General introduction to mathematics: set theory, boolean logic, algebraic structures, and category theory

Tristan Duquesne

July 3, 2021

Prerequisites: None; but you can (or should) use the summary data sheets provided. They explain different mathematical spaces, and thus present concrete cases for the abstract concepts introduced here. Some things you'll have seen in school already, and they'll be recontextualized in a unifying framework here.

Objective: The goal of this document is to introduce you to a rather general vision of the fundamental constructs of mathematics; without going too far in erudition and theorem-proving. Think of it as a first step to understanding the rich and interrelated vocabulary of mathematics, and the positioning, roles and usefulness of the mathematical ideas that this language describes. The goal is not to understand everything presented here on your first read-through (though congrats if you do manage to), but reading this document will greatly help you prepare how to structure, work and integrate all the rest of mathematics within a common unifying framework. Generally, mathematical ideas, as they are taught in schools, are unrelated and thus useless; like an archipelago without a boat. We take the complete opposite point of view: the goal is to describe to you the map of a continent. I greatly encourage to return to this document often: it'll be very useful in helping you draw out the necessary links between concepts.

Method: This text ignores many of the conventions of a "classical" introduction to mathematics: *first-and-foremost*, just about every proof is omitted. Note that an educated mathematician will surely recognize insights from standard proofs in the way we approach some of the problems, even when it is just a matter of describing a problem and the elements that make it up.

However, the main objective of this text is to apprehend the vocabulary of modern mathematics in an ecosystemic way (for it to be both interrelated and well-founded), and to do so quickly, for a public that might be moderately-to-not adept at mathematics, but that still has to deal with them on a daily basis. We operate by pushing geometrical intuition above all, and by expliciting the links of this geometrical intuition with its expression in both formal language, and in vernacular language. We have chosen this directing "pedagogical thesis" so that the reader can quickly grasp many reflexes of a mathematician's thinking in practice (swimming in analogies and definitions) and can pursue their further research autonomously.

To accomplish this, *the reader should not be afraid to read with a paper on hand, to write down every new concept, to make cheatsheets/flashcards, and to reread often.*

<u>ALWAYS RETURN TO THE DEFINITION</u>: I cannot stress this enough. Why ? Well, what is a definition, in mathematics? On the one hand, the abstract and formal description; on the other hand, concrete examples and counter-examples. That's what defining is all about. If you don't do this work, you will start daydreaming while you are reading because the words won't have something concrete to catch on to, as sentences become more technical! You know, when you try to read a passage, start daydreaming, realize it, go back to the last moment you remember, reread, restart to daydream... and go nowhere ? I call this the "reading illusion" (in the sense of an optical illusion). Do this work of going back to the definition, and you will see that this "reading illusion" will no longer occur. Better yet, if you realize the reading illusion is happening, know that it indicates that you have read a word you don't really understand! This mental glitch can be harnessed to become one of your strongest learning tools.

This short format seemed to be more efficient, without sacrificing the real keys to understanding, which are more the result of getting acquainted with a breadth of knowledge while always linking intuition to formalism, than either formalism or intuition alone, or even the technical details of a mathematician's work: the art of proof which is learned by problem-solving and practice.

I therefore apologize to any professional mathematician for my well-meaning heresy!

1 Introduction, or why we should start with ”abstract” subjects

At the basis of everything, you have algebra. Algebra, in its primary sense (the most used today at least), is the field of mathematics which is interested in the representation and study of mathematical spaces from the point of view of symbols and formal processes. It’s symbolic mathematics and its ruleset. ”Al-jabr”, in its original sense, is the ”method of reunion/equilibrium/equivalence of quantities”, today it is more clearly ”what one is allowed to do with mathematics without risking being wrong”. Algebra is the branch of mathematics that tries to establish ”what are the properties of such and such a set, and what does it imply” or more stupidly ”what do I have the right to do with that mathematical object/structure, if I start from this?

For example on ”the space of integers with addition, subtraction and multiplication” (one of these aforementioned ”mathematical structures”), one can define the ”Euclidean division”, which you already know since grade school (it’s the ”division with remainder”, sometimes called ”long division”). But did you know that on polynomials (contained in another structure with its own, very similar rules), one can also construct a form of Euclidean division? For example, $4x^4 + 7x^3 - 4x^2 + 3x + 12$ divided by the polynomial $x + 2$ will give you :

```
  4x^4 + 7x^3 - 4x^2 + 3x + 12 = A(x) | x + 2 = B(x)
- 4x^4 - 8x^3                         +-------
-------------                         | 4x^3 - x^2 - 2x + 7 = Q(x)
    0  -  x^3 - 4x^2 + 3x + 12        |
       +  x^3 + 2x^2                  |
         -----------                  |
            0 - 2x^2 + 3x + 12        |
              + 2x^2 + 4x             |
               ----------             |
                   0 + 7x + 12        |
                     - 7x - 14        |
                      --------        |
                         0 - 2        |
```

There is no way to make x+2 fit in -2: that means $R(x) = -2$.

Here, the remainder of the Euclidean division is $R(x) = -2$ (a constant considered as a ”constant polynomial”), and the quotient is $Q(x) = 4x^3 - x^2 - 2x + 7$, and we find $A(x) = B(x) \times Q(x) + R(x)$ like in regular Euclidean division (ex: 51 divided by 10 is equal to 5 and the remaining 1 corresponds to $a = b \times q + r \Rightarrow 51 = 10 \times 5 + 1$).

Notice in particular that each new monomial added to $Q(x)$ at each step is multiplied to $B(x)$ then subtracted from $A(x)$, like the method you know for integers, except that here the monomials are usually digits added to a single q number that is built up gradually. Note also that if $R(x) = 0$, the remainder polynomial is null, then we say that the polynomial $A(x)$ is divisible by the polynomial $B(x)$, just like we would for the integers.

Here is a small summary table :

$A(x)$	$4x^4 + 7x^3 - 4x^2 + 3x + 12$	a	51
$B(x)$	$x + 2$	b	10
$Q(x)$	$4x^3 - x^2 - 2x + 7$	q	5
$R(x)$	-2	r	1
$A(x) = B(x) \times Q(x) + R(x)$	Verify by yourself.	$a = b \times q + r$	$51 = 10 \times 5 + 1$

No need to flawlessly understand this example for the time being, the rest of the text will allow you to link everything here with what you already understand: just understand that complex mathematical structures will often ”behave” like simpler structures. This fundamental idea is the subject of the first part of this text. We will use it to explore polynomials in depth (well, relatively, for a brief introduction in only a few dozen pages).

Add to this the fact that **for any algebraic space, one can construct an underlying geometry that works like the calculations (algebraic symbol manipulations) in that space**, and you have the spearhead of modern mathematical research: **improving our understanding of complicated symbols that reveal the universe to us (but which our brains have trouble handling without the right reflexes), by returning to simpler geometric concepts**.

Such fundamental commonalities between structures that initially seem so different put the mathematicians of the 19th and 20th centuries on the trail of something very profound, a more abstract way of understanding mathematics, by analyzing structures (”`algebraic structures`”) and their properties, their behavior, rather than dwelling on their elements. With such a tool, a better understanding of the integers allows a better understanding of polynomials, ie, one starting from something simpler, but similar. This allows one to translate the tools between different branches of mathematics. It is an exceedingly powerful idea, for all science and especially for computer science: this use of equivalences between structures allows us to find the most interesting branch of mathematics to efficiently solve a given computation/code/modeling problem, and find the most appropriate simple branch to help us ease our mental manipulation, without loss of generality.

2 Some key ideas on sets and formal language for beginners

2.1 A few words to understand the concept of a mathematical ”set”

Sorry to do you dirty from the get-go, but unfortunately, a set is a kind of undefinable in a formal way (at least without reference to the concept of set itself; or something analogous to it, itself poorly defined). Which poses a few problems in terms of our mathematical foundations. For example, they say in textbooks ”the fundamental objects are the sets” but that doesn't really explain anything. Since the question of the nature/definition of sets has been raised, a lot of progress has been made - but that's another subject, which we'll only talk about briefly in this course.

Consequently, since a ”set” is a somewhat ambiguous concept for certain contentious cases, we will talk about some basic things to keep in mind when thinking about sets, which should help you come to grips with them:

– a set is basically **a *collection of elements* which are all different**. This collection is expressed as an ”idea expressed between curly braces” or as a ”list of elements between curly braces”, or as a ”drawing of a potato with stuff in it”: an abstract ”bag” with a name. These elements can be integers, vectors, giraffes, whatever.

– a subcollection of elements of a given set is called a subset, and is also another set in-and-of-itself.

— This collection of elements **can be empty, finite or infinite**, and there are different types and **different sizes of infinities** (*discrete infinity vs. continuous infinity* for the most important distinction in practice). The empty set is denoted $\emptyset$.

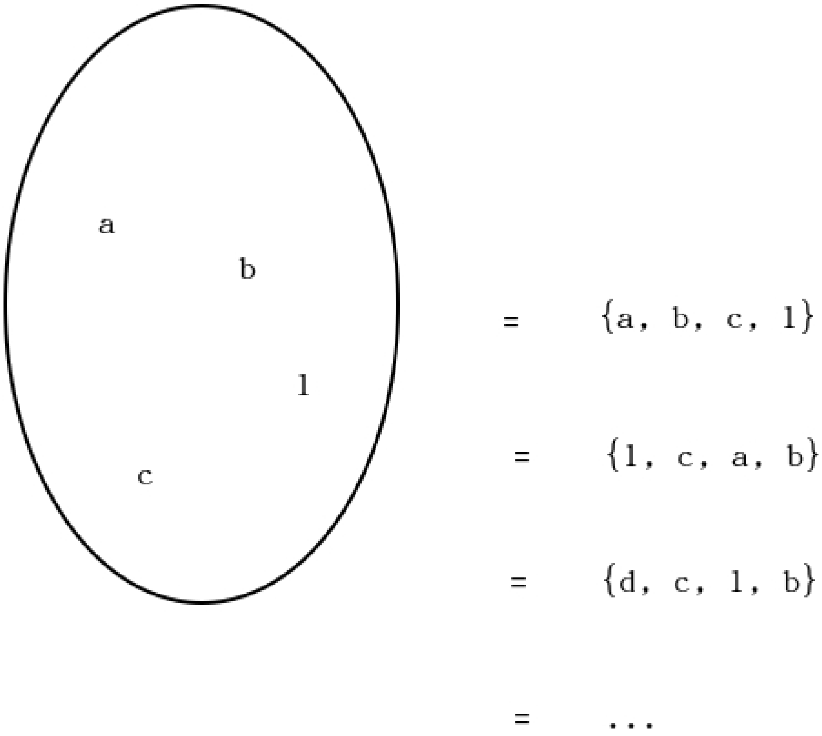


Figure 1: A set as a potato/bag, and as a collection of objects between curly braces. NB: the order of the elements does not matter.

— **there are sets which may contain other sets** (for example, the ”powerset” (which contains all possible combinations of elements of a given set as its elements, ie, all possible subsets of a set); ”sigma-algebras” (also called ”set of measurable sets”), or ”topologies” (also called ”set of open sets”)); **but you are not allowed to make a set which contains itself** (otherwise, you are really playing with fire), nor a ”set of all sets”, often not even ”a set of all sets of a certain type”. [Dealing with this problem at the level of the foundations of mathematics is one of the many reasons for the current interest in category theory, which is the most popular modern view of an abstract framework for the totality of mathematics, as was set theory for the XXth century].

— when doing set theory (and most mathematics), one generally places oneself in the context of a global "container" set: a universe within which all operations take place. Note that even if two sets have similar or even "identical" operations, one should not mix the elements of two different sets at the level of the operations internal to the set. If one wants to make sets "communicate" with each other, there are a multitude of different constructions "between two sets" to achieve the desired result. Since "knowing where is what" is a *fundamental* subject and one that is often ignored in introductory texts to mathematics, we emphasize it throughout this one.

NB: Except for a few mathematicians who are really interested in the subject, the absence of an absolutely rigorous definition for sets does not really pose a problem in practice, since one generally manipulates finite sets or sets of a type of infinity that we know well (discrete and continuous). [You will surely hear about the "axiom of choice" (or one of its equivalent versions such as "Zorn's lemma") through practice of mathematics, but this is probably the only "foundational" problem with which you may one day be confronted.] We can therefore simplify how we'll learn about sets by keeping our definition of a set as a "potato of elements", or "list of elements", or a "bag of different elements"! It's a very apt and convenient image to think about sets, and sufficient for the vast majority of our needs.

Do note that it is a common practice to describe sets "conceptually", meaning as an "idea between curly braces". Generally this is written as the pattern: "objects of such and such form — that verifiy such and such constraint". For example:

$$B = \{2^i \mid i \in [[0, 10]]\}$$

means "B is the set of numbers of the form '2 to the power i' such that i is a number between 0 and 10, inclusive", or in simpler, but longer, form:

$$B = \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024\}$$

The part to the left of the "|" bar describes the *form* of the elements of the set ("form", as in "formula" or "formally", i.e. the "way of writing with symbols neatly and rigorously"). The part to the right expresses the conditions that these elements must verify. And the vertical bar itself (sometimes it is a comma) between the two parts is read as "such as". The advantages of this way of writing sets as "an idea between braces" are multiple: it can represent an infinite number of elements respecting a certain pattern, it can make explicit the shared pattern of various elements, it is often shorter to write down, and often it is the only way to express things both clearly and rigorously.

2.2 Some important sets

Here is a (non-exhaustive) list of important classical sets, and/or sets to which you will see in these documents.

- $\mathbb{N}$, the set of natural numbers, $\mathbb{N} = \{0, 1, 2, 3, 4, \dots\}$.

- $\mathbb{Z}$, the set of integers, $\mathbb{Z} = \{\dots, -4, -3, -2, -1, 0, 1, 2, 3, 4, \dots\}$

- $\mathbb{Q}$, the set of fractions, also called rational numbers (as in "ratio"), $\mathbb{Q} = \{\frac{a}{b} \mid a \in \mathbb{Z} \text{ and } b \in \mathbb{N}^*\}$

- $\mathbb{R}$, the set of real numbers, defined as the set of limits of numerical sequences of rational numbers; often represented as the "real line". Put bluntly, $\mathbb{R}$ "plugs in the holes" left by the numbers that can't adequately be written as a fraction (such as $\sqrt{2} = 1.4142\dots$ or $\pi = 3.1415\dots$).

- $\mathbb{C}$, set of complex numbers, a 2D version of $\mathbb{R}$, given a form of multiplication which is a generalization of that which you find in $\mathbb{R}$ to a 2 dimensional plane.

- $\mathbb{R}^n$, the standard n-dimensional vector space over $\mathbb{R}$.

- $\mathbb{C}^n$, the standard n-dimensional vector space over $\mathbb{C}$

- $\mathbb{R}[X]$, space of polynomials of a single variable over $\mathbb{R}$.

- $\mathbb{C}[X]$, space of polynomials of a single variable over $\mathbb{C}$

- $\mathbb{R}(X)$, space of (polynomial) rational functions (sometimes called rational fractions, which can also refer to a closely related concept) of a single variable on $\mathbb{R}$.

- $\mathbb{C}(X)$, space of (polynomial) rational functions (sometimes called rational fractions, which can also refer to a closely related concept) of a single variable on $\mathbb{C}$.

- $\mathbb{R}^{\mathbb{N}}$, space of the real-valued numerical sequences, i.e. the space of the functions from $\mathbb{N}$ to $\mathbb{R}$.

- $\mathbb{R}^{\infty}$, space of real-valued numerical sequences, with null values starting from a certain rank, a subset of the previous set.

- $\mathbb{C}^{\mathbb{N}}$, space of numerical sequences with complex values, i.e. the space of the functions from $\mathbb{N}$ to $\mathbb{C}$.

- $\mathbb{C}^\infty$, space of numerical sequences with complex values, null from a certain rank, subset of the previous one.

- $\mathbb{R}^\mathbb{R}$, space of functions from $\mathbb{R}$ to $\mathbb{R}$.

- $\mathcal{L}(\mathbb{R}, \mathbb{R})$, space of linear functions from $\mathbb{R}$ to $\mathbb{R}$, subset of the previous set.

- $\mathbb{C}^\mathbb{C}$, space of functions from $\mathbb{C}$ to $\mathbb{C}$.

- $\mathcal{L}(\mathbb{C}, \mathbb{C})$, space of linear functions from $\mathbb{C}$ to $\mathbb{C}$, subset of the previous set.

NB: the reason behind the term "space" rather than "set" as used above is linked to that of "algebraic structure" that we see below. Technically, it is only a question of point of view: a "space" (or similarly an "algebraic structure") generally refers to a set to which we have given additional structure, typically a choice of properties that define the algebraic behavior of this set. By abuse of language, more complex structures tend to be called "space" more frequently, as they are more often found in contexts where advanced algebraic language is required (and also, physics and mathematics have evolved symbiotically throughout human history).

NB: *Discrete refers to what you count, continuous refers what you measure.* Discrete infinity corresponds to a "cardinal" (number of elements in a set) denoted $\aleph_0$ (aleph zero. This is the infinite order of magnitude of $\mathbb{N}$, $\mathbb{Z}$ and $\mathbb{Q}$. With $\aleph_0$, we can make an "infinite list" (the example of Hilbert's hotel, which I suggest you go research a bit). This infinity is strictly smaller than $\mathfrak{c}$, the cardinal of $\mathbb{R}$ and $\mathbb{C}$, which corresponds to continuous infinity. This continuous infinity is too big and too dense to be listed. A good illustration: there is more infinity in numbers between 0 and 0.001 in the real numbers than in the whole rational numbers from $-\infty$ to $+\infty$. The proof of this quirk is called "Cantor's diagonal argument". The fact that infinities have divers natures and sizes has very interesting ramifications on many important distinctions between the discrete and the continuous.

2.3 Some pointers concerning how to read symbolic mathematical language

Here are the symbols and fundamental concepts of the language of set theory (in truth, of mathematical language in general):

– a truth statement, ie, a sentence or mathematical formula that can be either true or false, is called a **proposition**. For example, "it's snowing" and "2 is an odd number" are both propositions; but "why am I hungry ?" is not. The previous sentence is also a proposition. Most mathematical ideas (most notably theorems) are expressed as propositions.

– "$\in$" is read as "belongs to/belonging to" or "in" and means that the element represented by the letter/value on the left is in the set represented by the letter on the right (ex: $a \in A$). It is a relation of an element to a set. This also applies to a set belonging to its powerset. Do notice that $a \in A$ is a proposition.

– "$\subset$" is read as "(is) included in" and means that all the elements of the set on the left can also be found in the set on the right (ex. $A \subset B$). It is a relationship between sets. It's an "order-like relationship" (it works like $\leq$; generally called a "partial order"). Equivalently, we say that A is included in B "if and only if" (iff) $\forall x \in A, x \in B$ (all elements of A are also in B). Graphically, A is a potato that is enclosed by the potato of B. Two sets of A and B are equal iff ("A is included in B" AND "B is included in A"). It is geometrically clear that if two potatoes contain each other, they are the same potato. Formally, we can write this idea as $A = B \Leftrightarrow (A \subset B) \cap (B \subset A)$ Ex : $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$

– "$\forall$" which is read as "for all", or "for any". It is called the universal quantifier. "$\forall x \in E$" reads "for all x belonging to E" and means "we can take any element x (all elements must be valid!) from our set E, and the rest of the formula will work for x". Ex: If $\mathbb{N}$ is the set of natural integers $\{0, 1, 2, 3, \dots\}$ then $\forall x, y \in \mathbb{N}, x + y = y + x$ is a property of $\mathbb{N}$ (this formula is true for any pair of elements chosen at random or arbitrarily in $\mathbb{N}$). [Side note: this property is called the "commutativity of addition in $\mathbb{N}$".]

– "$\exists$" which is read as "there exists". It is called the existential quantifier. $\exists x \in E$ means "we can find at least one element x of our set E that is suitable for the rest of the formula to work". Ex: If n is not prime, then there exists b, an element of $\mathbb{N}$ different from 0 and 1, such that b is a divisor of n. This would be written:

$$n \in \mathbb{N}, n \text{ not prime } \Rightarrow \exists b \in \mathbb{N}, b \geq 2, b \neq n, b|n$$

– "$\exists!$" is read as "there is a single element"/"there is one and only one". Uniqueness is an important subject in mathematics. $\exists! x \in E$ means "we can find exactly one x element of our set E such that the rest of the formula works for x".

– if a variable is declared in a formula but has neither $\forall$ nor $\exists$ preceding it, we must understand that it is a fixed constant (which is actually just a hidden $\forall$, but which is more intuitively understood if we understand it as a fixed case; it is a choice of style).

– "iff" reads "if and only if" and corresponds to the $\Leftrightarrow$ symbol, i.e. the concept of logical **equivalence**. This means a synonymy at the level of ideas; that one idea cannot exist without the other. The idea of "square" is equivalent to the idea of "four vertices with sides of equal length with at least one right angle" or to the idea of "four vertices each at a 90 degree angle, with at least two consecutive sides of equal length". This is to be contrasted with the **implication** $\Rightarrow$, which transcribes the idea that "all thumbs are fingers, but not all fingers are necessarily thumbs". In other words, "square $\Rightarrow$ quadrilateral" $\Leftrightarrow$ "all squares are quadrilaterals, but not all quadrilaterals are necessarily squares" $\Leftrightarrow$ "if it's not even a quadrilateral, it's impossible for it to be a square".

