Modern Computation

A Unified Approach

S. S. Chandra
Ph.D
February 2019

Draft
Version: 0.1

“Computing Science is no more about computers than
Astronomy is about telescopes.”

Edsger W. Dijkstra (1930-2002)

CHAPTER 2

Mathematical Preliminaries

"God made the integers, all else is the work of man."

Leopold Kronecker (1823-1891)

Mathematics is the basis of everything that computer science is built upon. Indeed, the first "computer" was invented not as circuits or hardware, but as an idea and abstract concept 30 years before its practical realisation. Sir Alan Turing [1937] built this simple computer, a basic machine now called a Turing machine, out of pure logic and mathematical theorems. He then extended its scope to include stored instructions, introducing computers programs in the process, and the ability to run other machines, a concept now know as universality, in order to solve one of the Hilbert's problems[1]. These concepts are discussed briefly at the end of this chapter in section 2.6 on page 15 and in detail within later chapters.

In this chapter, we will cover the mathematical preliminaries required to understand these results and the use of theorems to construct logical ideas. These will be used as a basis to describe and design computers and their languages just as Turing [1937] did with his machines. To get a better fundamental understanding of the concepts, let us begin with most basic of all mathematical concepts, numbers.

2.1 Numbers

Counting is a fundamental concept we learn in our early childhood. Yet some of the most amazing results in science and some of the most basic questions remain unanswered about the numbers used for counting, namely the integers

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \ldots \tag{2.1}$$

[1] A set of 23 important mathematical problems proposed by the renowned mathematician David Hilbert, the solutions of which became the defining achievements of twentieth century mathematics.

2.1.1 The Natural Numbers

We can create a collection of all possible integers and refer to them as a single entity called a set. This set of all possible positive integers is referred to as the natural numbers $\mathbb{N}$

$$\mathbb{N} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \ldots\} \tag{2.2}$$

Then a number of objects, say apples on a tree, would be a number in this set. An orchid, which would be a collection of apple trees each having a number of apples, would be a subset of this set. For example, an apple tree could have 5 apples and an orchid $\mathcal{A}$ could have 4 trees with

$$\mathcal{A} = \{2, 3, 5, 6\} \tag{2.3}$$

Thus, $\mathcal{A}$ is a subset of $\mathbb{N}$. For more compact notation, we can write the symbol $\in$ to represent if an element is part of a set. For example, we can write $2 \in \mathbb{N}$, since 2 is a natural number, but also $2 \in \mathcal{A}$.

We can draw the set $\mathbb{N}$ as distances from the origin (i.e. zero) as shown in figure 2.1. In theory, we

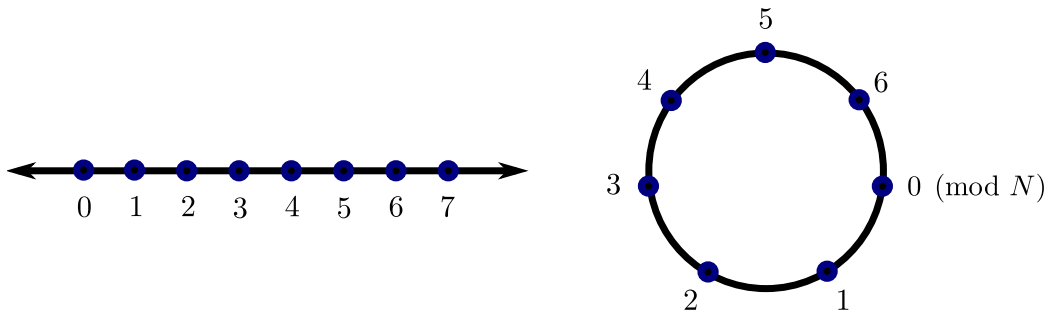


Figure 2.1: Two different ways to represent the natural numbers $\mathbb{N}$. The number line (left) and the finite circle (right).

could count forever, so this number line representation of $\mathbb{N}$ goes on forever. Because of this, we say that the set of natural numbers $\mathbb{N}$ has an infinite number of elements.

2.1.2 Operations on Numbers

You may ask, what else can we do with this set, and indeed the number line representation, that is useful? Well, we can represent the process of counting by adding a unit repeatedly. In fact, we can construct every other number in $\mathbb{N}$ from zero just by repeatedly adding one. This is easily seen by observing that we 'hop' from one point on the number line to the one immediately on the right of this point until we stop counting. If we continue forever, then we get our number line of figure 2.1. In later chapters, we will use this simple, yet mundane operation to construct an entire universal computational system!

What if we use another number other than one? Let's say we use the number two. We would then obtain the set of even numbers, but we may define a more convenient operation than repeatedly adding a chosen number x called multiplication. If we have a number b and we wish to add it a times, then we write $c = a \cdot b$ or $c = ab$ for short. We can then create a grid representation of the operation such as

$$\begin{matrix} c & c & c \\ c & c & c \end{matrix}$$

for $a = 2$ and $b = 3$. Then the number of c elements are evident from how many times it appears in the grid by relating the concept to how the area of a square is computed. The number c is referred to as the product of a and b, where the numbers a and b are called factors of c.

