Chapter 1

Sets and Logic

The use of math usually falls into two categories:

- *Developing mathematical tools that let us make better predictions.* This is how engineers and scientists use math. It is usually referred to as *applied math*.
- *Creating interesting statements and proving them to be true or false.* This is known as *pure math*.

Many mathematical ideas start out as pure math, and eventually become useful. For example, the field of number theory is devoted to proving things about prime numbers. The mathematicians who created number theory were certain that it could never be used for any practical purpose. After a century or two, number theory was used as the basis of most cryptography systems.

Conversely, some ideas start out as a ”rule of thumb” that engineers use, and are eventually rigorously defined and proven.

This course tends to emphasize applied math, but you should know something about the tools of pure math.

You can think of all the mathematical proofs as a tree. Each proof proves some statement true. To do this, the proof uses logic and statements that were proven true by other truths. So, the tree is built from the bottom up. However, the tree has to have a bottom. At the very bottom of the tree are some statements that we just accept as true without proof. These are known as *axioms*.

All of modern mathematics can be built from:

- A short list of axioms.
- A few rules of logic.

There have been several efforts to codify a small but complete axiomatic system. The most popular one is known as *ZFC*. ”Z” is for the Ernst Zermelo, who did most of the work. ”F” is for Abraham Fraenkel, who tidied up a couple of things. ”C” is for The Axiom of Choice. As a community, mathematicians debate whether the Axiom of Choice should be an axiom; we get a couple of strange results if we include it in the system. If we do not, there are a few obviously useful ideas that we can’t prove true.

ZFC has 10 axioms. We simply accept these 10 statements as true, and all the proofs of modern mathematics can be extrapolated from them. The 10 axioms are all stated in terms of sets.

1.1 Sets

A *set* is a collection. For example, you might talk about the set of odd numbers greater than 5, or the set of all protons in the universe.

We have a notation for sets. For example, here is how we define S to be the set containing 1, 2, and 3:

$$S = \{1, 2, 3\}$$

We say that 1, 2, and 3 are *elements* of the set S. (Sometimes we will also use the word ”member”)

If you want to say ”2 is an element of the set S” in mathematical notation, it is done like this:

$$2 \in S$$

If you want to say ”5 is *not* an element of the set S”, it looks like this:

$$5 \notin S$$

We have notation for a few sets that we use all the time:

Set	Symbol
The empty set	$\emptyset$
Natural numbers	$\mathbb{N}$
Integers	$\mathbb{Z}$
Rational numbers	$\mathbb{Q}$
Real numbers	$\mathbb{R}$
Complex numbers	$\mathbb{C}$

The empty set is the set that contains nothing. It is also sometimes called *the null set*.

Often, when we define a set, we start with one of these big sets and say ”The set I'm talking about is the members of the big set, but only the one for which this statement is true". For example, if you wanted to talk about all the integers greater than or equal to -5, you could do it like this:

$$A = \{x \in \mathbb{Z} \mid x \geq -5\}$$

When you read this aloud, you say "A is the set of integers x where x is greater than or equal to negative 5."

1.1.1 And and Or

Sometimes you need the members to satisfy two conditions; for this, we use "and":

$$A = \{x \in \mathbb{Z} \mid x > -5 \text{ and } x < 100\}$$

This is the set of integers that are greater than -5 *and* less than 100. In this book, we usually just write "and", but if you do a large amount of set and logic work, you will use the symbol $\wedge$:

$$A = \{x \in \mathbb{Z} \mid (x > -5) \wedge (x < 100)\}$$

Sometimes, you want a set that satisfies at least one of two conditions. For this, you use "or":

$$A = \{x \in \mathbb{Z} \mid x < -5 \text{ or } x > 100\}$$

These are the number that are less than -5 or greater than 100. Once again, there is a symbol for this:

$$A = \{x \in \mathbb{Z} \mid (x < -5) \vee (x > 100)\}$$

1.1.2 How simple are sets?

Sets are so simple that some questions just don't make any sense:

- "What is the first item in the set?" makes no sense to a mathematician. Sets have no order.

- "How many times does the number 6 appear in the set?" makes no sense. 6 is a member, or it is not.

1.1.3 Subsets

If every member of set A is also in set B, we say that "A is a subset of B."