NB: do not confuse these symbols and notations! When you work on sets of sets (multiset, hypergraph, powerset, sigma-algebra, topology) you might get it wrong otherwise. You can write either "$a \in A$", or (equivalently) "$\{a\} \subset A$", where $\{a\}$ is a set called **singleton** containing a single element: a. But we don't write "$a \subset A$" because it is changing the level of "content" inappropriately. It is a bit like forgetting to dereference a C pointer in a function call. Either the compiler grunts or your program segfaults.

NB: the order of the terms is important in a formula! For example :

$$\forall x \in A, \exists y \in B, y = f(x)$$

$$\exists y \in B, \forall x \in A, y = f(x)$$

In the first case, we are saying that A is the appropriate well-defined input domain for the f function because "for all x at the start, we can find a y image at the end of the function" (ie: there will be no case of an input for which your function has an undefined output). In the second case, we are saying that f is a constant function that at any x associates the value y, because "there is a y of the arrival set B (codomain) such that every x of A gives this same y that we can choose precisely". Try to understand this example well, it is one of the most useful examples to get an idea of how to read mathematical language in the right order.

NB: The rule for maintaining the meaning of a mathematical formula is that quantifiers are intervertible with quantifiers of the same type only (i.e. universal with universal, existential with existential, but never interchangeable). The justification for this principle can be found in category theory, in formal logic, and in type theory. So it is true that $\forall x \forall y \exists z = \forall y \forall x \exists z$, but $\forall x \forall y \exists z$, $\forall x \exists z \forall y$ and $\exists z \forall x \forall y$ all have a different meaning. I'll try to add the expression "such as" in the right places in the formulas in what follows, but it is ABSOLUTELY necessary to be able to do this "reading" of the mathematical formulas by yourself, it requires practice and taking the time to decipher early on! Really, this is very important in order to be able to understand new abstract ideas quickly. It is as important as geometric intuition, I think, which is not a small statement. Indeed, if "a good drawing is worth a thousand words", then "a well-understood formula is worth an *infinite* number of drawings". That doesn't mean you can just forsake geometry; it is essential when learning a new concept if your goal is to better understand the formulas! Everywhere, intuition and formalism complete and enrich each other.

3 Operations over sets and boolean logic

There exist operations over sets, that give new sets, or express some properties of sets. Additionally, these operations have an equivalent in the domain of logic. Here are the most important. (Note that in what follows, we place ourselves in the context of an all-emcompassing container set E.)

3.1 General list of operators over sets

— $A \setminus B$, or $A - B$, called **subtraction** of A by B, and read "A deprived of B", is the set of elements belonging to A and not to B. Formally, $A \setminus B = \{x \in A \mid x \notin B\}$. This can also be defined with the operators below as $A \setminus B = A \cap \overline{B}$

— $\neg A$, or $\overline{A}$, or A^c called the **negation**, or the **complementary**, of A, and read "not-A" or "neg-A" or "non-A", is the set of elements of E, the encompassing universe, not belonging to A. Formally, $\overline{A} = \{x \in E \mid x \notin A\} = E \setminus A$.

— $A \cap B$, called **intersection** (or, sometimes, "meet") of A and B, and read "A inter B" or "A and B" or "A cap B", is the set of elements common to A and B (the '&&' operator in C programming language for its logical version).

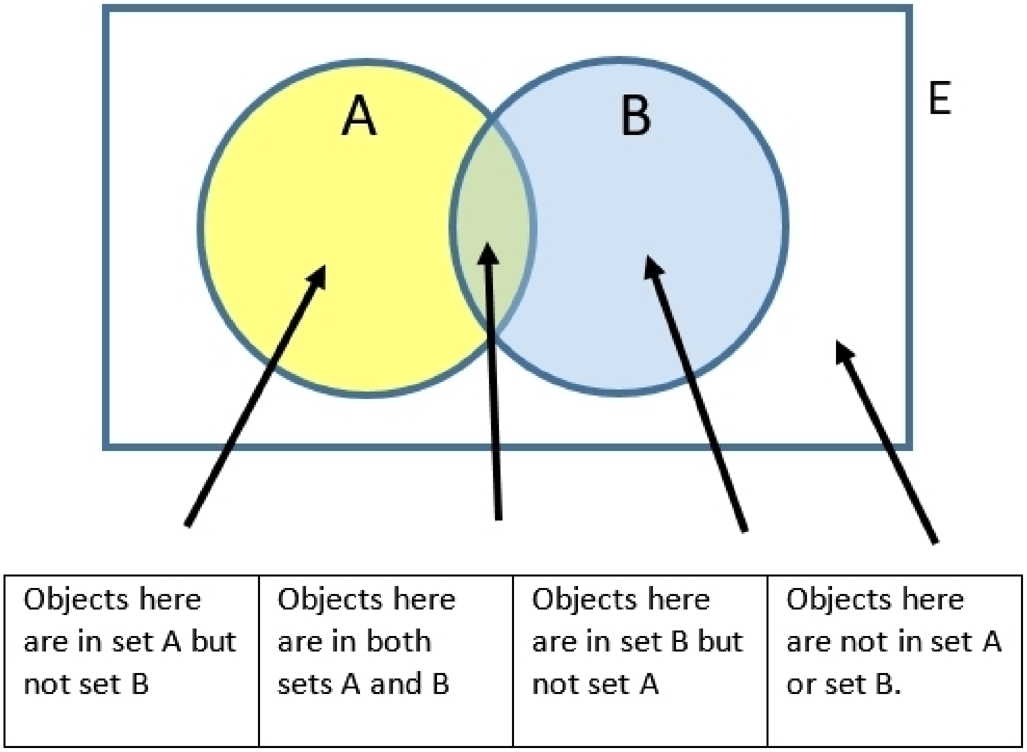
— $A \cup B$, called **union** (or, sometimes, "join") of A and B, and read "A union B", or "A (inclusive) or B" or "A cup B", is the set of elements either in A, or in B, or in both, keeping only one copy of any eventual duplicate (the "||" in programming language C, for its logical version).

— $A \Delta B$ (or more rarely $A \oplus B$ or $A \otimes B$), called the **symmetric difference** (sometimes "disjunctive union") of A and B, and read "A xor B" or "A delta B" or "A (exclusive) or B", corresponds to elements that belong to either A or B, but not to both. Formally, $A \Delta B = (A \cup B) \setminus (A \cap B) = (A \setminus B) \cup (B \setminus A) = (A \cap \overline{B}) \cup (B \cap \overline{A})$.

3.2 Venn diagrams

All this is pretty unintuitive without more concrete examples and visualizations. That's why we'll now link all these concepts to logic with an extremely useful tool to visualize sets and their operations. It's the "clean" version of our "sets-as-potatoes" intuition, and it's one of the dumbest, yet most prolific, tools of mathematics: Venn diagrams. You'll encounter basic

Venn diagrams most frequently in set theory, logic, and probability theory (measure theory); but you could argue that fields like algebraic geometry are fundamentally just complexified Venn diagrams. Here is a telling example of a Venn diagram.



Objects here are in set A but not set B	Objects here are in both sets A and B	Objects here are in set B but not set A	Objects here are not in set A or set B.

Figure 2: Two sets A and B, within an encompassing set E — as a Venn diagramm.

The principle is simple, isn't it? Each potato/circle is a set, and the different regions that are cut out will allow us to analyze the behavior of the operators.

Now, what is interesting is to ask the question of what these sets can represent, concretely. Especially in logic. So let's take concrete examples.

$$A = \{\text{Set of computers that run Linux}\}$$

$$B = \{\text{Set of computers that run Windows}\}$$

$p = \{$"It is currently raining", set of 'universes' in which this idea is true for example$\}$

$q = \{$"I am currently wearing a coat", ditto$\}$

You can think of p as A and q as B, or vice versa. I'm just giving them different names to be clearer, and to stay within the respective writing conventions for each domain (set theory *vs* boolean logic). Most of the operations are commutative (symmetrical, like $5+3$ which equals $3+5$) anyways, so how you put your letters on the diagram doesn't matter as long as you have the right geometric intuition.

To expand on what follows, these Wikipedia pages are fine and so are the detailed articles to which they refer:

https://en.wikipedia.org/wiki/Truth_table

https://en.wikipedia.org/wiki/Algebra_of_sets

I'm stealing their diagrams, by the way.

In what follows, A is the disk on the left, B is the disk on the right. The red coloring means "true", "T", or "1", the white means "false", "F" or "0". The red coloring also means "result of the set operation just performed".

And finally, before we move on to the details, here are two truth tables that synthesize all the set operators as seen in their "logical" form, for boolean calculus. The first one shows unary operators, the second one shows binary operator. You might want to keep coming back to these tables as you read what follows.

p	$id(p)$	$\neg p$	$\top$	$\bot$
F	F	T	T	F
T	T	F	T	F

p	q	$p \cap q$	$p \cup q$	$p \Delta q$	$p \cap \neg q$	$p \Rightarrow q$	$p \Leftrightarrow q$
F	F	F	F	F	F	T	T
F	T	F	T	T	F	T	F
T	F	F	T	T	T	F	F
T	T	T	T	F	F	T	T

3.2.1 Empty set ($\emptyset$); "FALSE" constant (F, 0, $\bot$)

The empty set $\emptyset$ is the set containing no element. It is also the representation of a proposition p that is always false, generally written F, or 0, sometimes $\bot$. Such a proposition is called a "negalogy" or a "contradiction".

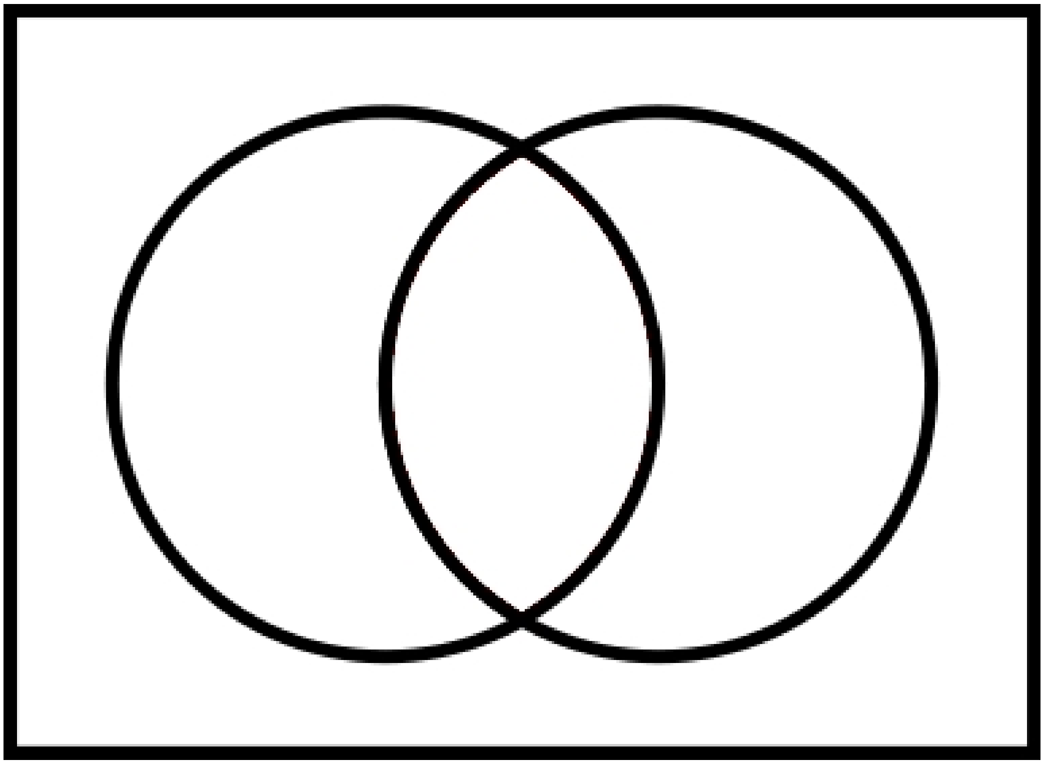


Figure 3: Empty set, containing no element.

3.2.2 Total set (here called E); "TRUE" constant (T, 1, $\top$)

The total set is the set containing all the elements (within the given context). It is also the representation of a proposition p which is always true; generally written T, 1, and sometimes $\top$. Such a proposition is called a "tautology" or a "(logical) theorem".

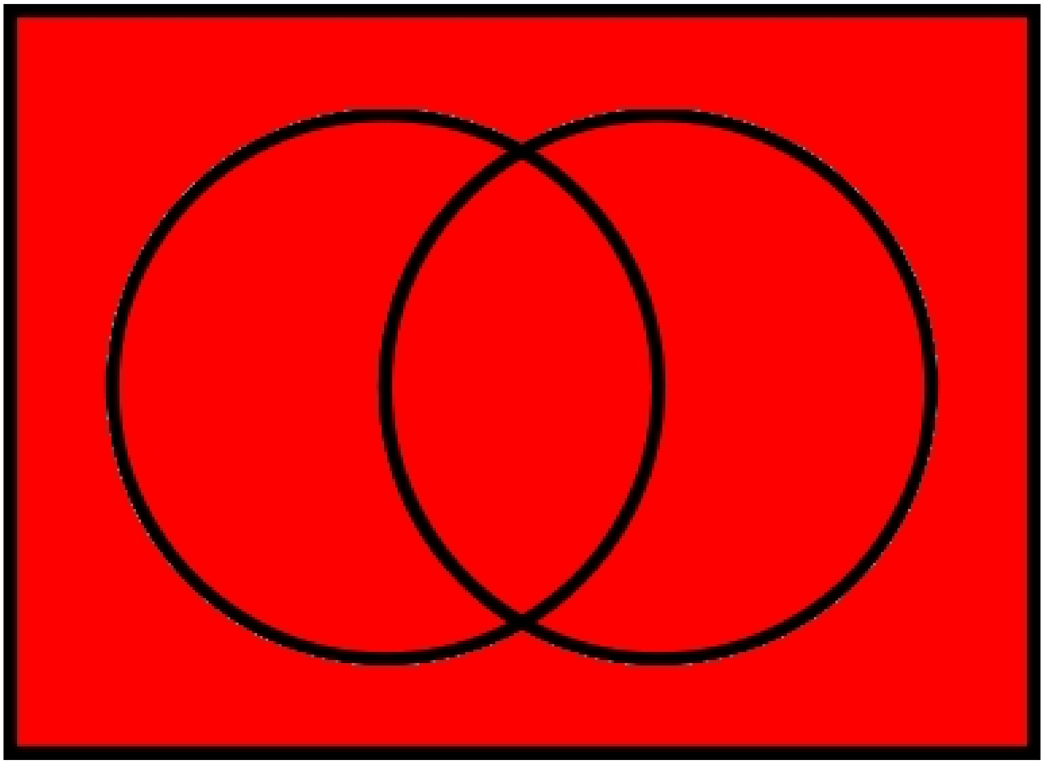


Figure 4: Total set, containing all elements.

[TODO: add "ID" identity/no change operator on A for the sake of exhaustivity ? Or is it overkill ?]

3.2.3 "NOT/NON/NEG": complementarity, negation ($\neg$, !)

The negation is a unary operator: it only applies to one set at a time, here A. The area in red is $\overline{A}$ (for set notation) or $\neg p$ (for its notation in logic). In computer programming, you'll generally see this as the symbol "!".

If A is the set of Linux computers, then $\overline{A}$ is the set of computers that are not Linux at all. This means a computer that is either Windows (the B potato) or Mac (a third circle one could imagine), but without "dual-booting with Linux" (a Linux dual-boot is a computer that contains both a Linux installation and an installation of another operating system). A Windows-Linux dual-boot would be a computer that belongs to the area common to A and B. Therefore, it would not belong to $\overline{A}$. If p is the proposition "it's raining", then $\neg p$ is the proposition "it is not raining".

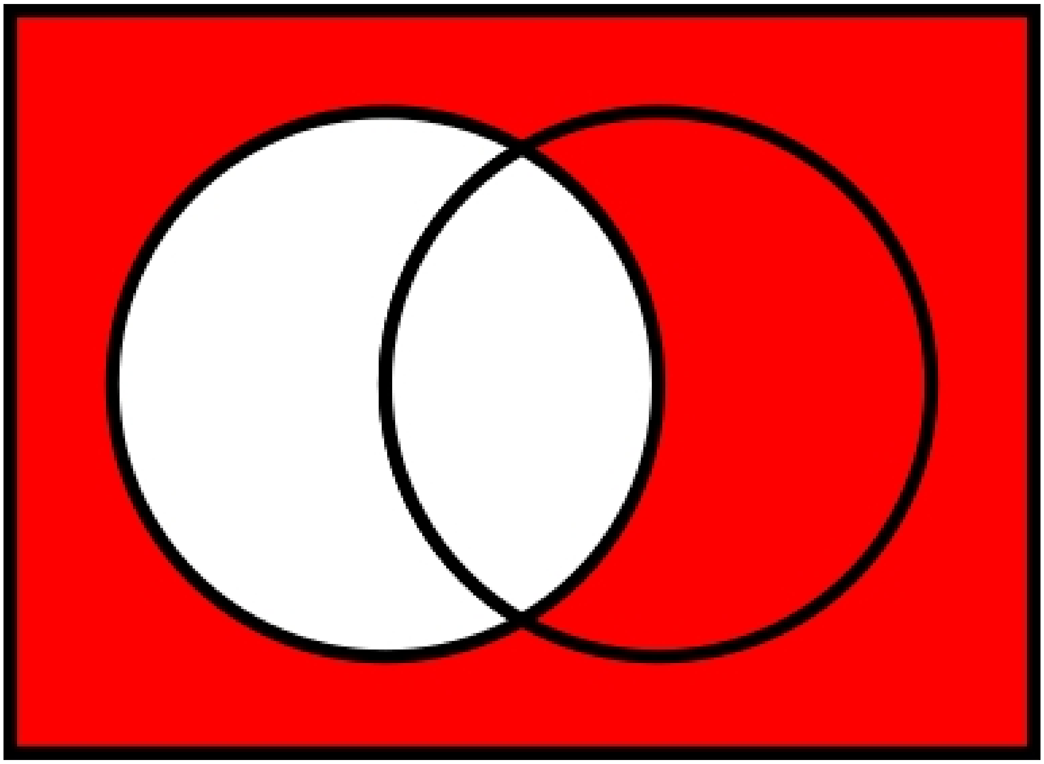


Figure 5: Set $\overline{A}$: set of all elements that are not in A

3.2.4 "AND": intersection, conjunction ($\cap$, $\wedge$, &&)

Set intersection is a binary operator: it applies to two sets, here A and B.

If A represents Linux computers, and B represents Windows computers, then the red area $A \cap B$ corresponds to the set of computers that can run both Linux and Windows, i.e. Linux+Windows dual-boots.

Logically, $p \cap q$ means that "p AND q"" is true only when "p is true" AND "q is true". For example, if $p =$ "It's raining" and $q =$ "I have a coat on my back", then $p \cap q$ is true only when "it's raining and I have a coat on my back", true only when p is true and q is true.

A typical example in info: we want to make sure that a number n is between 0 and 10 before continuing in our code. So we write the condition ""$0 \leq n \cap n \leq 10$"" - another way of expressing "$0 \leq n \leq 10$" but with binary operators that serve as base blocks ($\leq$ and $\cap$). Usually, in programming, "$\cap$" is written "&&", and has a bitwise version "&".

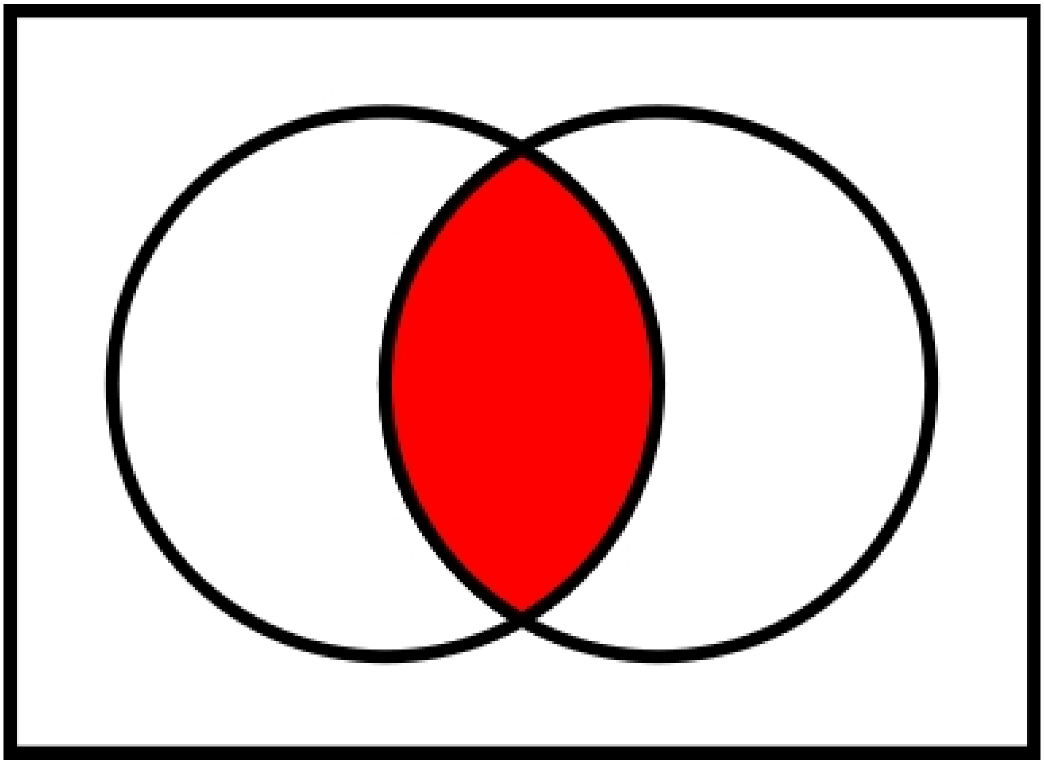


Figure 6: Set $A \cap B$: set of all elements in both A and B

3.2.5 "OR": union, (inclusive) disjunction ($\cup$, $\vee$, $||$)

Set union is a binary operator, it takes as input two sets A and B and returns a set $A \cup B$.