This poses two very important questions:

1. Can we create the number line (as we did for the addition of the unit) using products and their factors?

2. How can we undo the product of two numbers?

The later is the basis of all financial transactions because this property is an integral part of most secure encryption schemes. The former will lead us into the realm of the longest unsolved problem in all science - the distribution of prime numbers.

2.1.3 Prime Numbers

Let us begin by trying to create the number line of figure 2.1 on the preceding page using the multiplication of integers. Starting with the set containing the number two as an element of our set $\mathcal{P}$, since multiplication involving zero and one is trivial, gives us all the numbers that involve repeatedly multiplying two to itself.

We can define a more convenient operation for repeated multiplication as exponentiation or powers. The power of b raised to a power a can be defined as multiplying b a total of a times or simply written as b^a. Thus, making two an element of our set allows one to construct all powers of two written as 2^a, such as 4 and 8 etc. Our set $\mathcal{P} = \{2\}$ now gives our resulting set of numbers

$$\mathcal{N} = \{2, 4, 8, 16, 32, \ldots\}\,. \tag{2.4}$$

We have far from succeeded, since the next number 3 is missing, including all of its powers. By making $\mathcal{P} = \{2, 3\}$ improves our resulting set $\mathcal{N}$, since it contains not only powers of 3, but also numbers whose factors involve 2 and 3

$$\mathcal{N} = \{2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 27, 32, \ldots\}\,. \tag{2.5}$$

The next number missing from $\mathcal{N}$ is 5 and by making $\mathcal{P} = \{2, 3, 5\}$ improves our number set even further

$$\mathcal{N} = \{2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 20, 24, 25, 27, 30, 32, \ldots\}\,. \tag{2.6}$$

Our set $\mathcal{N}$ is slowly becoming complete with powers of 5 and all numbers with factors 2, 3 and 5 are joined to the number set. The same process can be repeated by making the missing numbers 7 and 11 so that $\mathcal{P} = \{2, 3, 5, 7, 11\}$. However, the set $\mathcal{N}$ remains incomplete with the numbers 13 and 17 missing despite filling in more missing numbers. We can continue this process, but we note that the set $\mathcal{P}$ has some interesting properties.

The set $\mathcal{P}_n$ is the set of prime numbers up to the integer n. These numbers have no factor other than itself and unity, while numbers with multiple factors referred to as composite. In a way, prime

numbers, or simply 'primes', are the multiplicative building blocks of the natural numbers $\mathbb{N}$. The process described above is known as the Sieve of Eratosthenes, where the primes are sieved out from other numbers as those that do not have factors other than themselves. If you continue the process till all primes are found for $n = 32$, then

$$\mathcal{P}_{32} = \{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31\} . \tag{2.7}$$

There is seemingly no pattern to the primes, and indeed no explicit pattern or formula for producing has been found. Given a number n, it is also difficult to predict how many primes there will be less than n. Significant progress has been made in determining an approximation for the number of primes less than n and is known as the Prime Number Theorem. Predicting the distribution of prime numbers is the Holy Grail of mathematics for which there is a million dollar prize [2].

But what about undoing a multiplication? Given numbers a, b and $c = ab$, how can be solve for say b? We need the concept of integer divisibility.

2.2 Divisibility

We introduce a new operation called division to initially undo the operation of multiplication, where c divide by a to re-obtain b would be written as

$$b = c/a. \tag{2.8}$$

Clearly, something strange happens in equation (2.8) when $a > c$. We no longer have an integer but a fraction or a number less than unity. Thus, if we want a number system that is self contained, i.e. a system where no new mysterious numbers appear without warning, we either need to expand our set of numbers or define a proper way of dividing only integers. In the next section, we define new number sets until we have a complete system, and in section 2.2.3 on page 8 we define a system where integer division is always possible under some constraints.

2.2.1 Rational Numbers

Let us begin making the system of $\mathcal{N}$ more complete by adding the numbers a/b for all the possible values of a and b to the set. This creates the new set of numbers called the rational numbers $\mathbb{Q}$ or simply rationals. It is clear that the set $\mathbb{N}$ is a subset of $\mathbb{Q}$, because $a \in \mathbb{N}$ and $b = 1$ for all elements of $\mathbb{N}$. The rationals are usually easy to spot when using decimal representation as those numbers that either terminate or repeat a sequence of digits after the decimal point.

For example, $1/2 = 0.5$ and $1/3 = 0.\overline{3}$, where the bar represents the digits that repeat. Also note that we can always 'convert' a rational number $c = a/b$ to an integer by multiplying by b, so that $cb = a$ by construction. These two reasons are why the integers and rationals are sometimes used inter-changeably and why integers are called rationals in some mathematical texts, particularly when defining variables within equations.

