Extracted from:

Good Math

A Geek's Guide to the Beauty of Numbers, Logic, and Computation

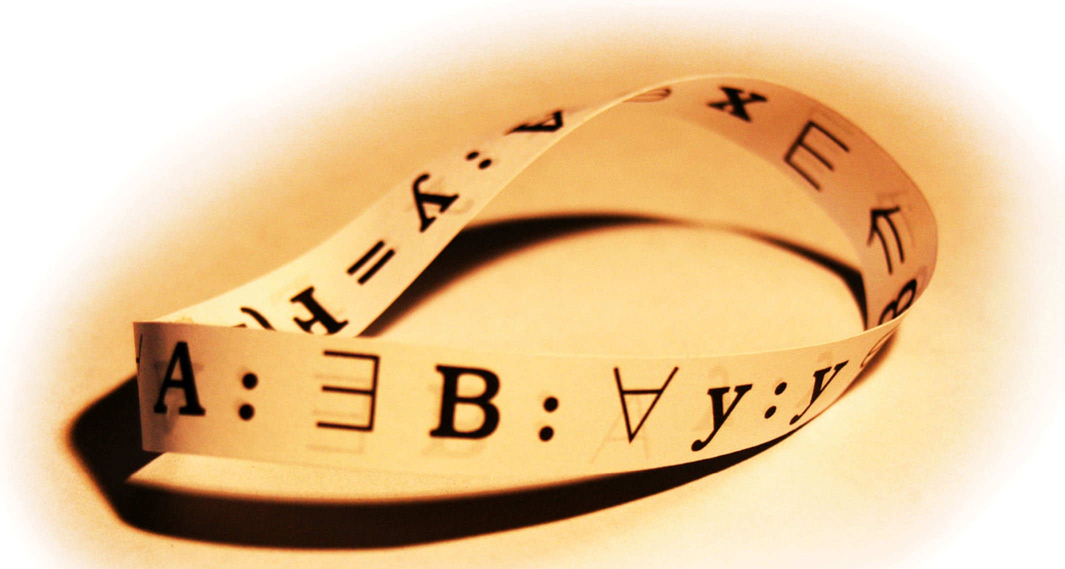
The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

Good Math

A Geek's Guide to the Beauty of Numbers, Logic, and Computation



Mark C. Chu-Carroll

Edited by John Osborn

Good Math

A Geek's Guide to the Beauty of Numbers, Logic, and Computation

Mark C. Chu-Carroll

The Pragmatic Bookshelf
Dallas, Texas • Raleigh, North Carolina

Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this book, and The Pragmatic Programmers, LLC was aware of a trademark claim, the designations have been printed in initial capital letters or in all capitals. The Pragmatic Starter Kit, The Pragmatic Programmer, Pragmatic Programming, Pragmatic Bookshelf, PragProg and the linking *g* device are trademarks of The Pragmatic Programmers, LLC.

Every precaution was taken in the preparation of this book. However, the publisher assumes no responsibility for errors or omissions, or for damages that may result from the use of information (including program listings) contained herein.

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at http://pragprog.com.

The team that produced this book includes:

John Osborn (editor)
Candace Cunningham (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Printed in the United States of America.
ISBN-13: 978-1-937785-33-8
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—July 2013

This book is dedicated to the memory of my father, Irving Carroll (zt"l). He set me on the road to becoming a math geek, which is why this book exists. More importantly, he showed me, by example, how to be a mensch: by living honestly, with compassion, humor, integrity, and hard work.

16

Cantor's Diagonalization: Infinity Isn't Just Infinity

Set theory is unavoidable in the world of modern mathematics. Math is taught using sets as the most primitive building block. Starting in kindergarten, children are introduced to mathematical ideas using sets! Since we've always seen it presented that way, it's natural that we think about set theory in terms of foundations. But in fact, when set theory was created, that wasn't its purpose at all. Set theory was created as a tool for exploring the concept of infinity.

Set theory was invented in the nineteenth century by a brilliant German mathematician named Georg Cantor (1845–1918). Cantor was interested in exploring the concept of infinity and, in particular, trying to understand how infinitely large things could be compared. Could there possibly be *multiple* infinities? If there were, how could it make sense for them to have different sizes? The original purpose of set theory was as a tool for answering these questions.

