MA132: Foundations Lecture Notes

Jo Evans

2024-09-20

Contents

Chapter 1

About

These are the 2024-25 lecture course for the Warwick undergraduate course MA132: Foundations. The material up to chapter 7 is also relevant to joint degree students taking Sets and Numbers.

These notes are heavily based on previous notes of Saul Schleimer and Dave Wood, though all mistakes are my own!

Please send me any typos (or possible typos) that you find. And be on the look out for typos.

1.1 How to use these notes

These notes are designed to generate both a html version (which should work well with screen readers) and a latex generated pdf and epub (which will work better if you want to print the notes). You can access the pdf/epub by pressing the download button in the top right (little down arrow).

Everything you need to know for the exam and future courses should be contained in these notes. There will also be examples, some discussions and some non-examinable sections.

Similarly, almost everything you need to know will be written by me on the board in lectures. There may be some exceptions which I will warn you about!

There are three main ways that undergraduates typically use printed lecture notes:

1. They take their own notes during the lecture (or take none) and refer to these notes outside the lecture to supplement those notes, to check things, when doing exercises, or during revision.

2. They print out a copy of these notes/or have them available as a pdf on a tablet and annotate them during the lectures.

3. They take a copy of these notes to lectures to refer to but don't annotate them.

Chapter 2

Sets and Functions

2.1 What is a set?

We begin with sets which are one of the basic objects of mathematics. Set theory becomes very complicated very quickly when you begin to explore the subtleties. We will discuss some of the pitfalls and paradoxes in a non-examinable section later but first we focus on when things are simple.

Definition 2.1 (set)**.** A *set* is a collection of mathematical objects.

To make sense of this definition we need to consider some examples of things that are and are not sets.

Example 2.1. The function $y = x^2, x \in \mathbb{R}$ is *not* a set (it's a function as we wrote). However we can form a *set* of the form

$$\{(x, y) \, : \, x \in \mathbb{R}, y \in \mathbb{R}, y = x^2\}.$$

Example 2.2. The natural numbers $\mathbb{N} = \{0, 1, 2, 3, 4, ... \}$ is a set. We can see in this and the previous example that you can often recognise sets because they have curly brackets around them. However, this is not foolproof. We can write this set as $\mathbb{N}$ and there are no curly brackets.

If you haven't seen it before remember this notation for the natural numbers. It will come up a lot!

This example also has another bit of common mathematical notation in its. The set of three dots at the end of the sequence of numbers. This indicates that the sequence will continue as you would expect it to.

Example 2.3. Sets can also have a finite number of elements. For example the following set $\{12\}$ which contains only the integer 12. As with the first example with a function, we make a distinction between the object which is the integer 12 and a set that contains only the integer 12.

Example 2.4. All the examples of sets above involve mathematical objects which are numbers (or pairs of numbers in the first example). We aren't limited to this. We might consider the set of all polynomials with integer coefficients (where all the elements are functions) or the set of all sequences of real numbers tending to zero (where all the elements are sequences). You can also consider sets with a mixture of different types of elements. e.g.

$$\{4, (\pi, \pi^2), \{1/n : n \in \mathbb{N}, n \neq 0\}, \text{the function } f(x) = x^2\}.$$

Here notice that one of the elements of this set is a set itself. This is perfectly possible.

We need to be able to talk and write about sets. We often give sets names (usually a letter) and we write

$$A = \{1, 2, 7\}.$$

We then want to be able to say whether something is or isn't in the set so we write

$$1 \in A$$

to mean 1 is in the set A or 1 is an element of the set A. We also write

$$3 \notin A$$

to mean that 3 is not an element of the set A or that 3 isn't in A.

Definition 2.2. There are some important sets which have their own symbols and names. You have probably met them before:

- The natural numbers, $\mathbb{N} = \{1, 2, 3, \dots\}$,
- The integers, $\mathbb{Z} = \{\dots, -3, -2, -1, 0, 1, 2, 3, \dots\}$,
- The rationals, $\mathbb{Q} = \{p/q : p \in \mathbb{Z}, q \in \mathbb{N}\}$,
- The real numbers, $\mathbb{R} = (-\infty, \infty)$,
- The complex numbers, $\mathbb{C} = \{x + iy : x \in \mathbb{R}, y \in \mathbb{R}\}$ where i is the complex unit.

Definition 2.3 (equality of sets/axiom of extension)**.** Two sets are equal (the same) if they have exactly the same elements. We call this the axiom of extension. We can write it in formal language as: if for every $x \in A$ we have $x \in B$ and for every $y \in B$ we have $y \in B$ then $A = B$.

Remark. It might seem obvious at this point that any two sets with the same element are the same. However there are two important ways this comes up.

In a proof we might write a set in two very different ways for example $[0, \infty) = \{x \in \mathbb{R} : \text{there exists} \, y \in \mathbb{R} \, \text{s.t.} \, y^2 = x\}$.

We might end up writing a set in a way that means some element appears in the representation multiple times e.g. $\{0\}$ and $\{0, 0\}$. The axiom of extension makes it clear that these are both the same set. It also tells us that there aren't multiple different sets containing only the element 0 there is just one the set $\{0\}$.

At this point it is useful to introduce some notation that you may or may not have seen before. We will talk about this notation more thoroughly in the section on proof. In my opinion it is useful to see all this notation a bit before we think about it too thoroughly.

Definition 2.4. There are several shorthand notations used in maths and particularly in logic. Some are more common than others and it is always okay and often wise not to overuse symbols. We might use the following:

-We use the symbol $\forall$ as a shorthand for the phrase *for every* or *for all*.

-We use the symbol $\exists$ to mean *there exists* or *there is at least one*.

-We use the symbol $\Rightarrow$ to mean *implies* and (much less often) the symbol $\Leftarrow$ to mean *is implied by*.

-We use the symbol $\Leftrightarrow$ to mean *if and only if* which we also sometimes abbreviate to iff. If and only if is a common phrase in pure maths but it might sometimes be easier to say *exactly when* to mean the same thing.

Using this we can write the axiom of extension as

$$\forall A \forall B((x \in A \Leftrightarrow x \in B) \Leftarrow (A = B)).$$

Which is a good illustration of why its often better to use words!

Given that sets are defined by their elements we sometimes need to consider that set that doesn't have any elements at all.

Definition 2.5 (the empty set (Axiom))**.** There exists a set which contains no elements. We call this the *empty set* and write it with the symbol $\emptyset$.

Remark. Here if you've been paying attention you'll notice that $\emptyset \neq \{\emptyset\}$. The first contains no elements, the second contains one element which is the empty set. You can also have $\{\emptyset, \{\emptyset\}\}$ and $\{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}\}$ and so on if you would like to reassure yourself that there exists a set containing k elements for any $k \in \mathbb{N}$ but you are unconvinced of the existence of the natural numbers. (If you are in this position you might be Bertrand Russell.)

2.2 Subsets

You might be interested in looking at only part of a set. This is called a subset.

Definition 2.6 (Subset)**.** If A and B are sets and for every $x \in A$ we have that $x \in B$ then we say A is a *subset* of B which we write

$$A \subset B.$$

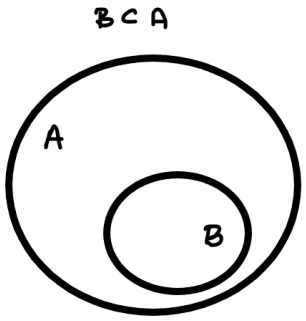


Figure 2.1: Picture showing A as a subset of B with sets indicated by circles

Example 2.5.
- 1 is *not* a subset of $\{1\}$.
- The even numbers are a subset of $\mathbb{N}$.
- $\{1\}$ is a subset of $\mathbb{N}$.
- $\{1\}$ is *not* a subset of $\{\{1\}\}$.

Remark. Sometimes when you are writing you wish to specify the set and the subset in a different order. We write $B \supset A$ and this expresses exactly the same information as $A \subset B$. When we read the expression $B \supset A$ we say B contains A or B is a superset of A.

The following is also always true

Lemma 2.1. *For any set A we have*

- $\emptyset \subset A$,
- $A \subset A$

Proof. Remember to show that $B \subset A$ we need to show that for every $y \in B$ we have that $y \in A$.

For the first statement since there are no elements of the empty set so absolutely any statement about every element of the empty set is true.

For the second statement if $x \in A$ then tautologically $x \in A$ so we have $A \subset A$. □

Following from this we have

Lemma 2.2. *For two sets A, B the following are equivalent:*

1. $A = B$,
2. $A \subset B$ *and* $B \subset A$.

Proof. Exercise! □

Definition 2.7. We call a set which contains only one element a *singleton set*.

We also define some notation

Definition 2.8. We write $[[n]] = \{k \in \mathbb{N}, k < n\}$.

2.3 Power sets and specification

Definition 2.9 (Power set (axiom))**.** Given a set A there exists another set $\mathcal{P}(A)$ called the *power set* of A which is the set of all possible subsets of A.

Example 2.6. The power set of $\{0, 1, 2\}$ is

$$\{\emptyset, \{0\}, \{1\}, \{2\}, \{0, 1\}, \{0, 2\}, \{1, 2\}, \{0, 1, 2\}\}.$$

Definition 2.10 (specification (axiom))**.** *Specification* is a way of constructing subsets of a set (we've done this a lot already). Suppose $P(A)$ is a property that an element x of A could have. Then we can define the set

$$B = \{x \in A : P(A)\}.$$

The *axiom of specification* is the set theory axiom positing that such a set exists. In this we would need a more precise notion of what a property is.

Example 2.7. The very first set we defined was defined using specification

$$\{(x, y) : x \in \mathbb{R}, y \in \mathbb{R}, y = x^2\}.$$

2.4 Functions

Definition 2.11 (function)**.** A function is comprised of three objects, a domain A which is a set, a co-domain B which is another set and a rule f which assigns an element $f(x) \in B$ to each element x of A.

We write $f : A \to B$.

Remark. This is a slightly informal definition. This is because we don't want to create an axiom saying functions exist. We are going to build functions out of more fundamental objects soon but first we want to have a useable definition.

Example 2.8. $f(x) = x^2 : \mathbb{R} \to \mathbb{R}$ is a function and technically $f(x) = x^2 : \mathbb{R} \to [0, \infty)$ is a different function.

A function can only take one value so we have to be careful when dealing with things like square roots. Similarly a function needs to take exactly one value so we also need to make sure it is defined everywhere.

Definition 2.12. If $f : A \to B$ is a function then if $y = f(x)$ we call y the *image* of x under f. We also call x a *preimage* of y under f.

Notice that an element of X can have only one image but an element of Y can have multiple or zero preimages.

Definition 2.13. The identity function on A is written $Id_A : A \to A$ and is defined by $Id_A(x) = x$.

Definition 2.14 (restriction to a subset)**.** Suppose that A and B are sets and $f : A \to B$ is a function, and suppose further that $C \subset A$. Then we can define a new function called the restriction of f to C which we write $f|_C$. This is a function with domain C and codomain B and for $x \in C$ we have $f|_C(x) = f(x)$.

Definition 2.15 (Indicator function)**.** Given a set A and a subset $B \subset A$ we can define the indicator function of B, from A to $\{0, 1\}$ by

$$1_B(x) = \begin{cases} 0 & x \notin B \\ 1 & x \in B \end{cases}$$

2.5 Properties of functions

Definition 2.16 (injectivity)**.** A function $f : A \to B$ is called *injective* if $f(x) = f(x')$ implies that $x = x'$. That is to say there are no two elements of X where $f(x)$ takes the same value, or y has at most one preimage under f.

Example 2.9. The function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2$ is not injective because for every $y > 0$ there are two possible values of x such that $x^2 = y$.

However, the function $f : [0, \infty) \to \mathbb{R}$ is injective because now for every $y \geq 0$ there is exactly one x such that $x^2 = y$ and for every $y < 0$ there are no elements x in the set such that $x^2 = y$ (so for any y in the codomain there is never more than one element x in the domain so that $x^2 = y$).

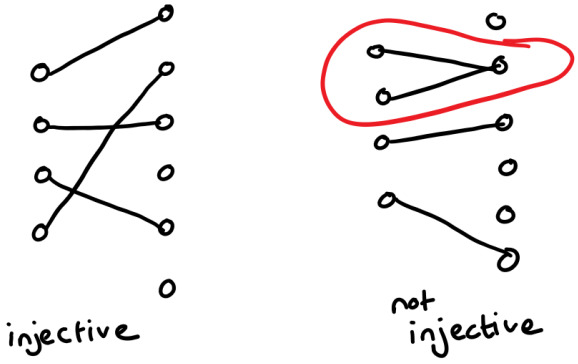


Figure 2.2: example of an injective function

Definition 2.17 (surjectivity)**.** A function $f : A \to B$ is called *surjective* if for every $y \in B$ there exists $x \in A$ with $f(x) = y$. That is to say the function f hits every element of the set B or that y has at least one preimage under x.

Example 2.10. The function $f : \mathbb{R} \to \mathbb{R}$ defined by $f(x) = x^2$ is not surjective because for $y < 0$ there is no element x of the domain so that $x^2 = y$.

However, the function $f : \mathbb{R} \to [0, \infty)$ is surjective because for every element y of the codomain we have some x in the domain with $x^2 = y$.

Definition 2.18 (bijectivity)**.** A function is called *bijective* if it is both *surjective* and *injective*.

Remark. Bijective functions are often called matchings because if $f : A \to B$ is a bijection then we *match* every element of A with an element of B.

2.6 Cardinality

If $f : A \to B$ is a bijection then that tells us something important about the relationship between A and B.

Definition 2.19 (Cardinality)**.** We say that A and B have the same *cardinality* (informally the same size) if there exists a bijection between A and B. We often write $|A| = |B|$.

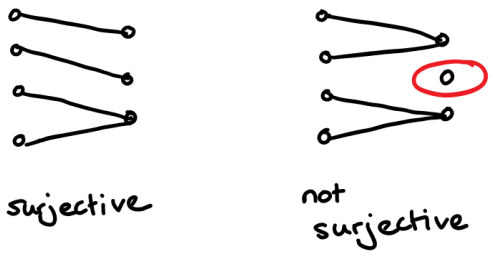


Figure 2.3: example of a surjective function

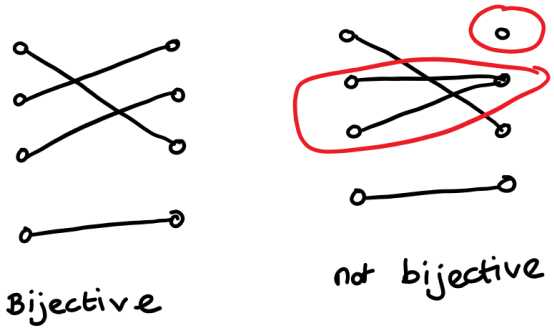


Figure 2.4: example of a bijective function

Definition 2.20 (Finiteness)**.** Cardinality allows us to give a rigorous notion for a set to be *finite*/have a finite number of elements.

Given a set A, if there is some natural number n such that $|A| = |[[n-1]]|$ then we say A has size n.

If A has size n for some n then we say A is finite.