The ancient Greek mathematicians thought that these numbers were the perfect numbers and that all of mathematics could be done using only integers, as our quote from Kronecker at the beginning of

[2] A millennium problem prize offered by the Clay Mathematics Institute.

the chapter suggests. However, a simple construction involving triangles spoiled their perfect number system. As a side note though, another approach involving integers developed long after the ancient Greeks can indeed create a perfect number system that can even represent floating point numbers exactly. This system is known as the p-adic numbers, where p is prime, but its discussion is outside the scope of this book.

2.2.2 Irrational Numbers

Consider the simple act of constructing a right-angled triangle whose two sides are of length one. What is the length of the third side? This brings us to one of the most fundamental of mathematical results, Pythagorean theorem. Although we will discuss what a theorem actually means later in section 2.5, for now imagine that you have an equation that always holds true, namely that for *any* given right-angled triangle with sides a, b and c with c being the longest side or hypotenuse, then

$$c^2 = a^2 + b^2. \tag{2.9}$$

This is shown in figure 2.2(a). The simple construction of a right-angled triangle with unit lengths

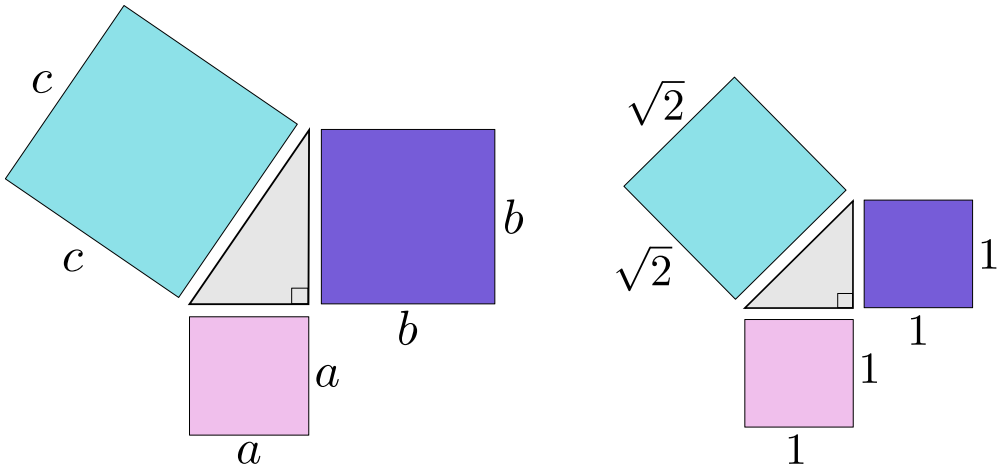


Figure 2.2: The Pythagorean theorem (a) for any right-angled triangle. The construction of a unit sided right-angled triangle (b) leading to the hypotenuse of $\sqrt{2}$.

shown in figure 2.2(b) immediately causes a problem because $\sqrt{2}$ cannot be represented as a ratio of integers. In terms of decimal representation of $\sqrt{2}$, the digits neither terminate nor have a fixed repeating sequence throughout. In fact, the sequence of digits encountered is unique to $\sqrt{2}$. The ancient Greek referred to these numbers as irrational, because it did not fit into their vision for a perfect number system.

Placing the irrational numbers $\sqrt{2}$, $\sqrt{3}$ etc. together with the rationals $\mathbb{Q}$ creates the real number system $\mathbb{R}$. We finally have a system where we can add, subtract, multiple and divide without encountering any elements that is outside the realm of $\mathbb{R}$. This type of system, one that is 'closed' under all these operations is called a number field, or simply a field. Using a field in anything we do mathematically ensures that all arithmetic computations can be done, while not creating anything unexpected.

However, many consider this to be an inelegant solution to the problem of constructing a field. In the next section, we construct a number field with only integers!

2.2.3 Congruences

Let us begin by using the number line representation of $\mathbb{N}$, and define some additional operators to help represent integer division. When an integer b divides a, denoted as $b \mid a$, then the distance from the origin a can be divided into b equal parts (see part (a) of figure 2.3). When b does not divide a,

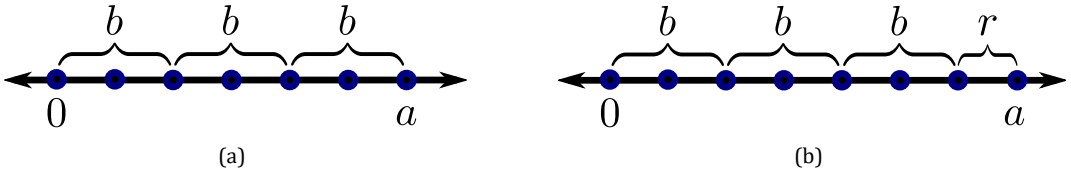


Figure 2.3: Illustration of integer division. (a) shows exact integer division. The integer a is a multiple of integer b so that $a = b \cdot c$. (b) shows integer division with a remainder r. The integer a has to be expressed as equation (2.10).

denoted as $b \nmid a$, the nearest multiple q being the quotient of b is found smaller than a so that

$$a = b \cdot q + r. \tag{2.10}$$

The remainder r is the extra distance left after division (see part (b) of figure 2.3). Note that equation (2.10) is an equation of a line whose values are restricted to the realm of integers. Such equations are called Diophantine equations, named after the work of Diophantus [100] who first studied them.