The logical OR is not the usual "or" in English, which usually corresponds to the XOR operator (just after). The logical OR is called "inclusive OR". If A and B represent Linux and Windows computers respectively, then $A \cup B$ represents the set of computers that can run either only Linux, or only Windows, or those the "dual-boots" that can run both. One often uses "and/or" in normal language to signify inclusive disjunction.

To make the distinction: - inclusive or: "you can become a millionaire by winning the lottery or by starting up your own business"; nothing prevents both from being true at the same time. - exclusive or: ”you have a choice: take a piece candy or a caramel”; it’s one or the other, but not both.

The rationale for the importance of the inclusive OR is mostly seen in logic. $A \cup B$ is true if A is true or if B is true. Either one of the two may be true, even if the other is false, the statement $A \cup B$ as a whole is still valid. If I say ”it’s raining or I have a coat on my back”, with an inclusive ’or’, then it can be raining, or I can have a coat on my back, or both. As soon as one of the two is true, the whole is true. That’s fundamental in computer science. For example, if you want the code to end, but there are several possible end conditions (e.g. checkmate, or stalemate, or a player giving up, in chess), any of these conditions is suitable to make the game stop (start the ”end of game” protocol) at the end of a turn. Usually, in programming, ”$\cup$” is written ”||”, and has a bitwise version ”|”.

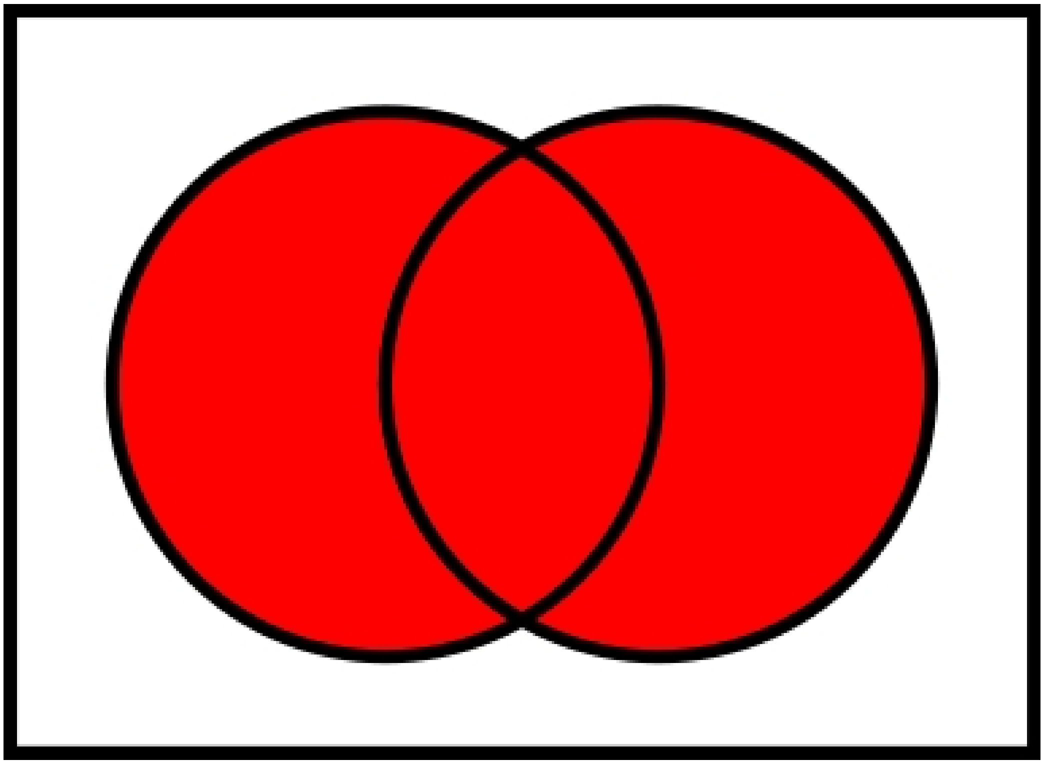


Figure 7: Set $A \cup B$: set of all elements in either A or B, or both.

3.2.6 "XOR": symmetrical difference, exclusive disjunction (Δ, $\oplus$, $\otimes$)

Symmetric difference, the more frequent usage of the word "or" ("I can give you a chocolate XOR a piece of candy; your choice"), is a binary operator. It is especially useful in electronics, a bit in cryptography, but not so frequent in logic. We would be more inclined to do $((A \cup B) \setminus (A \cap B))$, if needed in logic, where the "\" operator is the set subtraction operator, just below. The notation of the XOR operator varies a lot because it is rarely used. We recommend the notation Δ, because $\oplus$ is often reserved for the direct sum of vector spaces, and $\otimes$ for the tensor product; even if the probability of coming across an example where you have to deal with a symmetrical difference and a direct sum/tensor product of vector spaces is low, it is better to avoid it.

If A is the set of Linux, and B is the set of Windows, then $A \Delta B$ is the set of computers that are either pure Linux or pure Windows: i.e. no dual-boot.

Logically, it's pretty rare that something that works when A is true and when B is true doesn't work when both are true, so XOR is pretty rarely used, but it happens. I'll give a very important example below, because it's one of the first steps in computing, translating from the mathematical/logical world to the physical world with electrical currents. Also, there exists a bitwise version of the XOR operator, generally written ˆ, with the circumflex operator symbol.

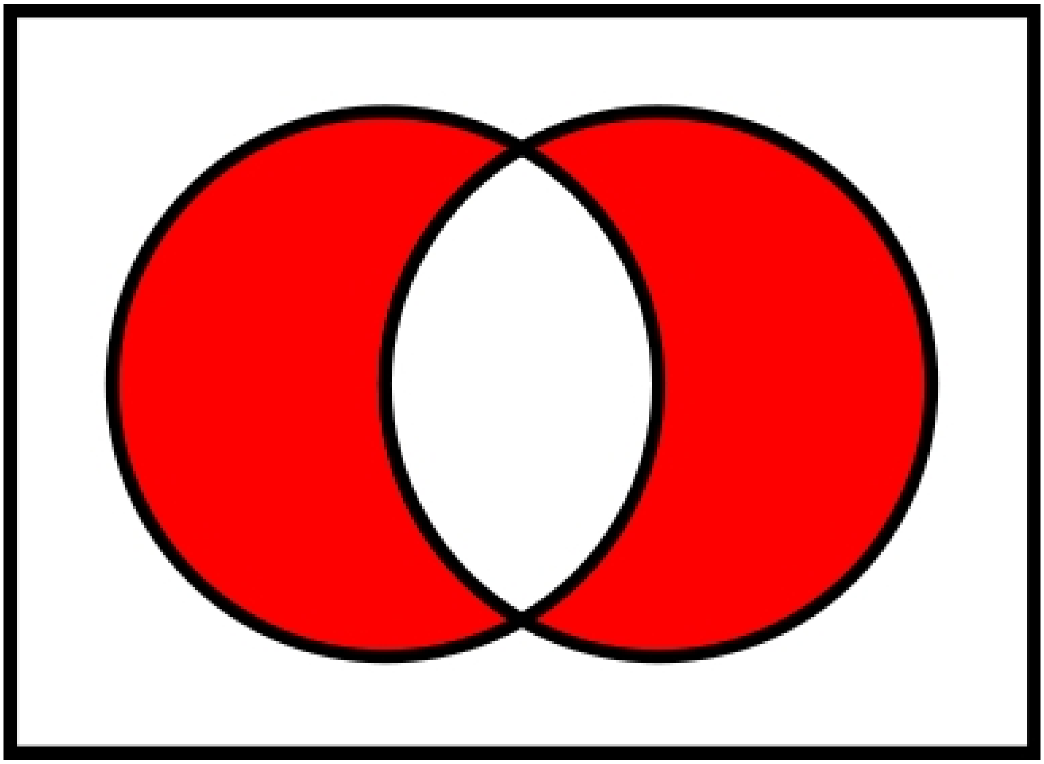


Figure 8: $A \Delta B$: set of all elements belonging to either A or B, but that do not belong to both

3.2.7 "A (AND) NOT B", set subtraction, $A \setminus B$

Here is the representation of the binary operator of set subtraction, $\setminus$. A is the disk on the left, B is the disk on the right, and $A \setminus B$ is the red area. This operation is read as "A deprived of B".

If p is the proposition "it's raining", and q is the proposition "I'm wearing a coat", then $p \setminus q$ means "it's raining and I'm not wearing a coat". It would be written as $p \cap \neg q$ in logic, in general.

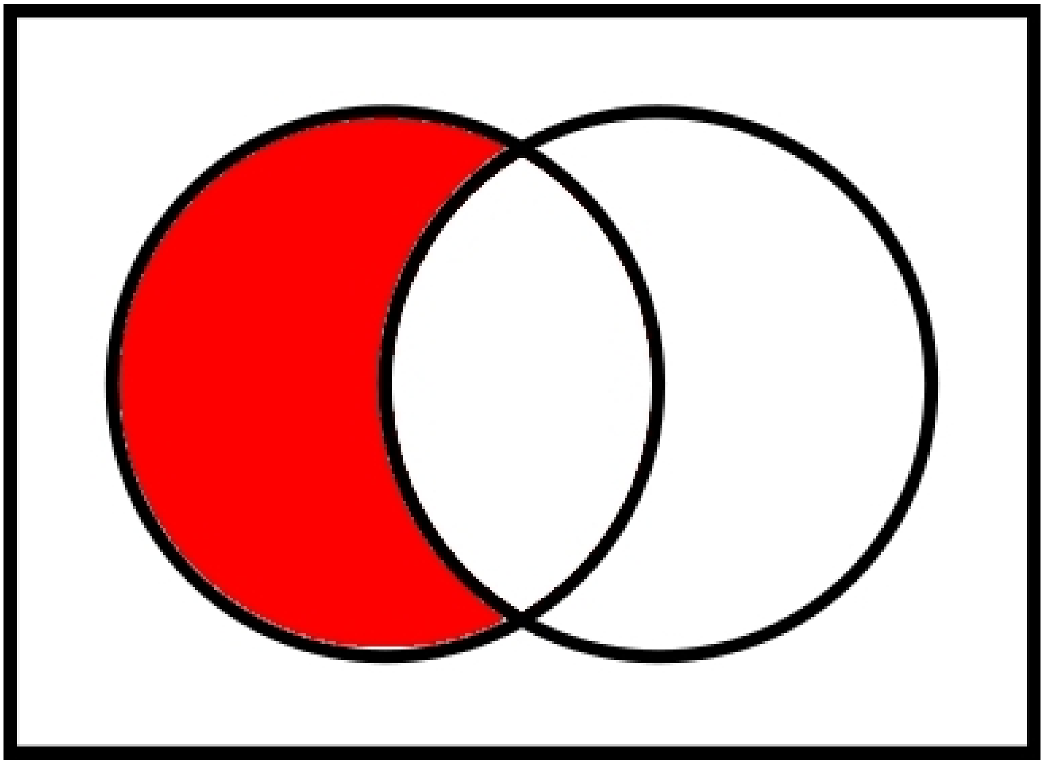


Figure 9: Set $A \setminus B$: the result of taking a set A and removing the elements of the set B

3.2.8 "If . . . , then . . . ": implication ($\Rightarrow$)

We come to the operation that is perhaps the most complicated of all in logic, but the one that is most often used intuitively without understanding its formalism. We don't really have a set-theoretic symbol to represent it, but that's okay because we can construct the same idea with other basic set operators (see the exercise at the bottom of this section). It is quite rare in computer science (except if you are doing pure logic for proof software or critical systems, such as software for nuclear power plants, for example). On the other hand, it is perhaps most important logical operator in everyday language and mathematics. This operator is called implication, and it is written $p \Rightarrow q$, and read "p implies q", or "if p, then q", or "so that p, it must be that q".

In particular, to understand the quirks of implication, look at the result when p is false and q is true in the truth table above, $p \Rightarrow q$ remains true. But when p is true and q is false, then $p \Rightarrow q$ is false. Seems strange, doesn't it? The problem with implication is that you symbolize it with an arrow, as if it only goes one way, but that's a misunderstanding. The logical link expressed by implication is really "between p and q", not only "from p to q".

Also, this idea "$p \Rightarrow q$" is an independent proposition, in its own right, with its own values of truths and so on. We could call it r and define it as $r := (p \Rightarrow q)$ to show that it is its own proposition ("logical assertion"). This is also the case for $\neg p$ which is its own proposition, for $p \cap q$ which is its own proposition, etc. This is important because sometimes, you'll know (the truth values of) p and q, but not (that of) $p \Rightarrow q$. Other times, you'll know q and $p \Rightarrow q$, but not p, etc.

If $p \Rightarrow q$ is true, we say that : - p is a **sufficient condition** for q. - q is a **necessary condition** for p

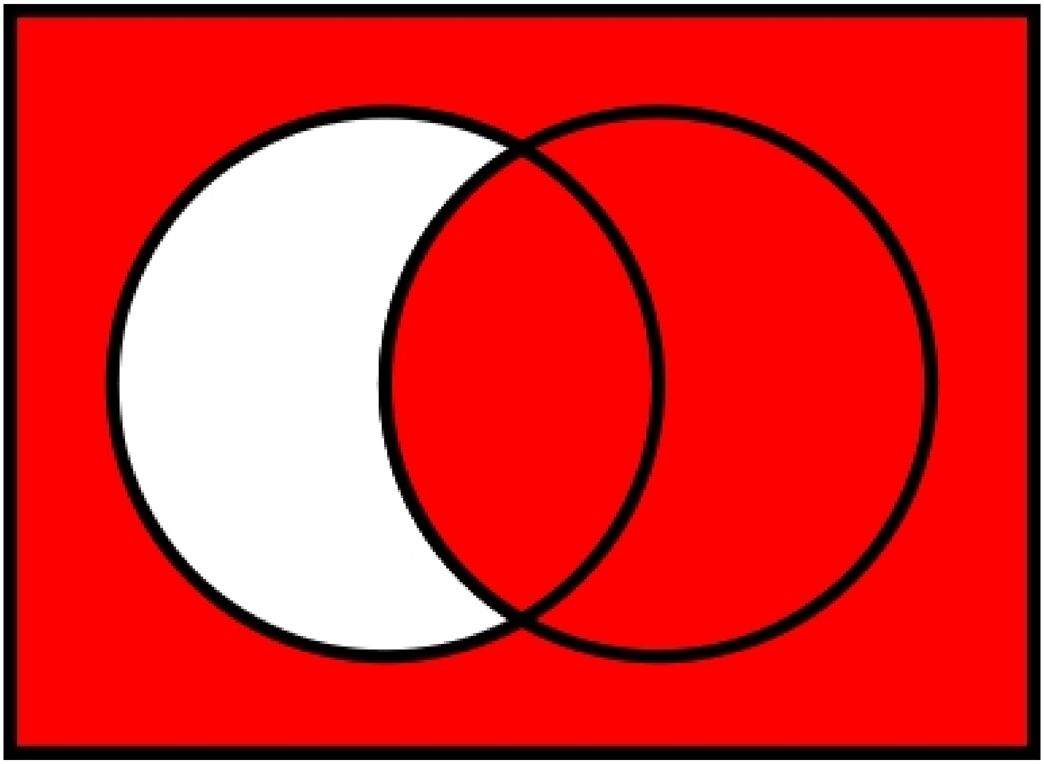


Figure 10: $A \Rightarrow B$: set of all elements for which the idea "if A then B" is valid

Let's say that if $p :=$ this object is a thumb and $q :=$ this object is a finger (and therefore $p \Rightarrow q$ is the idea that "if an object is a thumb, then that object is a finger"), then, in the most basic way I can explain it, the technical quirks of implication is the idea that "all thumbs are fingers, but not all fingers are necessarily thumbs". If it's a thumb, then it's a finger. Replace "thumb" with "square" and "finger" with "quadrilateral"; or "thumb" with "situation where it's raining" and "finger" with "situation where I have a coat on my back", etc: you'll get the equivalent examples for other propositions p and q.

It is "enough" to be a thumb to be sure you are a finger; it is "necessary" to already be a finger to even consider being a thumb. Saying that something is a thumb but is not a finger doesn't make sense, given how we define "thumb" and "finger" here. With "thumb $= p = A =$ true" and "finger $= q = B =$ false" (we are in A, but not in B: left crescent), the case that makes no sense is the white area in our diagram. But on the other hand, the fact that some fingers are not thumbs (the part $B \setminus A$, right crescent) does not prevent at all the fact that "all thumbs are fingers", thus becomes red zone (true) in the diagram. So when A (or p) is false, but B (or q) is true, $p \Rightarrow q$ remains true.

The example of the "coat on my back" may seem odd, but the mathematical world is much purer and more rigid in its structure than reality, or the human language that describes it. That's why we will use it as an example to build a translation between mathematical purity and human language in order to better understand all this.

Let's say that we are sure that $r := p \Rightarrow q$ is true, we are sure that "p implies q" is true. But we don't know the truth value of either p or q. We say that p is a sufficient condition of q. As soon as we have p true, we know (we can infer) that q is true. If p is false, one cannot say anything about q. Maybe it is true, maybe it is false. Imagine that I have a super power: as soon as it rains, even if I don't want to, I have a coat that appears on my shoulders. This superpower represents our $p \Rightarrow q$. p is true: it rains. Poof, I have no choice, the coat appears, and stays on as long as it rains. On the other hand, even when it's not raining (p false), nothing prevents me from still wearing a coat. Or not. Nothing prevents either scenario. We can simply deduce that if it rains, then Tristan will have a coat on his back, but that is limited: it takes p and $p \Rightarrow q$ both being true to deduce something about q (that q is true).

Now (still assuming that $p \Rightarrow q$ is true), we say that q is a necessary condition of p. And this is where we realize that the relationship goes both ways; not only with p and $p \Rightarrow q$ we can make deductions about q as we have just seen, but also with q and $p \Rightarrow q$ we can make deductions about p. We know that I have this power ($p \Rightarrow q$), but now we know the value of q, and we want to know if p is true or not. If q is false, you are SURE that p is false (because as soon as p is true, q is true). On the other hand, if q is true, you can't deduce anything. p can be true or false, it doesn't influence the truth of the link between p and q, of the implication.

Coming back to our scenario: imagine that someone tells you "I saw Tristan in a T-shirt yesterday" (q is false, I'm not wearing a coat); you know I have my superpower ($p \Rightarrow q$): you can be sure it wasn't raining (p is false), otherwise it would have been impossible (contradictory!) to see me without my coat. On the other hand, if they say they saw me with a coat on (q is true), it could very well just have been cold and not raining (p=?). We can simply deduce that if Tristan doesn't have his coat, then it wasn't raining, but this is limited: you need q false and $p \Rightarrow q$ true, to deduce something for p (that p is false).

These "limited pathways for proofs" help explain the weirdness: it's just the "behavior" of implication; i.e., we use an arrow to illustrate it, but it's just a logical "link" between two ideas. It tells us what relationship exists between ideas, what patterns we can draw between our ideas, which ones are bound to be true, which ones are bound to be false, and which ones we can't know anything about. The F's and T's are organized as they are in the truth table, even if it seems strange, so that your logical language can work and describe this phenomenon of link between the idea of the square and the idea of the quadrilateral, the thumb and the finger, which is neither an equivalence nor a total independence.

Also, from a set-theoretic perspective, to be "sure that $p \Rightarrow q$" (to suppose "$p \Rightarrow q$"), is really very simple! It's just having $q \subset p$ be true (the set of all thumbs is contained within the set of all fingers).

BONUS: We come to a funny consequence: a tautology (a logical theorem; a logical formula that remains true no matter what the values of the atoms p, q, etc. that make it up) is implied by all the ideas in your theory (your tautology is necessarily true, so you have to have it in your theory for your theory to hold up; it is a necessary condition for all the other ideas). A contradiction (or "negalogy", an idea that is necessarily false whatever the value of the atoms that compose it) on the contrary, implies all the ideas of your theory (it is necessarily false, so it does not pose any problem, does not affect the truth value of any implication)! It is on this idea that rests the principle of explosion (also called the law of non-contradiction), which is probably one of the most important laws of epistemology (the study of the rational scientific method). If in a theory, you can prove a contradiction (as true), then anything can be deduced, and your whole theory falls apart.

Exercise: show with Venn diagrams, a truth table and by boolean calculus that: $(A \Rightarrow B) \Leftrightarrow \neg(A \setminus B)$ and that $(p \Rightarrow q) \Leftrightarrow ((\neg p) \cup q)$

3.2.9 "If and only if", "XNOR": logical equivalence, ($\Leftrightarrow$, $\underline{\triangle}$)

Equivalence is a form of logical synonymy between two propositions. $p \Leftrightarrow q$ is true if and only if p and q have the same truth value. To show a mathematical equivalence is to show that one cannot have p without q - nor q without p - nor $\neg q$ without $\neg p$ - nor $\neg p$ without $\neg q$. The two ideas, p and q, are fundamentally linked and necessarily come together in your paradigm, your knowledge system. They are either both present, or both absent.