Definition 2.21 (infinite)**.** We say a set is *infinite* if it isn't finite.

Suppose A and B are finite sets with $|A| > |B|$ and $f : A \to B$ is a function then there exists some $b \in B$ for which there are at least two elements a_1, a_2 of A for which $f(a_1) = f(a_2) = b$.

The name for this fact comes from the idea that if you have a dovecote with n holes and you have more than $n + 1$ pigeons then however you arrange the pigeons at least one hole must contain more than one pigoen.

Lemma 2.3. *Suppose that A, B are sets and B is finite.*

- *If there exists an injection $f : A \to B$ then A is finite.*
- *If there exists a surjections $g : B \to A$ then A is finite.*

Proof. If B is finite then there is a bijection between B and some $[[n]]$ and so composing f and this bijection gives an injection form A to some subset of $[[n]]$. Let us call this injection j. Now let us create a bijection from A to some $[[m]]$ as follows. The image of j is $\{j_0, \dots, j_m\}$ so let us map $j^{-1}(j_k)$ to k for $k = 0, \dots, m$. This shows that A is finite.

Now considering the second point. We can choose a right inverse to g which we call h. This will be an injection since g is a function so the first point proves that A is finite also in this case. □

Theorem 2.1 (Cantor's Theorem)**.** *Let A be a set and $f : A \to \mathcal{P}(A)$ then f cannot be a surjection.*

You could also say: there is no surjection between a set and its power set.

Remark. One implication of this theorem is that a set cannot be the same size as its power set. This is obvious for finite sets; if $|A| = n$ then $|\mathcal{P}(A)| = 2^n$ but it isn't clear for infinite sets.

Proof. We notice that for every $x \in A$ we have $f(x)$ which is a subset of A. This brings up two possibilities we could have $x \in f(x)$ or $x \notin f(x)$. We can form a set C by writing

$$C = \{x \in A : x \notin f(x)\}.$$

Now suppose for contradiction that f is surjective. This implies that there exists some $c \in A$ such that $f(c) = C$.

Now there are two possibilities.

1. $c \in C$ which is a contradiction because we defined C to be the set of x for which $x \notin f(x)$.
2. $c \notin C$ which is also a contradiction because then if C is the set of all x for which $x \notin f(x)$ so should contain c.

Therefore we have a contradiction to f being surjective. □

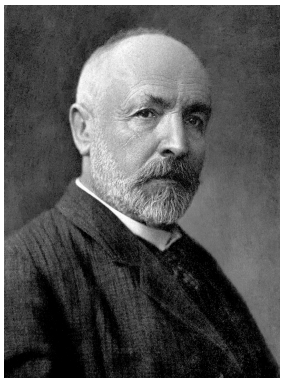


Figure 2.5: A picture of Cantor

Cantor was one of the pioneers of the foundations of mathematics and in particular set theory. His work was astonishingly controversial at the time. One of the implications of Cantor's theorem which we haven't explored is the existence of an infinity which is *larger* in some sense than the infinity that is the cardinality of the natural numbers. Some theologians believed this as a step towards pantheism. He was also described as a "corrupter of youth".

Remark (non-examinable). The proof of Cantor's theorem is is strongly related to *Russel's paradox". Let us consider the set $R = \{\text{all sets } x \text{ such that } x \notin x\}$ then the question is whether $R \in R$?

It becomes clear from this that we cannot define the set R so we do not want to build an axiom set which would allow us to define something like R.

2.7 Ordered pairs and Cartesian products

When we think about functions we often think about the graph of a function. Therefore is is useful to enhance our set theory in a way that allows us to talk about graphs.

Definition 2.22 (ordered pairs)**.** If we have two sets X and Y and $x \in X, y \in Y$ then we can form an ordered pair of these two elements that we write (x, y).

Remark. Here we say *ordered* pair because the order matters. So $(1, 2) \neq (2, 1)$. This is different to how sets behave where $\{1, 2\} = \{2, 1\}$. Also unlike setes we keep repeats. We can have a pair x, x if $x \in X$ and $x \in Y$ and this is different to the element x.

Definition 2.23 (cartesian product)**.** Given two sets X and Y we can form a new set $X \times Y$ called the *Cartesian product* of X and Y and defined by

$$X \times Y = \{(x, y) \, : \, x \in X, y \in Y\}.$$

Remark. We have a special notation for the Cartesian product between a set and itself. We write

$$X^2 = X \times X,$$

and

$$X^n = X \times X^{n-1}.$$

You will have probably seen this before e.g. $\mathbb{R}^d$.

2.8 Graphs and a better definition of functions

Relations are an important mathematical object that you might not have thought about before. At first they seem quite similar to functions but they can appear in a very different settings. Since this is a section a about functions we are just going to talk enough about relations to give a better definition of a function and then return to them later.

Definition 2.24 (Graphical relations)**.** A *graphical relation* is formed of three objects

- A domain X
- A codomain Y
- A subset $G \subset X \times Y$ which saitsfies that for every $x \in X$ there exists exactly one $y \in Y$ such that $(x, y) \in G$.

If $(x, y) \in G$ we write xGy.

Example 2.11. The relation defined by $(x, y) \in G \Leftrightarrow x \leq y$ is not graphical from $\mathbb{R}$ to itself because for every x there are many y with $y \leq x$.

The relation defined by $(x, y) \in G \Leftrightarrow x = y^2$ is not graphical from $\mathbb{R}$ to itself because if x is negative then it isn't the square of any real number so there are no ys with xGy, and also because for $x \geq 0$ there are two ys with $x = y^2$.

The relation defined by $(x, y) \in G \Leftrightarrow x = y^3$ on $\mathbb{R}$ *is* graphical because for every $x \in \mathbb{R}$ there is exactly one $y \in \mathbb{R}$ such that $x = y^3$.

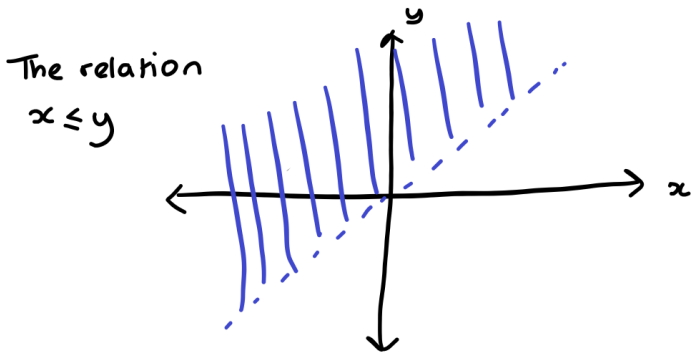


Figure 2.6: A picture of the less than relation on the reals

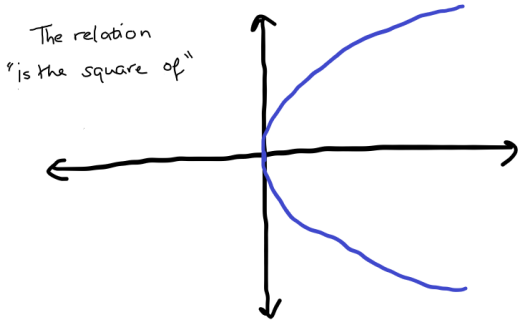


Figure 2.7: A picture of the is the square of relation

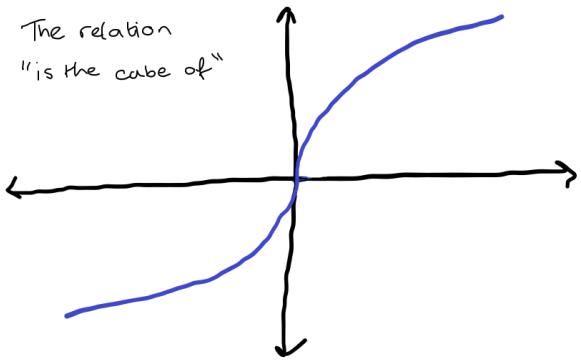


Figure 2.8: A picture of the is the cube of relation

Using this we can give a better definition of a function

Definition 2.25 (function)**.** Given a graphical relation (x, Y, G) we can define a function f with domain X and co-domain Y by setting $f(x) = y$ for the unique y such that $(x, y) \in G$.

Remark. It is interesting *and subtle* to think about why this is a more satisfactory definition of a function. People who worked on the foundations of mathematics wanted to build all mathematical objects from sets using a fairly small set of axioms.

We have done something in this direction, but starting only with sets and axioms it would take a very long time to define everything we need for this course. It is also challenging and not to everybodies taste. Here we have skipped some steps, hidden some subtleties, added axioms to make it simpler etc. For example, in the axiom of separation we have not really defined what we mean by a property, and we have used the natural numbers without defining them starting from sets.

Most of the time when we've introduced new axioms, we are asserting that some set exists (e.g. a power set, union, etc.). If we wanted to add an axiom saying functions exist/make sense this would mean adding an axiom that doesn't just say "another kind of set exists" it would say "a completely new kind of object exists" and this is in some sense very unsatisfactory. Therefore, this later more formal definition of function is *better* because it allows us to say what a function is using only concepts about sets and subsets.

Chapter 3

Operations on sets and functions

3.1 Set operations

We have already seen a few ways of making new sets from old like specification or taking Cartesian products. We are now going to look at some common *set operations* which allow us to make lots of new sets.

Definition 3.1 (union)**.** The union of two sets A and B is set containing all the elements that are in A or B (or both). We write it $A \cup B$.

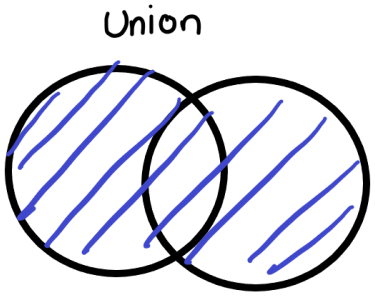


Figure 3.1: picture of a union of two sets, sets represented as overlapping circles

We don't just have to take unions over pairs of sets. In fact we can take a union over almost any collection of sets. Formally, suppose that C is a set all of whose

elements are sets then we can define a new set

$$\bigcup C = \{x : \exists S \in C \text{ s.t. } x \in S\}.$$

Remark. In practice most unions we take over larger collections of sets won't be written like they are in the formal definition. It is common so see a union taken of a sequence of sets $A_1, A_2, A_3, ...$ then we write $\bigcup_n A_n$ to be the union of all these sets.

Example 3.1.

$$\{1, 2, 3\} \cup \{1, 2, 4\} = \{1, 2, 3, 4\}.$$

Example 3.2.

$$\bigcup_{n \in \mathbb{N}} [[n]] = \mathbb{N}.$$

Lemma 3.1. *Suppose that A, B and C are sets then the following are true*

- $A \cup \emptyset = A$,
- $A \cup (B \cup C) = (A \cup B) \cup C$,
- $A \cup B = B \cup A$,
- $A \cup B = B$ *if and only if* $A \subseteq B$,
- $A \cup A = A$

Another very important *set operation* is taking intersections.

Definition 3.2 (intersection)**.** Given two sets A and B the *intersection* of A and B is the set containing the elements in *both* A and B.

As with unions, we don't have to do this with a pair of sets. If C is a set all of whose elements are sets we can write

$$\bigcap C = \{x : x \in S \,\forall\, S \in C\}.$$

An important piece of notation is that if A and B are sets with $A \cap B = \emptyset$ then we say that A and B are *disjoint*.

Remark. Again as with unions, we will more often see this definition applied to sequences of sets using notation like

$$\bigcap_n A_n.$$

Example 3.3.

$$\{1, 2, 3\} \cap \{1, 2, 4\} = \{1, 2\}.$$

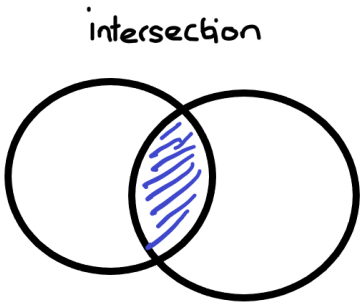


Figure 3.2: picture of the intersection of two sets, sets represented by overlapping circles

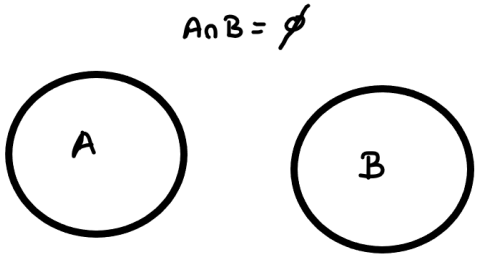


Figure 3.3: picture showing two disjoint sets

Example 3.4.

$$\bigcup_{n\in\mathbb{N}} [[n]] = 0.$$

Lemma 3.2. *Given sets A, B, C the following are true*

- $A \cap \emptyset = \emptyset$,
- $(A \cap B) \cap C = A \cap (B \cap C)$,
- $A \cap B = B \cap A$,
- $A \cap B = B$ *if and only if* $A \supset B$,
- $A \cap A = A$.

Definition 3.3 (set difference)**.** If A and B are sets the we define the *set difference* which we write $A - B$ (or sometimes $A \setminus B$) by

$$A - B = \{x \,:\, x \in A, x \notin B\}.$$

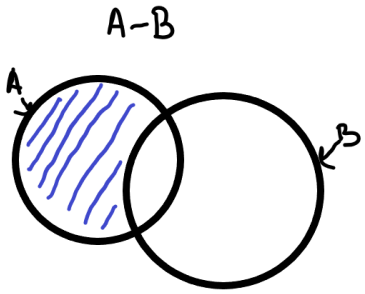


Figure 3.4: picture of setminus with sets represented by overlappying circles

Remark. WARNING: Unlike union and intersection, set difference is not commutative.

Example 3.5.

$$\{1,2,3\} - \{1,2,4\} = \{3\}, \quad \{1,2,4\} - \{1,2,3\} = \{4\}.$$

3.2 Using set operations together

There are some rules for how set operations interact with each other. Usually these are easy to remember/prove by drawing pictures or by writing out exactly what each operation means.

Lemma 3.3. *Suppose that A, B and C are sets then we can distribute intersections and unions with each other in the following way*

- $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$

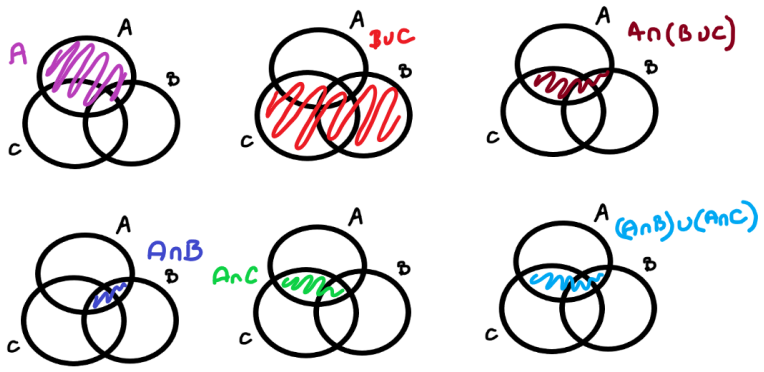


Figure 3.5: picture of distibutivity of union

- $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$

Proof. These results are very straightforward to prove. We remember that that if $x \in A \Rightarrow x \in B$ and $x \in B \Rightarrow x \in A$ then $A = B$.