In equation (2.10), one may ask two separate, but ultimately equivalent questions. What property will the numbers a and b always share? What is the result of repeatedly applying equation (2.10) to any of the remainders? The answer to both questions is the greatest common divisor (GCD), also called the highest common factor. The GCD of two numbers a and b, which will be denoted as $\gcd(a, b)$, can be found as the following using Euclid's algorithm [Euclid, 300BCE]. Consider the rectangle $a \times b$ as shown in part (a) of figure 2.4 on the next page. Tile as many whole $b \times b$ squares into this rectangle. Next, tile the remaining $r_0 \times b$ rectangle into as many $r_0 \times r_0$ squares as possible. Repeat this using a $r_1 \times r_1$ square within the remaining $r_0 \times r_1$ square and so on. When the remaining rectangle is tiled exactly by a square, the length of this last square is the GCD of the numbers. This GCD square therefore, tiles the entire initial $a \times b$ rectangle. The tiling process is shown in part (b) of figure 2.4 on the facing page.

Gauss [1801, see Articles 1 & 2] introduced another way to view the remainder via the notion of a congruence and congruent integers. Then the integer a is congruent to integer b modulo M when $M \mid a - b$ (see figure 2.5 on the next page). Alternatively, if we view the number line as a circle with M points as shown in figure 2.1 on page 4, then the remainder is only the extra bit required to move around the circle ignoring the repeatedly cycles around the circle akin to clock arithmetic. It is analogous to 2 pm on a Thursday looking the same on a clock as 2 am or pm on a Friday.

We can then write linear congruences as

$$a \cdot m \equiv b \pmod{M}. \tag{2.11}$$

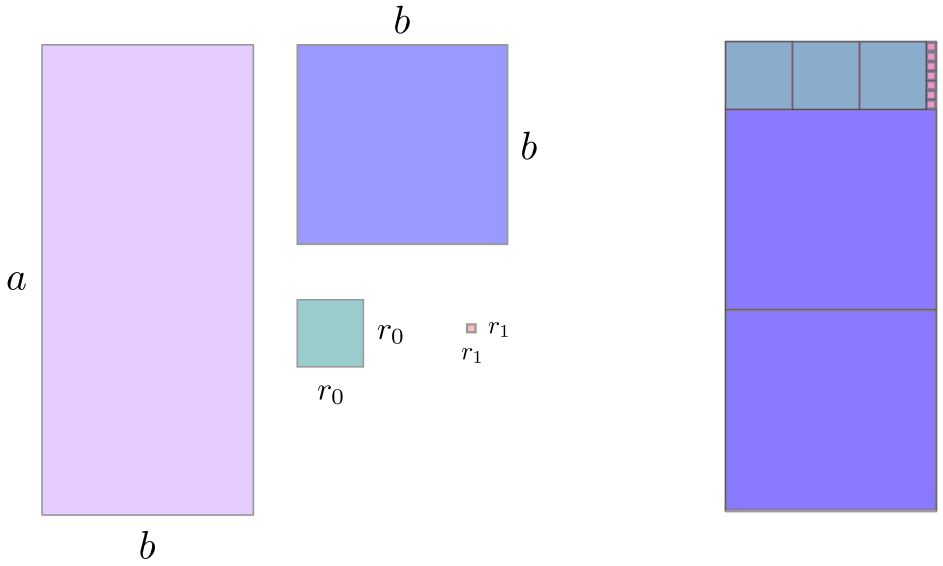


Figure 2.4: Euclid's Algorithm. (a) shows the rectangles used to tile the main rectangle $a \times b$. (b) shows the tiling of the main rectangle $a \times b$.

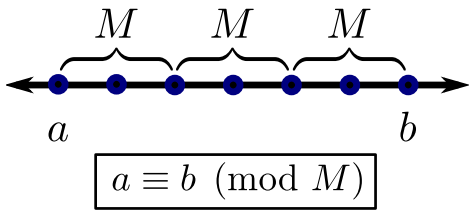


Figure 2.5: The congruence of integers. Integer a is congruent to integer b when the distance between them is a multiple of M.