The answers come from Cantor's most well-known result, known as *Cantor's diagonalization*, which showed that there were at least two different sizes of infinity: the size of the set of natural numbers and the size of the set of real numbers. In this chapter, we're going to look at how Cantor defined set theory and used it to produce the proof. But before we can do that, we need to get an idea of what set theory is.

Sets, Naively

What Cantor originally invented is now known as *naive set theory*. In this chapter, we'll start by looking at the basics of set theory using naive set theory roughly the way that Cantor defined it. Naive set theory is easy to understand, but as we'll see in Section 16.3, *Don't Keep It Simple, Stupid*, on page 15, it's got some problems. We'll see how to solve those problems in the next chapter; but for now, we'll stick with the simple stuff.

A *set* is a collection of things. It's a very limited sort of collection where you can only do one thing: ask if an object is in it. You can't talk about which object comes first. You can't even necessarily list all of the objects in the set. The only thing you're guaranteed to really be able to do is ask if specific objects are in it.

The formal meaning of sets is simple and elegant: if an object is a member of a set S, then there's a predicate P_S, where an object o is a member of S (written $o \in S$) if and only if $P_S(o)$ is true. Another way of saying that is that a set S is a collection of things that all share some property, which is the defining property of the set. When you work through the formality of what a property means, that's just another way of saying that there's a predicate. For example, we can talk about the set of natural numbers: the predicate *IsNaturalNumber(n)* defines the set.

Set theory, as we can see even from the first definition, is closely intertwined with first-order predicate logic. In general, the two can form a nicely closed formal system: sets provide objects for the logic to talk about, and logic provides tools for talking about the sets and their objects. That's a big part of why set theory makes such a good basis for mathematics—it's one of the simplest things that we can use to create a semantically meaningful complete logic.

I'm going to run through a quick reminder of the basic notations and concepts of FOPL; for more details, look back at Part IV, *Logic*, on page ?.

In first-order predicate logic, we talk about two kinds of things: *predicates* and *objects*. Objects are the things that we can reason about using the logic; predicates are the things that we use to reason about objects.

A *predicate* is a statement that says something about some object or objects. We'll write predicates as either uppercase letters or as words starting with an uppercase letter (*A*, *B*, *Married*), and we'll write objects in quotes. Every predicate is followed by a list of comma-separated objects (or variables representing objects).

One very important restriction is that *predicates are not objects*. That's why this is called *first-order* predicate logic: you can't use a predicate to make a statement about another predicate. So you can't say something like *Transitive(GreaterThan)*: that's a second-order statement, which isn't expressible in first-order logic.

We can combine logical statements using *and* (written $\wedge$) and *or* ($\vee$). We can negate a statement by prefixing it with *not* (written $\neg$). And we can introduce a variable to a statement using two logical quantifiers: for all possible values ($\forall$), and for at least one value ($\exists$).

When you learned about sets in elementary school, you were probably taught about another group of operations that seemed like primitives. In fact, they aren't really primitive: The only things that we need to define naive set theory is the one definition we gave! All of the other operations can be defined using FOPL and membership. We'll walk through the basic set operations and how to define them.

The basics of set theory give us a small number of simple things that we can say about sets and their members. These also provide a basic set of primitive statements for our FOPL:

Subset

$$S \subseteq T$$

> S is a subset of T, meaning that all members of S are also members of T. Subset is really just the set theory version of implication: if S is a subset of T, then in logic, $S \Rightarrow T$.
>
> For example, let's look at the set N of natural numbers and the set N_2 of even natural numbers. Those two sets are defined by the predicates *IsNatural(n)* and *IsEvenNatural(n)*.

When we say that N_2 is a subset of N, what that means is $\forall x: IsEvenNatural(x) \Rightarrow IsNatural(x)$.

Set Union

$$A \cup B$$

Union combines two sets: the members of the union are all of the objects that are members of either set. Here it is in formal notation:

$$x \in (A \cup B) \lneqq x \in A \vee x \in B$$

The formal definition also tells you what union means in terms of logic: union is the logical *or* of two predicates.