For example, if $A = \{1, 4, 5\}$ and $B = \{1, 2, 3, 4, 5, 6\}$, then A is a subset of B. There is a symbol for this:

$$A \subseteq B$$

Remember the table of commonly used sets? We can arrange them as subsets of each other:

$$\emptyset \subseteq \mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q} \subseteq \mathbb{R} \subseteq \mathbb{C}$$

Note that that subsets have the transitive property: $\mathbb{N} \subseteq \mathbb{Z} \subseteq \mathbb{Q}$ thus $\mathbb{N} \subseteq \mathbb{Q}$

Note that if A and B have the same elements, $A \subseteq B$ *and* $B \subseteq A$. In this case, we say that the two sets are equal.

We also have a symbol for "is not a subset of": $A \not\subseteq B$

1.1.4 Union and Intersection of Sets

If you have two sets A and B, you might want to say "Let C be the set containing element that are in *either* A or B." We say that C is the *union* of A and B. There is notation for this too:

$$C = A \cup B$$

For example, if $A = \{1, 3, 4, 9\}$ and $B = \{3, 4, 5, 6, 7, 8\}$, then $A \cup B = \{1, 3, 4, 5, 6, 7, 8, 9\}$.

You also want to say "Let C be the set containing elements that are in *both* A and B." We say that C is the *intersection* of A and B. There is notation for this too:

$$C = A \cap B$$

For example, if $A = \{1, 3, 4, 9\}$ and $B = \{3, 4, 5, 6, 7, 8\}$, then $A \cap B = \{3, 4\}$.

1.1.5 Venn Diagrams

When discussing sets, it is often helpful to have a Venn diagram to look at. Venn diagrams represent sets as circles. For example, the sets A and B above could look like this:

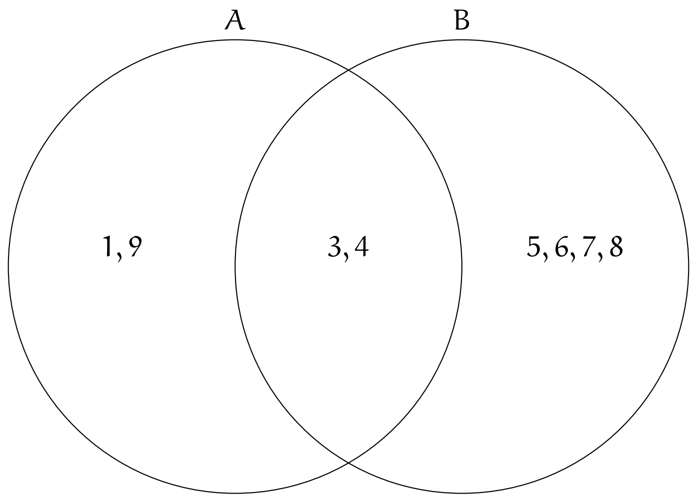

Figure 1.1: A Venn Diagram of the set of numbers $\{1, 9, 3, 4, 5, 6, 7, 8\}$

It makes it easy to see that A and B have a non-empty intersection, but they are not subsets of each other.

Often we won't even show the individual elements. For example, in the universe of all polygons, some rectangles are squares. Here's the Venn diagram:

As the combinations get more complex, we sometimes use shading to indicate what part we are talking about. For example, imagine we wanted all the rectangles with area greater than 5.0 that are not squares. The diagram might look like this:

1.2 Logic

We use a lot of logic in set theory. For example, the shaded region above represents all the polygons for which all the following are true:

- It is a rectangle.
- It is *not* a square.

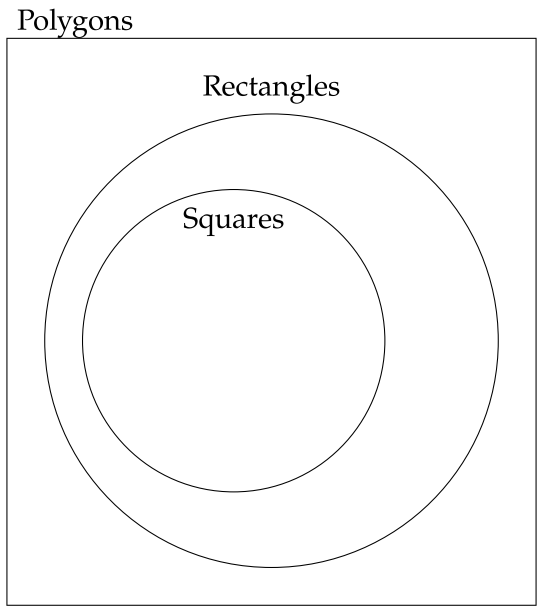

Figure 1.2: A Venn Diagram of polygons, rectangles, and squares.