If M is prime, then $a \cdot m$ will go through all possible values $\{0, \ldots, M-1\}$ uniquely in some order when a goes through $\{0, \ldots, M-1\}$ since the $\gcd(m, M) = 1$ always for a fixed $m \in \mathbb{N}$ (see figure 2.6 on the next page). When $\gcd(m, M) = 1$, division by m is possible within equation (2.11). This is done via a multiplicative inverse, so that one can determine a if only b is known, according to

$$a \equiv b \cdot m^{-1} \pmod{M}. \tag{2.12}$$

The act of division is replaced by multiplying b by the multiplicative inverse of m. We can find this inverse for any given modulus M and integer m via an extended version of Euclid's algorithm. This allows equation (2.11) to be a field, i.e. addition, subtraction, multiplication and division can be computed without leaving the set $\mathbb{N}$.

Congruences and modulo properties of composite numbers forms the basis of key-based encryption schemes such as RSA used by financial institutions around the world. The circle representation of figure 2.1 on page 4 is the how the encryption can be undone. A modulus M is found that is a product of two very large prime numbers, therefore a composite number, that is known only to you. Two powers

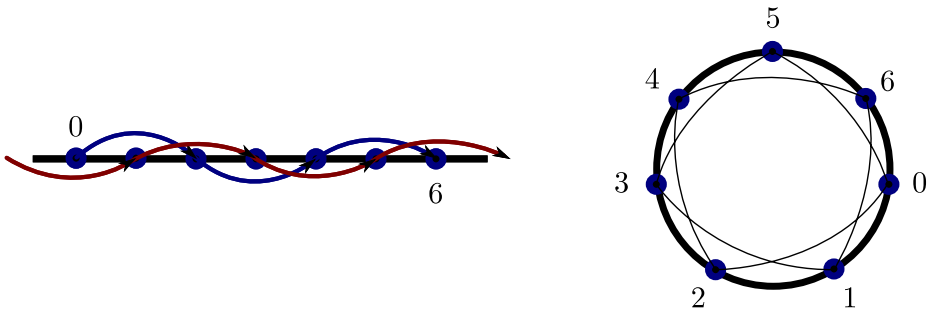


Figure 2.6: Linear congruence equation (2.11) when M is prime, shown as a number line (left) and its equivalent finite circle (right). The example shows $M = 7$ and $m = 2$, where equation (2.11) cycles through all possible values called residue classes $0, \ldots, M-1$ uniquely in some order when a goes through $0, \ldots, M-1$.

e, d modulo M are found that act as public and private keys. Exponentiation by the public key e results in an encryption of the character represented as a number. This encryption is a point on the finite circle, albeit one with very many points. The decryption is computed via the exponentiation with the private key d that effectively traverses the finite circle 'back' to the original starting point on the circle, analogous to a multiplicative inverse but for exponentiation.

As long as the modulus M is large and composite, it is very difficult to find the private key d, therefore making the encryption 'secure'. The term secure is relative here, since with a supercomputer one could compute the decryption given enough time. Fortunately (or unfortunately depending on your point of view) it would take the lifetime of the universe to crack it with the current computational power. However, a new paradigm of computation called quantum computation could change that and this will be covered in later chapters.

Now that we have defined numbers and number systems, we need to be able to compute and do things with them. This does not always involve only arithmetic but concepts of computation and structures as we shall see in the next section.

2.3 Functions

Given a number system, such as any of the ones defined previously $\mathbb{N}$, $\mathbb{Q}$ etc., we can define functions over numbers. A function is an object that produces an output based on the input provided. A function f can also be thought of a mapping from a to b when written as $f(a) = b$. The set of inputs to a function are also referred to as the domain D of the function and the set of outputs as its range R. We can express this as

$$f : D \to R. \tag{2.13}$$

We can also think of a function as a black box that gives us a transformation of the input to an output.

We have already seen examples of functions in previous sections, such as the GCD of two numbers a and b written as $\gcd(a, b)$. It returns the number that is the greatest common divisor of a and b. When the GCD of two numbers is unity ($\gcd(a, b) = 1$), the numbers are defined as being coprime to each other. Euler [1763] described the properties of a function which defines the number of coprime integers less than a number n. This function is denoted as $\phi(n)$ and is known as Euler's Totient or phi

function with $\phi(n) : \mathbb{N} \to \mathbb{N}$. For example, a prime number p is a number where $\phi(p) = p - 1$, i.e the $\gcd(j, p) = 1$ for all $0 \leqslant j < p$ and $j \in \mathbb{N}$.

Functions do not have to take exactly one or two inputs however. It is common to use the notation $\mathcal{A}_0 \times \mathcal{A}_1 \times \mathcal{A}_2 \times \ldots \times \mathcal{A}_k$ of sets to represent a tuple $(a_0, a_1, a_2, \ldots, a_k)$ of inputs or arguments to a function. An example of tuples is how we use the Cartesian coordinates in two dimensional (2D) or 3D (x, y, z) is formed from $\mathbb{R} \times \mathbb{R} \times \mathbb{R}$, which we use to plot arithmetic functions.

The concept of a function may seem simple enough and maybe even uninteresting, but in later chapters we will use it to construct machines capable of any possible computation you can devise! In the next section, we show a mathematical way to represent functions visually or more generally the structure of any arbitrary concept.