If $x \in A \cap (B \cup C)$ then we know $x \in A$ and $x \in B$ or $x \in C$. Therefore $x \in A$ and $x \in B$ or $x \in A$ and $x \in C$ so $x \in (A \cap B) \cup (A \cap C)$.

The second result is proved similarly.

As there are only 3 sets involved the pictures probably provide a clearer (and still rigorous) proof for most people. However with four or more sets it becomes impossible to draw sets with all the possible intersections, so we need to be able to use symbols too. □

We also have a similar result involving setminuses.

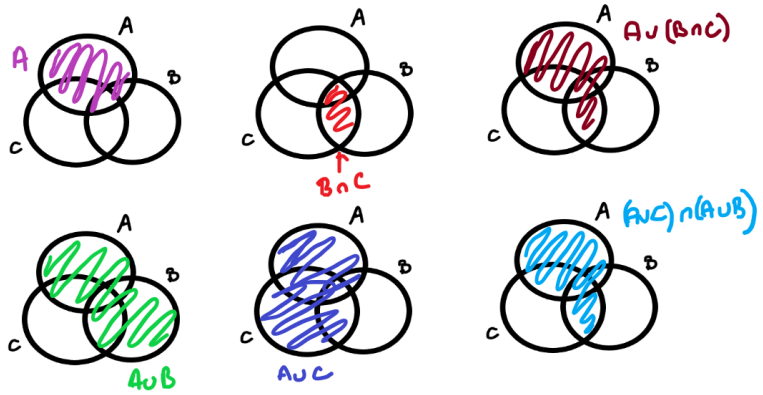


Figure 3.6: picture of distibutivity of intersection

Lemma 3.4 (De Morgan's Laws)**.** *Suppose A, B and C are sets then the following are true*

- $A - (B \cup C) = (A - B) \cap (A - C)$,

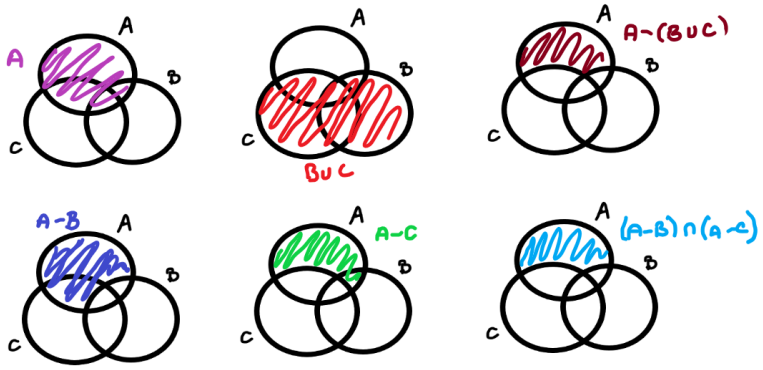


Figure 3.7: picture of De Morgan's Laws 1

- $A - (B \cap C) = (A - B) \cup (A - C)$.

All these operations can be understood in terms of indicator functions as well

Lemma 3.5. *Given a set A and subsets B, C, D we have the following:*

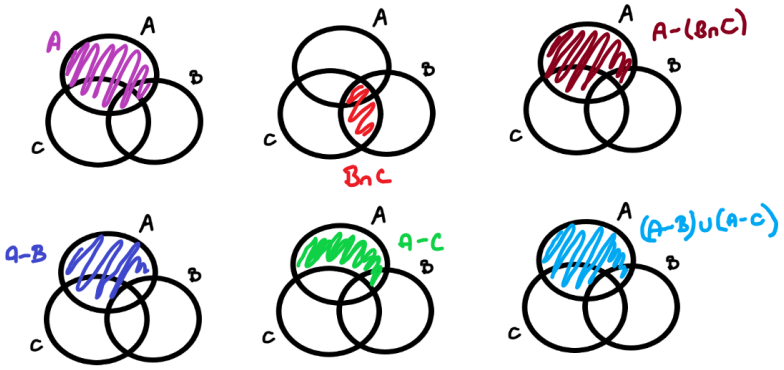


Figure 3.8: picture of De Morgan's Laws 2

- $1_{B\cap C} = 1_B 1_C$.
- $1_{B\cup C} = 1_B + 1_C - 1_B 1_C$.
- *If* $B \subset C$ *then* $1_{C\setminus B} = 1_C - 1_B$.

Proof. If you want to you can check these yourself. It is mainly just symbol pushing. A more exciting thing to do is try and prove De Morgan's laws or distributative laws using these facts. □

3.3 Operations on functions

Definition 3.4 (composition)**.** Given sets A, B and C and functions $f : A \to B$ and $g : B \to C$ we can define a new function $f \circ g$ from A to C by

$$f \circ g(x) = g(f(x)).$$

Example 3.6.

Example 3.7. Another example would be if $f : \mathbb{R} \to [0, \infty)$ is defined by $f(x) = x^2$ and $g : [0, \infty) \to [0, \infty)$ is defined by $g(y) = \sqrt{y}$ then $f \circ g(x) = |x|$ and is defined from $\mathbb{R}$ to $[0, \infty)$.

Remark. An important example of composition is if A is a set and $f : A \to A$ then we can compose A with itself. We often write $f \circ f = f^2$ and $f^n = f \circ f^{n-1}$.

Lemma 3.6 (Associativity of composition)**.** *Composition of functions is associative. That is to say, if* A, B, C *and* D *are all sets and* $f : A \to B, g : B \to C$ *and* $h : C \to D$ *are all functions then*

$$f \circ (g \circ h) = (f \circ g) \circ h$$

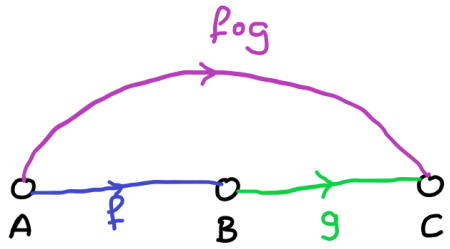


Figure 3.9: diagram of function composition

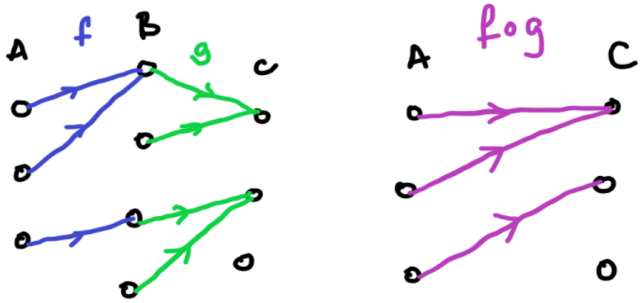


Figure 3.10: example of function composition

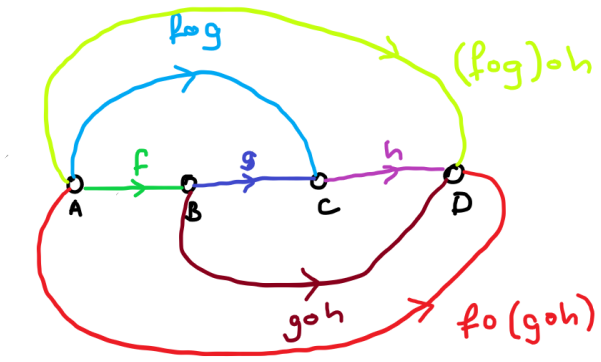


Figure 3.11: picture of associativity of composition of function

Proof. To prove this we can evaluate the functions at a given $x \in A$.

$$f \circ (g \circ h)(x) = (g \circ h)(f(x)) = h(g(f(x))).$$

$$(f \circ g) \circ h(x) = h(f \circ g(x)) = h(g(f(x))).$$

□

We can relate composition of functions to injectivity and surjectivity

Lemma 3.7. *Suppose that A, B and C are sets and $f : A \to B$ and $g : B \to C$ are functions then*

- *If both f and g are injective then so is $f \circ g$,*
- *If both f and g are surjective then so is $f \circ g$.*

Proof. If both f and g are injective then given $z \in C$ there is at most one $y \in B$ with $g(y) = z$ then for this y there is at most one $x \in A$ with $f(x) = y$ therefore there is at most one $x \in A$ with $f \circ g(x) = z$.

If both f and g are surjective then given $z \in C$ there is at least one $y \in B$ with $g(y) = z$ and for this y there is at least one $x \in A$ with $f(x) = y$ therefore there is at least one $x \in A$ with $f \circ g(x) = z$. □

This next set of results is about what it means to be the *inverse of a function*. This can be a subtle and quite complicated issue.

Example 3.8. As we just saw above if $f : \mathbb{R} \to [0, \infty)$ is defined by $f(x) = x^2$ and $g : [0, \infty) \to [0, \infty)$ is defined by $g(y) = \sqrt{y}$ then $f \circ g(x) = |x|$. So even though we think of square root and squaring as inverses of each other in this case $f \circ g$ is not equal to the identity function.

On the other hand if $f : [0, \infty) \to [0, \infty)$ defined by $f(x) = x^2$ and $g : [0, \infty) \to [0, \infty)$ is defined by $g(y) = \sqrt{(}y)$ then $f \circ g(x) = x$ so if we change the domain of f we can think for these functions as inverse to each other.

We also have that if $f : \mathbb{R} \to [0, \infty)$ defined by $f(x) = x^2$ and $g : [0, \infty) \to [0, \infty)$ defined by $g(y) = \sqrt{(}y)$ (as in the first part of the example) then $g \circ f)(y) = y$. we can think of these as inverse to eachother in one order but not in the other order.

Definition 3.5 (left and right inverses)**.** Let A and B be sets and let $f : A \to B$ and $g : B \to A$.

- We call g a *left inverse* of f if $g \circ f = Id_B$,
- We call g a *right inverse* of f if $f \circ g = Id_A$.
- We call g an *inverse* of f if it is both a left inverse and a right inverse. If an inverse exists we often write $g = f^{-1}$.

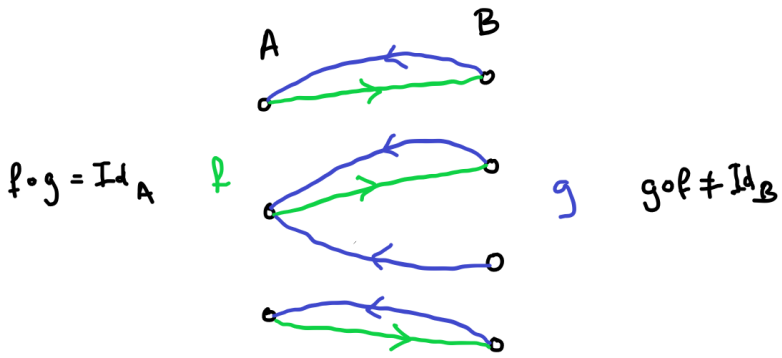


Figure 3.12: example of a function with a right inverse but no left inverse

Lemma 3.8. *Given two sets A and B and a function $f : A \to B$, we have the following equivalences*

1. *f is injective if and only if f has a left inverse,*
2. *f is surjective if and only if f has a right inverse,*

3. f is bijective if an only if f has an inverse.

Proof. Let us begin with point 1. in the direction *injective* $\Rightarrow$ *left inverse*. Injectivity means that for every $y \in B$ there is at most one $x \in A$ with $f(x) = y$. So we can define a left inverse as follows: if there exists an $x \in A$ with $f(x) = y$ then set $g(y) = x$. If there exists no $x \in A$ with $f(x) = y$ then choose an arbitrary element $x_0 \in A$ and set $g(y) = x_0$. This ensures that for every $x \in A$ we have $g(f(x)) = x$.

Now point 1 in the direction *left inverse* $\Rightarrow$ *injective*. So there exists g with $g(f(x)) = x$ for all $x \in A$. Suppose f isn't injective then there exists $y_0 \in B$ and $x_1 \neq x_2 \in A$ such that $f(x_1) = f(x_2) = y_0$. Then we have that $g(y_0) = g(f(x_1)) = x_1 = g(f(x_2)) = x_2$ which is a contradiction. Therefore f must be injective.

Now point 2 in the direction *surjective* $\Rightarrow$ *right inverse*. For every $y \in B$ there exists at least one x such that $f(x) = y$ by surjectivity. So define a function g by choosing $g(y)$ to be equal to one of the $x \in A$ with $f(x) = y$. This means that $f(g(y)) = y$ so g is a right inverse to f.

Now point 2 in the direction *right inverse* $\Rightarrow$ *surjective*. So we have a function g such that $f(g(y)) = y$. So for every $y \in B$ there exist one element in A, namely $g(y)$, such that $f(g(y)) = y$ so f is surjective.

Now for point 3 it looks at first like we can just apply the previous results. We can in one direction. If f has an inverse then it has both a left inverse and a right inverse so by points 1 and 2 f must be both injective and surjective so it is bijective.

Now if we want to show bijectivity of f implies we must have an inverse we not that if f is bijective then for every $y \in B$ there exists exactly one $x \in A$ such that $f(x) = y$ so we can define $g(y)$ to be this unique x and this ensures that $g(f(x)) = x$ and $f(g(y)) = y$. □

Definition 3.6 (image and preimage)**.** Suppose that A, B are sets and $f : A \to B$ is a function. Suppose further that $C \subset A, D \subset B$ then we write

- $f(C) = \{y \in B : y = f(x), \text{for some}\, x \in C\}$ and we call $f(C)$ the *image* of C under f.
- $f^{-1}(D) = \{x \in A : f(x) \in D\}$ and we call $f^{-1}(D)$ the *preimage* of D under f.

Example 3.9. We have to be particularly careful with preimages as this example demonstrates

Let $f : \mathbb{R} \to \mathbb{R}$ given by $f(x) = x^2$ then $f^{-1}(\{2\}) = \{-\sqrt{2}, \sqrt{2}\}$.

Let $f : \mathbb{Q} \to \mathbb{Q}$ given by $f(x) = x^2$ then $f^{-1}(\{2\}) = \emptyset$. Because $\pm sqrt2$ is irrational so 2 has no square roots in the rationals.

Let $f : [0, \infty) \to [0, \infty)$ given by $f(x) = x^2$ then $f^{-1}(\{2\}) = \{\sqrt{2}\}$.

Lastly in this section we have a deeper theorem whose proof is more complicated that those we have encountered before.

Theorem 3.1 (Cantor-Schoeder-Bernstein)**.** *Let A, B be sets and let $f : A \to B$ be an injection and $g : B \to A$ be an injection. Then there exists a bijection h between A and B.*

NONEXAMINABLE. Let us call $C = f(A) \subset B$ and $D = g(B) \subset A$. Since f and g are injective we can define $f^{-1} : C \to A$ and $g^{-1} : D \to B$.