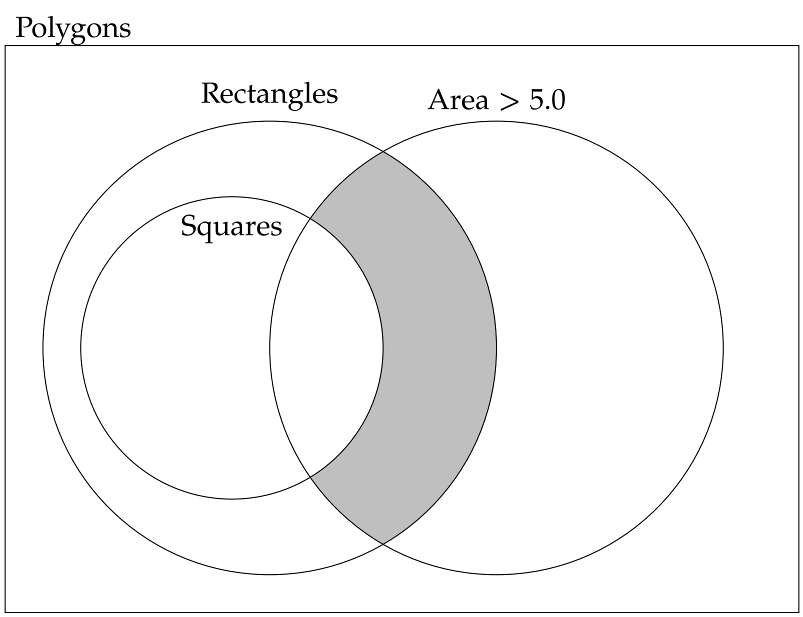

Figure 1.3: A Venn Diagram of the polygons, rectangle, and square number, and rectangles that have an area greater than 5.0 that are not squares.

- It has an area greater than 5.0.

1.3 Implies

In logic, we will often say "a implies b". That means "If the statement a is true, the statement b is also true." For example: "p is a square" implies "p is a rectangle".

There is notation for this: an arrow in the direction of the implication.

$$p \text{ is a square} \implies p \text{ is a rectangle}$$

Notice that implication has a direction: "p is a rectangle" does *not* imply "p is a square".

Implications can be chained together: If $A \implies B$ and $B \implies C$, then $A \implies C$.

1.4 If and Only If

If the implication goes both ways, we use "if and only if". This means the two conditions are equivalent. For example: "n is even if and only if there exists an integer m such that $2m = n$".

There is a notation for this too:

$$p \text{ is even} \iff \text{there exists an integer } m \text{ such that } 2m = n$$

There is even notation for "there exists". It is a backwards capital E:

$$p \text{ is even} \iff \exists\, m \in \mathbb{Z} \text{ such that } 2m = n$$

1.5 Not

The not operation flips the truth of an expression:

- If a is true, $\text{not}(a)$ is false.
- If a is false, $\text{not}(a)$ is true.

We sometimes talk about "notting" or "negating" a value. We won't use it much, but there is a symbol for this: $\neg$.

We might create a *logic table* for negation that shows all the possible values and their negation:

A	$\neg$A
F	T
T	F

This table says "If A is false, $\neg$A is true. If A is true, $\neg$A is false."

Most logic tables are for operations that take more than one input. For example, this logic table shows the values for and-ing and or-ing:

A	B	A and B	A or B
F	F	F	F
F	T	F	T
T	F	F	T
T	T	T	T

Notice that we have to enumerate all possible combinations of the inputs of A and B.

When a variable like A can only take two possible values, we say it is a *boolean* variable. (George Bool did important work in this area.)

Exercise 1 Logic Table

Make a logic table that enumerates all possible combinations of boolean variables A and B and shows the value of the two following expressions:

- $\neg$ (A or B)
- ($\neg$A) and ($\neg$B)

Working Space

Answer on Page 15

1.6 Cardinality

Informally, the *cardinality* of a set is the number of elements it contains. So, $\{1, 3, 5\}$ has a cardinality of 3. The null set has a cardinality of zero.

Things get a little trickier if a set is infinite. We say two infinite sets A and B have the same cardinality if there is some mapping that pairs every member of A with a member of B and mapping that pairs every member of B with a member of A.