In practice we often demonstrate $p \Leftrightarrow q$ by showing $(p \Rightarrow q) \cap (q \Rightarrow p)$, hence the notation.

Note that definitions often hide an "if and only if" even when using the formula "It is said that ... is a ... if...". Why is that? Because a definition is something axiomatic, arbitrary. It is decided that this implication between usual expression and formal language is reciprocal, that these two entities are "mathematically synonymous". One chooses a word arbitrarily, being very formal about what it describes, in order to manipulate it with human language, which is one of the strong points of our brain. A definition in math is the work of translating a concept from mathematical language in order to use it in visual or vernacular language.

From a set point of view, to be sure that $p \Leftrightarrow q$ (suppose $p \Leftrightarrow q$), is to have $p = q$, quite simply (because we have $p \subset q$ and $q \subset p$, by reciprocal implications).

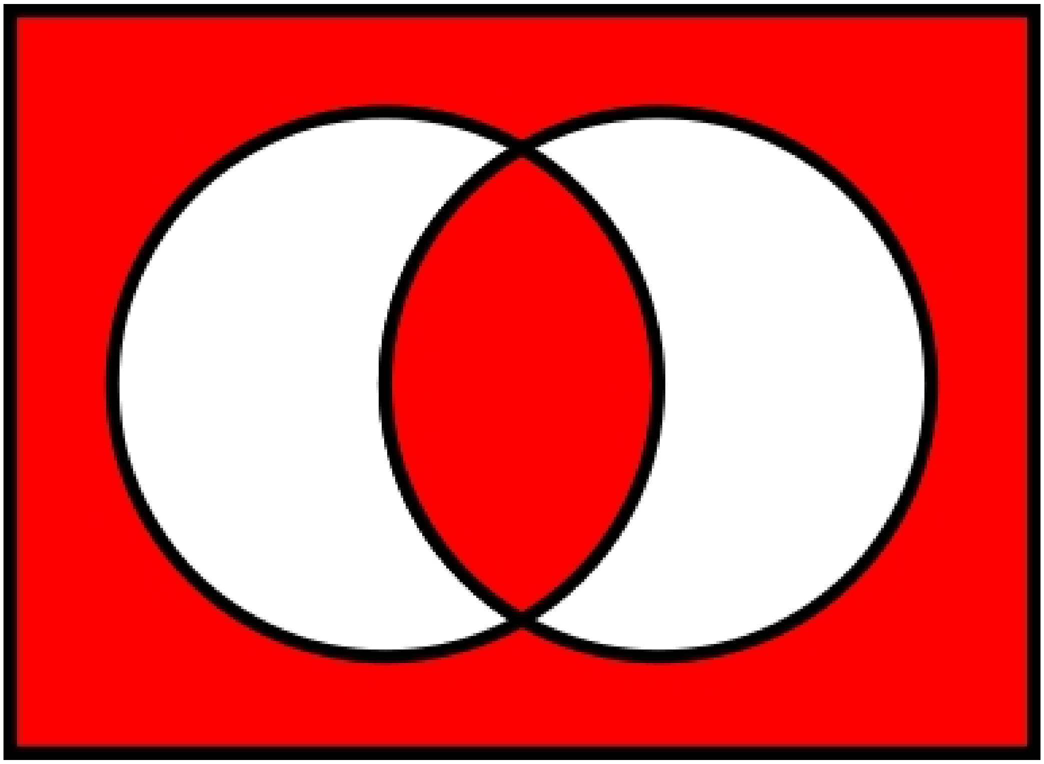


Figure 11: $A \Leftrightarrow B$: set of all elements representing ”A equivalent to B” as true

Bonus exercise: think and try to see how logical equivalence is an isomorphism (see below).

3.3 Practice of boolean logic: truth tables, logical calculus and logical theories

[Start with the tables, then explain the notion of tautology, and the principle of non-contradition from a formal point of view]. TODO: De Morgan's laws; absorption laws; properties of set algebras (associativity, commutativity, distributivity in both directions) ? put commutative diagrams ?] [TODO: explain models; speak quickly about multivalued logics, non-standard logics (notably modal logics), operation ”nand” which is totipotent and therefore automatically ”generates” the other operators; electronics, reduction to dis/conjunctive form, Karnaugh tables ?]

3.3.1 Some exercises

Exercise: Using a truth table and/or logical calculus, show the intuitive idea that if two things imply each other mutually, then they are equivalent. Formally: show that $(p \Leftrightarrow q) \Leftrightarrow ((p \Rightarrow q) \cap (q \Rightarrow p))$ is a tautology. Note the use of parentheses for those who would use parentheses intuitively without being aware of the underlying intellectual process: what is most inside is done first (I know that some people can get lost in their programmatic "ifs", which usually causes avoidable bugs and wasted time, so it's not bad to practice all these little logical manipulations...).

Exercise: Search online for "normal conjunctive/disjunctive form" and negate an example you find. Check your result with a truth table, or possibly Venn diagrams, even if in this case I would advise to have fun trying to code a solver for this problem. Also practice expressing your 'if' more intelligently with negation (there is often an optimization to be found on complex AND and OR mixes when negations are involved).

Bonus exercise: try to draw (or represent in your head) $A \star B \star C$ with Venn diagrams, where $\star$ represents in each case either the intersection, the union, or the symmetrical difference. Hint: AND, OR and XOR are associative and commutative (see below for definition), so the order is not important, and the notation without parenthesis is legitimate. One can always give a new name to a set obtained by a complicated calculation to make it a shortcut in the following calculations (ah, sweet variables).

Bonus exercise: For which logical operators is the absence of parentheses ambiguous or even contradictory?

Bonus exercise: same question, but analyzing combinations of operators.

[TODO: add Venn diagram web applet here]

3.3.2 To go further

1) Find a list of tautologies to prove, and prove them.

2) Try to design a basic binary calculator, a "Full Adder". Because if you know binary, you are finally able to know how a computer "thinks", basically. How you go from logic to arithmetic. It's all based on binary. True or False. 1 or 0. When you make a sum in binary, if you focus on only one column of your addition's inputs, you only have four possible cases: 0 AND 0, 0 AND 1, 1 AND 0, 1 AND 1. Let's replace the AND with +. We get "$0 + 0 = 0$", "$0 + 1 = 1$", "$1 + 0 = 1$, and "$1 + 1 = 10$", knowing that

$[10]_2$ is the binary representation of the number $[2]_10$ in our usual decimal base. Your ”carry” in the addition is not ten, but two, and each following column corresponds to each power of two (which are written as $[10]_2 = [2]_10$, $[100]_2 = [4]_10$, $[1000]_2 = [8]_10$, etc, because, yes, bases are interesting).

Now look at the units column. When your input numbers are the same, the output units digit is 0, when they are different, it's 1. Here is the addition: a XOR logic gate on binary. The carry, if it's there (recognized by an AND gate, we only have a carry if we have 1 AND 1 for our input), is sent to an identical block that will do the same job for the ”2-s”, then the ”4-s”, then the ”8-s”, as we know how to do with our tens and hundreds since grade school. Here, summarily put, is the core of computer science: *Through logic, we have tricked rocks into thinking.*

Optimize your Full Adder electronic design with Karnaugh tables. Google/Duckduckgo/Etc (+Images) Full Adder, Substractor, etc.

4 Set theory

4.1 Preliminary vocabulary, reminders

We call **empty set** and we note $\emptyset$ (or $\{\}$), the only set without elements. The empty set is always a subset of any set.

It is said that two sets A and B are **disjoint** if their intersection is empty (the two potatoes don't even touch each other). Formally, $A \text{and} B \text{disjoint} \Leftrightarrow A \cap B = \emptyset \Leftrightarrow \neg(\exists x \in A, x \in B)$

Let E a set. We call **powerset** of E, and note $\mathcal{P}(E)$, the set of all subsets of E. For example, if $A = \{a, b, c, d\}$, then

$$\begin{aligned}\mathcal{P}(A) = \{&\emptyset, \{a\}, \{b\}, \{c\}, \{d\},\\ &\{a, b\}, \{a, c\}, \{a, d\}, \{b, c\}, \{b, d\}, \{c, d\},\\ &\{a, b, c\}, \{a, b, d\}, \{a, c, d\}, \{b, c, d\}, \{a, b, c, d\}\}\end{aligned}$$

NB: $\mathcal{P}(\emptyset) = \emptyset$

We call **partition** of a set a division of this set into disjoint subsets. Example: $A = \{1, 2, 7, 12, 15, 21\}$, a set of numbers, then (A_1, A_2, A_3), where $A_1 = \{1, 2\}$, $A_2 = \{15\}$ and $A_3 = \{7, 12, 21\}$, is a partition of A.

One calls **cardinal** of a E set, noted $card(E)$ or $\#(E)$ the number of elements contained in E.

Ex: $card(\{1, 2, 7, 10\}) = 4$.

NB :

- $card(\mathbb{N}) = card(\mathbb{Z}) = card(\mathbb{Q}) = \aleph_0$, also called "discrete infinity" (read "aleph null" or more rarely "aleph zero")

- $card(\mathbb{R}) = card(\mathbb{C}) = card(\mathbb{R}^n) = \mathfrak{c}$, also called "continuous infinity". Note that we have $card(\mathcal{P}(\mathbb{N})) = \aleph_0^{\aleph_0} = 2^{\aleph_0} = \mathfrak{c}$.

Reminder :

- $\mathbb{N}$ is the set of natural numbers $\{0, 1, 2, 3, 4, 5, 6 \ldots\}$. $\mathbb{N}^*$ is $\mathbb{N}$ without 0.

- $\mathbb{Z}$ is the set of integers $\{\ldots, -6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6, \ldots\}$.

- $\mathbb{Q}$ is the set of rational numbers (=fractions) $\{\frac{a}{b} \mid a \in \mathbb{Z} \text{ and } b \in \mathbb{N}^*\}$. We speak of "rationals" because any fraction expresses a proportion, a "ratio", between two numbers.

- $\mathbb{R}$ is the set of real numbers; roughly speaking, if we make $(u_n)_{n\in\mathbb{N}}$, the numerical sequences with (output) values $\mathbb{Q}$, then the set of all the possible sequence limits builds the real line; we fill the "holes" between the rationals to make a clear line. The square root of 2 or the number π are "irrationals", examples of numbers that are real but not expressible as a rational, only as a sequence of rationals that get closer and closer to them, at infinity. The number π, for example, can be written: $3/1+1/10+4/100+1/1000+5/10000\ldots$, and be approximated afterwards $(u_n)_{n\in\mathbb{N}} = (3, 3.1, 3.14, 3.141, 3.1415, \ldots)$.

- $\mathbb{C}$ describes the set of complex numbers (also called imaginary or more rarely transversal numbers, even if it would have been a much better name, since better represents what they are); it is a $\mathbb{R}$-algebra (you will see that below) built on $\mathbb{R}^2$ with vectors $\{1, i\}$ as a basis and $i^2 = -1$ as a multiplicative property. Any number of $\mathbb{C}$ can be expressed $x + i * y$ with x and y two real numbers.

NB: we have $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$. The set of decimal numbers is a subset of $\mathbb{Q}$ (see also p-adic numbers).

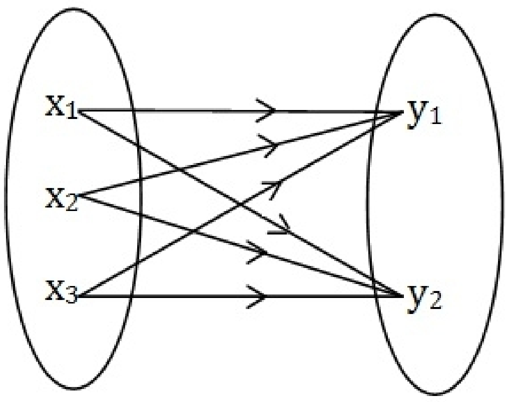
NB: notice the oddity here: there are as many natural (positive) integers as there are relative integers, and as many as there are fractions. But on the other hand, there are so many more real, irrational "non-fractional" numbers, that it becomes a "larger" type of infinity... but which remains the same, even when extended to n dimensions! We'll see how much lower down when we work on the notion of bijection.

[TODO: a gif which shows the evolution of N to C ?]

4.2 Relations between sets

We call the **Cartesian product** of two sets A and B, and note $A \times B$, the set containing all pairs of elements of A and B (where the order matters, and the element of A is given first). These "ordered pairs" are generally simply called "**pairs**" (more rarely "couples"), and 2D vectors are a special case of this concept.

For example: $A = \{a, b\}$ a set with 2 elements, and $B = \{1, 2, 3\}$ a set with 3 elements. Then $AtimesB = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), (b, 3)\}$, a set with $2 \times 3 = 6$ elements.



$A=\{x_1, x_2, x_3\}$ $B=\{y_1, y_2\}$

$A*B=\{(x_1, y_1), (x_1, y_2), (x_2, y_1), (x_2, y_2), (x_3, y_1), (x_3, y_2)\}$

Figure 12: Example of the Cartesian product of two sets

For $A \times A$, one can write A^2, the same applies for A^n in general (for the sequence of successive Cartesian products involving n times the set A). In general, to specify that we are selecting n elements of a set A, we prefer to write that we're choosing a single element of the n-fold Cartesian product of A with itself, which is denoted A^n. Elements of A^n are called n-tuples. 2-tuples are often called "pairs" and 3-tuples "triples".

NB: if you do category theory, $|\mathbb{R}^n|$ designates the underlying set the vector space of dimension n based on real numbers, $\mathbb{R}^n$. In computer terms, we can think that $\mathbb{R}^n$ is the set of all imaginable arrays of float of size n (even if the computer must be satisfied with an approximation of this pure mathematical space). If you are coding video games, or making 2D sprites or 3D models, you have worked (maybe without knowing it) on a computer version approximating $\mathbb{R}^2$ or $\mathbb{R}^3$.

Exercise: Sometimes sets have an infinite number of elements. This is the case of real intervals. If we take the interval [0.1] (all x such as "$0 \leq x \leq 1$") and the interval [4.6], what is a good way to display $[0.1] \times [4.6]$, the set of all pairs with as first coordinate an element of [0.1] and as second coordinate an element of [4.6]? Hint: We have two real intervals... Use $\mathbb{R}^2$, the real plane! Bonus: How can this visualization be extended to larger dimensions?

A "**relation** between two sets A and B" is a set containing a choice of pairs of elements where the first member of each pair comes from A and the second member of each pair comes from B. Another way of looking at it: a relation $\mathcal{R}$ between two sets is simply a subset of their Cartesian product.

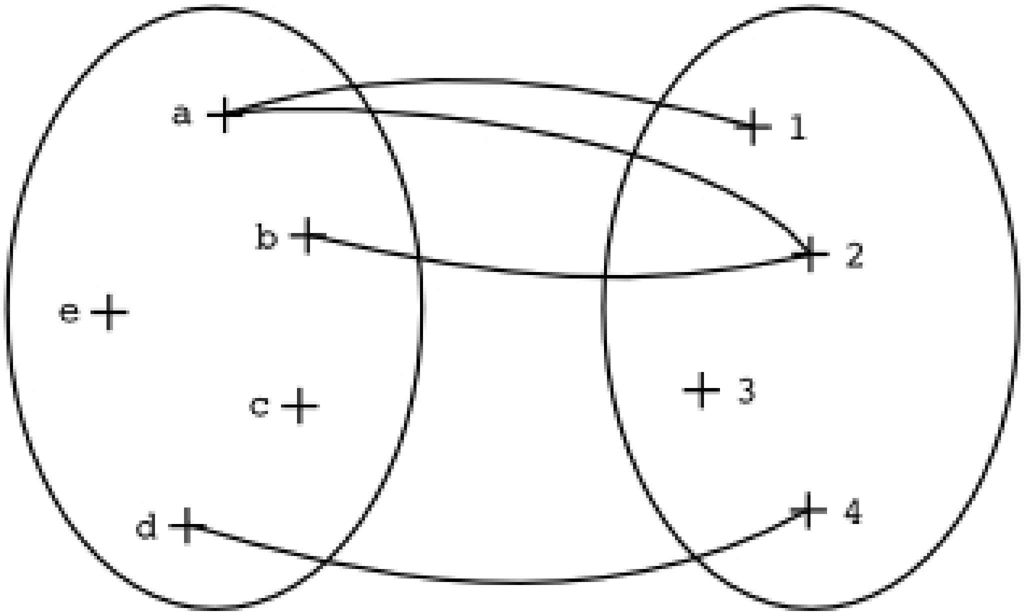


Figure 13: In the diagram above, we have two sets $A = \{a, b, c, d, e\}$ and $B = \{1, 2, 3, 4\}$. The pair $(a, 1)$ can be understood as the link between $\{a\}$ and $\{1\}$. Consequently, if we had to describe the set $\mathcal{R}$ of links in the diagram above, we would write $\mathcal{R} = \{(a, 1), (a, 2), (b, 2), (d, 4)\}$.

A relation of a set E to E itself, is said to be "binary" (nothing to do with binary in computer science, beware). Any binary relationship is a subset of E^2.

A binary relation $\mathcal{R}$ can be :

– **reflexive**: $\forall x \in A$, on a: $x\mathcal{R}x$ (ex: $=$, $\geq$ and $\leq$ are reflexive, because for example "$x \leq x$" is always true)

– **irreflexive** : $\forall x \in A$, we have "$\neg(x\mathcal{R}x)$". (ex: $<$ and $>$ are irreflexive, because "$x < x$" is always false; $\neg$ is read as "not...")

– **symmetric** : $\forall (x,y) \in A^2$, we have $x\mathcal{R}y$ => $y\mathcal{R}x$ (ex: equality, because $x = 2 \Rightarrow 2 = x$)

– **antisymetric** :

$$\forall (x,y) \in A^2,\ \text{we have: } \begin{cases} x\mathcal{R}y \\ \\ y\mathcal{R}x \end{cases} \Rightarrow x = y$$

– **transitive** :

$$\forall (x,y) \in A^2,\ \text{we have: } \begin{cases} x\mathcal{R}y \\ \\ y\mathcal{R}z \end{cases} \Rightarrow x\mathcal{R}z$$

NB: $=, \geq, \leq, >, <$ and parallelism // are transitive, perpendicularity $\perp$ is not.

NB: for the definition of the symmetry of a relation, notice that we have in formula an implication "$\Rightarrow$" which seems to go only in one direction, but as this property is valid for all couples of A^2, it is also true in the opposite direction if we start with the couple (y,x), we would have $2 = x \Rightarrow x = 2$, and so it is in fact also an obvious equivalence ("$\Leftrightarrow$")! But it's always cleaner and better in math to keep the version of a definition that makes the least assumptions, which is why it's defined this way.

NB: for the definition of the antisymmetry of a relation, the idea is that the ONLY case where it can go both ways ($x\mathcal{R}y$ AND $y\mathcal{R}x$) is the EVENTUAL case of equality (e.g.: $\leq$ is antisymmetric because if "$x \leq y$ AND $y \leq x$" then necessarily "$x = y$"). This does not mean that this case of equality exists ! Just that if we see $x\mathcal{R}y$ and $y\mathcal{R}x$ with a relation that we know to be antisymmetric, we can deduce that it is the case of equality (e.g.: $<$ and $>$ are antisymmetric but are nevertheless irreflexive so have no case of equality, so you would have reached a contradiction in case of ($x\mathcal{R}y$ AND $y\mathcal{R}x$) where $\mathcal{R}$ is a 'strict partial order').

We call "**equivalence relation**" any relation which is : **reflexive, symmetric, transitive** (examples: equality, congruence modulo n, parallelism). These play a fundamental practical role in higher mathematics, but less so in computer science, unless you want to push your functional programming further (it's worth it).

If $\sim$ is an arbitrary equivalence relation, we note $[x]$, and we call "equivalence class of an element x modulo the relation $\sim$" the set of elements which are "equal" to x if we consider that this relation is a form of equality, i.e. all y such as $x \sim y$. For example, for parallelism, if x is a horizontal line, all other horizontal lines (which we would name $y_1, y_2, \dots$) are "equal" to x within the context of parallelism because they are parallel to x. The "equivalence class of the line x modulo the parallelism relation //" is the set of all lines whose direction is horizontal. Any line in this equivalence class can be used to illustrate "horizontality" (x's direction): we say that x is a representative of the class $[x]$ and any element (either x, or y_1, or y_2,...) is suitable to be a representative of its class (here, class of lines with a certain direction).

NB: The representative x is in the space with the initial lines (let's name it E), as well as the y_i lines. On the other hand, $[x]$ is in another space, called "quotient space of E by the equivalence relation $\sim$", and noted $E/\sim$. For such a "quotient by equivalence relation" (construction of the space where equivalent elements are reduced to a single element) to be allowed, it is just necessary to verify that the same objects of an equivalence class have the same role, function perfectly identically, in the arrival set. A good example, on a clock with 12 hour notches, advancing by 16 hours and advancing by 4 hours gives the same result (so 16 and 4 are equivalent in this context). The word "modulo" for equivalence relations also refers to the "congruence modulo n" operator, used in the quotienting from which the clock's arithmetic (modular arithmetic) is defined, but we must not confuse these uses of the term modulo (one is a lot more general).