2.4 Graphs

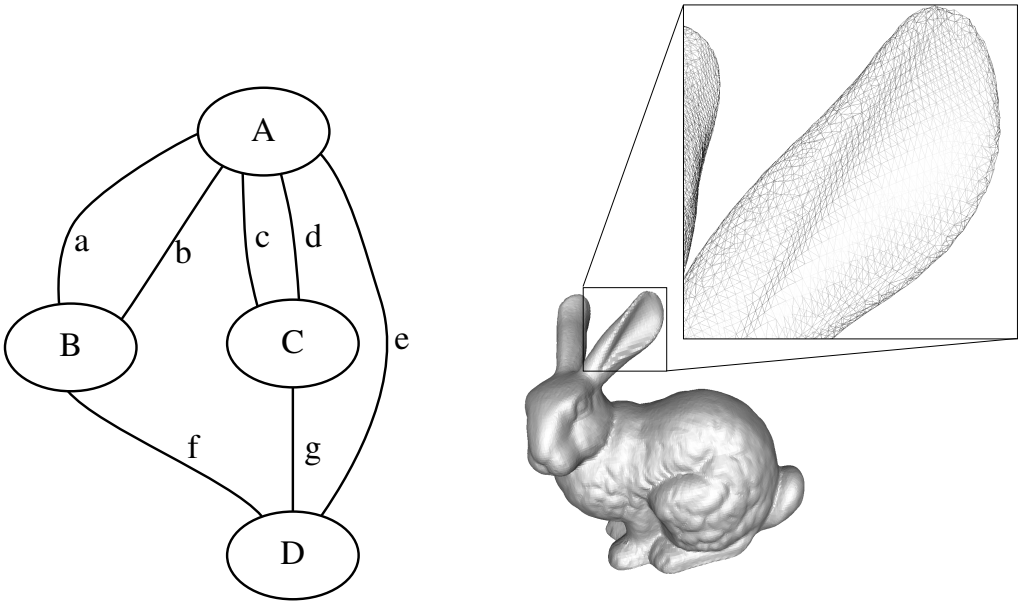
A graph is a useful mathematical construct to help visualise and analyse connections between objects or concepts and represent the inter-dependencies among them. Although they can be used to represent functions or formulas, they are powerful enough to warrant their own field in mathematics called graph theory.

A graph is a structure made out of nodes or vertices V and edges E. Much like drawing a polygon such as a triangle, the vertices usually represent the 'corners' of the graph and the edges connect these vertices together that encode the relationship between the nodes depending on what the graph represents. Indeed, this polygon analogy is used as an efficient way to represent 3D models as 2D surfaces in computer graphics. Figure 2.7 on the next page shows examples of two models representing two different structures. Models in this form are represented as vertices, edges and cells with the latter required if the model is solid and not a wireframe.

The number of vertices in a graph is called its degree and the edges can be directed or undirected. In a directed graph, all the edges have a direction or point to the next vertex to traverse, which allows for more complex relationships and constraints to be encoded into a graph.

Graphs have many uses other than arithmetic and 3D models. The graph structure is the basis of how auto-merging is done by the version control software called Git. Git tracks content not files like many more traditional version control packages. The result is that Git can automatically merge changes to source files without user intervention, a feature that is critical to the fork and distributed version control workflow of software engineering.

Sophisticated mathematical algorithms are also available on graphs. One such algorithm is known as Dijkstra's shortest path algorithm, which finds the shortest paths between vertices on graphs. For example, if the nodes represent interconnected cities and towns, Dijkstra's algorithm determines the shortest route required to navigate from one city or town to another. Another algorithm is one that finds the minimal energy among nodes to separate nodes on a graph into two parts called graph cuts. We can use this algorithm to create a separation between two structures close together. These features make graphs a useful tool in visualising and analysing structures for a number of applications. We will use graphs to represent machines and their components to analyse how they process incoming input data and what states they occupy.



(a) (b)

Figure 2.7: Examples of graphs as (a) the Dollar game and (b) as a 3D model used in computer graphics.

So far we have used a few results that seem to be very important and we have used the terms 'theorem' or algorithm and we have assumed that they always work. We have even made assumptions about numbers, but how do we know that they are always true? In the next section, we describe what a mathematical proof is and what the consequences are for science and computation.

2.5 Theorems and Proofs

We all know the definition of a fact: A concept or something that is true or proven to be true. A theorem is a mathematical fact, a result that is proven to be true using logic and mathematical results. Unlike facts in everyday life however, a theorem is as close to scientific fact or truth as possible that is beyond any doubt and *always* true. The term 'always' is important here, as the fact is true regardless of the size of numbers or limits and conditions involved. A proof is a series of infallible logical arguments that shows the theorem to be true. Any special conditions required by the fact are provided in the theorem itself and also shown in the proof. Let us examine theorems with a simple example involving even numbers, which are numbers that are multiples of two.