For example, the set of all natural numbers is $\mathbb{N} = \{1, 2, 3, 4, \ldots\}$. The set of all even numbers is $\{2, 4, 6, 8, \ldots\}$. These have the same cardinality because we can pair each natural number n with an even number $2n$.

1.7 Complement of a Set

Most sets exist in a particular universe, for example you might talk about the even numbers as a set in the integers. You can then talk about the set's *complement*: the set of everthing else. For example, the complement of the even numbers (inside the integers) is the odd numbers.

If you have a set A, its complement is usually denoted by A'.

See the Venn Digram shown in Figure 1.4.

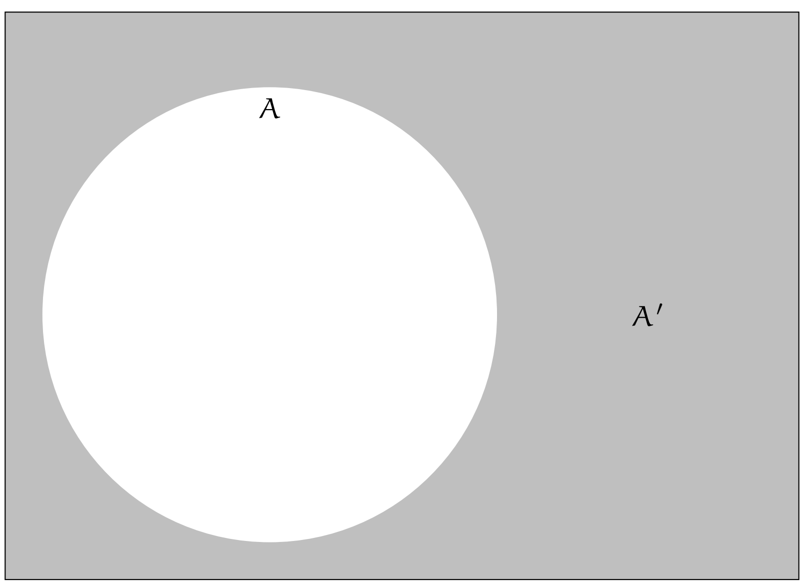

Figure 1.4: The complement of A is A'.

Outside of logic, the concept of a set

1.8 Subtracting Sets

If you have sets $A = \{1, 2, 3, 4\}$ and $B = \{1, 4\}$, it makes sense to subtract B from A by removing 1 and 4 from A.

If A and B are sets, we define $A - B$ to be $A \cap B'$. Take a second to look at this diagram and convince yourself that the white region represents $A - B$ and that it is the same as $A \cap B'$.

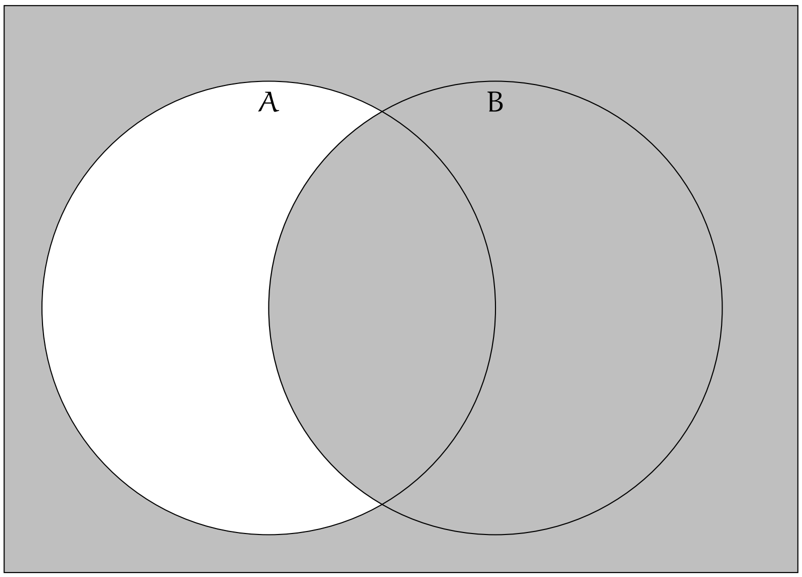

Figure 1.5: The subtraction of sets A and B.

1.9 Power Sets

It is not uncommon to have a set whose elements are also sets. For example, you might have the set that contains the following two sets: $\{1, 2, 3\}$ and $\{2, 3, 4\}$. You might write it like this: $\{\{1, 2, 3\}, \{2, 3, 4\}\}$. (Note that this set has a cardinality of 2 — it has two members that are sets.)