So we end up with two bijective functions going from parts of A to parts of B namely $f : A \to D$ and $g^{-1} : C \to B$

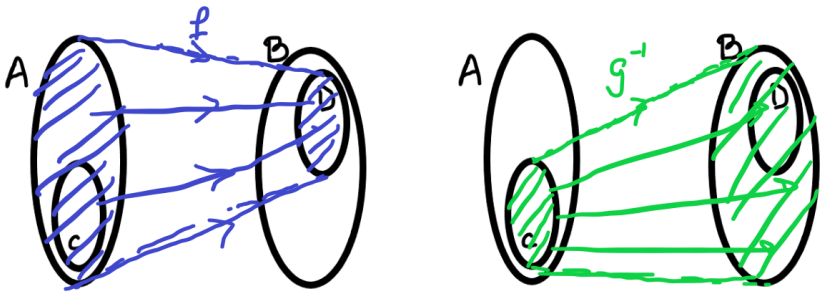


Figure 3.13: Our two functions going from A to B

Now we want to create h from both f and g^{-1}. To do this we want to split A into two sets A_1 where we use f to get to points in B and A_2 where we use g^{-1} to get to points in B.

Our challenge is to find suitable sets A_1 and A_2. We can see that $A_2 \subset C$ since g^{-1} must be defined on A_2. We can also see that in some situations A_2 could be the whole of C because doing this we could hit some elements of D twice and break the injectivity.

Let us write $i = f \circ g$. This function is injective on A and it's range is C. We can similarly define $j = g \circ f$ which will also be injective on B and whose range is D.

Now let us create some sequences of sets $C_0 = A, C_1 = i(A), C_2 = i(C_1), \dots, C_n = i(C_{n-1}), \dots$ and $D_0 = B, D_1 = j(B), \dots, D_n = j(D_{n-1}), \dots$. Then let us define $C_\infty = \bigcap_n C_n$ and $D_\infty = \bigcap_n D_n$. So C_∞ are points which will keep being in the range of i^n for any n and D_∞ similarly.

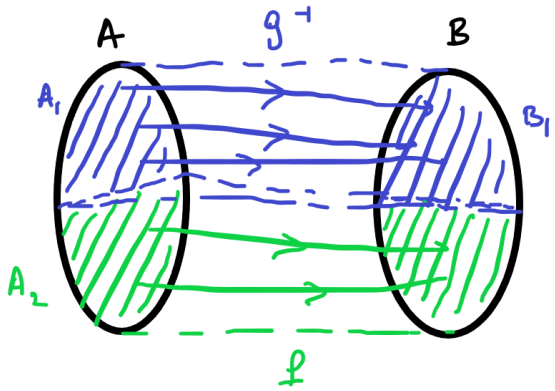


Figure 3.14: A picture of how we want to split up A

We make the following claim:

$$g(y) \in C_\infty \Leftrightarrow y \in D_\infty, \tag{3.1}$$
$$f(x) \in D_\infty \Leftrightarrow x \in C_\infty. \tag{3.2}$$

Let us call

$$A_1 = \{x \in A : x \notin C_\infty, \text{for the least } n \text{ s.t. } x \notin C_n, i^{-n+1}(x) \notin C\}$$

and

$$A_0 = \{x \in A : x \notin C_\infty, \text{for the least } n \text{ s.t. } x \notin C_n, i^{-n+1}(x) \in C, g^{-1}(i^{-n+1}(x)) \notin D\}.$$

Then A_1 and A_2 are disjoint.

Similarly,

$$B_1 = \{y \in B : y \notin B_\infty, \text{for the least } n \text{ s.t. } y \notin D_n, j^{-n+1}(y) \notin D\}$$

and

$$B_0 = \{y \in B : y \notin B_\infty, \text{for the least } n \text{ s.t. } y \notin D_n, j^{-n+1}(y) \in D, f^{-1}(j^{-n+1}(y)) \notin C\}.$$

The sets are also disjoint.

Now we have our second claim:

- $x \in A_1$ if and only if $f(x) \in B_0$
- $x \in A_0$ if and only if $g^{-1}(x) \in B_1$

This second claim shows that

- $f|_{A_1} : A_1 \to B_0$ is a bijection and
- $g^{-1}|_{A_0} : A_0 \to B_1$ is a bijection.

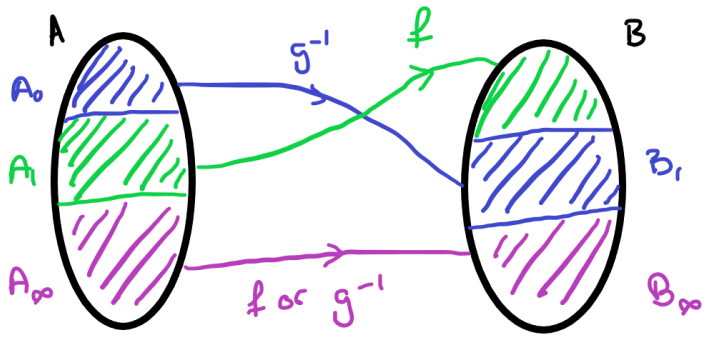


Figure 3.15: A picture of the bijections we've constructed

Now we have deconstructed into different bijections we can define h by saying

$$h(x) = \begin{cases} f(x) & x \in A_1 \cup A_\infty \\ g^{-1}(x) & x \in A_0 \end{cases}$$

□

Chapter 4

Relations

Before we define them formally let us look at some examples.

For a generic relationship R we write x and y are related under R by xRy. Here the ordering matters as you can see in the example.

Example 4.1. We can relate two real numbers x and y with the relationship *is less than*. So we can write xRy if $x \leq y$.

Example 4.2. Is equal to is also a relation. This can be more complicated that just trivially equating elements of the same set. For example we might want to write a relation between $\mathbb{N}$ and $\mathbb{R}$ by equating integers with their counterpart in the natural numbers.

Remark. Something that isn't a relation but might seem similar is a *property*. So for example the statement 3 is a prime number is just talking about a property that may or may not hold for the integers. We can come up with a relation to the set $\{1\}$ by saying $nR1$ if and only if n is prime.

The formal definition of a relation is as follows

Definition 4.1 (relation)**.** A relation consists of three parts

- A set X called the *domain*,
- A set Y called the *co-domain*
- A subset of $X \times Y$ often given the name R.

Using the notation from before we write xRy iff (x, y) is in the subset of $X \times Y$ defining the relation.

We can represent the relation *is less than* with a picture as we saw in week 1.

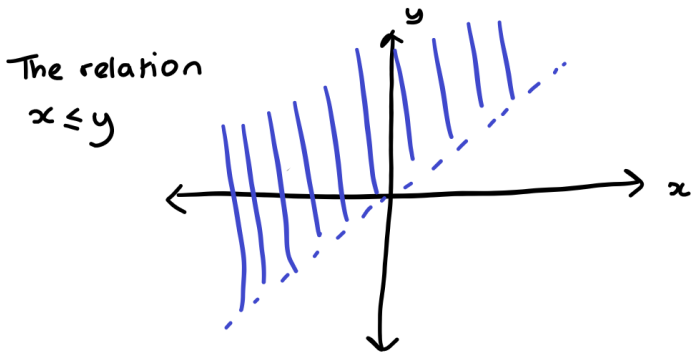


Figure 4.1: A picture of the relation less than

4.1 Equivalence relations, Equivalence classes and Quotients

In this section we look at some particular properties a relation can have when it relates elements of the same sets. That is to say, we are interested in a set X and a relation R on X defined by a subset of X^2.

Definition 4.2 (reflexivity)**.** We call a relation, R, on X *reflexive* if xRx.

Example 4.3. The relation defined by $<$ is *not* reflexive; the relation defined by $\leq$ *is* reflexive.

Definition 4.3 (transitivity)**.** We call a relation R on X *transitive* if xRy and yRz implies that xRz.

Example 4.4. Both the relations given above $<, \leq$ are transitive but a relation like *is the square of* is not.

Definition 4.4 (symmetric)**.** A relation R on X is *symmetric* if xRy implies that yRx.

Example 4.5. The relation on $\mathbb{R}$ given by xRy if an only if $|x - y| = 1$ is symmetric.

The relation on $\mathbb{R}$ given by xRy if and only if $x - y = 1$ is not symmetric.

Definition 4.5 (equivalence relations)**.** A relation on X is called an *equivalence relation* if it is reflexive, symmetric and transitive. We often denote equivalence relations with $\sim$ rather than R.

Remark. Equivalence relations are a very important object in mathematics. We will see more about splitting up sets using equivalence relations soon. Equivalence relations are supposed to represent the properties of equality.

Example 4.6.
- The relation defined by $x \sim y$ if and only if $x - y \in \mathbb{Q}$ is and equivalence relation.
- The relation defined by $x \sim y$ if and only if $x = y$ is an equivalence relation.

We can use equivalence relations to divide sets into chunks. In order to talk about this let us first give a proper definition of what it would mean to define a set into chunks.

Definition 4.6 (Partitions)**.** Given a set X a partition of X is a subset $\mathbb{P}$ or $\mathcal{P}(X)$ (so a set of subsets of X) satisfying the following:

- If P, Q in $\mathbb{P}$ and $P \neq Q$ then $P \cap Q = \emptyset$, i.e. any two sets in $\mathbb{P}$ are disjoint,
- $\bigcup_{P \in \mathbb{P}} P = X$, i.e. for every $x \in X$ there is a $P \in \mathbb{P}$ such that $x \in P$,
- Every $P \in \mathbb{P}$ is non-empty.

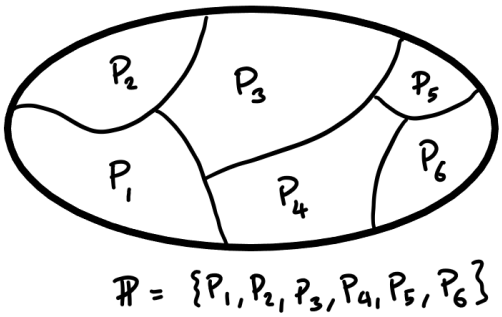


Figure 4.2: A picture of a partition

Example 4.7. For any X we can define the partition into the set of singletons, $\mathbb{P} = \bigcup_{x \in X} \{\{x\}\}$.

We can also define a trivial partition $\mathbb{P} = X$.

We can use equivalence relations to form partitions. First let us talk about each of the chunks separately.

Definition 4.7 (equivalence class)**.** Given a set X and an equivalence relation $\sim$ and an element $x \in X$ we define the equivalence class of x by

$$E_x = \{y \in X : x \sim y\}.$$

Another common notation for this is $[x]$ or $[x]_\sim$.

Lemma 4.1. *Given a set X and an equivalence relation $\sim$ then the equivalence classes under $\sim$ form a partition of X.*

Equally, given a set X and a partition $\mathbb{P}$ of x the relation defined by $x \sim y$ if x and y are both in the same P.

Proof.
- For every $x \in X$ we know that $x \in E_x$ so the union of all equivalence classes is all of X.
- If E, F are two equivalence classes then if there exists $x \in E \cap F$ then $y \sim x$ for every $y \in E$ and $z \sim x$ for every $z \in F$. So by transitivity of $\sim$ we have $y \sim z$ for every $y \in E, z \in F$ therefore we must have $E = F$.
- Every equivalence class is not empty as $x \sim x$ so $x \in E_x$.

This proves that the equivalence classes form a partition of the set.

Now considering $\sim$ defined in the lemma.

- It is reflexive since $x \in P$ so $x \sim x$.
- It is transitive since if $x \sim y$ and $y \sim z$ then $x \in P$ implies $y \in P$ which in turn implies $z \in P$ so $x \sim z$.
- It is symmetric since if $x \sim y$ then for some P,$x, y \in P$ so $y \sim x$.

□

Example 4.8. If we consider the equivalence relation on $\mathbb{Z}$ given by $x \sim y$ when $|x-y|$ is divisible by 2. Then under this relation we have two equivalence classes. The even integers and the odd integers.

4.1.1 Quotients

While it probably won't be obvious in this course quotients are one of the most important concepts in mathematics. Quotienting by an equivalence relation is the act of considering two objects to be *the same* if they lie in the same equivalence class. In further areas of mathematics you will be considering sets with structures on them (e.g. groups in algebra) and when you quotient by things you will want to do so in such a way that you can preserve that structure. At the moment we are only interested in the structure of being a set. That is crucially the axiom of extension that a set is defined by its elements.

Definition 4.8 (quotient)**.** Given a set X and an equivalence relation $\sim$ then we define the quotient

$$X/\sim = \{\text{the set of equivalence classes under } \sim\} = \{E_x : x \in X\}.$$

Example 4.9. If we consider the set $\mathbb{R}$ and the equivalence relation $x \sim y$ when $x - y \in \mathbb{Z}$. Then for any x the set E_x is the set of all real numbers who have the same decimal expansion after the decimal point we often call this set $\mathbb{T}$ and think about it as wrapping the real numbers repeatedly around the set $[0, 1)$.

Example 4.10. Quotienting by equivalence relations is one example of using mathematical abstraction. It is something you will already have been doing very frequently. A good example from school mathematics is congruent triangles. We call triangles similar if they have the same angles and side length. Regarding two triangles as *the same* if they are congruent is an example of quotienting by an equivalence relation.

4.2 Integers via quotients

Quotients are a key way of constructing new things from old. A good example is constructing $\mathbb{Z}$ starting from $\mathbb{N}$.

We work from the starting point that we have defined $\mathbb{N}$ and addition and multiplication on $\mathbb{N}$ already.

Definition 4.9 (Integers)**.** In this setting we want to think about integers as the possible differences between two natural numbers. So, for example, we want to define $-1 = 2-3$, or anything representing taking one step to the left on the numberline. So we want to construct the integers from ordered pairs of natural numbers $\mathbb{N}^2$. We can make a direct equivalence between $\mathbb{N}^2$ and $\mathbb{Z}$ because there are many possible pairs of natural numbers whose difference will be equal to the same integer e.g. $-1 = 2-3 = 4-5$. So we need to put an equivalence relation on $\mathbb{N}^2$ so we have

$$(p, q) \sim (s, t) \text{ when } p + t = s + q.$$

Then we can identify $\mathbb{Z}$ with $\mathbb{N}^2/\sim$.

Now we want to further define the arithmetic operations on $\mathbb{Z}$ we do this as follows.

- We define a function called *negation* by $- : \mathbb{Z} \to \mathbb{Z}$ by $-E_{(p,q)} = E_{(q,p)}$.
- We define *addition* as a function $+ : \mathbb{Z}^2 \to \mathbb{Z}$ by $E_{(p,q)} + E_{(s,t)} = E_{(p+s,q+t)}$.
- We define *multiplication* as a function $\times : \mathbb{Z}^2 \to \mathbb{Z}$ by $E_{(p,q)} \times E_{(s,t)} = E_{(pr+qs,ps+qr)}$.

These definitions show a very common subtlety. We have a function whose domain is a quotient and we specify the function by looking at one element of an equivalence class and specifying the function on that. What we want to do is:

Definition 4.10 (well defined function)**.** Suppose we have a set X, and equivalence relation on X given by $\sim$ and we wish to define a function from $f : X/\sim \to Y$ by $f(E_x) = \tilde{f}(x)$ for some other function $\tilde{f}$. We call f a *well-defined function* if $f(x) = f(x')$ whenever $x \sim x'$.