1 Theorem (Product of Even Numbers)
The product of any two even numbers is always an even number.

Proof: *Every even number n_i can be represented as $n_i = 2 \cdot k_i$, where $k, i \in \mathbb{N}$. The product m of two even numbers can then be written as*

$$\begin{aligned} m &= n_1 * n_2 && (2.14)\\ &= (2 \cdot k_1) \cdot (2 \cdot k_2) && (2.15)\\ &= 2 \cdot (2 \cdot k_1 \cdot k_2) && (2.16)\\ &= 2 \cdot k_3 && (2.17) \end{aligned}$$

But the term $2 \cdot k_1 \cdot k_2$ can be substituted as a natural number k_3, so that we get $m = 2 \cdot k_3$, which is also an even number. Since k_1, k_2 and k_3 can be any natural number $\mathbb{N}$, the substitution is always possible resulting in an even number for all and any k_1 and k_2. ◀

The above is an example of an algebraic proof, where if we can equate the left-hand side with the right, our proof is complete. The text following the equations are not really necessary, but for the benefit of the reader. Notice that the theorem applies to any and *all* even numbers regardless of size. This cannot be emphasized enough, there are no counter examples possible and absolute nature and finality of the theorem are indicative of all mathematical proofs. We will use the word 'proof' in certain times in the book and by this we will mean that the result is absolute and final with no doubt.

Not all proofs need be algebraic in nature however, or indeed be possible algebraically. There are many other types of proofs that are available including proof by construction and by induction. A proof by induction assumes you have a starting statement called as basis that can be extended to all possible cases or scenario by an induction. In a proof by construction, logical steps in creating mathematical object, such as a polygon or shape, make the proof self-evident from these steps. More information about these types of proofs can be found in Sipser [2013, Chapter 0]. In the subsequent sections, we will discuss some of the other major ways to prove mathematical results.

2.5.1 Visual Proofs

One of the easiest proofs to understand are those that visually represent the proof as a figure and the theorem is almost self-evident. These types of proofs are the hardest to construct though and generally not utilised very often. An example of such a proof is the proof of the Pythagorean theorem using figures of triangles as shown in figure 2.8 on the following page. A more common type of proof is via the use of creating contradictions from your initial assumption.

2.5.2 Proof by Contradiction

Often we would like a certain assumption to be true and we would like to prove it. It is usually tricky to prove that something is true because either there are too many aspects to prove correct or it does not provide us with anything glaringly obvious to show that the proof is done. What would be nice is if we set up a set of logical statements and it led to an obvious outcome that completes the proof for us.

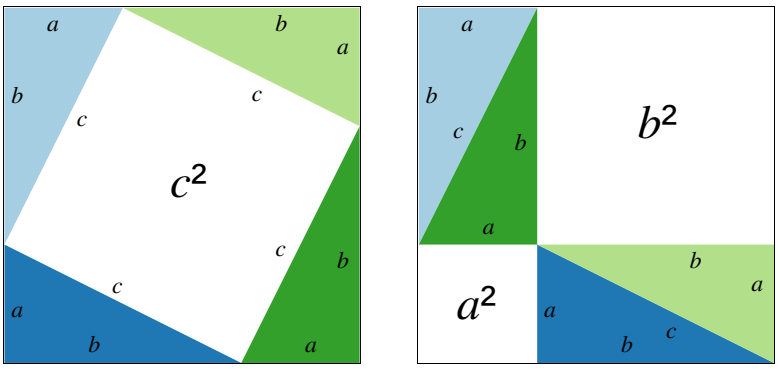


Figure 2.8: A visual proof of the Pythagorean theorem. The total square minus the area of the triangles gives c^2 on the left and a^+b^2 on the right. Original figure by by William B. Faulk.

Proof by contradiction is one of those techniques, whereby we make an assumption, set out a set of logical statements and see if it leads to a obvious erroneous outcome like a contradiction to our initial assumption. A popular way to use this approach is when we assume that the opposite of what we would like to prove is true and if if end up with a contradiction, then the opposite must be true so it proves the statement that we wanted to prove.

For example, in the case of prime numbers, we would hope that there are an infinite number of primes because there are an infinite set of numbers in $\mathbb{N}$. To prove this, we assume that there are in fact a finite number of primes and check to see if we encounter a contradiction, see theorem 2.

2 Theorem (Infinitude of Primes)
There are an infinite number of primes in $\mathbb{N}$.

Proof: *We begin by assuming the contrary, there are a finite number of primes* p_i*, where* $i < N$ *and* N *is some large number, i.e. our set* $\mathcal{P}$ *is finite. Then let us multiply all the known finite primes together to create the product*

$$m = p_0 \times p_1 \times p_2 \times \ldots \times p_N \tag{2.18}$$

This is the largest number we can represent using our known set of primes $\mathcal{P}$*. But then we can always add unity, so that* $m' = m + 1$*. Clearly* m' *is not a product of the known primes, since it is bigger than* m *and* $p_i \nmid m'$ *for* $i < N$*. But* m' *is on the number line like any other and therefore must be a product of primes. Thus, there must be another prime number that is not in our set* $\mathcal{P}$*. We can repeat this process ad infinitum and we arrive at a contradiction. Therefore there must be an infinite number of primes.*◀

This is a well known proof, since a similar proof was known to Euclid [300BCE], but it also relates to an important concept of completeness, which we will discuss in the next section.