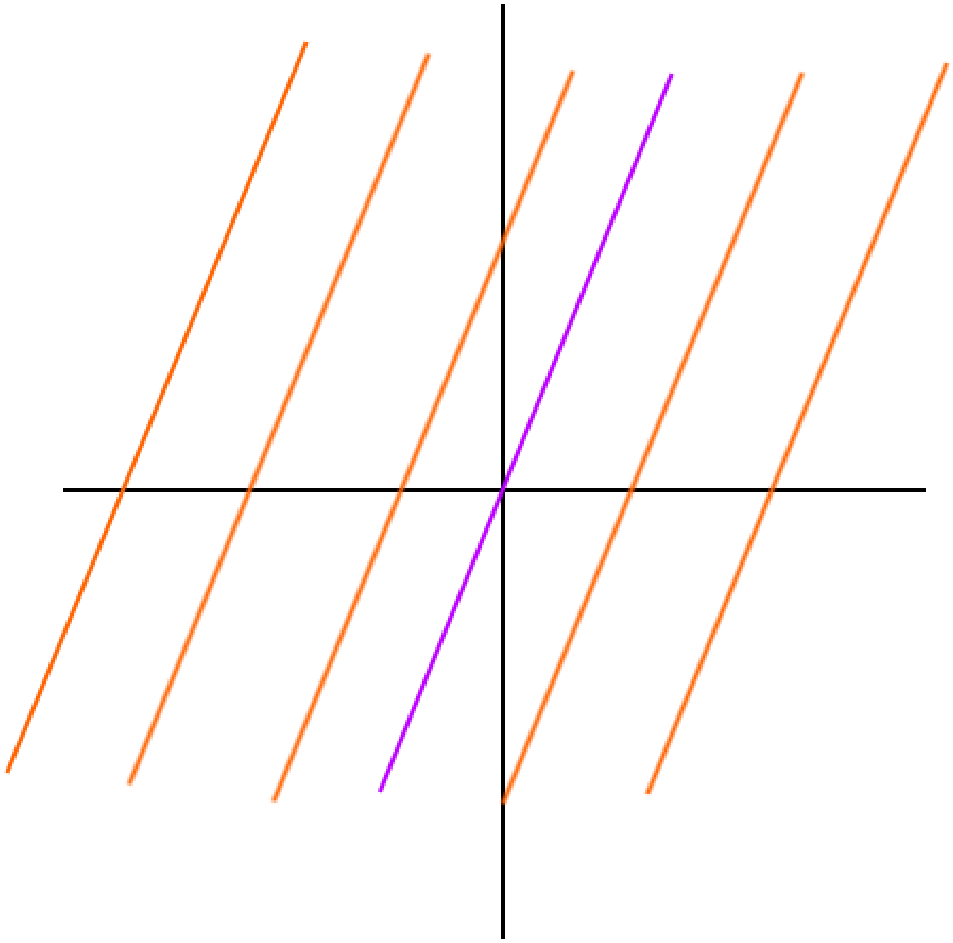


Figure 14: Example of a quotient space by the equivalence relation corresponding to parallelism, part 1. We have $E = \{ax + by = c \mid (a, b, c) \in \mathbb{R}^3 \text{and} (a, b, c) \neq (0, 0, 0)\}$, the set of lines in the plane (note that the same line can have several ways of being represented as an equation; precisely, a line d_1 defined by the equation $ax + by = c$ is equal to a line d_2 defined by $Ax + By = C$ iff $\exists k \in \mathbb{R}^*, A = ak, B = bk, C = ck$). In the figure above, we have different members of the same equivalence class for the equivalence relationship representing parallelism. These parallel lines d_i differ algebraically from each other only for their respective c_i parameter. The equivalence relation of parallelism could therefore be expressed algebraically here as $d_1 \sim d_2 \Leftrightarrow \exists k \in \mathbb{R}^*, a_1 = ka_2 \text{et} b_1 = k * b_2$. We choose only one representative of the equivalence class: the purple line.

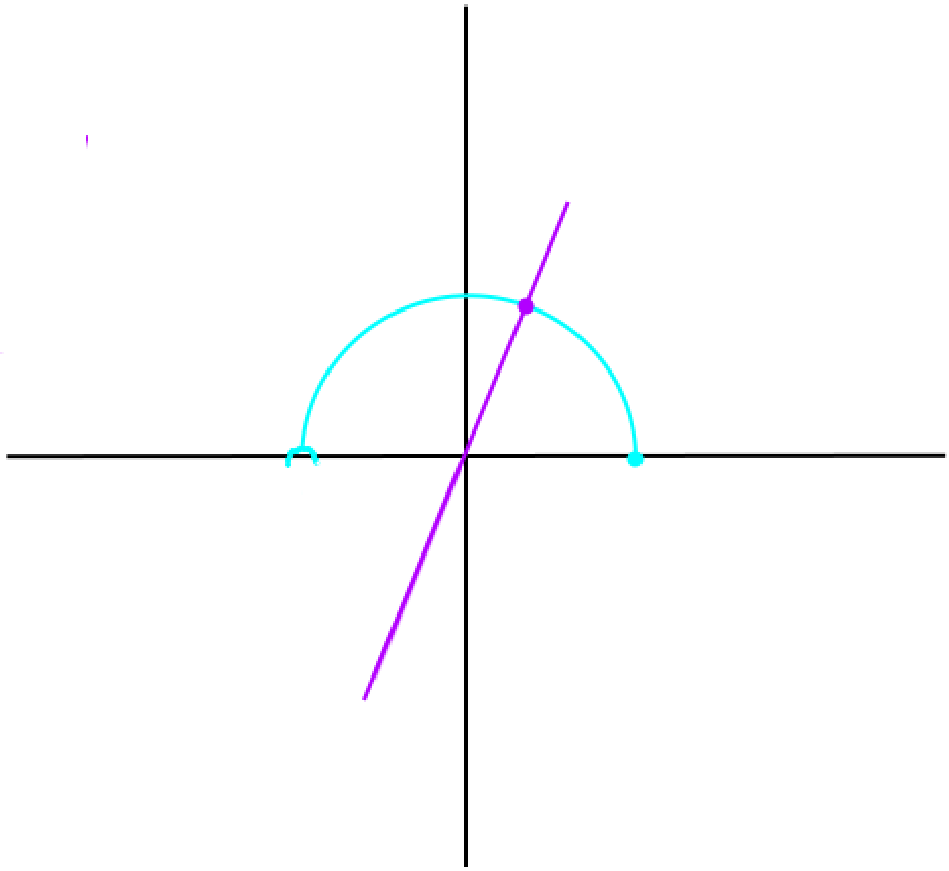


Figure 15: Example of a space quotient by the equivalence relation corresponding to parallelism, part 2. We can do what we did in part 1 of the example for any possible direction. If, for every direction, we reduce each equivalence class to a single element (choose the purple line among the orange lines), and represent each element as a single point (the purple point), we obtain the quotient space (the cyan semicircle, with one edge included, one edge excluded), which is the set of all such purple points. There are precisely as many points on the cyan semicircle as there are distinct directions that our straight lines can take, and each point on this semicircle corresponds to one and only one direction. Note that other representatives/points could be chosen for the quotient space; ie, this geometric representation of the quotient space is not the "truth", but just a representative of it. The actual properties and behavior of the quotient space (and thus, also which representation of this quotient space that you should choose) will depend on "where" this quotient takes place; but to explain this, we will have to explain the notion of an "algebraic structure" and that of a "category".

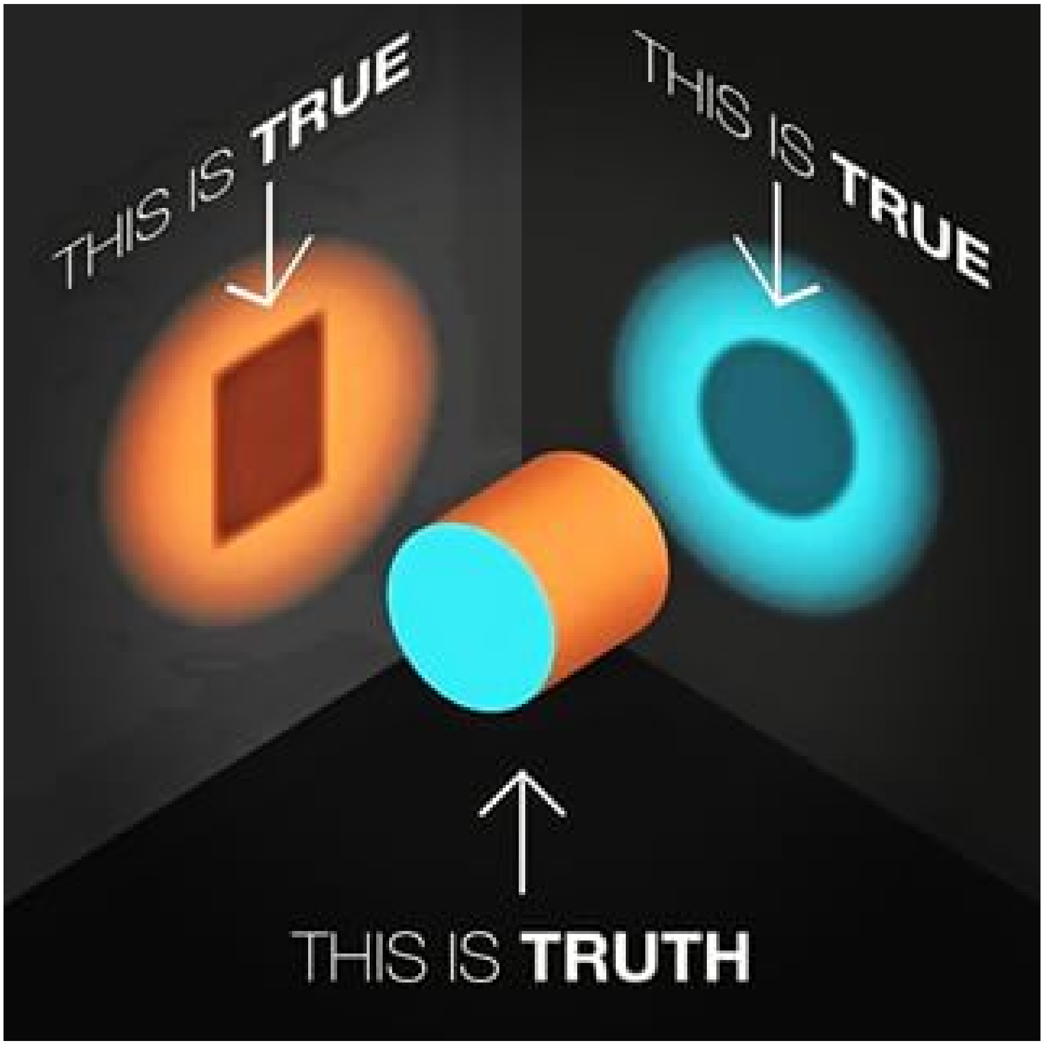


Figure 16: Distinction between a "true" representation, and the "truth" it represents. A representation is generally a "choice of point of view", and there are often points of view that will be more appropriate in some cases and not in others. The belief in the existence of an objective mathematical "truth" is called mathematical "idealism" or "platonism"; it is a debate that has been raging for more than two millennia. Even if mathematicians are divided on this philosophical question, all agree that "the map is not the territory": that a representation is never absolutely identical to the represented object.

A **partial order** (more rarely, an "order relation"), generally denoted "$\leq$", refers to any relation which is: **reflexive, antisymmetric, transitive** ($\geq$, $\leq$, or the inclusion of $\subset$ between sets, are partial orders). A total order is a partial order for which every pair of distinct elements are related by the relation (ex: $\leq$ and $\geq$ are total orders on $\mathbb{R}$, but not on $\mathbb{C}$). Partial orders play an important role in the foundations of theoretical computer science, especially for the definition of recursion. Indeed, how can we define a "bottom" case that can be processed in real time, if we have no notion of what is the "bottom" of a data structure?

We call a strict partial order any relation: irreflexive, antisymmetric, transitive (ex: $<$ and $>$). Strict orders are much rarer, in general non-strict partial orders are more efficient to construct relevant things, to the point where if you hear "order relation" or "partial order" without further specification, it is because we are talking about a non-strict order. On the other hand, we will specify whether the order is total or partial.

NB: Many relations have no particular characterization (e.g.: perpendicularity which is irreflexive and symmetrical, but is not "special" because of that; just like any other arbitrary relation with not special property).

We call **poset** or **partially ordered set** the pair $(X, \leq)$ of a X set with a partial order on its elements. An order is **total** on X if any element of X can be ordered with any other, i.e. if $\forall(x, y) \in X^2, x \leq y$ or $y \leq x$.

Exercise: $\leq$, $\geq$ are total orders on $\mathbb{R}$, but what about the $\mathbb{C}$ set of complex numbers, or $\mathbb{R}^n$? Why ? Try to invent a total order on $\mathbb{C}$ - there are an infinite number of possible correct answers! [A classical example is the lexicographical order.]

NB: The inclusion operator "$\subset$" defines a partial order. If we take $E = \{1, 2\}$, we have $\{1\} \subset E$ and $\{2\} \subset E$, but we don't have $\{1\} \subset \{2\}$, nor $\{2\} \subset \{1\}$. Exercise: Draw the poset representing inclusion in $\mathcal{P}(\{a, b, c\}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}$, the powerset of $\{a, b, c\}$. One can however define a 'strict' inclusion of the sets. We then speak of "proper subset".

NB: posets, which can be represented as acyclic oriented graphs, play an important role in category theory, because they are a way of analyzing categories and their behavior.

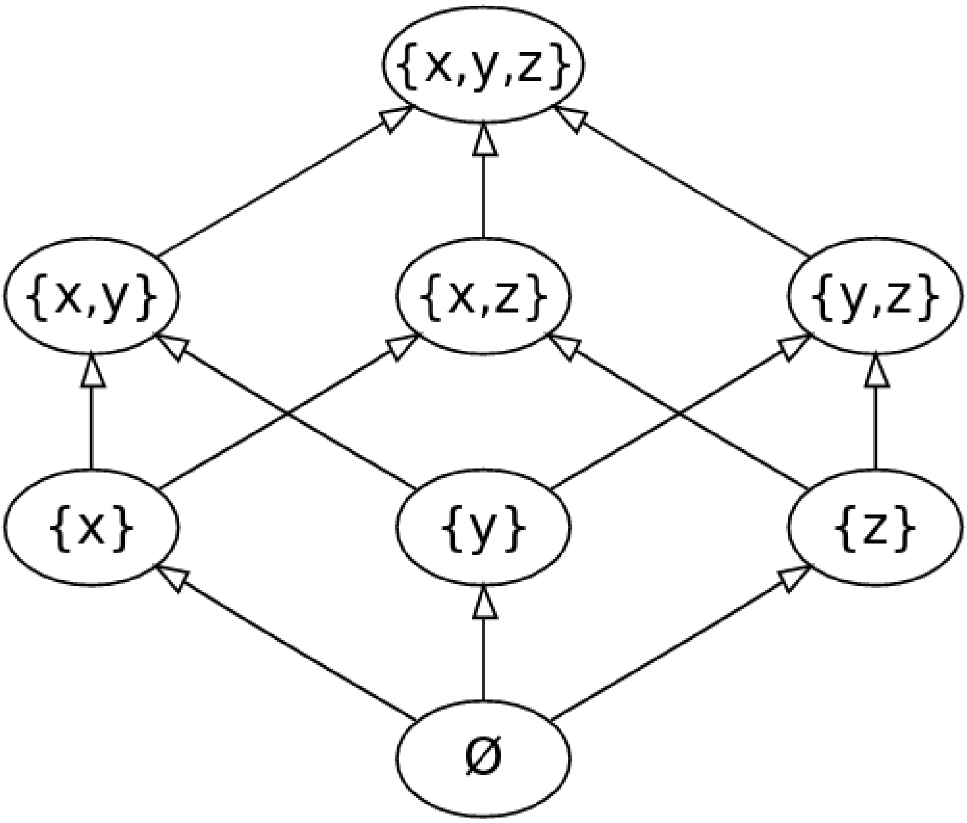


Figure 17: Example of Hasse diagram of the powerset $\mathcal{P}(\{x, y, z\})$, representing set inclusion poset as an acyclic directed graph.

4.3 Functions

The defining characteristic of set functions, *a contrario* of a computer function like "rand()", is that they are unambiguous: giving it 4 as input that gives you 16 as your output once will always give a 16 as output for 4 as input. If your function is fixed/defined, the case where you give 4 as input then get a 16 as output once, and a -21 as output at another time is NOT possible. At least not with what is called a "function" in set theory.

Technically, a (set) function is just a particular type of relation between sets $A\mathcal{R}B$ where $\mathcal{R}$ is generally replaced by an arrow, and the relation's/function's name precedes the rest of the formula (ie, $f : A \to B$). In this context, the element of A, called the **argument**, plays the role of input; the element of B, called the **image**, plays the role of output. A is called the **domain** of f, and B its **codomain**. Formally, we say that F is "a **function** of the A set in the B set" if F is a relation between A and B such as :

$$\forall(a_1, a_2) \in A^2, \forall(b_1, b_2) \in B^2, (a_1 F b_1 \text{ AND } a_2 F b_2 \text{ AND } a_1 = a_2) \Rightarrow b_1 = b_2$$

or, in a simpler but different notation, more appropriate for function, you have the equivalent formula:

$$\forall(x, y) \in A^2, x = y \Rightarrow F(x) = F(y)$$

NB: if F contains the pair (a, b) (a is the argument, and $b = F(a)$ the image) then, equivalently, one can write aFb. It's just that $a \to F(a)$ is a more explicit notation with respect to the role of functions, which is to transform one object into another according to a fixed protocol. Moreover, $[a]F[F(a)]$ is needlessly wordy.

NB: if you see the term **"map"**, or "application", you can understand it as "function". The technical distinction that is sometimes made, is that a function is a map iff it has an image for all the elements of A (its domain $dom(f) = A$). For example, $x \to \frac{1}{x}$ is a function that can be defined on $\mathbb{R}$, but is a map only on $\mathbb{R}^*$ because 0 is not invertible, and thus has no image in $\mathbb{R}$. We will generally not be making the distinction unless necessary.

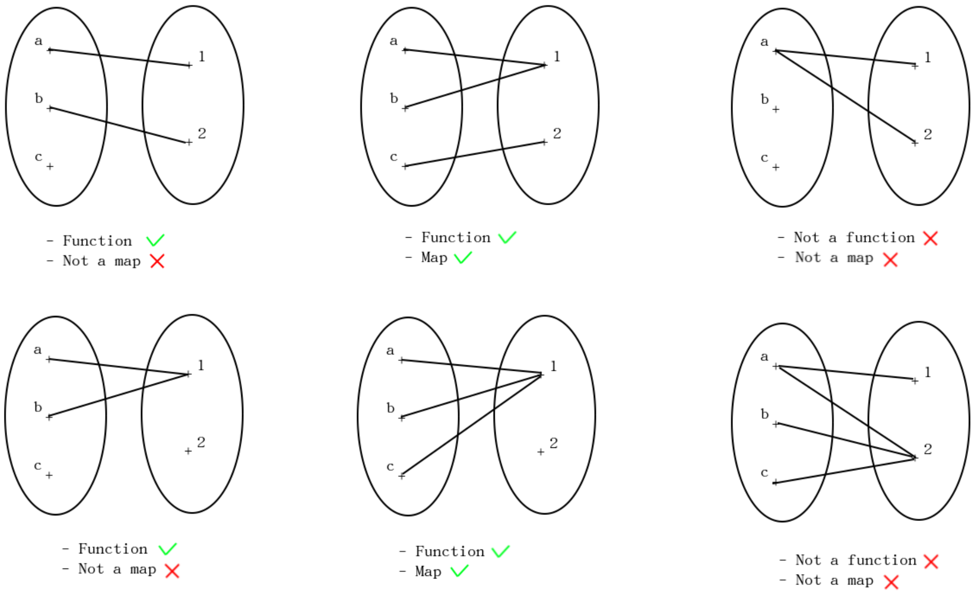


Figure 18: Examples of relations between sets that are either functions, or both functions and applications, or neither. Note the critical diagram in the top right corner: if there exists an argument that can have several images, what you have cannot be a function.

4.3.1 Image, preimage of a set by a function

If you see $f(x)$, where x is an element, then $f(x)$ is a single element of the output set. If, on the other hand, you see $f(E)$ with E a set, then $f(E)$ is also a set, called the **image set** (sometimes simply ”image”) of E by the function f. This set contains the elements of the output set for which there exists a corresponding argument by the function f (the set of ”possible” results of your f function for the elements of E, basically). There also exists a concept which corresponds to ”all the possible inputs for a given set of outputs”, called the **inverse image**, or **preimage**.