For example, if we have the set of even naturals and the set of odd naturals, their union is the set of objects that are either even naturals or odd naturals: an object x is in the union $(EvenNatural \cup OddNatural)$ if either $IsEvenNatural(x)$ or $IsOddNatural(x)$.

Set Intersection

$$A \cap B$$

The intersection of two sets is the set of objects that are members of both sets. Here it is presented formally:

$$x \in A \cap B \lneqq x \in A \wedge x \in B$$

As you can see from the definition, intersection is the set equivalent of logical *and*.

For example, $EvenNatural \cap OddNatural$ is the set of numbers x where $EvenNatural(x) \wedge OddNatural(x)$. Since there are no numbers that are both even and odd, that means that the intersection is empty.

Cartesian Product

$$A \times B$$
$$(x, y) \in A \times B \lneqq x \in A \wedge y \in B$$

Finally, within the most basic set operations, there's one called the *Cartesian product*. This one seems a bit weird, but it's really pretty fundamental. It's got two purposes: first, in practical terms, it's the operation that lets us create ordered pairs, which are the basis of how we can create virtually everything that we want using sets. In purely theoretical terms, it's the way that set theory expresses the concept of a predicate that takes more than one parameter. The Cartesian product of two sets *S* and *T* consists of a set of *pairs*, where each pair consists of one element from each of the two sets.

For example, in 12, *Mr. Spock Is Not Logical*, on page ?, we defined a predicate *Parent(x, y)*, which meant that *x* is a parent of *y*. In set theory terms, *Parent* is a set of *pairs* of people. So *Parent* is a subset of the values from the Cartesian product of the set of people with itself. *(Mark, Rebecca)* ∈ *Parent*, and *Parent* is a predicate on the set *Parent* × *Parent*.

That's really the heart of set theory: set membership and the linkage with predicate logic. It's almost unbelievably simple, which is why it's considered so darned attractive by mathematicians. It's hard to imagine how you could start with something simpler.

Now that you understand how simple the basic concept of a set is, we'll move on and see just how deep and profound that simple concept can be by taking a look at Cantor's diagonalization.

Cantor's Diagonalization

The original motivation behind the ideas that ended up growing into set theory was Cantor's recognition of the fact that there's a difference between the size of the set of natural numbers and the size of the set of real numbers. They're both infinite, but they're not the same!

Cantor's original idea was to abstract away the details of numbers. Normally when we think of numbers, we think of them as being things that we can do arithmetic with, things that can be compared and manipulated in all sorts of ways. Cantor said that for understanding how many numbers there were, none of those properties or arithmetic operations were needed. The only thing that mattered was that a kind of number like the natural numbers was a collection of objects. What mattered is which objects were parts of which collection. He called this kind of collection a *set*.

Using sets allowed him to invent a new way of defining a way of measuring size that didn't involve counting. He said that if you can take two sets and show how to create a mapping from every element of one set to exactly one element of the other set, and if this mapping didn't miss any elements of either set (a *one-to-one mapping* between the two sets), then those two sets are the same size. If there is no way to make a one-to-one mapping without leaving out elements of one set, then the set with extra elements is the *larger* of the two sets.

For example, if you take the set *{1, 2, 3}*, and the set *{4, 5, 6}*, you can create several different one-to-one mappings between the two sets: for example, $\{1 \Rightarrow 4, 2 \Rightarrow 5, 3 \Rightarrow 6\}$, or $\{1 \Rightarrow 5, 2 \Rightarrow 6, 3 \Rightarrow 4\}$. The two sets are the same size, because there is a one-to-one mapping between them.

In contrast, if you look at the sets *{1, 2, 3, 4}* and *{a, b, c}*, there's no way that you can do a one-to-one mapping without leaving out one element of the first set; therefore, the first set is larger than the second.

This is cute for small, finite sets like these, but it's not exactly profound. Creating one-to-one mappings between finite sets is laborious, and it always produces the same results as just counting the number of elements in each set and comparing the counts. What's interesting about Cantor's method of using mappings to compare the sizes of sets is that mappings can allow you to compare the sizes of infinitely large sets, which you *can't* count!

