the paper the entire time.

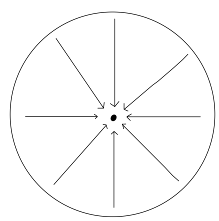

FIGURE 38. A circle can be "contracted" to a point if it is embedded in a piece of paper.

However, if a circle is drawn around the loop of a surface of a doughnut, it cannot be shrunk until it becomes a point without the circle breaking or leaving the surface of the doughnut. This can actually be done in two different ways. None of the following circles can be transformed into each other without tearing them or pulling them off the surface of the doughnut (surface meaning that you cannot pull the loops through the inside of the doughnut).

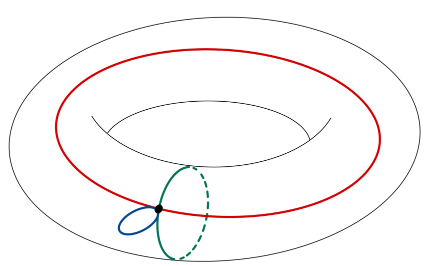

FIGURE 39. These circles, embedded on the surface of a doughnut, cannot be transformed into each other.

However these two, which are both looped around the inside of the doughnut, can be transformed into each other.

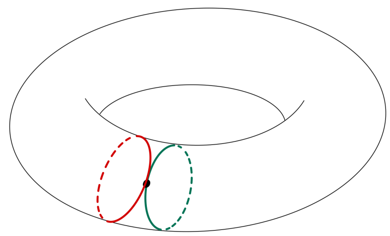

FIGURE 40. These circles, both running through the inner loop of a doughnut, can be transformed into each other.

This difference between circles embedded on the plane and circles embedded on the doughnut indicates an important difference between the doughnuts and the plane: doughnuts have a hole (actually, three holes mathematically speaking). The idea of CCCMs is that all maps into them are contractible. This is why we forbid any gluings which could allow holes.

My goal is to develop a homotopy theory of graphs that is similar to the homotopy of topological spaces in terms of algebraic power. That is, it should describe the "holes" in a graph. I believe this is possible, using CCCMs as the fundamental building blocks. By using induction for my definitions, I am able to produce compact and elegant descriptions of the necessary components. Importantly, these definitions can also be understood by computers, enabling me to use digital tools to help explore my theories.

5.6. **Recommended Readings.** For a basic description of relations and graphs, I recommend Epp's textbook[4] on discrete mathematics. For a book focused on graphs, I recommend Chartrand[5] from where I originally learned the subject. For a discussion of simplicial complexes, I recommend Gallier's textbook[7] on compact surfaces. This describes complexes from a topological viewpoint, so it will require many topological preliminaries. For those, you should see a topology textbook, but I do not have a particular recommendation.

Finally, I will cite the graph theory papers [6] and [3] which inspired my work on graph homotopy. These papers are not intended for the non-mathematician, but they are very good. I would recommend them to those who have studied the preliminary materials.

6. Summary

Let us recall what we have accomplished in this thesis. First, we described what it looks like to read mathematics. Then, we used induction to define the language of logic, the semantics of logic, and the language of formal proofs. Once we had logic, we discussed the axioms of set theory, using induction of propositions to prove theorems like the substitution theorem, and showing how induction arises within set theory using the natural numbers. With the language of set theory, we discussed geometry. We showed how the various types of sets in set theory can be used to codify relationships and shapes. Finally, we showed how induction can be used to define certain shapes, including the particularly complicated structure of compact contractible combinatorial manifolds.

Induction is one of the most powerful tools in all of mathematics. Not only does it appear in these topics, but also in effectively every other area of mathematics. Although we have not truly used induction to its full potential, which is proving complicated results with ease, we shown have how it can be used to create complicated structures. If you are interested in more uses of mathematical induction, I hope that you seek out content on constructive mathematics, where it has the highest importance.

References

[1] Samuel R. Buss, *An introduction to proof theory*, Elsiver Science B.V., 1998.
[2] Keith Devlin, *Introduction to mathematical thinking*, Keith Devlin, 2012.
[3] Anton Dochtermann, *Hom complexes and homotopy in the category of graphs*, Electronic Notes in Discrete Mathematics **31** (2008), 131 –136. The International Conference on Topological and Geometric Graph Theory.
[4] Susanna S. Epp, *Discrete mathematics with applications, fourth edition*, Brooks/Cole Cengage Learning, 2011.
[5] Ping Zhang Gary Chartrand, *A first course in graph theory*, Dover Publications, 2010.
[6] Alexander Grigor'yan, Yong Lin, Yuri Muranov, and Shing-Tung Yau, *Homotopy theory for digraphs*, Pure and Applied Mathematics Quarterly **10** (201407).
[7] Dianna Xu Jean Gallier, *A guide to the classification theorem of compact surfaces*, Springer, 2013.
[8] Thomas Jech, *Set theory*, Springer, 2003.
[9] P.D. Magnus, *An introduction to formal logic*, Fecundity, 2012.
[10] Grigori Mints, *A short introduction to intuitionistic logic*, Springer, 2000.
[11] Adam Brooks Webber, *Formal language: A practical introduction*, Franklin, Beedle & Associates Incorporated, 2011.

Belmont University, Nashville, TN 37212
Email address: davis.deaton@pop.belmont.edu