You can now "have fun" by checking that negation, addition and multiplication as defined above are well defined.

4.3 Order relations

Definition 4.11 (antisymmetry)**.** We call a relation R on X *antisymmetric* if xRy and yRx implies that $x = y$.

Example 4.11. The relation above xRy if and only if $x - y = 1$ is antisymmetric.

Definition 4.12 (partial orders)**.** Partial orders are another special kind of relation on a set. A relation R on a set X is a partial order if it is antisymmetric, transitive and reflexive.

Example 4.12. The most classical example of a partial order is the normal sense of order given by $\leq$ on some set of numbers $\mathbb{R}, \mathbb{Q}, \mathbb{Z}, ...$

Another example of a partial order is if A is a set and $X = \mathcal{P}(A)$ then we can put a partial order on X with BRC iff $B \subset C$. You can check this satisfies all the conditions.

Remark. These two examples show two key types of partial order. In the first for any two numbers x, y either $x \leq y$ or $y \leq x$. However for any two subsets B, C it is not the case that either $B \subset C$ or $C \subset B$.

Definition 4.13 (total order)**.** A total ordering on a set x is a partial order where for every x, y either xRy or yRx.

4.4 Modular arithmetic

We now return to $\mathbb{N}$ we can define a relation called *is a divisor of* which we write $n|m$ as n is a divisor of m.

Definition 4.14 (divisor)**.** Given two natural numbers n and m we say n is a divisor of m (or $n|m$) if there exists $k \in \mathbb{N}$ such that $m = n \times k$.

Remark. This relation gives us a new example of a partial order on $\mathbb{N}$. You can check this!

Definition 4.15 (prime number)**.** While we define divisor it is worth defining a prime number. We call $p \in \mathbb{N}$ a *prime number* if the only divisors of p are $1, p$.

Definition 4.16 (congruence modulo n)**.** Given an number $n \in \mathbb{N} - \{0\}$ we can define an equivalence relation on $\mathbb{Z}$ called congruence modulo n (written $\equiv \mod n$) by

$$p \equiv q \,(\mathrm{mod}\, n) \quad \text{when} \quad n \Big| |p - q|.$$

Remark. You can check that congruence modulo n is an equivalence relation.

We write $[a]_n$ for the equivalence class of an integer a under congruence modulo n.

We have a slightly different notation for the quotient we write

$$\mathbb{Z}/n\mathbb{Z} = \{[a]_n \,:\, a \in \mathbb{Z}\}.$$

We can straightforwardly see that $\mathbb{Z}/n\mathbb{Z}$ has n elements

We can check that given n if $p, q \in \{0, 1, \ldots, n-1\}$ then $p \not\equiv q \mod n$. We have $|p-q| < n$ so n cannot divide $|p-q|$. Therefore we often think of $\{0, 1, \ldots, n-1\}$ as the most important representatives of the equivalence classes.

Definition 4.17 (arithmetic modulo n)**.** We can define arithmetical operations modulo n by using our previous notion of well defined-ness.

For example if we want to define $+ : (\mathbb{Z}/n\mathbb{Z})^2 \to \mathbb{Z}/n\mathbb{Z}$ by using our notion of addition on $\mathbb{Z}$ then we need to check that if $a \equiv a' \mod n$ and $b \equiv b' \mod n$ then $a + b \equiv a' + b' \mod n$.

In this case there exists $k, j \in \mathbb{Z}$ such that $a' = a + kn$ and $b' = b + jn$ so then $a' + b' = a + b + (k + j)n$ and since $k + j \in \mathbb{Z}$ we have $a' + b' \equiv a + b \mod n$. We can check similar facts for multiplication.

There is a lot of richness in modular arithmetic once you have learnt more group theory and number theory.

Example 4.13. Suppose we want to calculate 7^{12} modulo 10. The we can do it by working in $\mathbb{Z}$ and going back to modulo 10 beforehand. $7^{12} = 13841287201$ so we can see that $7^{12} \equiv 1 \mod 10$. However it is more efficient (especially without a calculator) to work in a different way

$$\begin{aligned} 7^2 &\equiv 49 \equiv 9 \mod 10 \\ 7^3 &\equiv 7^2 \times 7 \equiv 9 \times 7 \equiv 63 \equiv 3 \mod 10 \\ 7^4 &\equiv 3 \times 7 \equiv 21 \equiv 1 \mod 10 \\ 7^{12} &\equiv (7^4)^3 \equiv 1^3 \equiv 1 \mod 10. \end{aligned}$$

Now if we supposed we wanted to work out $7^{2025} \mod 10$ this is not really more difficult than working out the earlier example

$$7^{2025} \equiv 7^{1006 \times 4 + 1} \equiv 1 \times 7 = 7 \mod 10.$$

It isn't always the case that there is some k such that $m^k \equiv 1 \mod n$. An example of this is that $5^k \equiv 5 \mod 10$ for every k.

Example 4.14. Modular arithmetic can sometimes be used to show equations cannot have solutions in the integers. An example is

$$2a^2 + 3b^3 = 1.$$

If we had a solution then $2a^2 \equiv 1 \mod 3$ but $2 \times 0^2 \equiv 0 \mod 3$, $2 \times 1^2 \equiv 2 \mod 3$ and $2 \times 2^2 \equiv 2 \mod 3$ so no such a can exist.

Definition 4.18 (linear congruences)**.** We call the equation

$$ax \equiv b \mod n$$

for given $a, b, n \in \mathbb{Z}$ and $x \in \mathbb{Z}$ a free variable a *linear congruence.*

We wish to solve for x in this type of equation. From above we only need to look for an equivalence class of x.

Example 4.15. If we are interested in the congruence

$$2x \equiv 3 \mod 5,$$

our only way of doing this is to check all of $x = 0, 1, \ldots 4\}$ we have

$$\begin{aligned}
2 \times 0 \equiv 0 \not\equiv 3 \mod 5,\\
2 \times 1 \equiv 2 \not\equiv 3 \mod 5,\\
2 \times 2 \equiv 4 \not\equiv 3 \mod 5,\\
2 \times 3 \equiv 6 \equiv 1 \not\equiv 3 \mod 5,\\
2 \times 4 \equiv 8 \equiv 3 \mod 5,
\end{aligned}$$

So our solutions are all $x \in [4]_5$.

We will get some better tools for solving linear congruences once we have done some number theory.

Chapter 5

Logic

5.1 Booleans

This section is about Boolean operators. These are a way of describing how the truth of one statement are contingent of the truth of statements it is made of. An example is if P and Q are two statements and we are interested in whether the statement P and Q is true. This is only the case if both P and Q are true. So we can think of this as a function from the truth values of P and Q to another truth value.

Definition 5.1 (Boolean)**.** Booleans are elements of the set $\mathcal{B} = \{T, F\}$. Where T is *true* and F is *false*.

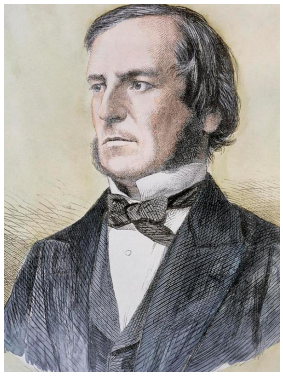


Figure 5.1: A picture of George Boole

Booleans are named after George Boole who was an English mathematician. He is notable for a few things including writing a book with the impressive title *The*

Laws of Thought. He also became a professor of mathematics at the university of Cork despite being largely self taught after primary school. His wife, Mary Everest Boole is also interesting and an example of a woman who made a career on the borders of academic mathematics when it was extremely hostile. Its worth looking them both up!

Definition 5.2 (Boolean operator)**.** A Boolean operator is a function from $f : \mathcal{B}^n \to \mathcal{B}$.

The value of n is called the *arity* of f.

Example 5.1. There are some key examples of *arity* 1 which we write by P and $\neg P$.

P is the identity operator under which $T \mapsto T$ and $F \mapsto F$.

$\neg P$ is the negation operator under which $T \mapsto F$ and $F \mapsto T$. We say that $\neg P$ is true if P is false and $\neg P$ is false if P is true.

Example 5.2. The operators of arity 2 are very helpful for understading what is going on. A first example is $P \wedge Q$ or P and Q. Under this function

$$\begin{aligned}(T,T) &\mapsto T,\\ (T,F) &\mapsto F,\\ (F,T) &\mapsto F,\\ (F,F) &\mapsto F.\end{aligned}$$

We have further *basic* operators of arity two, these are $P \vee Q$ (spoken P or Q), $P \Rightarrow Q$ (spoken P implies Q) of $P \Leftrightarrow Q$ (spoken P is equivalent to Q).

We can express the way these functions work in a table

P	Q	$(P \vee Q)$	$(P \Rightarrow Q)$	$(P \Leftrightarrow Q)$
T	T	T	T	T
T	F	T	F	F
F	T	T	T	F
F	F	F	T	T

(5.1)

We can compose Boolean operators and rewrite them in various different ways

Example 5.3. The operator $\neg(P \wedge Q)$ is given by

$$(T,T) \mapsto F, (T,F) \mapsto T, (F,T) \mapsto T, (F,F) \mapsto T.$$

The operator $(\neg P) \vee (\neg Q)$ is given by

$$(T,T) \mapsto F, (T,F) \mapsto T, (F,T) \mapsto T, (F,F) \mapsto T.$$

Therefore, $\neg(P \wedge Q)$ and $(\neg P) \vee (\neg Q)$ are in some sense the same function. We can say $\neg(P \wedge Q) = (\neg P) \vee (\neg Q)$.

More generally we have the following definition

Definition 5.3. We say two Boolean operators f and g or arity n are the same if they are equal as functions (they map the same elements to the same elements).

We call a way of writing a Boolean operator f in terms of the basic operators $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$ an *expression* for f.

Example 5.4. Both P and $P \wedge P$ are expressions for the identity operator.

Theorem 5.1. *Every Boolean operator has an expression in terms of the basic operators.*

We don't quite have all the technology to prove this. If you are curious you can get a sense of why this is true by working out how you would go from knowing it is true for all operators of arity two to knowing it is true for all operators of arity three.

We can actually do better than this

Theorem 5.2. *Every Boolean operator has an expression in terms of $\neg$ and $\vee$*

Proof. First we check that we can express all our basic operations like this.

$$\begin{aligned} P \wedge Q &= \neg((\neg P) \vee (\neg Q)), \\ P \Rightarrow Q &= (\neg P) \vee Q, \\ P \Leftrightarrow Q &= (P \Rightarrow Q) \wedge (Q \to P) \\ &= \neg((\neg(P \Rightarrow Q)) \vee (\neg(Q \Rightarrow P))) \\ &= \neg((\neg((\neg P) \vee Q)) \vee (\neg((\neg Q) \vee P))). \end{aligned}$$

Now suppose that we have an expression for f in terms of our basic operations we can then replace all instances of $\wedge, \Rightarrow, \Leftrightarrow$ by their expression in terms of $\neg, \vee$ in exactly the way we have when expanding out the expression of $P \Leftrightarrow Q$. □

5.2 Boolean algebra

The basic Boolean operators interact with eachother in much the same way set operations do. We have the following results whose proofs are omitted.

Lemma 5.1. *Suppose that P, Q, R are Booleans then we have the following about $\vee$:*

- $P \vee T = T$ *and* $P \vee F = P$,
- $P \vee (Q \vee R)) = (P \vee Q) \vee R$,

- $P \vee Q = Q \vee P$,
- $(P \Rightarrow Q) = T$ *if and only if* $P \vee Q = Q$,
- $P \vee P = P$.

Lemma 5.2. *Suppose that P, Q, R are Booleans then we have the following about $\wedge$:*

- $P \wedge T = P$ *and* $P \wedge F = F$,
- $P \wedge (Q \wedge R) = (P \wedge Q) \wedge R$,
- $P \wedge Q = Q \wedge P$,
- $(P \Rightarrow Q) = T$ *if and only if* $P \wedge Q = P$,
- $P \wedge P = P$.

Lemma 5.3. *Here are some distributive laws. Suppose that P, Q, R are Boolean's then*

- $P \wedge (Q \vee R) = (P \wedge Q) \vee (P \wedge R)$,
- $P \vee (Q \wedge R) = (P \vee Q) \wedge (P \vee R)$.

Lemma 5.4. *And finally we get to De Morgan's law's again: Suppose P and Q are Booleans then*

- $\neg(P \vee Q) = (\neg P) \wedge (\neg Q)$,
- $\neg(P \wedge Q) = (\neg P) \vee (\neg Q)$.

Proof. We give just one example of how you would prove such a statement

P	Q	$(P \vee Q)$	$\neg(P \vee Q)$	$\neg P$	$\neg Q$	$((\neg P) \wedge (\neg Q))$
T	T	T	F	F	F	F
T	F	T	F	F	T	F
F	T	T	F	T	F	F
F	F	F	T	T	T	T

Observing that the fourth column $\neg(P \vee Q)$ and the seventh column $((\neg P) \wedge (\neg Q))$ are always the same proves that these expressions are the same as functions. □

Definition 5.4 (tautologies)**.** If $f : \mathcal{B}^n \to \mathcal{B}$ is a Boolean operator then we call f a *tautology* if $f(x) = T$ for all $x \in \mathcal{B}^n$. We call f an *antinomy* if $f(x) = F$ for all $x \in \mathcal{B}^n$.

Tautologies are useful because they describe ways in which we can make logical arguments. For example, if we are arguing by contradiction (more on this later) we wish to prove P. We assume $\neg P$ and arise at a contradiction, so we know $\neg P$ is false then we move from this to saying P must be true.

Here are some useful tautologies and their names

$$\begin{array}{ll}
\neg(\neg P) \Leftrightarrow P & \text{double negation elimination} \\
(P \Rightarrow Q) \Leftrightarrow ((\neg Q) \rightarrow (\neg P)) & \text{contraposition} \\
(P \Rightarrow Q) \Leftrightarrow ((\neg P) \vee Q) & \text{definition of implication} \\
(P \Leftrightarrow Q) \Leftrightarrow ((P \Rightarrow Q) \wedge (Q \Rightarrow P)) & \text{definition of equivalence} \\
(P \vee \neg P) & \text{law of the excluded middle} \\
(P \wedge (P \Rightarrow Q)) \Rightarrow Q & \text{modus ponens} \\
((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R) & \text{transitivity of implication} \\
((\neg P) \Rightarrow F) \Rightarrow P & \text{argument by contradiction}
\end{array} \tag{5.2}$$

5.3 Truth tables

Definition 5.5 (Truth tables)**.** A truth table is table which allows you to look up the output of a Boolean operator given its variables. Given a Boolean operator f of arity three the truth table will look like

P	Q	R	an expression for f
T	T	T	$f(T,T,T)$
T	T	F	$f(T,T,F)$
T	F	T	$f(T,F,T)$
T	F	F	$f(T,F,F)$
F	T	T	$f(F,T,T)$
F	T	F	$f(F,T,F)$
F	F	T	$f(F,F,T)$
F	F	F	$f(F,F,F)$

(5.3)

We extend this in the way you would expect to Boolean operators of different arity. We also often evaluate more than one expression.