For example, let's look at the set of natural numbers (N) and the set of even natural numbers (N_2). They're both infinite sets. Are they the same size? Intuitively, people come up with two different answers for whether one is larger than the other.

1. Some people say they're both infinite, and therefore they must be the same size.

2. Other people say that the even naturals must be half the size of the naturals, because it skips every other element of the naturals. Since it's skipping, it's leaving out elements of the naturals, so it must be smaller.

Which is right? According to Cantor, both are wrong. Or rather, the second one is completely wrong, and the first is right for the wrong reason.

Cantor says that you can create a one-to-one mapping between the two:

$$\{(x \gg y) : x, y \in N, y = 2 \times x\}$$

Since there's a one-to-one mapping, that means that they're the same size—they're not the same size because they're both infinite, but rather because there is a one-to-one mapping between the elements of the set of natural numbers and the elements of the set of even natural numbers This shows us that some infinitely large sets are the same size as some other infinitely large sets. But are there infinitely large sets whose sizes are *different*? That's Cantor's famous result, which we're going to look at.

Cantor showed that the set of real numbers is larger than the set of natural numbers. This is a very surprising result. It's one that people struggle with because it *seems* wrong. If something is infinitely large, how can it be smaller than something else? Even today, almost 150 years after Cantor first published it, this result is *still* the source of much controversy (see, for example, *this famous summary [Hod98]*.) Cantor's proof shows that no matter what you do, you can't create a one-to-one mapping between the naturals and the reals without missing some of the reals; and therefore, the set of real numbers is larger than the set of naturals.

Cantor showed that every mapping from the naturals to the reals *must* miss at least one real number. The way he did that is by using something called a *constructive proof*. This proof contains a procedure, called a *diagonalization*, that takes a purported one-to-one mapping from the naturals to the reals and generates a real number that is missed by the mapping. It doesn't matter what mapping you use: given *any* one-to-one mapping, it will produce a real number that isn't in the mapping.

We're going to go through that procedure. In fact, we're going to show something even stronger than what Cantor originally did. We're going to show that there are more real numbers between zero and one than there are natural numbers!

Cantor's proof is written as a basic proof by contradiction. It starts by saying "Suppose that there is a one-to-one mapping from the natural numbers to the real numbers between zero and one." Then it shows how to take that supposed mapping and use it to construct a real number that is missed by the mapping.

Example: Prove that there are more real numbers between 0 and 1 than there are natural numbers.

1. Suppose that we can create a one-to-one correspondence between the natural numbers and the reals between 0 and 1. What that would mean is that there would be a total one-to-one function R from the natural numbers to the reals. Then we could create a complete list of all of the real numbers: $R(0)$, $R(1)$, $R(2)$, ...

2. If we could do that, then we could also create another function, D (for digit), where $D(x,y)$ returns the yth digit of the decimal expansion of $R(x)$. The D that we just created is effectively a table where every row is a real number and every column is a digit position in the decimal expansion of a real number. $D(x,3)$ is the third digit of the binary expansion of x.

   For example, if $x = 3/8$, then the decimal expansion of x is 0.125. Then $D(3/8,1) = 1$, $D(3/8,2) = 2$, $D(3/8,3) = 5$, $D(3/8,4) = 0$, ...

3. Now here comes the nifty part. Take the table for D and start walking down the diagonal. We're going to go down the table looking at $D(1,1)$, $D(2,2)$, $D(3,3)$, and so on. And as we walk down that diagonal, we're going to write down digits. If the $D(i, i)$ is 1, we'll write a 6. If it's 2, we'll put 7; 3, we'll put 8; $4 \Rightarrow 9$; $5 \Rightarrow 0$; $6 \Rightarrow 1$; $7 \Rightarrow 2$; $8 \Rightarrow 3$; $9 \Rightarrow 4$; and $0 \Rightarrow 5$.

4. The result that we get is a series of digits; that is, a decimal expansion of some number. Let's call that number T. T is *different* from every row in D in at least one digit—for the ith row, T is different at digit i. There's no x where $R(x) = T$.