2.6 Incompleteness

In previous sections, we have constructed mathematical systems, such as the natural numbers $\mathbb{N}$, where we defined its set of elements and its operations. There were a few rules in this system called axioms that we took for granted about the operations in $\mathbb{N}$, a few of which include axioms such as

1. multiplication and addition are commutative, i.e. $ab = ba$ or $a + b = b + a$,
2. associative law applies, i.e. $c(a + b) = ca + cb$.

These axioms, though seemingly obvious, are important for the system to function properly. In general, any formal mathematical system consists of a set of axioms that define the rules for the system and allow for the creation of new theorems. These axioms govern the way the system behaves and define everything else within it, including the theorems that are possible.

For example, consider plane geometry that we use whenever we construct geometric figures such as circles and triangles on the page. This geometry is actually governed by a set of axioms initially defined by Euclid [300BCE], which is why it is usually referred to as Euclidean geometry. These axioms include how circles, lines and points are defined, and how lines interact, but the exact details are not important here. The fact that a fixed set of axioms is required for this system to function is important for the idea of completeness. Euclid's axioms are sufficient to perform any operation within the geometry, so Euclidean geometry is considered to be complete. Note that defining a system is not always unique however as Klein [1893] found a more general set of axioms that unified both Euclidean and non-Euclidean (i.e. curved) geometries with a single set of axioms.

In an important result that turned mathematics upside down, Gödel [1931] proved as a series of theorems that a system that is complete cannot be consistent and vice versa. We mentioned that a system is complete if the axioms govern all of the possible theorems. A formal system is consistent if all the theorems can be either proved or disproved, i.e. there are no mathematical inconsistent results possible. In other words, the incompleteness theorems of Gödel [1931] show that a consistent formal system has to be incomplete.

To understand why this is the case, let us examine our Theorem 2 on the facing page on the infinitude of the primes. When the set of primes $\mathcal{P}$ is finite, we can only create a finite set of numbers up to the product of all the primes in our set. Likewise, by having a finite set of axioms in our formal system, we can only create a finite set of theorems and therefore there will always be theorems that cannot be proved using these axioms, just like there will be numbers which we cannot create as a product of the primes in our limited set $\mathcal{P}$.

In the following chapters, we will discuss how representing mathematics and formal systems as a 'code' of symbols enacted on by 'machines', can prove the same result and much more leading to the creating of computers in a formal sense. This result by Turing [1937] is the basis of state-based computation and forms the theoretical foundation of most digital computers we use today.

Abbreviations

2D two dimensional 11
3D three dimensional iv
GCD greatest common divisor 8

Bibliography

Berlekamp, E., J.Conway, R.Guy, 1982. Winning Ways for your Mathematical Plays. Vol. 2.

Church, A., 1932. A set of postulates for the foundation of logic. Annals of Mathematics 33 (2), 346–366.
URL https://doi.org/10.2307/1968337

Diophantus, 100. Arithmetica. Springer-Verlag Berlin and Heidelberg GmbH & Co. KG (December 31, 1982).

Euclid, 300BCE. The Elements.

Euler, L., 1763. Theoremata Arithmetica Nova Methodo Demonstrata. Novi Commentarii Academiae Scientiarum Petropolitanae 8, 74–104.

Gardner, M., 1970. Mathematical games: The fantastic combinations of john conway□s new solitaire game □life□. Scientific American (223), 120–123.

Gauss, C. F., 1801. Disquisitiones Arithmeticae. Yale Univeristy Press.

Gödel, K., Dec 1931. Über formal unentscheidbare sätze der principia mathematica und verwandter systeme i. Monatshefte für Mathematik und Physik 38 (1), 173–198.
URL https://doi.org/10.1007/BF01700692

Klein, F., Mar 1893. Vergleichende betrachtungen über neuere geometrische forschungen. Mathematische Annalen 43 (1), 63–100.
URL https://doi.org/10.1007/BF01446615

Langton, C. G., 1986. Studying artificial life with cellular automata. Physica D: Nonlinear Phenomena 22 (1), 120 – 149, proceedings of the Fifth Annual International Conference.
URL https://doi.org/10.1016/0167-2789(86)90237-X

Sipser, M., 2013. Introduction to the theory of computation / Michael Sipser., 3rd Edition. Cengage Learning, Andover.

Turing, A. M., 1937. On computable numbers, with an application to the entscheidungsproblem. Proceedings of the London Mathematical Society s2-42 (1), 230–265.
URL https://doi.org/10.1112/plms/s2-42.1.230