You have already seen a lot of truth tables in the previous section without me having given them a name.

We often wish to compute truth tables by breaking expressions down to their constituent parts. For example if we want to check that the transitivity of implication is indeed a tautology we can do as follows

P	Q	R	$(P \Rightarrow Q)$	$(Q \Rightarrow R)$	$(P \Rightarrow R)$	$((P \to Q) \wedge (Q \to P))$	$(((P \to Q) \wedge (Q \to P)) \Rightarrow (P \Rightarrow R))$
T	T	T	T	T	T	T	T
T	T	F	T	F	F	F	T
T	F	T	F	T	T	F	T
T	F	F	F	T	F	F	T
F	T	T	T	T	T	T	T
F	T	F	T	F	T	F	T
F	F	T	T	T	T	T	T
F	F	F	T	T	T	T	T

(5.4)

5.4 Quantifiers

Quantifiers are $\forall$ and $\exists$ we want to start using these in our logical expressions.

We can use quantifiers to turn the sentence "for all $n \in \mathbb{N}$ there exists a $p > n$ such that p is prime. First we define the function $Prime : \mathbb{N} \to \{T, F\}$ by $Prime(n) = T$ when n is prime, and F otherwise. Then we can write

$$\forall n(\exists p((p > n) \wedge (Prime(p)))).$$

The order of quantifiers is very important

$$\exists p(\forall n((p > n) \wedge (Prime(p))))$$

means that there exists a prime p that is bigger than every natural number n. Which definitely isn't true.

Remark. When we are using quantifiers there is an ambient set sitting behind our language. We often surpress this set but in the previous expressions above we are always saying *for all n in $\mathbb{N}$* and *for every p in $\mathbb{N}$*.

As with earlier we also have rules for negation and distribution with quantifiers.

Lemma 5.5. *Quantifiers distribute as follows:*

- $(\forall a \in A, S(a) \wedge T(a)) = (\forall a \in A, S(a)) \wedge (\forall a \in A, T(a))$,
- $\exists a \in A, S(a) \vee T(a)) = (\exists a \in A, S(a)) \vee (\exists a \in A, T(a))$.

We also have the negation rules:

- $\neg(\forall a \in A, S(a)) = (\exists a \in A, \neg S(a))$,
- $\neg(\exists a \in A, S(a)) = (\forall a \in A, \neg S(a))$.

As with earlier results these are fairly straightforward to prove just by writing out exactly what everything means.

Chapter 6

Proof

Proof can be a difficult and subtle concept. During your first year you develop a good sense of proof by *seeing lots of proofs*. In this section we will work towards a rigorous notion of what a proof is and study some common proof techniques.

First, to show this is really necessary let us look at some false proofs.

Theorem 6.1 (untheorem)**.**

$$e^i = 1$$

unproof.

$$e^i = (e^i)^{2\pi/2\pi} = (e^{2\pi i})^{1/2\pi} = 1^{1/2\pi} = 1.$$

□

Theorem 6.2 (untheorem)**.** *All triangles are isoceles*

unproof. Consider the triangle ABC. Draw the angle bisector at A and the perpendicular bisector of BC. Call the point where these two intersect P. Now draw a line from P to AB which is perpendicular to AB and call the intersection Y and draw a line from P to AC which is perpendicular to AC and call the intersection point Z.

Now the triangles AYP and AZP are reflections of each other since they share two angles and a length. This means that the length AY is equal to the length AZ and the length YP is equal to ZP.

Now the triangles BPX and APX are also reflections of each other since they both have right angles at X and share the two side lengths either side of the right angle. This implies that the lengths BP and CP are the same.

Then the triangles BPY and APZ are reflections of each other since they share two side lengths and a right angle. This means that the lengths BY and AZ are the same.

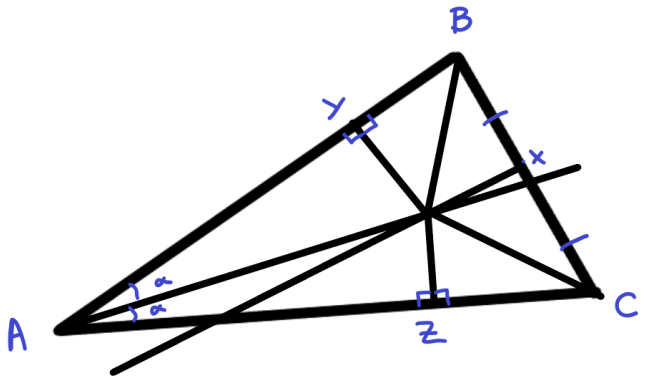


Figure 6.1: A picture of a triangle

Now the length AB is equal to the lengths AY plus YB and the length AC is equal to the lengths AZ plus ZC. Therefore $AB = BC$.

□

Can you spot the problems in the proofs above? If they didn't give obviously false results do you think you would notice that they were false.

6.1 Patterns of proof

In this section I am going to write proofs out using largely the language of logic. My hope is that it will make the different patterns of proof clear. In general it is neither necessary nor desirable to do this.

Definition 6.1. Proof by *direct implication* is the most straightforward kind of proof. Here we want to prove $P \Rightarrow Q$.

Example 6.1. Suppose we want to prove that n being even implies that n^2 is even. We can do this directly:

$$\begin{aligned} &n \text{ even } \Rightarrow \exists\, k \in \mathbb{N}\, s.t. n = 2k, \\ &n = 2k \Rightarrow n^2 = (2k)^2 = 2(2k^2), \\ &n^2 = 2(2k^2) \Rightarrow n^2 \text{ even}. \end{aligned}$$

So you see we just flow from one implication to the next.

Definition 6.2. Proof by *contraposition* is when we use the equivalence $(P \Rightarrow Q) \Leftrightarrow ((\neg Q) \Rightarrow (\neg P))$.

Example 6.2. Suppose we want to prove that if n is odd then n^2 is odd by contraposition. We need to use the following facts:

- (H1) If p is a prime and $a, b \in \mathbb{N}$ then $p|ab \Rightarrow (p|a) \vee (p|b)$.
- (H2) 2 is a prime

We have:

$$\begin{aligned}
&\neg(n^2 \text{ odd}) \Rightarrow (n^2 \text{ even}),\\
&(n^2 \text{ even}) \Rightarrow (\exists\, k \in \mathbb{N}\, s.t.\, n^2 = 2k)\\
&((n^2 = 2k) \wedge (H1) \wedge (H2)) \Rightarrow (2|n)\\
&(2|n) \Rightarrow \neg(n \text{ odd})\\
&((\neg(n^2 \text{ odd})) \Rightarrow (\neg(n \text{ odd})) \Rightarrow ((n \text{ odd}) \Rightarrow (n^2 \text{ odd}))
\end{aligned}$$

Definition 6.3 (proof by contradiction)**.** Proof by contradiction is when we use the fact that $(\neg P \Rightarrow F) \Rightarrow P$. So we assume the opposite of what we are trying to prove and get to a contradiction.

Example 6.3. Suppose we want to prove there does not exist any $r \in \mathbb{Q}$ such that $r^2 = 2$. Again first we need another results:

- We say $p, q \in \mathbb{Z}$ are coprime if there is no prime number k such that $k|p$ and $k|q$. Let us write the function $coprime : \mathbb{Z}^2 \to \{T, F\}$ to tell us if two numbers are coprime.
- If $r \in \mathbb{Q}$ then there exists $p, q \in \mathbb{Z}$ with $coprime(p, q) = T$ such that $r = p/q$

$$\begin{aligned}
&\neg(\nexists\, r \in \mathbb{Q}\, s.t.\, r^2 = 2) \Rightarrow (\exists\, r \in \mathbb{Q}\, s.t.\, r^2 = 2)\\
&(\exists\, r \in \mathbb{Q}\, s.t.\, r^2 = 2) \Rightarrow (\exists\, p, q, \in \mathbb{Z}\, s.t. (p^2 = 2q^2) \wedge ((coprime(p, q))))\\
&(p^2 = 2q^2) \Rightarrow (2|p^2) \Rightarrow (2|p)\\
&(\exists\, p, q, \in \mathbb{Z}\, (p^2 = 2q^2) \wedge (2|p)) \Rightarrow (\exists\, k \in \mathbb{Z} s.t.\, p = 2k) \wedge (4k^2 = 2q^2))\\
&(4k^2 = 2q^2) \Rightarrow (2k^2 = q^2) \Rightarrow (2|q)\\
&((2|p) \wedge (2|q) \wedge (coprime(p, q))) \Rightarrow F\\
&\neg(\nexists\, r \in \mathbb{Q}\, s.t.\, r^2 = 2) \Rightarrow F
\end{aligned}$$

Definition 6.4. Proof by construction. This method of proof is about showing that something exists.

Example 6.4. Suppose we want to prove that every quadratic equation with integer coefficients has at least one solution in $\mathbb{C}$. i.e. we want to show if $a, b, c \in \mathbb{Z}$ then the equation

$$ax^2 + bx + c = 0$$

has at least one solution. We can check (and I'm sure you have before) that

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

is a solution. So we have demonstrated the existence of at least one solution.

Another good example of proof by construction is the proof of Cantor-Schroeder-Bernstein.

6.2 Proof by induction

A particularly important pattern of proof is *proof by induction*. Induction is a key property of how the natural numbers work.

Definition 6.5 (proof by induction)**.** Suppose that P is a property that could or could not hold for each natural number. We can think of P as a function $\mathbb{N} \to \{T, F\}$. Suppose the following hold

- $P(0) = T$ (alternatively we say $P(0)$ holds) - this is called the base case,
- $\forall\, n \in \mathbb{N}\, P(n) \Rightarrow P(n+1)$ - this is called the inductive step,

then we can conclude that $P(n)$ holds for every n. We can write this as $\forall n \in \mathbb{N}\, P(n)$.

Remark. Proof by induction works equivalently (by relabling things) if we start at $n = 1$ or in fact from any number k we just have to alter the conclusion to say $\forall n \geq k\, P(n)$.

Example 6.5. Let $P(n)$ be the statement that the sum of the first n odd numbers is n^2.

Base case: For $n = 1$ the sum of the first odd number is $1 = 1^2$.

Inductive step: Suppose that

$$\sigma_{k=1}^{n}(2k-1) = n^2$$

then

$$\sigma_{k=1}^{n+1}(2k-1) = n^2 + 2(n+1) - 1 = n^2 + 2n + 1 = (n+1)^2.$$

So we have shown $P(n) \Rightarrow P(n+1)$.

Therefore we have shown $P(n)$ holds for all $n \geq 1$.

Definition 6.6 (well ordering principle)**.** Suppose that $S \subset \mathbb{N}$ and $S \neq \emptyset$ then S has a *smallest element.*

A classic example of using the well ordering principle is the prime factorisation theorem.

Theorem 6.3 (prime factorisation)**.** *Every $n \in \mathbb{N} - \{0\}$ is the product of prime factors.*

Proof. Let C be the set of all natural numbers that aren't the product of prime factors. We want to show C is empty.

We assume for contradiction that $C \neq \emptyset$ then by the well ordering principle C has a least element. Let us call this element m.

If m only has divisors 1 and m then m is prime so is the product of prime factors which would be a contradiction.

If m has another divisor $k \neq 1, m$ then we must have $m = kj$ for some $k, j \in \mathbb{N} - \{0\}$. This implies $k, j < m$ so both k and j are the product of prime factors. This implies m is the product of prime factors which is a contradiction.

Therefore C must be empty. □

Definition 6.7 (strong induction)**.** Suppse P is a property that could or could not hold for each of the natural numbers. Suppose that the following are true.

- $P(0)$ holds,
- If $P(k)$ holds for all $k < n$ then $P(n)$ holds,

then we can conclude that $P(n)$ holds for all $n \in \mathbb{N}$.

Theorem 6.4 (unique prime factorisation)**.** *Every natural number $n \geq 2$ has a* unique *prime factorisation.*

Proof. In this proof we are going to assume the fact that if p is a prime and $p|ab$ then $p|a$ or $p|b$. If you are doing ma132 you will see this proved later.

For the base case 2 has a unique prime factorisation.

Suppose that for every $k < n$ that k has a unique prime factorisation.

We already know that n has a prime factorisation from the prime factorisation theorem. It remains to show that it is unique.

Suppose that there are two prime factorisations

$$n = p_1 \dots p_k = q_1 \dots q_j.$$

By re-ordering we can assume that $p_1 \leq p_2 \leq ...$ and $q_1 \leq q_2 \leq ....$

If $p_1 = q_1$ then n/p_1 has a unique prime factorisation so we must have $p_2 = q_2$ etc.

If $p_1 \neq q_1$ then without loss of generality $p_1 < q_1$.

$$p_1 p_2 ... p_k - p_1 q_2 q_3 ... q_j = n(1 - p_1/q_1) = (q_1 - p_1)q_2 ... q_n.$$

If we call $m = (q_1 - p_1)q_2 ... q_n$ then the expression on the left tells us that $p_1|m$ and the expression on the right tells us that $m < n$ since $q_1 - p_1 < q_1$. Therefore m has a unique prime factorisation and $p_1|m$. We know that $q_k > p_1$ for all k so we must have $p_1|(q_1 - p_1)$ but this would imply that $p_1|q_1$ which is a contradiction to q_1 being prime with $p_1 < q_1$. So we cannot have $p_1 \neq q_1$. □

Theorem 6.5. *Induction, the well ordering principle and strong induction are all equivalent.*

Proof. (Induction ⇒ Well ordering principle): Given a set $S \subset \mathbb{N}$ let us assume that S has no least element. Then let $P(n)$ be the property than $S \cap [[n]] = \emptyset$.

If $0 \in S$ then 0 would be the least element of S so $S \cap [[0]] = \emptyset$. This is the base case.

If $[[n]] \cap S = \emptyset$ then if $n + 1 \in S$ then $n + 1$ would be the least element of S so $[[n + 1]] \cap S = \emptyset$.

Therefore induction implies that $S = \bigcup_n (S \cap [[n]])$ is empty.

(Well ordering principle ⇒ strong induction):

Suppose that the well ordering principle holds and we have a property P such that $P(0)$ holds and for every n, $(P(k) \forall k < n) \Rightarrow P(n)$. Then set S be the set where P doesn't hold. If $S \neq \emptyset$ by well ordering it has a least element m. By definition of S we must have $P(k)$ holding for all $k < m$ therefore we must have $P(m)$ holds. This shows that S must be empty so $P(n)$ holds for all n.

(Strong induction ⇒ induction):

Suppose that strong induction holds and we have some property P such that $P(0)$ holds and $P(n) \Rightarrow P(n + 1)$ for all n. Then we also have $P(k) \forall k < n \Rightarrow P(n)$ since $P(k) \forall k < n \Rightarrow P(n - 1) \Rightarrow P(n)$ so by strong induction $P(n)$ holds for all n. □

Chapter 7

Some number theory

7.1 Divisors and prime numbers

Let us recall the definition of divisor and prime numbers

Definition 7.1 (divisor)**.** Given two natural numbers n and m we say n is a divisor of m (or $n|m$) if there exists $k \in \mathbb{N}$ such that $m = n \times k$.