Given any set A, you can construct its *power set*, which is the set of all subsets of A. For example, if you have a set $\{1, 2, 3\}$, its power set is $\{\{1, 2, 3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1\}, \{2\}, \{3\}, \emptyset\}$.

If a set has n elements, its power set has 2^n elements. Note that the last element must be the empty set, because there are no subsets of an empty set. Recall that order does not matter in sets, so $\{1, 2\}$ and $\{2, 1\}$ are considered the same.

1.10 Booleans in Python

In Python, we can have variables hold boolean values: `True` and `False`. We also have operators: `not`, `and`, and `or`.

For example, you could find out what the expression "a and not b" is if both variables are false like this:

```
a = False
b = False
result = a or not b
print(f"a={a}, b={b}, a or not b = {result}")
```

This would print out:

```
a=False, b=False, a or not b = True
```

What if you wanted to try all possible values for `a` and `b`? You could use `itertools`.

```
import itertools

all_combos = itertools.product([False, True], repeat=2)
for (a, b) in all_combos:
    result = a or not b
    print(f"a={a}, b={b}: a or not b = {result}")
```

Type it in and run it. You should get the whole logic table:

```
a=False, b=False: a or not b = True
a=False, b=True: a or not b = False
a=True, b=False: a or not b = True
a=True, b=True: a or not b = True
```

If you had three inputs into the expression, your truth table would have eight entries. For example, if you wanted to know the truth table for $a and not(b and c)$, here is the code:

```
all_combos = itertools.product([False, True], repeat=3)
for (a, b, c) in all_combos:
    result = a and not(b and c)
    print(f"a={a}, b={b}, c={c}: a and not (b and c) = {result}")
```

Type it in and run it. You should get:

```
a=False, b=False, c=False: a and not (b and c) = False
```

```
a=False, b=False, c=True: a and not (b and c) = False
a=False, b=True, c=False: a and not (b and c) = False
a=False, b=True, c=True: a and not (b and c) = False
a=True, b=False, c=False: a and not (b and c) = True
a=True, b=False, c=True: a and not (b and c) = True
a=True, b=True, c=False: a and not (b and c) = True
a=True, b=True, c=True: a and not (b and c) = False
```

1.11 The Contrapositive

Here is a statement with an implication: “If it has rained in the last hour, the grass is wet.”

This is *not* equivalent to “If the grass is wet, it has rained in the last hour.” (After all, the sprinkler may be running.)

However, it is exactly equivalent to its *contrapositive*: “If the grass is not wet, it has not rained in the last hour.”

The rule can be written using symbols:

$$(A \implies B) \iff (\neg B \implies \neg A)$$

1.12 The Distributive Property of Logic

Many ideas from integer arithmetic have analogues in boolean arithmetic. For example, there is a distributive property for booleans. These two expressions are equivalent:

- A and $(B$ or $C)$
- $(A$ and $B)$ or $(A$ and $C)$

So are these:

- A or $(B$ and $C)$
- $(A$ or $B)$ and $(A$ or $C)$

1.13 Exclusive Or

The expression “a or b” is true in any of the following conditions:

- a is True and b is False.
- a is False and b is True.
- Both a and b are True.

Sometimes engineers need a way to say "Either a or b is true, but not both." For this, we use *exclusive OR* (or XOR). You may see XOR written symbolically as $\oplus$ or just used as XOR. indexXOR

Here, then, is the logic table for XOR

A	B	XOR(a,b)
F	F	F
F	T	T
T	F	T
T	T	F

In Python, Logical XOR is done using `!=`:

```
just_one = (a != b)
```

(Take 10 seconds to confirm that this is the same as the logic table above.)

This is a draft chapter from the Kontinua Project. Please see our website (https://kontinua.org/) for more details.

Appendix A

Answers to Exercises

Answer to Exercise 1 (on page 8)

A	B	$\text{not}\,(A \text{ or } B)$	$(\text{ not } A) \text{ and } (\text{ not } B)$
F	F	T	T
F	T	F	F
T	F	F	F
T	T	F	F

Notice that the two expressions are equivalent!

DeMorgan's Rule says "not $(A \text{ or } B)$" is equivalent to "$(\text{not } A)$ and $(\text{not } B)$".

It also says "not $(A \text{ and } B)$" is equivalent to "$(\text{not } A)$ or $(\text{not } B)$".

Index