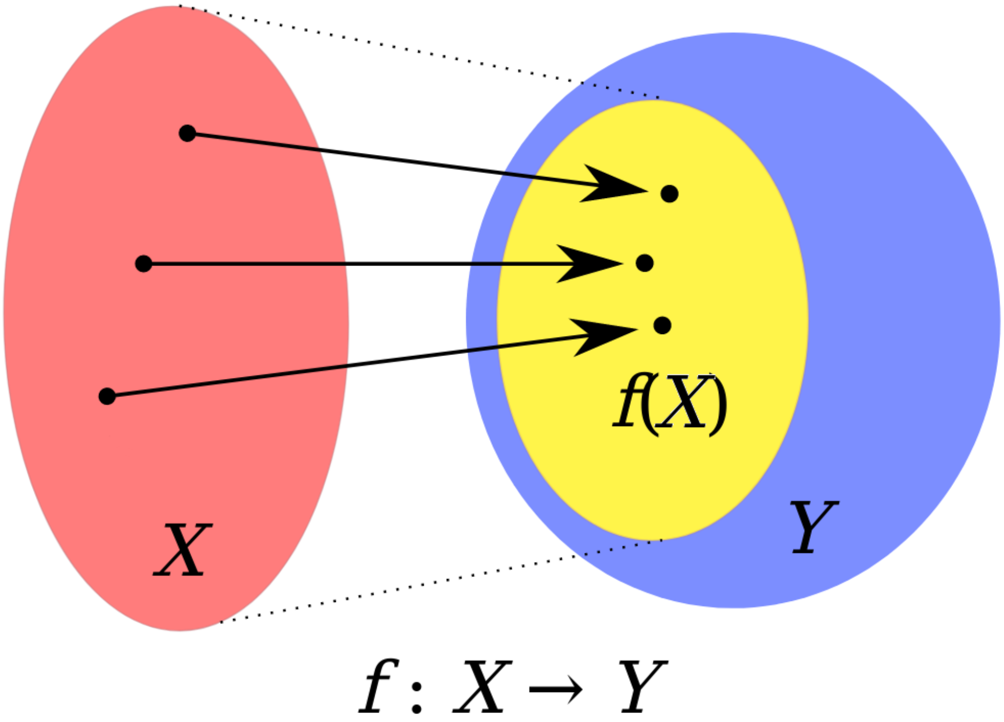


Figure 19: Visualization of the image $f(X)$ (included in a codomain Y) of a set X. The image is defined as $f(X) = \{y \in Y \mid \exists x \in X, f(x) = y\}$, or equivalently, $f(X) = \{f(a) \in Y \mid a \in X\}$.

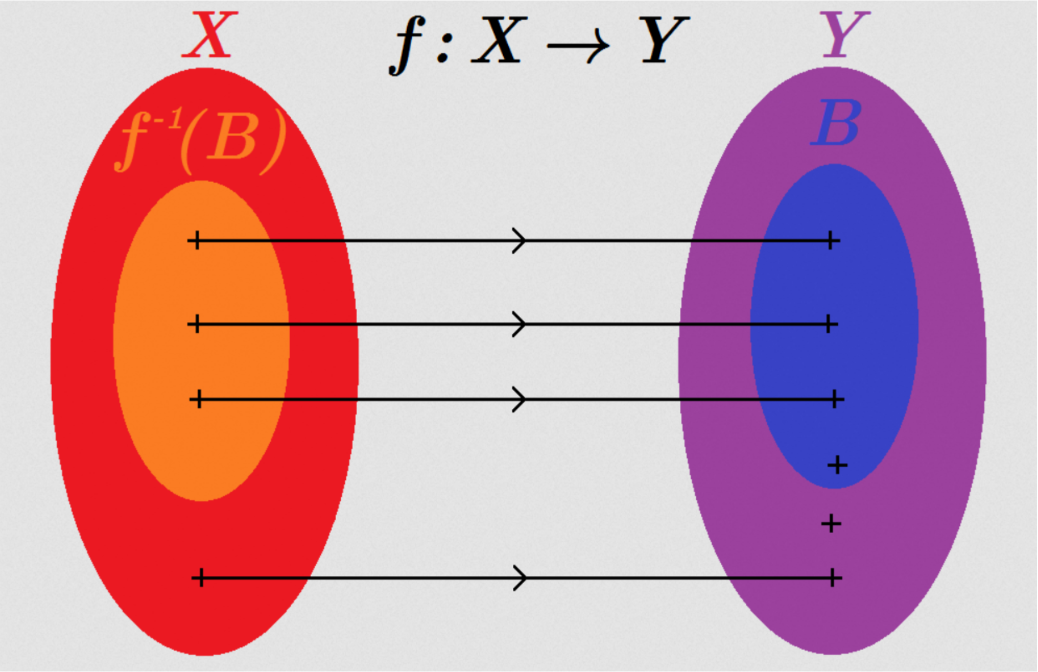


Figure 20: Visualization of the preimage $f^{-1}(B)$ (included in a domain X) of a set B (included in a codomain Y). The pre-image is defined by $f^{-1}(B) = \{x \in X \mid f(x) \in B\}$.

4.4 Function composition

Let f be a function from A to B and g be a function from B to C, then there is a function h (denoted $h = g \circ f$) from A to C such that $\forall a \in A, h(a) = g(f(a))$. Beware, I have simplified the definition a bit, technically, it is also necessary that g be well defined at least in any point of $f(A)$ in case $f(A)$ is a subset of B distinct from B as a whole. For you coders out there, this means it must be possible to give all returns from f as an argument to g without g returning an error. Remember that $f(A)$ is a SET. For the concrete definition, the definition set of h consists of the largest subset A' of A such that $f(A')$ is included in the definition set of g. In other words, $A' = f^{-1}(g^{-1}(C))$, the preimage of the preimage of the final codomain.

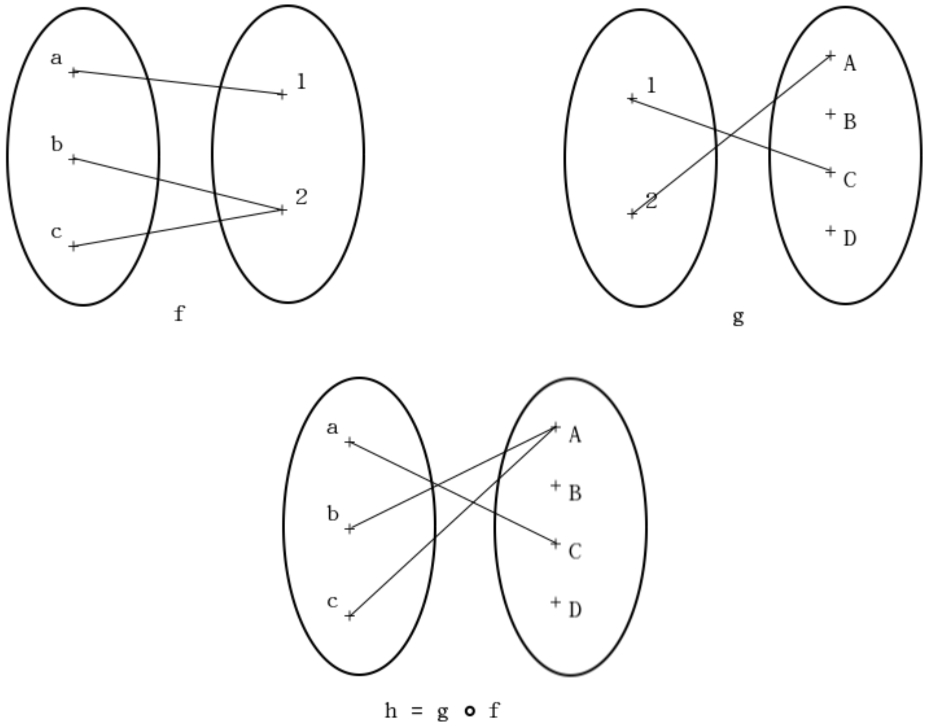


Figure 21: Visualization of a composition of functions $h = g \circ f$ on simple sets.

.

4.5 Set of functions

We write F^E, or $E \to F$, or more rarely $\mathcal{F}(E, F)$, the set of functions from E to F. E.g.: take the sequences of real numbers $(u_n)_{n\in\mathbb{N}}$; they are the functions from $\mathbb{N} \to \mathbb{R}$, one can thus write $\mathbb{R}^{\mathbb{N}}$ the set of real-valued sequences. NB: the notation F^E is explained by the fact that in the case where E and F are finite, $card(F^E) = card(F)^{card(E)}$.

4.5.1 Injectivity, surjectivity, bijectivity

Here are 3 fundamental notions about functions. A map f from A to B is said : - **injective** ssi

$$\forall (x, y) \in A^2, f(x) = f(y) => x = y$$

NB: this is the opposite implication as in the of the definition of a function, basically. **A function is injective if we have no image element that has several possible arguments. EVERY element in the destination set has AT MOST an arrow/link arriving to it**. There can still be elements in the destination set that have no argument (that can't be reached), of course. The notion of injectivity is useful to be able to find the input from the output, unambiguously. This is in particular the reason why we choose that "the square root of a number is always its positive root" since otherwise we would always have two possible arguments (one positive and one negative), except for 0. Intuitively, an injection goes from a "smaller" set into a "larger" set (see NB below, one must be wary of what "smaller/larger" means because of the different types of infinity as well as the dimensions of vector spaces, but it's still a good mnemonic to think of a medical "injection": the insertion of something small into a larger body).

- **surjective** ssi

$$\forall y \in B, \exists x \in A, f(x) = y$$

. NB: this means that **EVERY element of the target set has AT LEAST one arrow coming towards it**. In general, we can consider that an overhang is a function of a "larger" set in a "smaller" set (see NB below, think of the word "superimpose": everything is covered, and "sur-" and "super-" share the same latin root).

- **bijective** ssi

$$\forall y \in B, \exists ! x \in A \textbf{such as} f(x) = y$$

NB: f is bijective $\Leftrightarrow$ f is injective AND surjective. This means that **all the elements of the starting set have EXACTLY ONE link with the finish set, and vice versa**. For this reason, a bijective function can be considered as a "map" that can go both ways (since neither of the two sets ever has a double-branch), and so we can define a function called **reciprocal** noted f^{-1}.

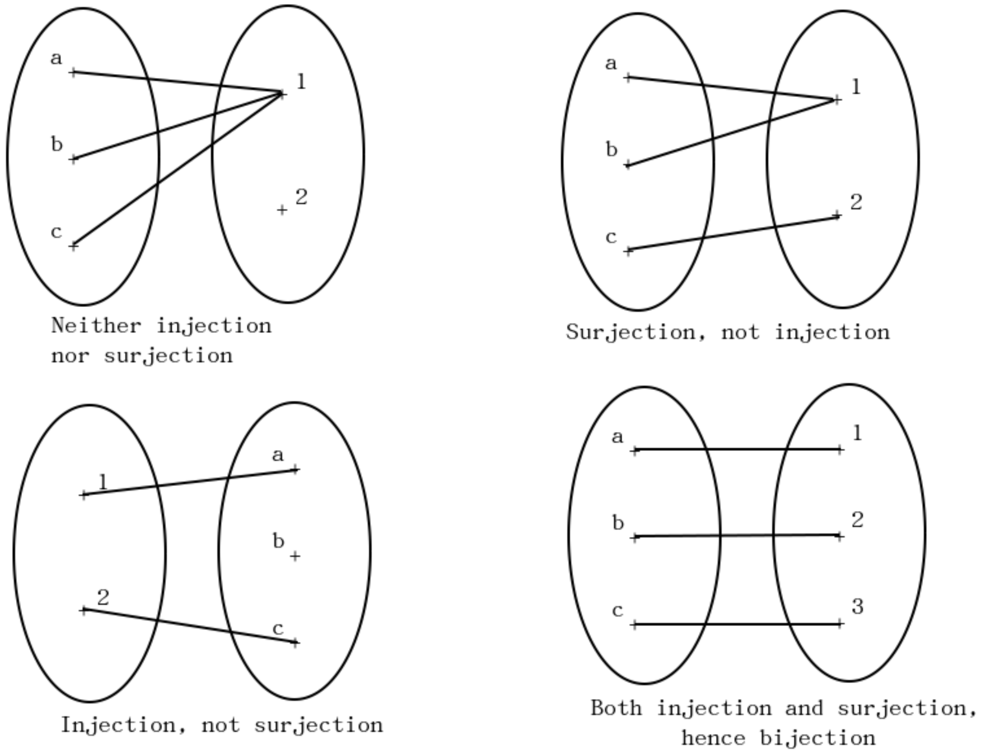

a
b
c
1
2
Neither injection
nor surjection
a
b
c
1
2
Surjection, not injection
1
2
a
b
c
Injection, not surjection
a
b
c
1
2
3
Both injection and surjection,
hence bijection

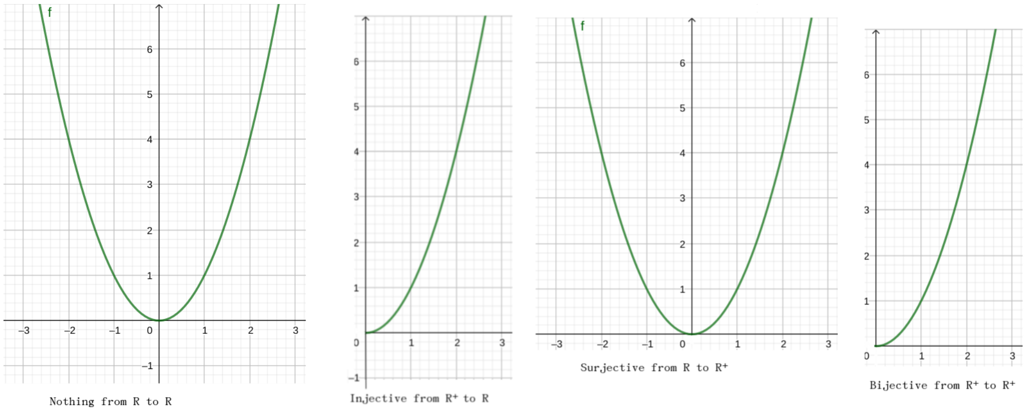
Figure 22: Examples of functions that are injective, surjective, bijective, or nothing.

NB: THESE NOTIONS ARE COMPLETELY DEPENDENT ON THE STARTING SET (DOMAIN) AND ARRIVAL SET (CODOMAIN). THE SAME FUNCTION MAY OR MAY NOT BE INJ/SUR/BIJ DEPENDING ON THE CONTEXT !. Very important example: by noting $\mathbb{R}^+ = [0, +\infty[$ and with $f = x \to x^2$, we see that f is :

- surjective of $\mathbb{R}$ in $\mathbb{R}^+$, but not injective

- injective of $\mathbb{R}^+$ in $\mathbb{R}$, but not surjective

- bijective of $\mathbb{R}^+$ in $\mathbb{R}^+$



Nothing from R to R

Injective from R^+ to R

Surjective from R to R^+

Bijective from R^+ to R^+

Figure 23: Example of the same function being injective, surjective, bijective, or nothing, depending on the context (its domain and codomain).

.

NB: "smaller set" and "larger set" take on a very particular meaning here which has to do with the notion of the cardinal of a set and with the different forms of infinity. We say that a (finite or infinite) set F is **discrete if there is an injection of this set into** $\mathbb{N}$, set of natural numbers (it is then "equal in size or smaller than natural integers", F in $\mathbb{N}$, small in large). In this case, if F is discrete, and moreover infinite, then this ratio is not only injective but also bijective "F is an infinity as large as $\mathbb{N}$". $\mathbb{Z}$, set of (relative) integers, and $\mathbb{Q}$, set of rational numbers (set of all fractions), are in this type of bijection with $\mathbb{N}$, so have a cardinal precisely as large as $\mathbb{N}$'s (as strange as it may seem, this is something that makes complete sense and is one of the major reasons why set theory was adopted: it has given us a better understanding of how infinity works, showing that there are different types of infinity and allowing their analysis). On the other hand, there exists no injection of $\mathbb{R}$ into $\mathbb{N}$, so $\mathbb{R}$ is a strictly larger infinity! **We call the infinity of $\mathbb{R}$ "continuous" infinity**. Basically, **the discrete is what we count, the continuous is what we measure**. If these digressions on infitinity interest you, look up Hilbert's first problem: the continuum hypothesis.

Now, to really blow your mind: there is a bijection between any interval of $\mathbb{R}$ and $\mathbb{R}$ as a whole (example, $]-\pi/2, \pi/2[$ in bijection with $\mathbb{R}$ by the tangent function ($tan(x)$))! This means that there is strictly "more infinity" in $[0, 1]$ than there is "infinity of natural integers"; and "as much infinity" in $[0, 1]$ as there is "infinity in $\mathbb{R}$". It's crazy, isn't it? Yet it is true. Or at least, following these principles of set theory, we've made math that is solidly applicable to physics (and other fields), which is both quite impressive and intriguing.

4.5.2 Family of sets

We call $\mathcal{F}$ a **family** if $\mathcal{F}$ is an indexed collection (of sets or elements) on a set I serving as an indexer. For example, a list of n real numbers can be written as $\{x_i \in \mathbb{R} \mid i \in I\}$ where $I = [[1; n]]$.

4.6 Operators

4.6.1 Binary operator

We call **binary operator** a function which takes two arguments as input, and returns a single return as output (nothing to do with computer binary, it's just the idea that your operator takes 2 inputs, a ternary operator would take 3, etc). Formally, we say that $\star : A \times B \to C$, i.e. $\star$ (star) is a function of the Cartesian product of A and B into a set C.

NB: The choice of the symbol $\star$ to denote operators is to have a symbol for an abstract operator, able to represent many others. Since it is a symbol that you don't know, it becomes easier to avoid errors related to your habits, reflexes and assumptions concerning mathematics. We will also use the $\perp$ symbol ('perp') for a second operator when we need to express the links between 2 different operators, like we will to describe addition and multiplication.

You already know a lot of binary operators! $+, -, \times, \div, \hat{}$, modulo $(\%), \cap, \cup, \&\&, ||, \Rightarrow, \Leftrightarrow, \leq$ ($\leq$ can also be an operator to build logical propositions, so a function of ($A^2 \to \{True, False\}$), we often see it in computer science), etc. But beware, the same symbol can refer to different operators, it all depends on the context ! Matrix multiplication does not work like regular multiplication as one could expect (matrix multiplication is an assortment of dot products) - neither does matrix addition, although it looks/feels a bit more like the addition we would "expect". To be perfectly rigorous, even subtraction in $\mathbb{N}$ is not the same as the subtraction in $\mathbb{Z}$, you'll see why below.

Now, why should you understand this as two inputs, one output? Basically because any element in $A \times B$, "ordered pairs", can be understood to work as two elements.

Examples:

– $12 + 5 = 17$, we can also write it as $+(12, 5)$ =17 or $add(12, 5) = 17$ or $f(12, 5) = 17$ to understand the link with computing: two arguments as input (12 and 5), which can be considered as a couple, with here an element of an argument in output (17). Here, $+ : N \times N \to N$

– In a $\mathbb{K}$-vector space E, the dot product of two vectors u and v, denoted $\langle u|v\rangle$, is an operator of $E \times E \to K$.

– Let be $x \in E$ and $f \in F^E$ [also written $(E \to F)$], the set of functions from E to F. Let $\circ$ be the evaluation operator of a function, such as :

$$\circ : \begin{array}{ccc} \big((E \to F) \times E\big) & \to & F \\ (f, x) & \to & f(x) \end{array}$$

that is, $\circ(f, x) = f \circ x = f(x)$ $\circ$ is also called the function composition operator.

– We take the binary alphabet $A_b = \{0, 1\}$ and the Latin alphabet $A_l = \{a, b, c, ..., y, z\}$. Let L_b be a language over A_b and L_l a language on A_l. A language is defined as a set of words over an alphabet. A word is defined as a string (a string of symbols) over an alphabet. For example L_l could be the set of words that contain only groups of 5 letters, so "aaaaaaaaaauuuuuccc" is a word on L_l. We note "+" the operator representing the concatenation of words ("0101" + "dog" = "0101dog"), which is non-commutative. Then $+ : L_b \times L_l \to L_a$, where $L_a = +(L_b \times L_l)$ is a language, image set of (which is computed from its arguments) over the alphabet $A_a = \{0, 1, a, b, c, \dots, y, z\} = A_b \cup A_l$.

Bonus exercise: How would you describe what $+(L_b \times L_l) = L_a$ (the image of the set $L_b \times L_l$ by the concatenation function denoted +) represents? Is this language L_a equal to the language L on the alphabet A_a such that $L = A_a^* =$ all possible words on the A_a alphabet, or is the image set L_a a proper subset of L (i.e. included in L but different from L)?

NB: the Kleene star A^* is a unary operator that transforms an alphabet into the set of all possible words on this alphabet. It plays a very important role in category theory when we look into the concept of monoid (see below): it is the "free functor" from Set, the category of sets to Mon, the category of monoids (this free functor is a tool to transform any set into a working monoid).

4.6.2 n-ary operator

An n-ary operators is an operator that repeats the same $n-1$ binary operation in a row on a set with n elements. For addition, the n-ary operator is usually written as Σ (”sigma”, for ”sum”); for multiplication Π (”pi”, for product). The part below the symbol defines the starting point of the index, the part above its stop condition; if nothing else is specified, the index values are considered to increase by 1 for each value. If there is no top expression, the bottom shouls describe a full index set on its own.