Definition 7.2 (prime number)**.** While we define divisor it is worth defining a prime number. We call $p \in \mathbb{N}$ a *prime number* if the only divisors of p are $1, p$.

Now we have some new definitions

Definition 7.3 (greatest common divisor)**.** Given two natural numbers n and m a number q that divides both of them is a common divisor and the largest such number is called the *greatest common divisor* we write $gcd(n, m)$.

This allows us to write a better definition of coprime

Definition 7.4 (coprime)**.** If $gcd(n, m) = 1$ then n and m are coprime.

Theorem 7.1 (division with remainder)**.** *Suppose that $a \in \mathbb{Z}, b \in \mathbb{N} - \{0\}$ then there exists $q \in \mathbb{Z}$ and $r \in [[b]]$ such that*

$$a = bq + r.$$

Proof. If $b = 1$ we just take $q = a, r = 0$ so we can work in the case $b > 1$.

Let us fix b and prove the result by induction on a. If $a = 1$ then $a = 0 \times b + 1$. This is the base case.

Now if we assume that there exists q, r such that $a - 1 = qb + r$ then either $r \in [[b - 1]]$ in which case $a = qb + (r + 1)$ is a solution to our problem, or $r = b - 1$ in which case we write $a = (q + 1)b$. □

Example 7.1. How can we find the greatest common divisor of two numbers. One way to do it is by repeated division.

We can apply repeated division with 81 and 51. We have

$$\begin{aligned} 81 &= 1 \times 51 + 30, \\ 51 &= 1 \times 30 + 21, \\ 30 &= 1 \times 21 + 9, \\ 21 &= 2 \times 9 + 3, \\ 9 &= 3 \times 3. \end{aligned}$$

We can also do this backwards to get

$$\begin{aligned} 9 &= 3 \times 3, \\ 21 &= 2 \times 9 + 3 = (2 \times 3 + 1) \times 3 = 7 \times 3, \\ 30 &= 1 \times 21 + 9 = (1 \times 7 + 3) \times 3 = 10 \times 3, \\ 51 &= (1 \times 10 + 7) \times 3 = 17 \times 3, \\ 81 &= (1 \times 17 + 10) \times 3 = 27 \times 3. \end{aligned}$$

And we deduce from the that $gcd(81, 51) = 3$.

Why does this work? We notice that if $c|a, c|b$ then if $a = qb + r$ then we must have that $c|r$. Continuing on if $c|b$ and $c|r$ and $b = q_2 r + r_2$ then $c|r_2$ and so on. If eventually you end up with a remainder term which is 0 then we terminate. This shoes that if we terminate with remainder r_k then $r_k|a$ and $r_k|b$ and that nothing larger than r_k can divide both a and b.

7.2 Euclid's Algorithm

Taking inspiration from calculations like the one above we can write down a procedure to find the greatest common divisor of a, b.

Definition 7.5 (Euclid's Algorithm)**.** Given $a, b \in \mathbb{N}$ with $a < b$ we can write

$$\begin{aligned} b &= q_1 a + r_1, \quad q_1 \in \mathbb{N}, r_1 \in [[a]], \\ a &= q_2 r_1 + r_2, \quad q_2 \in \mathbb{N}, r_2 \in [[r_1]], \\ r_1 &= q_3 r_2 + r_3, \quad q_3 \in \mathbb{N}, r_3 \in [[r_2]], \\ r_2 &= ... \end{aligned}$$

Eventually this process will terminate because we will have $r_k = 0$ for some k and then we have $r_{k-1} = gcd(a, b)$.

Theorem 7.2 (Euclid's Algorithm)**.** *Given the algorithm above we have the following - This algorithm will terminate. i.e. eventually we have $r_{k-1} = q_{k+1} r_k + 0$. - In this case $r_k = gcd(a, b)$ - There exists $x, y \in \mathbb{Z}$ such that $gcd(a, b) = xa + by$.*

Proof. We take each point in turn.

Firstly we can see that, for every j, $r_j > r_{j+1} > r_{j+2}$ so eventually we must get to 0. Therefore the sequence terminates.

Secondly, we prove this by showing that in each step of the algorithm we preserve the set of common divisors. If $m|r_j$ and $m|r_{j+1}$ then since $r_j = q_{j+2}r_{j+1} + r_{j+2}$ so $m|r_{j+2}$ as well. Equally if $n|r_{j+1}$ and $n|r_{j+2}$ then we must have $n|r_j$ as well. So the set of divisors of r_j, r_{j+1} is the same as the set of divisors of r_{j+1}, r_{j+2}. From this it follows that $gcd(r_j, r_{j+1}) = gcd(r_{j+1}, r_{j+2})$. Therefore itterating backwards we have that for every j that $gcd(r_j, r_{j+1}) = gcd(a, b)$. Consequently at the point where the algorithm terminates, $r_k = gcd(r_k, 0) = gcd(r_k, r_{k+1}) = gcd(a, b)$.

For the last point let us make the claim: For every j we can write $r_j = x_j a + y_j b$ for some $x_j, y_j \in \mathbb{Z}$. Then we can prove this recursively. First we know that $b = q_1 a + r_1$ so $r_1 = -q_1 a + 1 * b$. Now suppose that r_{j-1}, r_{j-2} can be expressed as above. We know $r_{j-2} = q_j r_{j-1} + r_j$ so $r_j = r_{j-2} - q_j r_{j-1} = x_{j-2}a + y_{j-2}b - q_j(x_{j-1}a + y_{j-1}b) = (x_{j-2} - q_j x_{j-1})a + (y_{j-2} - q_j y_{j-1})b$. So this shows the claim by induction. Now the claim implied in particular that $r_k = x_k a + y_k b$ proving the third point in the theorem. □

7.2.1 Geometric interpretation of Euclid's algorithm

We can think of this algorithm pictorially by drawing a rectangle of length a and height b and then q_1 squares of side length b inside it leaving a rectangle of length r_1 and height b and so on…

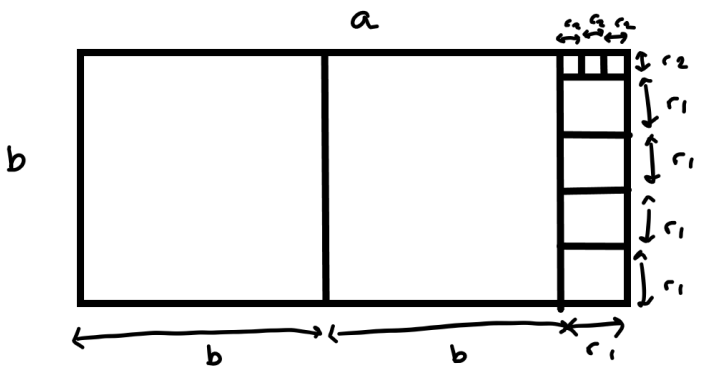
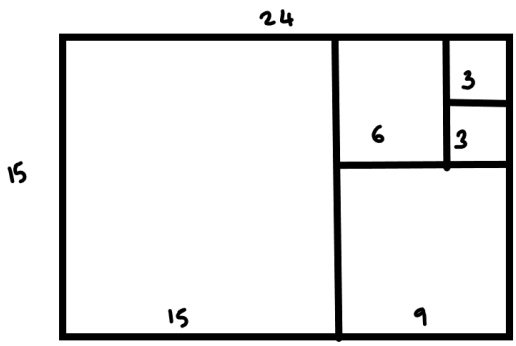


Figure 7.1: Picture showing Euclid's algorithm pictorially

Here is a specific example with the numbers 24 and 15



Continued fractions

Another way of looking at Euclid's algorithm is continued fractions. If $a = qb+r$ then

$$\frac{a}{b} = q + \frac{r}{b}.$$

If $b = q_2 r + r_2$ then

$$\frac{b}{r} = q_2 + \frac{r_2}{r},$$

so

$$\frac{a}{b} = q + \frac{1}{q_2 + \frac{r_2}{r}}.$$

Continuing on like this we can express

$$\frac{a}{b} = q + \frac{1}{q_2 + \frac{1}{q_3 + \ldots}}.$$

In a similar way if we take some real number x we can write it in continued fraction form by writing $q_1 = [x]$ and $q_2 = [1/(x - [x])]$ and so on. Unlike when expressing a rational number this process might not terminate.

In both situations we call this the continued fraction representation of a number.

7.2.2 Bezout's lemma

The final statement in Euclid's algorithm is called Bezout's lemma

Lemma 7.1 (Bezout's Lemma)**.** *If a, b are two natural numbers then there exists $x, y \in \mathbb{Z}$ such that*

$$gcd(a, b) = xa + bc.$$

As a result of this if a, b are coprime and n is any integer then there exists $x, y \in \mathbb{Z}$ such that

$$n = xa + yb.$$

Proof. We have already proved the first part in the discussion of Euclid's algorithm.

For the second if a, b are coprime then $gcd(a, b) = 1$ (this is the definition of being coprime). Then by the first part of the theorem there exists $\tilde{x}, \tilde{y} \in \mathbb{Z}$ such that

$$1 = \tilde{x}a + \tilde{y}b,$$

using this

$$n = (n\tilde{x})a + (n\tilde{y})b.$$

□

Using this we can show a powerful result which we have already seen

Suppose that $a, b \in \mathbb{N}$ and p is a prime number and $p|ab$ then either $p|a$ or $p|b$.

If $p|a$ then we are done so suppose that p does not divide a. Then a, p are coprime so by Bezout's lemma we can write

$$1 = xa + yp$$

and hence

$$b = xab + ypb.$$

As $p|ab$ we know $p|xab$ and from the expression we can see that $p|ypb$ so $p|(xab+ybp)$ so $p|b$.

Now let us use this to prove the fundamental theorem of arithmetic again. Everything is a bit smoother now with more results.

Theorem 7.3 (Fundamental Theorem of Arithmetic)**.** *Any natural number n has a unique factorisation into prime numbers.*

Proof. First let us use strong induction to prove that there is a prime factorisation. The base case is $n = 2$ which is already in prime factorisation. Now suppose that every number less than n can be written as a product of prime factors. Then either n is prime, so it is in prime factorisation or $n = ab$ then $a, b < n$ so they have prime factorisations which allows us to write a prime factorisation for n.

Second we want to prove this factorisation is unique. Suppose $n = p_1p_2 \dots p_k = q_1q_2 \dots q_j$ where all the p_i, q_i are primes. Then $p_1|n$ so $p_1|q_1$ or $p_1|q_2 \dots q_j$. In the first case we must have $p_1 = q_1$ since q_1 is prime. In the second case we have $p_1|q_2$ or $p_1|q_3 \dots q_j$ and so on. We can keep itterating to show that p_1 must appear in the list $q_1, \dots q_j$. We can then repeat this with all the p_i. □

7.3 Chinese remainder theorem

Suppose we are interested in solving two or more linear congruences simultaneously.

Example 7.2. Suppose we would like to find x such that $x \equiv 1 \mod 3$ and $x \equiv 3 \mod 4$. Then we can work modulo 12 since if we have a solution we can see we will get another solution by adding multiples of 12. If $x \equiv 1 \mod 3$ then modulo 12 $x \equiv 1, 4, 7$ or 10. Similarly, if $x \equiv 3 \mod 4$ then $x \equiv 3, 7$ or 11 modulo 12. Therefore $x \equiv 7$ modulo 12 is our unique solution up to adding multiples of 12.

Slightly differently suppose we would like to have $x \equiv 2 \mod 6$ and $x \equiv 3 \mod 4$ then again we want to work modulo 12. From the first constraint we have $x \equiv 2$ or $x \equiv 8$ and from the second we have $x \equiv 3, 7$ or 11 as before. In this example we see there are no possible solutions.

Theorem 7.4 (Chinese Remainder Theorem)**.** *Suppose that $n_1, \dots, n_k$ are pairwise coprime integers and $a_1, \dots, a_k$ are integers with $a_i \in [[n_i]]$ for every i. Then there exists an integer x with*

$$x \equiv a_i \quad mod n_i, \quad i = 1, \dots, k.$$

Furthermore all the solutions are equivalent modulo $N = n_1 \dots n_k$.

Proof. Let $m_i = \Pi_{j \neq i} n_j$. Then n_i and m_i are coprime so by Euclid's algorithm there exists x_i, y_i such that $1 = x_i n_i + y_i m_i$ so $e_i = y_i m_i = 1 - x_i n_i$ so $e_i \equiv 1 \mod n_i$ and $e_i \equiv 0 \mod n_j$.

Then let $x = \sum_{i=1}^{k} a_i e_i$ then this satisfies the conditions of the theorem.

Now suppose we have another y satisfying the congruences. Then $x \equiv y \mod n_i$ for every i. Therefore, $n_i || x - y|$ for every i. Since all the n_i are coprime this means that $N || x - y|$. □

Remark. The Chinese remainder theorem is very old. It dates back to Sunzi in the 3rd to 5th Century. It can be used to do apparently complicated computations very quickly and is used in important algorithms today such as RSA cryptography and Fast Fourier transform.

7.4 Fermat's Little Theorem and Euler's Theorem

Definition 7.6. A number a is called a emph{unit} modulo n if there exists b such that $ab = 1 \mod n$. (We call b a multiplicative inverse modulo n.)

Lemma 7.2. *A number a is a unit modulo n if and only if a and n are coprime.*

Proof. If a and n are coprime then $1 = xa + yn$ in which case $xa \equiv 1 \mod n$.

If there exists x such that $xa \equiv 1 \mod n$ then $xa = 1 + yn$ and $gcd(a, n)|(xa - yn)$ so $gcd(a, n) = 1$. □

Definition 7.7 (Euler Totient Function)**.** The Euler Totient function $\phi(n)$ counts the number of natural numbers smaller than n that are coprime to n. Alternatively the number of units modulo n. Notice that $\phi(1) = 1$.

Lemma 7.3. *If p is prime then $\phi(p) = p - 1$.*

Example 7.3. Another example is $n = 12$ the numbers smaller than n which are coprime are $1, 5, 7, 11$ so $\phi(12) = 4$

Lemma 7.4. *If p is a prime then*

$$\phi(p^k) = p^k - p^{k-1}.$$

Proof. The only way that $gcd(m, p^k) \neq 1$ is if m is a multiple of p. There are p^{k-1} multiples of p less than p^k. □

Lemma 7.5. *If n and m are coprime integers then*

$$\phi(nm) = \phi(n)\phi(m).$$

Proof. Recall that the units modulo n are exactly the numbers which are coprime to n. Let us write the set of units modulo n as $(\mathbb{Z}/n\mathbb{Z})^\times$.