But T is clearly a real number between 0 and 1: the mapping can't possibly work. And since we didn't specify the structure of the mapping, but just assumed that there was one, that means that there's no possible mapping that will work. This construction will always create a counterexample showing that the mapping is incomplete.

5. Therefore, the set of all real numbers between 0 and 1 is *strictly larger* than the set of all natural numbers.

That's Cantor's diagonalization, the argument that put set theory on the map.

Don't Keep It Simple, Stupid

There's an old mantra among engineers called the KISS principle. KISS stands for "Keep it simple, stupid!" The idea is that when you're building something useful, you should make it as simple as possible. The more moving parts something has, the more complicated corners it has, the more likely it is that an error will slip by.

Looked at from that perspective, naive set theory looks great. It's so beautifully simple. What I wrote in the last section was the entire basis of naive set theory. It looks like you don't need any more than that!

Unfortunately, set theory in practice needs to be a lot more complicated. In the next section, we'll look at an axiomatization of set theory, and yeah, it's going be a whole lot more complicated than what we did here! Why can't we stick with the KISS principle, use naive set theory, and skip that hairy stuff?

The sad answer is, naive set theory doesn't work.

In naive set theory, *any* predicate defines a set. There's a collection of mathematical objects that we're reasoning about, and from those, we can form sets. The sets themselves are also objects that we can reason about. We did that a bit already by defining things like subsets, because a subset is a relation between sets.

By reasoning about properties of sets and relations between sets, we can define sets of sets. That's important, because sets of sets are at the heart of a lot of the things that we do with set theory. As we'll see later, Cantor came up with a way of modeling numbers using sets where each number is a particular kind of structured set.

If we can define sets of sets, then using the same mechanism, we can create infinitely large sets of sets, like "the set of sets with infinite cardinality," also known as the set of infinite sets. How many sets are in there? It's clearly infinite. Why? Here's a sketch: if I take the set of natural numbers, it's infinite. If I remove the number 1 from it, it's still infinite. So now I have two infinite sets: the natural numbers, and the natural numbers omitting 1. I can do the same for every natural number, which results in an infinite number of infinite sets. So the set of sets with infinite cardinalities clearly has infinite cardinality! Therefore, it's a member *of itself*!

If I can define sets that contain themselves, then I can write a predicate about self-inclusion and end up defining things like the set of all sets that include themselves. This is where trouble starts to crop up: if I take that set and examine it, does it include itself? It turns out that there are *two sets* that match that predicate! There's one set of all sets that include themselves that includes itself, and there's another set of all sets that include themselves that does *not* include itself.

A predicate that *appears* to be a proper, formal, unambiguous statement in FOPL turns out to be ambiguous when used to define a set. That's not fatal, but it's a sign that there's something funny happening that we should be concerned about.

But now, we get to the trick. If I can define the set of all sets that contain themselves, I can also define the set of all sets that do *not* contain themselves.

And that's the heart of the problem, called *Russell's paradox*. Take the set of all sets that do *not* include themselves. Does it include itself?

Suppose it does. If it does, then by its definition, it *cannot* be a member of itself.

So suppose it doesn't. Then by its definition, it *must be* a member of itself.

We're trapped. No matter what we do, we've got a contradiction. And in math, that's deadly. A formal system that allows us to derive a contradiction is completely useless. One error like that, allowing us to derive just one contradiction, means that every result we ever discovered or proved in the system is worthless! If there's a single contradiction possible anywhere in the system, then every statement—whether genuinely true or false—is provable in that system!

Unfortunately, this is pretty deeply embedded in the structure of naive set theory. Naive set theory says that *any* predicate defines a set, but we can define predicates for which there is no valid model, for which there is no possible set that consistently matches the predicate. By allowing this kind of inconsistency, naive set theory itself is inconsistent, and so naive set theory needs to be discarded. What we need to do to save set theory at all is build it a better basis. That basis should allow us to do all of the simple stuff that we do in naive set theory, but do it without permitting contradictions. In the next section, we'll look at one version of that, called Zermelo-Frankel set theory, that defines set theory using a set of strong axioms and manages to avoid these problems while preserving what makes set theory valuable and beautiful.