- sum of 2 to the power i for i ranging from 0 to 5:

$$\sum_{i=0}^{i=5} 2^i = 2^0 + 2^1 + 2^2 + 2^3 + 2^4 + 2^5 = 63$$

- product of the $2+i$ for i ranging from 0 to n (the exclamation mark is the ”factorial” operator representing successive multiplication of consecutive integers):

$$\prod_{i=0}^{i=n} 2 + i = (2+0) \times (2+1) \times (2+2) \times \ldots (2+n) = \frac{(n+2)!}{1!} = (n+2)!$$

- intersection of 3 sets, using an index set $I = \{1, 2, 3\}$:

$$A_1 = \{a, b, c\}, A_2 = \{a, c\}, A_3 = \{a, b\}, \bigcap_{i \in I} A_i = A_1 \cap A_2 \cap A_3 = \{a\}$$

You can have multiple indexes on n-ary operators. Often, when we have multiple indices, it is better to visualize the unwrapping of an n-ary operator as an object with as many dimensions as we have variables serving as indexes. So, for example for two indices, something rectangle-like:

$$\begin{aligned} \sum_{i=0,j=7}^{i=2,j=10} i * j = {} & 0*7 + 0*8 + 0*9 + 0*10 + \\ & 1*7 + 1*8 + 1*9 + 1*10 + \\ & 2*7 + 2*8 + 2*9 + 2*10 \end{aligned}$$

We can also have slightly more complicated indexing schemes. For example:

$$\sum_{i=0,\ j=0}^{i=4,\ 2j\leq i} a_{i,j}$$

can be read

$$\begin{aligned}
&a_{0,0}+\\
&a_{1,0}+\\
&a_{2,0}+a_{2,1}+\\
&a_{3,0}+a_{3,1}+\\
&a_{4,0}+a_{4,2}+a_{4,4}
\end{aligned}$$

NB: the ”fold” (also called ”reduce”) operation on a list/array type in computer science is generally equivalent to applying some n-ary operator.

NB: an n-ary operator with no input values is understood to return the identity for that operator. A Σ will return 0, a Π will return a 1 (this is coherent with: the idea that a number times 0 is also 0; the idea that a number to the power 0 is equal to 1; factorial 0 (0!) is equal to 1).

4.6.3 Closure of an operator

One last *very* important thing we'll need for the next part. We say that a binary operator ”**stable**” or ”**closed**” (or sometimes we call a the operator a ”binary operation”), if it is an operator of $E \times E \to E$. That is, an operator that always gives a result in the same set as its inputs. This notion of stability, or closure, is extremely important in mathematics. *It is a necessary condition for most of the properties of algebraic structures to work.*

We now arrive to what is probably the most interesting point of this course: algebraic structures.

5 Hierarchical construction of the fundamental algebraic structures

The idea here is to make a "cartography" of the spine, the tree trunk of the field of mathematics. We need an organized structure in order to sort the tools that mathematicians have developed over human history, as they needed make sense of this interrelated self-constructing system that is mathematics. The subject we'll talk about here is fairly hierarchical, and offers the keys necessary to explore the ecosystem of mathematics.

5.1 Algebraic structures

All the examples that given and not detailed in what follows are to be checked by yourself: this is how you will understand abstract notions, by comparing them to what you already know. Counterexamples are also very important. Feel free to look for extra examples on your own.

All the following cases are to be understood, those that should be most "understood by heart", those that you should be able to recognize as well as you recognize the difference between two parallel lines and two intersecting lines, are "**abelian group**" and "**field**", because they intervene in the definition of a vector space. Possibly "**monoid**" because it is simple, and for its role in language theory and category theory. Finally "**ring**" and "**algebra**" because they will be important for polynomials.

That being said, the rest of the vocabulary is still very, very useful to create a cartography of the "bestiary" of mathematics and to get an idea of the interactions between algebraic structures of different types. These interactions between structures are a crucially fundamental subject to understand math at a higher level (if it makes you curious, here is a good introductory article on the Yoneda Lemma http://www.math3ma.com/mathema/2017/8/30/the-yoneda-perspective). In addition to helping you tame your thought processes, and helping you create a machete to explore the mathematical jungle, the Yoneda perspective is useful especially for some protocols in high-level functional programming, including all questions of automatic proof (especially the logical infallibility of a code module, which is starting to become important in computer security) that are inspired by type theory.

An algebraic structure, or a "space" in the general sense of the term, is a set with fixed properties.

1. – Magma

   We call magma a pair $(E, \star)$ where E is a set and $\star$ is a closed binary operation over E.

   Ex:

   – $(\mathbb{N}, +)$ is a magma, because $+$ is stable in $\mathbb{N}$.

   – $(\mathbb{N}, -)$ is not a magma, because subtraction is not stable in $\mathbb{N}$ (e.g.: $7 \in \mathbb{N}$ and $12 \in \mathbb{N}$ but $7 - 12 = -5$ and $-5 \notin \mathbb{N}$).

   – $(\mathbb{Z}, -)$ is a magma

   – $(\mathbb{Z}, \div)$ is not a magma

   – $(\mathbb{N}, \times)$ is a magma

   – $(\mathbb{R}, \times)$ is a magma

   – $(\mathbb{R}^*, \div)$ is a magma, but not $(\mathbb{R}, \div)$ (because the operator must be an application, not just a function, and $a \div 0$ is indefinite for any real number a).

   NB: a magma is a kind of the basic block "without anything special" of the theory of algebraic structures (more frequently called "abstract algebra"). "Without anything special", except the stability of the operator, which "goes without saying" because the foundation of abstract algebra is to "build the mathematical language around the isolation of structures in order to better analyze them"; no wonder then that the basic block has a property like the stability of its operator.

2. – Monoid

   Let E be a set, $\star$ an operator. We say that $(E, \star)$ is a **monoid** if it has the following properties :

   – $(E, \star)$ is a magma

   – $\star$ is **associative**:

   $$\forall (x, y, z) \in E^3,\ (x \star y) \star z = x \star (y \star z)$$

   – existence of a **neutral element** (also called the **identity**, as in "identical") (noted e) for $\star$:

   $$\exists e \in E, \forall x \in E,\ x \star e = e \star x = x$$

Ex:

– $(\mathbb{N}, +)$ is a monoid (it is in fact a commutative monoid).

– $(\mathbb{N}^*, +)$ is not a monoid.

– $(\mathbb{Z}, -)$ is not a monoid (because subtraction is not associative)

– $(\mathbb{N}*, \times)$ is a monoid.

– Any (language-theoretic) language with closed concatenation and the empty word (=empty string) is a monoid.

NB : — in $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$, the neutral element for addition is 0, the neutral element for multiplication is 1.

— The null vector $(0, 0, 0, ..., 0)$ with n coordinates (=of dimension n) is the identity for the addition of $\mathbb{R}^n$.

— In function spaces, the identity for addition, if it exists, is the "identically null" function of this space (the constant $(x \to 0)$, the null matrix, etc). The identity for multiplication (if it is the point-by-point extension of basic multiplication) is the constant function $(x \to 1)$. The identity for the composition of functions ($\circ$ operator) is generally the function $Id = (x \to x)$.

— In matrix spaces, since matrices are functions (linear maps) and their product works like function composition, the identity function is the matrix I_n with 1s on the main diagonal and 0s everywhere else.

— The identity for string concatenation is the empty string ("", also noted ϵ).

— The identity for set union is the empty set, the identity for set intersection is the "maximum" set in which we place ourselves.

NB: Some neutral elements are neutral only on one side, for example 1 is neutral on the right for the power operator "$\hat{}$". ($a^1 = a$), but is not neutral to the left ($1^a = 1 \neq a$, in the general case where $a \neq 1$).

NB: If a neutral element exists, it is UNIQUE (exercise: demonstrate uniqueness; super classic example, assume that there are two different identities and show that they are necessarily equal).

NB: Note the order in the formula for the neutral element. Exercise: explain the difference if we alternate the "there is e" and the "for all x".

3. — Group

We say that $(E, \star)$ is a **group** if it has the following properties :

– $(E, \star)$ is a monoid

– existence of **symmetric elements** (or **inverse elements**) for $\star$:

$$\forall x \in E, \exists sym(x) \in E,\ x \star sym(x) = sym(x) \star x = e$$

We say that $(E, \star)$ is an **abelian group** (or **commutative group**) iff :

– $(E, \star)$ is a group

– $\star$ is **commutative**:

$$\forall (x, y) \in E^2,\ x \star y = y \star x$$

NB: an abelian group thus has 5 properties: stability (closure) of the operator, associativity of the operator, neutral/identity element for the operator, symmetry of the elements with respect to the operator, commutativity of the operator.

NB: for addition, we note $-x$ the symmetric of an element x and we call it "**opposite**"; for multiplication, we note $\frac{1}{x}$ or x^{-1} in general, and we call it "**inverse**"; for function composition, we note f^{-1} in general, and we call it "**reciprocal**". Generally, if we're talking about a single abstract operator like $\star$, we'll use notation similar to multiplication.

NB: the fact that subtraction and division are not associative explains why we define things this way: we use $+$ and $\times$ which work well, and we extend these operators' notations for to $-$ and $\div$ if symmetrical elements are present (*"to subtract is to add by the opposite, to divide is to multiply by the inverse"*).

NB: the $x \to sym(x)$ function is an involution: this means that if you apply it twice, you get back to where you started. Formally, $x \to sym(sym(x)) = Id = x \to x$. This means inverse elements always come in pairs (and rarely, are paired to themselves, like -1 for multiplication). Note that the identity is always its own inverse for the operator for which it is the identity.

Ex:

— $(\mathbb{N},+)$ is not a group

— $(\mathbb{Z},+)$ is an abelian group

— $(\mathbb{R}_+^*,\times)$, which we can also note $(]0,+\infty[,\times)$, is an abelian group. So is $(\mathbb{R}^*,\times)=(]\infty,0[\cup]0,+\infty[,\times)$.

— the set of symmetries of the square is a nonabelian group, called $D4$. https://www.cs.umb.edu/∼eb/d4/index.html

— $(bij(\mathbb{R}^{\mathbb{R}}),\circ)$ the set of bijective functions of $\mathbb{R}$ in $\mathbb{R}$, provided with the composition operator, is a non-abelian group.

— the circle of complex numbers of module (=radius) 1 is an abelian group for multiplication, called the unitary group, it is noted $U(1)$ and is isomorphic ("equal", the same, in the sense of algebraic structures) as an abelian group to the space of the rotations of a circle, called $SO(2)$. https://groupprops.subwiki.org/wiki/Circle_group

— Group theory is vast and fundamental to much modern mathematics. An example, quantum mechanics: groups of rotations and symmetries are used to greatly simplify the language of transformation possibilities in vector spaces, especially complex vector spaces and complex topological vector spaces, whose objects are for example continuous complex functions (functions over the complex plane that transform it continuously).

4. — Ring

Let $(E,\star,\perp)$ be a set with two stable binary operators (I don't know if it has a name, we could call it a "bimagma" for example; inventing concepts is certainly not forbidden in mathematics, so long as they have a clear and rigorous definition).

We say that $(E,\star,\perp)$ is a **ring** if it verifies the following properties:

– $(E,\star)$ is an abelian group

– $\perp$ is associative

– $\perp$ has its own neutral element, noted e'.

– $\perp$ is **distributed on both sides** on $\star$, i.e.:

$$\forall(x,y,z)\in E^3, x\perp(y\star z)=(x\perp y)\star(x\perp z)$$

.

$$\forall(x,y,z) \in E^3, (y \star z) \perp x = (y \perp x) \star (z \perp x)$$

.

We say that E is a **commutative ring** if $\perp$ is also commutative.

NB: in general, $\star$ corresponds to addition and $\perp$ to multiplication, or we can make parallels that come pretty close to that. For this reason, when there are two different types of inverse elements, for two different operators, we tend to use the notation $-x$ for $\star$ and $x^{-1} = \frac{1}{x}$ for $\perp$ and to note 0_E for the identity of $\star$ and 1_E for the identity of $\perp$. You'll also see some authors using $+$ and $\times$ by default instead of $\star$ and $\perp$, and you have to understand that it's not necessarily $+$ and $\times$ as you're used to them. So basically, a *ring is a structure with addition, subtraction and multiplication (not necessarily commutative) but not generalized division* (we ignore the Euclidean division which exists even in integers). We can have rings where CERTAIN inverses exist, but not for all elements, such as matrix or function rings.

NB: a **pseudo-ring** (or **rng**, pronounced "rung") is a ring without a multiplicative identity. The nomenclature of rings varies a lot depending on different authors' nationality or historical period. Some will say "ring" to mean "pseudo-ring" and on the other hand will specify "unit ring" for cases where there is a multiplicative identity. So be careful when you have doubts as to how a given author defines a specific concept.

NB: some operators are only distributive on one side, or distributive in a weird way (especially in sesquilinear algebra, in the vector space framework over $\mathbb{C}$, there is some mind-boggling and beautiful geometry in there, and the Fourier analysis that depends on it is the fundamental tool for high level signal processing).

NB: in some cases, distributivity can work in both directions, as for $\cap$ and $\cup$ (union and intersection) where $\cap$ is distributive over $\cup$ and $\cup$ is also distributive over $\cap$. In rings it is usually one-way, i.e. $\times$ over $+$, or $\perp$ over $\star$. Understanding two-way distributivity will be essential in logic, especially if you want to understand the composition and optimisation of logical electronic circuits.

NB: The definition of the ring can also be summarized as follows: $(E, \star)$ is an abelian group, $(E, \perp)$ is a monoid, and $\perp$ is distributive on $\star$.

Ex :

— $(\mathbb{Z}, +, \times)$ is a ring (this is the basic example of a commutative integral domain (and other classes of rings); see next section).

— $(\mathbb{R}[X], +, \times)$, simply noted $\mathbb{R}[X]$, is the ring of real-valued polynomials.

— $(\mathcal{M}_n(R), +, \times)$ set of square matrices of size $n \times n$ with real coefficients with matrix addition and multiplication is a non-commutative ring.

5. – Integral domain

We say that an element x of E is a **zero divisor** iff :

$$\begin{cases} x \neq 0 \\ \exists y \in E, y \neq 0, \text{ such that } x \perp y = 0_E \text{ or } y \perp x = 0_E \end{cases}$$

NB : watch out, as $\perp$ represents multiplication but 0_E is the identity for addition, noted $\star$!

Example : — Let be two functions f and g of the function space $E = \mathbb{R} \to \mathbb{R}$, f being null over $\mathbb{R}$ - but not on $\mathbb{R}_+$, and g null over $\mathbb{R}_+$ but not on $\mathbb{R}_-$ are divisors of 0 for the "pointwise" multiplication operator because $f \times g = g \times f = (x \to 0) = 0_E$, here the null function (neutral for the addition of functions).

— $A = [1.0; 0.0]$ and $B = [0.0; 0.1]$ two matrices of size 2×2, are zero divisors, because $AB = 0_{\mathcal{M}_2(\mathbb{R})}$, here the null matrix (neutral for matrix addition). More generally, any matrix with determinant 0 is a zero divisor.

NB: zero divisors are important because they correspond precisely to the elements that are not invertible for multiplication.

We say that $(E, \star, \perp)$ is an **integral domain** (also called an **entire ring**) iff :

– $(E, \star, \perp)$ is a ring

– $(E, \star, \perp)$ is not the null ring (ie, $(E, \star, \perp)$ has at least BOTH neutral elements, 0_E and 1_E)

– E has no zero divisors

NB: You will sometimes see people use just "*domain*", instead of "integral domain"... Which tends to be pretty confusing, since "domain" is in general reserved for the "starting set of a function". This is one of those cases where I find the English nomenclature quite poor; the French's "integral ring" is much better, in my opinion.

NB: this notion is useful to create systems where we can solve equations in which we have to deal with cases in zero, that is to say the basis of the vast majority of equations that occur in loooots of branches of mathematics. Why ? Generally, you can replace an equation between two elements $f = g$ with the equivalent formula $f - g = 0$, and study the cases where $h = f - g$ is null, which tends to be a simpler problem. It is also useful to be able to define how to divide rigorously, which we see immediately.

6. — Field

We often note E^* the E set deprived of its (multiplicatively) non-invertible elements (except in the context of vector spaces where the star is often reserved for the dual). If E is an integral domain, the only non-invertible element (by which one cannot divide) is the identity for addition (or $\star$), noted 0_E.

We say that a set $(E, \star, \perp)$ is a **field** iff:

– $(E, \star, \perp)$ is an integral domain

– $\perp$ is commutative

– $\forall x \in E^*$, x has a symmetrical element for $\perp$ noted x^{-1} called its inverse.

Ex:

— $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$, with the usual addition and multiplication, are fields. The letter chosen to refer to a field (usually $\mathbb{R}$ and $\mathbb{C}$) is $\mathbb{K}$ in many languages (for *Körper*, in German, since the letter 'C' is busy being the field of complex numbers) and generally $\mathbb{F}$ in English. Note that in English, you should take care not to confuse "(algebraic) field" as it's used in abstract algebra with "field" in physics, *magnetic field*, etc, which are generally "vector fields". Vector fields are vector spaces where each vector (each point in the space) is affixed with another vector (an arrow to see the direction of motion at that point in space, so to speak).

— $(\mathbb{R}(X), +, \times)$ set of rational fractions (strange nomenclature, but, well, it is fractions with polynomials in the numerator and denominator) is a field.

— $(bij(M_n(\mathbb{R})), +, \times)$ the set of square matrices of size $n \times n$ invertible is not a body because its multiplication is not commutative, and addition is not closed.

— The only fields with a finite cardinal are $\mathbb{Z}/p\mathbb{Z}$ (also noted $\mathbb{K}_p$, or $\mathbb{F}_p$; or GF_p for "Galois field") where p is a prime number. They are called "set of equivalence classes on $\mathbb{Z}$ modulo p". [Technically one should say "modulo the relation of 'congruence modulo p' " but that's a bit wordy.] Think of a clock with a prime number of hours and where there are only hours, not minutes. You are allowed to make your usual multiplications, additions, but you have to stay within the numbers on the clock.

Exercise: To understand the division in such a set, go back to the definition of symmetrical elements: define each inverse and multiply by the inverse. Clocks with a number n of hours where n is not prime contain divisors of zero.

Exercise: on your usual clock with 12 hours, if you go to bed at 10 o'clock in the evening and sleep for 9 hours, you get up at 7 o'clock in the morning. So $10 + 9 \equiv 7[12]$. Question: 12 is not prime, so $\mathbb{Z}/12\mathbb{Z}$ is not a field. But is it at least an integral domain? Why, or why not?

— The notation "$(\mathbb{Z}/n\mathbb{Z})^*$" is used to designate the multiplicative group for $\mathbb{Z}/n\mathbb{Z}$ where n is not prime. This means to keep only the elements of the clock that have an inverse. So sometimes the symbol "power star" is not just removing 0_E, as in $\mathbb{Q}$, $\mathbb{R}$ and $\mathbb{C}$. There are contexts where it is used to refer the (multiplicatively) invertible elements of a ring, and build its multiplicative subgroup from its multiplicatif monoid.

NB: Basically, a field is a structure on which we can use $+$, $-$, $\times$ and $/$, according to the usual rules (no division by 0).

NB: In summary, **a field $\mathbb{K}$ is a structure where $(\mathbb{K}, +)$ and $(\mathbb{K}^*, \times)$ are abelian groups and $\times$ is distributive over $+$.**

— Vector space

Reminder: by "$\mathbb{K}$", we generally mean $\mathbb{R}$, the real numbers, or $\mathbb{C}$, the complex numbers. However the definition also applies to $\mathbb{Q}$, to finite fields such as $\mathbb{Z}/p\mathbb{Z}$ (except $\mathbb{Z}/2\mathbb{Z}$ which is weird apparently, I'd wager it's because in it you have $-1 \equiv 1$), or other more complex examples. If you already have some experience with real-valued vectors (you should at this point!), like $\mathbb{R}^2$, try to imagine and visualize what that would give for vectors with values over a finite field, like $(\mathbb{Z}/5\mathbb{Z})^2$ for example.

Let $\mathbb{K}$ be a field, $(E, +)$ an abelian group. Let $K \times E$ have an operator in E named "scalar multiplication" (or "external operator", or "scalar operator", or "scaling product"; not to be confused with "scalar product"!) noted as you would for multiplication (i.e. with $\times$, a centered point "$\cdot$", or without notation, just by concatenating the letters). Clearly put, let be a function $\cdot : (\mathbb{K} \times E) \to E$.

We then say that $(E, +, \cdot)$, or E for short, is a **$\mathbb{K}$-vector space** iff it has the following properties :

– Pseudo-distributivity on vectors:

$$\forall \lambda \in \mathbb{K}, \forall (x, y) \in E^2, \ \ \lambda(x + y) = \lambda x + \lambda y = \lambda x + \lambda y$$

.

– Pseudo-distributivity on scalars:

$$\forall (\lambda, \mu) \in \mathbb{K}^2, \ \ \forall x \in E, (\lambda + \mu)x = \lambda x + \mu x =$$

.

– Multiplicative pseudo-associativity of the scalar multiplication:

$$\forall (\lambda, \mu) \in \mathbb{K}^2, \ \ \forall x \in E, (\lambda\mu)x = \lambda(\mu x)$$

.

– Compatibility of the multiplicative identity of the field with scalar multiplication:

$$\forall x \in E, \ \ 1_{\mathbb{K}} x = x$$

.