Suppose that k is a unit modulo n and j is a unit modulo m then by the Chinese remainder theorem there exists a unique l such that $l \equiv k \mod n$ and $l \equiv j \mod m$. Equally if l is a unit modulo mn then there is some q such that $lq \equiv 1 \mod mn$ so l is a unitt modulo n and a unit modulo m. This shows we have a bijection between $(\mathbb{Z}/n\mathbb{Z})^\times \times (\mathbb{Z}/m\mathbb{Z})^\times$ and $(\mathbb{Z}/nm\mathbb{Z})^\times$. Therefore the two sets have the same size. This gives our conclusion. □

Proposition 7.1. *We have a formula for Euler's totient function given by*

$$\phi(n) = n\Pi_{p\,prime, p|n}(1 - 1/p).$$

Proof. This follows from our previous results. By the fundamental theorem of arithmetic we can write $n = p_1^{k_1} ... p_j^{k_j}$ then

$$\begin{aligned}\phi(n) &= \phi(p_1^{k_1}) ... \phi(p_j^{k_j}) \\ &= p_1^{k_1}(1 - 1/p_1) ... p_j^{k_j}(1 - 1/p_j) \\ &= n(1 - 1/p_1) ... (1 - 1/p_j).\end{aligned}$$

□

Theorem 7.5 (Divisor sum)**.** *If n is an integer then*

$$n = \sum_{d|n} \phi(d)$$

Proof. We prove this by strong induction. We can see that it is true when $n = 1$ then the only divisor of 1 is itself and $\phi(1) = 1$. This gives us the base case.

For the inductive step we assume that for all $m < n$ we have

$$m = \sum_{d|m} \phi(d).$$

Then there are two cases. If n is prime then $n = n - 1 + 1 = \phi(n) + \phi(1)$ and we are done.

If n isn't prime then $n = pm$ where p is prime and $m < n$. From this we can compute:

$$\begin{aligned}\sum_{d|n} \phi(d) &= \sum_{d|m} (\phi(d) + \phi(pd)) \\ &= \sum_{d|m} \phi(d)(1 + \phi(p)) \\ &= \sum_{d|m} \phi(d)p \\ &= pm = n.\end{aligned}$$

□

Proposition 7.2. *If p is prime and a is not a multiple of p then*

$$a^{p-1} \equiv 1 \mod p.$$

Proof. Let us consider $a, 2a, 3a, \ldots, (p-1)a$ as elements in $\mathbb{Z}/p\mathbb{Z}$. We claim that they are all distinct. This is because since a, p are coprime, there exists x such that $xa = 1$ modulo p. So if $ka \equiv ja \mod p$ then $kab \equiv jab \mod p$ so $k \equiv j \mod p$. So if $k, j \in [[p]]$ this would imply $k = j$.

So $a, 2a, \ldots, (p-1)a$ considered modulo p must be $1, 2, \ldots, (p-1)$ in some order. Hence

$$a \times 2a \times \cdots \times (p-1)a \equiv (p-1)! \mod p.$$

Which rewriting is

$$a^{p-1}(p-1)! \equiv (p-1)! \mod p.$$

Now $(p-1)!$ is coprime to p since p is prime. So there exists c such that $c(p-1)! \equiv 1 \mod p$ hence multiplying the equation above by c we get

$$a^{p-1} \equiv 1 \mod p.$$

□

Theorem 7.6 (Fermat's Little Theorem)**.** *Suppose that* $gcd(a, n) = 1$ *then* $a^{\phi(n)-1} \equiv 1 \mod n$.

Proof. The proof is very similar to above so I will leave the details to you!

Consider the set of integers $\{\text{ka}\}$ where $gcd(k, n) = 1$ and $k \in [[n]]$ and multiply them all together. □

Definition 7.8 (primitive root)**.** We say a is a *primitive root* modulo n if for every $b \in \mathbb{Z}/n\mathbb{Z}$ with $gcd(b, n) = 1$ there exists an x such that $b \equiv a^x \mod n$.

You have hopefully seen in Algebra 1 that $(\mathbb{Z}/n\mathbb{Z})^\times$ is the set of elements in $\mathbb{Z}/n\mathbb{Z}$ which have multiplicative inverses. This set forms a group under multiplication and the primitive roots are the generators of this group.

Example 7.4. Working modulo 5 the primitive roots are $2, 3$ we can see that $4^2 = 1$ which means it cannot be a primitive root.

Lemma 7.6. *Suppose that* p *is prime and* $r|p-1$ *then there are exactly* r *elements,* b*, of* $(\mathbb{Z}/p\mathbb{Z})^\times$ *with* $b^r \equiv 1 \mod p$.

Proof. This is straightforward once you can talk about polynomials modulo p but we don't want to talk about it so we're leaving it unproved. □

Proposition 7.3. *If* p *is prime and there exists an element* a *of order* r *then there are exactly* $\phi(r-1)$ *elements of order* r.

Proof. Give an element a of order r then we have distinct elements $a, a^2, \ldots, a^{r-1}$. There are at most r elements b such that $b^r \equiv 1 \mod p$ so these are all the possible elements of order less than or equal to r. Now if $gcd(k, r-1) = 1$ then for every $m < r-1$ we have $mk \not\equiv 0 \mod r-1$ so $a^{mk} \not\equiv 1 \mod p$. However $a^{rk} \equiv 1 \mod p$. Therefore a^k is another element of order r. Therefore we have at least $\phi(r-1)$ elements of order r. □

Corollary 7.1. *If* p *is prime then there are exactly* $\phi(p-1)$ *primitive roots.*

Proof. We know that the total number of elements in $(\mathbb{Z}/p\mathbb{Z})^\times$ is $\sum_r$ number of elements of order r. We also know that the size of $(\mathbb{Z}/p\mathbb{Z})^\times$ is $p-1$ and so there can only be elements of order r if $r|p-1$.

We also know know that $p - 1 = \sum_{d|p-1} \phi(d)$. Putting all this together gives our result. □

Chapter 8

Algorithms and Algorithmic complexity

8.1 Algorithmic Complexity

Algorithmic complexity is a measure of how complicated a procedure is by counting roughly how many operations are required to complete it.

8.1.1 Big and little O notation

When we are talking about complexity (and in lots of other situations later in your degree) we are often only interested in the asymptotic behaviour of a function.

For example if we want to know the complexity of an algorithm to multiply two numbers with at most n digits together we have a function of n and we are only really interested in describing how it behaves when n is very large. Big and little O notation is a way of doing this.

Definition 8.1 (Big O)**.** Suppose $f(n)$ and $g(n)$ are function of n and there exist real numbers C, n_0 such that for $n \geq n_0$ then

$$|f(n)| \leq Cg(n)$$

then we say

$$f(n) = O(g(n)).$$

This is most commonly used when $g(n)$ is a simple function like n^k for some k or $\log(n)$ or a^n or something similar.

Example 8.1.

- The function $2n^3 + 3n - 102 = O(n^3)$.
- $\phi(n) = O(n)$ since $\phi(n) \leq n$.

While we are defining big O notation it is useful to define little o notation

Definition 8.2 (little o notation)**.** If $f(n), g(n)$ are functions and for every $c > 0$ there exists n_c such that if $n > n_c$ then

$$|f(n)| \leq cg(n).$$

We can think of this as saying $f(n)$ becomes insignificant compared to $g(n)$ as n becomes large.

Example 8.2. We have $n^2 = o(n^3)$.

8.2 Running times of algorithms

We use big O notation to talk about how long it will take to run an algorithm as the input becomes large. It is easiest to understand this by looking at some examples.

Example 8.3 (Addition)**.** Suppose we are adding two n digit numbers $a_1 ... a_n$ and $b_1 ... b_n$ using our standard method of addition where we do $a_n + b_n$ and enter the unit part of this as the unit part of the sum then possibly carry the 1 if $a_n + b_n \geq 10$. Then we add $a_{n-1} + b_{n-1}$ and possibly add the additional 1 we carried if necessary.

If we want to count the steps it depends on precisely what we list as a step. If we only think we are doing 1 step when adding and carrying for each place value then we are performing n steps. If we think of each addition and carrying step as a separate step then we do $3n$ steps. However in either case we say the algorithm is $O(n)$.

Example 8.4 (Multiplication)**.** Now suppose we think about multiplication using the long multiplication algorithm then for two n digit numbers we need to multiply a_k by b_j for each k, j (which is n^2 steps) and then perform n^2 addition steps of what are effectively at most two digit numbers. So this algorithm is O(n^2).

Example 8.5 (Checking divisibility)**.** Division is no worse than multiplication. The algorithms are a bit more complicated to state. One example is given an n digit number, a and an m digit number, b with $m \leq n$ and we want to know whether $b|a$ then if a/b were an integer it would have at most $n - m + 1$ digits and at least $n - m$. So we are have around 20 possible candidate for a/b. Now we just check multiplying all 20 candidates with m which is a multiplication operation with $m \times (n - m + 1)$ steps so $O(n^2)$ and we do this 20 times so checking divisibility is $O(n^2)$.

Example 8.6 (Naive primality testing)**.** Suppose we want to test whether a number n is prime. The most obvious way to do this is to check whether we can find any other $m \neq n$ such that $m|n$. We notice here that we only need to check $m \leq \sqrt{n}$ as if $ab = n$ then one of a, b must be smaller than $sqrtn$. If we suppose it is one step to check whether n is divisible by m. Now each division operation is a check and the number of digits in n is around $\log_{10}(n)$ so if we are interested in testing an N digit number to see if its prime we need around $10^{N/2}N^2$ operations so the algorithm is $O(10^{N/2}N^2)$.

Remark. Notice in all of these examples we are using our understanding of the complexity of less complicated operations to find the complexity of more complicated operations.

Definition 8.3 (Fibonacci Numbers)**.** The Fibonacci numbers are defined by

$$f(k+2) = f(k+1) + f(k),$$

and $f(0) = f(1) = 1$.

We discover that if we want to count how many steps Euclid's algorithm takes then the Fibonacci numbers appear.

Theorem 8.1 (Complexity of Euclid's Algorithm)**.** *If $a > b$ and Euclid's algorithm takes k steps to complete then $a \geq f(k+2)$ and $b \geq f(k+1)$.*

Proof. We prove this by induction on k.

In the base case if the algorithm takes 1 step then we must have $b \geq 1 = f(2)$ and $a > b$ so $a \geq 2 = f(3)$.

Now we assume it is true for $k-1$ steps. Then suppose we have a, b which need k steps for Euclid's algorithm. Then $a = qb+r$ and b, r need $k-1$ steps for Euclid's algorithm. So by our induction hypothesis we know that $b \geq f(k+1), r \geq f(k)$. Then $a = qb + r \geq b + r \geq f(k) + f(k+1) = f(k+2)$. □

Remark. The above theorem means that the Fibonacci numbers represent the worst case for Euclid's algorithm. i.e. $f(k+2), f(k+1)$ are the smallest numbers that will need k steps to complete Euclid's algorithm for.

We can show that for k large $f(k) \approx \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^k$ so $k \approx \log(f(k))/\log((1+\sqrt{5})/2)$. Therfore the complexity of Euclid's algorithm when the larger number is n is $O(\log(n))$.

Definition 8.4 (polynomial time)**.** We say that an algorithm takes *polynomial time* if the complexity is $O(n^k)$ for some k (in the input size n).

Remark. Problems where there is no known polynomial time algorithm are considered very hard for computers to do. These provide a good oportunity for creating codes and encryptions which are hard to break.

Chapter 9

Cryptography

In this chapter we are going to look at how tools from this course can be used in encrypting information.

Definition 9.1 (key in cryptography and public key)**.** When we encrypt something we often use a *key* to encrypt and decrypt it. This key is a number or a sequence of letters (or sometimes something more complicated) which allows us to decrypt a code.

A public key is used in a situation where A wants to send some information privately to B. They each have separate private information which the other doesn't know. There is also some public information which both can see. This is called the public key. We need this to work in such a way that A's private information and the public key are sufficient to encrypt the data so that B's private information and the key can decrypt the data but the public key alone is not sufficient to decrypt the data and A and B don't need each others private information to do the encryption or decryption.

I think it is much easier to understand this concept once we've seen some examples.

Remark. In this course we aren't really interested in how you encrypt or decrypt data once you have the key. There are lots of ways of doing this! We just want to come up with some good methods for creating these public keys. Most of these methods will involve understanding algorithmic complexity.

9.1 Discrete logarithms and Diffie-Hellman

A discrete logarithm is essentially taking the logarithm in modular arithmetic.

Example 9.1. Suppose we know that $2 \equiv 3^x \mod 5$ and we want to find x. We call x the discrete logarithm.

The only straightforward way to do this is to compute 3^y modulo 5 until we find a y satisfying this problem.

Definition 9.2 (Discrete Logarithm problem)**.** Suppose p is a prime and $a, b \in (\mathbb{Z}/p\mathbb{Z})^\times$. The discrete logarithm problem is to find n so that $a^n \equiv b \mod p$.

Definition 9.3 (Diffie-Hillman problem)**.** Suppose p is a prime and $a \in (\mathbb{Z}/p\mathbb{Z})^\times$ and n, m are integers. Suppose we know a, a^n, a^m but we don't know n or m the Diffie-Hillman problem is to compute a^{nm}.

Remark. At the moment nobody knows a polynomial time algorithm to solve either the discrete logarithm problem or the Diffie-Hillman problem. We can see that if we could solve the discrete logarithm then we could definitely solve Diffie-Hillman.

Definition 9.4 (Diffie-Hillman key exchange)**.** We are looking at encrypting and decrypting information using the key $a^{nm} \mod p$.

Person A and B both generate a prime p and an element $a \in (\mathbb{Z}/p\mathbb{Z})^\times$ which are made public.

Person A then decides their private number n and person B decides their private number m. They don't exchange these numbers or tell anyone else.

Then person A computes a^n and person B computes a^m and they make this information public. This means that person A can compute a^{nm} since they know a^m and n so can compute $(a^m)^n$. Similarly person B can also compute a^{nm}. However, no other person is able to compute a^{nm} without solving the Diffie-Hillman problem.

This means that A and B can safely use a^{nm} to encrypt and decrypt their secret messages without worrying about anyone being able to intercept them.

9.2 RSA Cryptography

RSA Cryptography is named for Rivest, Shamir and Adelman who invented it. It is used in a more asymmetric setting to the Diffie-Hillman problem. Here we want anyone to be able to encrypt a message but only one person to be able to decrypt any of the messages. This comes up a lot with passwords sent to websites for example.

Definition 9.5 (RSA Cryptography)**.** Person A choose two large prime numbers p, q they then compute $n = pq$ and $\phi(n) = (p-1)(q-1)$. Person A then chooses $ein[[\phi(n)]]$ which is coprime to $\phi(n)$ (it is often a good idea to choose e prime). Using Euclid's algorithm they compute d such that $ed \equiv 1 \mod n$. Now person A has the information $p, q, n, \phi(n), e, d$. They make n, e public. It is very hard to compute (i.e. no known polynomial time algorithm) any of the private information from the public information (essentially you would have to factorise n then you can do it easily but it is very hard to factorise n).

Now suppose person B wants to encrypt a message so that only person A can decrypt it. B doesn't need any of the private information. Person B encodes his message into a number m modulo n and then computes m^e modulo n.

Now if person A wants to decrypt the message they compute $(m^e)^d$ modulo n. Then since $de \equiv 1 \mod \phi(n)$ we know that $de = k\phi(n) + 1$ for some k. Then $m^{ed} \equiv m^{k\phi(n)+1} \equiv m \mod n$. Where here we used the fact that $m^{\phi}(n) \equiv 1 \mod n$ using Fermat's little theorem.