Here's also a version of the above that's a bit less legible, but has all the technical details:

– Pseudo-distributivity on the vectors:

$$\forall \lambda \in \mathbb{K}, \forall (x,y) \in E^2,\ \ \lambda \cdot_{\mathbb{K}\times E} (x +_E y) = (\lambda \cdot_{\mathbb{K}\times E} x) +_E (\lambda \cdot_{\mathbb{K}\times E} y)$$

– Pseudo-distributivity on scalars :

$$\forall (\lambda, \mu) \in \mathbb{K}^2, \forall x \in E,\ \ (\lambda +_E \mu) \cdot_{\mathbb{K}\times E} x = (\lambda \cdot_{\mathbb{K}\times E} x) +_E (\mu \cdot_{\mathbb{K}\times E} x)$$

– Multiplicative pseudo-associativity of the scalar multiplication :

$$\forall (\lambda, \mu) \in \mathbb{K}^2, \forall x \in E,\ \ (\lambda \cdot_{\mathbb{K}} \mu) \cdot_{\mathbb{K}\times E} x = \lambda \cdot_{\mathbb{K}\times E} (\mu \cdot_{\mathbb{K}\times E} x)$$

– Compatibility of the multiplicative identity of the field with the scalar multiplication :

$$\forall x \in E,\ \ 1_{\mathbb{K}} \cdot_{\mathbb{K}\times E} x = x$$

NB: the elements of $\mathbb{K}$ are called **scalars**, and are written with Greek letters, and the elements of E are called **vectors**, and are written with Latin letters (by convention).

NB: Do note confuse the "scalar multiplication" above, which scales a vector to a certain size, with the "scalar product" (which is more often called the "inner product" or "dot product"), and is an operation from $E \times E \to \mathbb{K}$ which encodes information about the lengths of, and the angle made by, two vectors into a single scalar. The term "vector product" (a rather pathological tool that exists only in $\mathbb{R}^3$ but which is widely used because of the choice of the mathematical language created by Gibbs rather than the one created by Clifford at the beginning of the 20th century) is called *cross product*. The "cross product" is a variation of a more important, and better constructed "exterior product" (also called "outer product"), which intervenes in the field of tensor algebra (the study of tensor spaces; which is the next step after linear algebra, itself the study of vector spaces).

NB: if E is a $\mathbb{K}$-vs, we generally note 0_E its additive identity, the famous "null vector". You can find more details in the section on monoids above.

NB: the scalar multiplication is commutative, but in practice, we note the scalars on the left (ex: $\frac{1}{3}v$).

NB: **There is NO MULTITIPLICATION BETWEEN VECTORS WHICH RETURNS A VECTOR IN A BASIC VECTOR SPACE. The subject of multiplication between vectors is addressed in the next section, with another algebraic structure, the notion of a "$\mathbb{K}$-algebra".**

Examples :

— $\mathbb{K}[X]$, the set of polynomials with values in $\mathbb{K}$ is a $\mathbb{K}$-vector space (it is also a $\mathbb{K}$-algebra, see below).

— $\mathbb{K}^n$ is a $\mathbb{K}$-vs (this is the fundamental example in finite dimension, because if we limit ourselves to the structure of vector space, i.e. without the multiplication of a $\mathbb{K}$-algebra, any vector space of dimension n is isomorphic to $\mathbb{K}^n$, so that one can for example say to oneself "Ah, if I limit myself to the addition of the matrices, without matrix multiplication, $\mathbb{R}^{m \times n}$ is the same thing as $\mathcal{M}_{m,n}(\mathbb{R})$! It'll behave in exactly the same way").

— $(F^E, +) = ((E \to F), +)$, an additive abelian group of functions can always be transformed into a $\mathbb{K}$-vector space (hence the idea that every function is a vector in a given context; note that for a different reason which I explain later, every vector is also a function). Some such spaces will be more interesting than others.

Exercise: if we consider the abelian group $(E, \times)$ of physical dimensions (E is generated by successive multiplications or divisions by elements of its generating subset, its "base", $B = \{m, kg, s, mol, cand, ^\circ K, A\}$ and the inversion operation, for example $m^3 * kg * s^{-2}$ is an element of E). Is E a vector space on $\mathbb{R}$, on $\mathbb{C}$? Why, or why not?

Exercise: Visualize $f = (x \to x^2)$ and $g = (x \to 3x)$, for example on Geogebra, or better, in your head. What would the function give (both visually and as an algebraic formula): $h_1 = f + g$? and $h_2 = 4f$? and $h_3 = 3h_1$? and $h_4 = 3f + 3g$? What conclusion do you draw from this?

The functions that preserve the structure of a vector space (the "morphisms" of vector spaces, cf. category theory below) are called **linear**

maps (or linear functions).

Let E and F be two $\mathbb{K}$-vs. We say that $f : E \in F$ is **linear** iff :

– $\forall (x, y) \in E^2, f(x +_E y) = f(x) +_F f(y)$

– $\forall \lambda \in \mathbb{K}, \forall x \in E, f(\lambda x) = \lambda f(x)$

This means that linear maps are commutative (in the sense of category theory diagrams) with addition and scalar multiplication. This means that if you add vectors in the domain first, then apply the function, or apply the function first, then add the results in the codomain: you'll get the same result in the end.

NB: We denote $\mathcal{L}(E, F)$ or $Hom_{Vec_{\mathbb{K}}}(E, F)$ the set of linear applications of $E \to F$.

NB: $\mathcal{L}(E, F)$, as a set of morphisms of vector spaces, is a vector space. Exercise: demonstrate that $\mathcal{L}(E, F)$ is a $\mathbb{K}$-vs.

NB: in category theory, we denote it as $Hom_{Vec_{\mathbb{K}}}(E, F)$ or just $Hom(E, F)$ if there is no ambiguity on the fact that E and F are vector spaces on the same field.

NB: we note $\mathcal{L}(E)$ rather than $\mathcal{L}(E, E)$ the set of endomorphisms over E.

NB: be careful, the notation for $\mathcal{L}(E, F)$ varies depending on the source. We often note $L(E, F)$ the set of "continuous linear maps", which are the morphisms of "topological vector spaces". But sometimes you will see the opposite notation: L for linear maps and $\mathcal{L}$ for continuous linear maps, so beware...

NB: we note $(\mathcal{GL}(E), \circ)$ or rather $\mathcal{GL}(E)$ the set of automorphisms (bijective morphisms) on E, provided with the composition operator, called "linear group". Exercise: demonstrate that it is indeed a group. For those who have explored a little bit the geometrical representation of vector spaces: this linear group corresponds to all the invertible matrices, i.e. all the valid base coordinate changes of our vector space.

— Algebra

Basically, an algebra (or $\mathbb{K}$-algebra, or "algebra over a field $\mathbb{K}$", not to be confused with the branch of mathematics called "algebra") is a *vector space to which we add a form of multiplication between vectors.* This multiplication is a third binary operator, also noted as multiplication (or sometimes as function composition), usually called "vector multiplication", sometimes "third operator". Here, I have chosen to reuse the $\perp$ operator we had for multiplication on rings. Here, it is used an internal operator (i.e. $\perp$: $E \times E \to E$ is a multiplication on vectors, and is also a stable multiplication, which returns a vector; I specify this because the Euclidean scalar product returns a scalar, element of $\mathbb{K}$, as output).

Formally, a set $(E, +, \cdot, \perp)$ is called a **$\mathbb{K}$-algebra** iff:

– $(E, +, \cdot)$ is a vector space on $\mathbb{K}$.

– $\perp$: $E^2 \to E$

– $\perp$ is bilinear (linear for each argument; compare this with distributivity):

– $\forall (x, y, z) \in E^3, \perp (x + y, z) = \perp (x, z) + \perp (y, z)$

– $\forall (x, y, z) \in E^3, \perp (x, y + z) = \perp (x, y) + \perp (x, z)$

– $\forall \lambda \in \mathbb{K}, \forall (x, y) \in E^2, f(\lambda x, y) = \lambda f(x, y)$

– $\forall \lambda \in \mathbb{K}, \forall (x, y) \in E^2, f(x, \lambda y) = \lambda f(x, y)$

=> if the vector multiplication operator has an identity, the algebra is said to be unitary

=> if the vector multiplication operator is associative, the algebra is said to be associative

=> if the vector multiplication operator is commutative, the algebra is said to be commutative

=> I'm not sure how to qualify algebras with (multiplicative) symmetric elements ("invertible algebra" doesn't seem to exist, but I'll use it), but I know they exist (for example, the spaces of rational (polynomial) fractions function like fields, so are multiplicatively invertible).

NB: Algebras look a lot like rings, but are richer. They are like a ring of vectors (possibly without associativity or identity for multiplication) over a field of scalars, roughly speaking. Polynomials are a very good example. Euclidean division and scalar multiplication can be done there. We discuss this in the next section.

NB: Often you will see "algebra" to mean "unitary associative algebra", because a unitary associative algebra is a ring (in the more common "non-rng" sense), and therefore is a typical example of algebra that we are going to use a lot - beware of this abuse of nomenclature, since it's not that rare.

NB: Lie algebras are an example of non-associative algebras (their third operator, called the commutator, respects instead a property called the Jacobi identity).

NB: If an algebra is associative, unitary, commutative, does not have zero divisors (divisors of the identity for vector addition, ie the null vector, for vector multiplication specific to the algebra), and has an inverse for all elements except the null vector, then it functions in a way like a "field of vectors over a field of scalars" (NB: an "algebraic" field of vectors, not a "vector field"). This is the case for the spaces of rational (polynomial) fractions.

NB: For A and B two $\mathbb{K}$-algebras, morphisms are the maps $f \in Hom(A, B)$. They verify linearity ($\forall (x, y) \in A^2, f(x +_A y) = f(x) +_B f(y)$ and $\forall \lambda \in \mathbb{K}, \forall x \in A, f(\lambda \cdot_A x) = \lambda \cdot_B f(x)$); but also the preservation of the third operator ($\forall (x, y) \in A^2, f(x \perp_A y) = f(x) \perp_B f(y)$). For morphisms of unitary algebras, we add the condition $f(1_A) = 1_B$, where 1_A (resp. 1_B) is the identity for the third operator of A (resp. B), although some consider this redundant since it is implied by the preservation of the third law, as long as a non-zero multiplicative identity exists in B (i.e. as soon as $B \neq (\{0_E\}, +, \cdot, \perp)$, the null ring/algebra). We often add conditions on morphisms when the structure becomes richer and more complex: always research the type of morphism for your structure!

NB: You can consider algebras in which there are multiple different possible multiplications. Hestenes' geometric algebras / Clifford's algebras have a fundamental product, called the geometric product, and this product is the composition of several sub-products that each encode different information. This is an example of what could possibly be called a "polymultiplicative algebra". Another example would be function spaces with both a "pointwise" multiplication product, a "convolution" product, or a "commutator" product.

Examples :

— $\mathbb{R}$ can be considered a unitary, associative, commutative algebra, with symmetric (invertible) elements, choosing the usual multiplication as both the scalar multiplication and the vector multiplication.

— $\mathbb{C}$ is a $\mathbb{R}$-unitary, associative, commutative, invertible algebra, with complex multiplication ($\times : \mathbb{C}^2 \to \mathbb{C}$, such that $\times(z_1, z_2) = z_1 z_2 = (x_1 + iy_1) \times (x_2 + iy_2) = (x_1 x_2 - y_1 y_2) + i(x_1 y_2 + x_2 y_1)$) as the vector multiplication. If one restricts its structure to that of a vector space (by ignoring multiplication), it is isomorphic to $\mathbb{R}^2$.

— $\mathbb{R}^n$ with the Schur product (*pointwise product*) is an associative, commutative, unitary (the vector $[1, 1, 1, ..., 1, 1]$ of size n), non-invertible algebra (it has zero divisors; find an example of a pair of zero divisors!).

— $\mathbb{R}^3$ with the *cross product* is an anticommutative algebra (ie, $\forall(a, b) \in (\mathbb{R}^3)^2, a \times b = -b \times a$), but it is not unitary, nor associative. The cross product can only be defined as an $(n-1)$-ary operator in a vector space of dimension n. The only case of such a binary (2-ary) operator is therefore the cross product of $\mathbb{R}^3$.

— $\mathbb{K}[X]$ with the multiplication $[P \times Q = P(x) \times Q(x)$ for all polynomials P and $Q]$ is an associative, unitary, commutative $\mathbb{R}$-algebra.

— $\mathcal{M}_n(\mathbb{R})$ with matrix multiplication is a unitary, associative $\mathbb{R}$-algebra.

— By algebra isomorphism $\mathcal{M}_n(\mathbb{R}) \cong \mathcal{L}(\mathbb{R}^n)$. This also means that $\mathcal{L}(\mathbb{R}^n)$ (with as vector multiplication, the function composition operator "round $\circ$" on linear maps) is also a unitary, associative $\mathbb{R}$-algebra (and thus a non-commutative ring of vectors over a scalar field).

7. — Substructures

One last useful definition before we can move on to rings/algebras (and thus polynomials). However, it will surely be clearer after having also read the next part.

It is said that F is a **sub-structure** of E if F and E belong to the same category $\mathcal{C}$ (that is, both E and F are the same type of algebraic structure) and F is included in E. One can simply prove the stability of the substructure for all the operators to justify that E and F are in the same category.

For example :

Let $(G, \star)$ be a group, we say that H is a **subgroup** of G iff :

- $H \subset G$

- $\forall x \in H, sym_G(x) \in H$ (ie, H is a closed/stable subset for the sym_G inversion automorphism)

- $\forall (x, y) \in H^2, x \star y \in H$ (ie, H is a closed/stable subset of G for $\star$)

Another example : F is a vectorial subspace (generally just called a "**subspace**") of E iff F is a vector space included in E and F is stable for addition and scaling in E, i.e. iff :

- F is included in E

- $0_E \in F$

- $\forall (x, y) \in F^2, x +_E y \in F$ (stability for vector addition)

- $\forall \mu \in \mathbb{K}, \forall x \in F, \mu x \in F$ (stabiity for scalar multiplication)

NB: If the three previous properties are verified, F is a $\mathbb{K}$-vs, all on its own. Its operators ($+_F$, $\cdot_F$) are just the restriction of those of E to the elements of F. It is often useful to show that F is a subspace of another known $\mathbb{K}$-vs to show that F is a $\mathbb{K}$-vs in-and-of-itself. This allows you to prove that F has the right properties, so that these can then be used.

NB: A subgroup is also group in-and-of-itself (i.e. it respects the same properties as all other groups). In fact, this is the case for any substructure, and is a key component of category theory.

NB: A group included in another is necessarily a subgroup. These notions extend to other structures; ex: $\mathbb{Q}$ is a subfield of $\mathbb{R}$, and $\mathbb{R}$ is a subfield of $\mathbb{C}$. Showing that $\mathbb{Q}$ is a subfield can be done by knowing that $\mathbb{R}$ is a field and showing that $\mathbb{Q}$ is closed for $+$, $\times$, the additive symmetry involution and the multiplicative symmetry involution (except 0), etc.

NB: We speak of a "proper" substructure when F is a substructure "strictly included" in E, i.e. if F is included in E and F is different from E. Remember that inclusion is a partial order!

NB: The inverse of a substructure is a structure extension. For example, $\mathbb{C}$ is a field extension of $\mathbb{R}$. A more interesting example is to take $\mathbb{Q}$, as well as the elements $\sqrt{2}$ and $\sqrt{3}$ from $\mathbb{R}$, and make all possible combinations of operations to make it a stable set again. This gives us

$\mathbb{Q}(\sqrt{2}, \sqrt{3}) = \{a + b\sqrt{2} + c\sqrt{3} + d\sqrt{6}6pt|5pt(a, b, c, d) \in \mathbb{Q}^4\}$, which is both a proper field extension of $\mathbb{Q}$ and a proper subfield of $\mathbb{R}$. These sorts of field extensions and subfields intervene in Galois theory.

Well, we've already done a pretty good job at this point. We have ignored some classical algebraic structures that are less frequent (semigroups: an associative magma, without the neutral element that would make it a monoid), or those with a bit of added complexity of their own (modules; which is what would happen if we took a ring A instead of a field $\mathbb{K}$ as a set of scalars, and tried to make a vector space with it: it's the potential non-commutativity of the ring that complicates things, because multiplying by a scalar to the right or to the left doesn't necessarily give the same result). Feel free to learn more about them in your own explorations!

6 Category theory

6.1 Introduction

The principal interest of category theory is that it is an excellent unifying theory of abstract algebra. It offers a general framework to talk about all types of algebraic structures with a common language for all of them, and explain clearly how these relate. It allows one to manipulate the algebraic structures in themselves, to completely ignore the elements of a set but rather to look at how its structural properties dictate the behavior of the elements. Basically, we call "category" the totality of all structures of a given type. Remember that one cannot have "a set of all sets", this is the problem that the notion of category addresses: it is "large enough" enough to be able to construct "categories of all sets of a certain type" but "small enough" enough not to fall into the contradictions of sets that contain themselves. Category theory thus allows us to be able to make operations between massive collections of sets (categories) that still make rigorous sense.

So basically, each element of a category is an algebraic structure, or can be interpreted as such a structure even if it is not the "usual" conception of the problem (e.g. each *logical proposition* can be understood as *the space of logical models in which this proposition can be demonstrated as true*). One can go further (more general or more precise) with categories – for example decide to analyze a single structure as a category – but in general the most interesting and frequently recurring categories are these *categories of algebraic structures* (also called "**concrete categories**"; a theorem says that every category is isomorphic to at least one concrete category). What this entails is that category theory goes beyond abstract algebra in what it can express; but it still contains it fully.

At its heart, category theory is a grand attempt to understand the notion of "function" in the purest, most abstract, and most general way possible. This might not seem like much, but given the ubiquity of the concept of function in mathematics, science, cognition... this makes category theory one of the most profound achievements of human knowledge.

Examples of concrete categories:
— Set, category of sets with set functions as morphisms
— Rel, category of sets with relations between sets as morphisms
— Fld, category of fields,
— Mon, category of monoids,

— Grp, category of groups,

— Ab, category of abelian groups,

— Rng, category of pseudo-rings (rings without a identity element for multiplication),

— Ring, category of rings,

— $Vect_{\mathbb{K}}$, or K-**Vect**, category of vector spaces over the field $\mathbb{K}$

— $Alg_{\mathbb{K}}$, or K-**Alg**, category of algebras over the field $\mathbb{K}$

— Top, category of topological vector spaces

— Mod_A, or A-**Mod**, category of right-modules over a ring or algebra A

— Many others.

Formally, it is said that $\mathcal{C}$ is a category if $\mathcal{C}$ consists of the following properties:

— $\mathcal{C}$ has a collection of objects A, B, C... noted $|\mathcal{C}|$.

— For any pair (A, B) of objects of $|C|$ there is another object of $\mathcal{C}$ noted $A \to B$, containing the morphisms (arrows, transformations, noted f, g, etc.) from A to B. If $f : A \to B$, then we note $dom(f) = A$ the start object of f, and $cod(f) = B$ its end object.

— For any pair of arrows (f, g) such that $f : A \to B$ and $g : B \to C$, there is a morphism $h : A \to C$ such that $h = g \circ f$, which consists in applying first f, then g.

— The composition operator thus generated is associative, i.e. for any triplet of morphisms (f, g, h) with compatible domains and codomains ($cod(f) = dom(g)$ and $cod(g) = dom(h)$, we have $h \circ (g \circ f) = (h \circ g) \circ f$.

— For any object A of $|\mathcal{C}|$, there is an identity morphism $1_A : A \to A$ such that for any f such as $cod(f) = A$, we have $1_A \circ f = f$ and for any g such as $dom(g) = A$, we have $g \circ 1_A = g$.

Such categories are often called "monoidal categories", since their arrows behave more or less like the elements of a monoid.

A category $\mathcal{C}$ can therefore be described as a collection of points (called objects, which are usually structures) and arrows (called [homo]morphisms, which correspond to transformations of the objects that maintain the structure/properties, i.e. stable transformations in the category). Just objects and arrows. Note that the Greek root "homo" means "similar", while "morphè" means "shape".

NB: the collection of objects and the collections of morphisms are not necessarily "small enough" to be sets. A category where all collections are "classic" sets is called "small". Categories in which the collection of objects is "larger" (or more self-referencing) than a set can be, but in which all the homsets (its morphism collections, the "$Hom(A, B) = (A \to B)$") are small enough to be "classical" sets, is said to be "locally small". For most of the cases treated here, we are in a context of locally small categories.

In a way, category theory is the study of algebraic structures and the functions that maintain this structure. "Function" is used here in a broad sense, they are not necessarily "set functions"; even if many of them are anyway, because of the importance of concrete categories – the categories whose objects are algebraic structures.

This "abstract function", generalized, in category theory (the "morphism") is defined as "a link between the objects of a category, and this link verifies three properties: stability, associativity and the existence of an identity morphism for the composition operation for each object".

Stability by composition is just the idea that if f and g are in the same category (like in the category of injective functions) and that $cod(f) = dom(g)$, then $g \circ f$ exists and is of the same type as f and g (thus injective).

The associativity of composition, you're starting know well by now :

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

Identity is that for any object A in our category, we have an identity function of A to A that does not change anything and is of the same type as other morphisms (e.g.: injective). Formally:

$$\forall A, B \in |\mathcal{C}|, \exists 1_A \in (A \to A), \exists 1_B \in (B \to B), \forall f : A \to B, f \circ 1_A = f = 1_B \circ f$$

If one of these points isn't verified, we are not in a category. On the other hand, any group of "functions" that verify these properties by being well defined for a given collection of objects can be considered as a possible choice of "morphisms" for this collection in order to make it a category. For example, over sets, injective functions, set functions, and set relations are all valid choices of morphisms, and give rise to different categories over sets.

Bonus exercise: explore these concepts by trying to give you an idea of how the Set category of sets with set functions differs from the Rel category of sets with set relations.

All this being of course very abstract, let's explain by taking the example of Grp.

Let $(G, \star)$ and $(H, \perp)$ be two groups. We call a "group morphism" a function $f : G \to H$ such that:

$$\forall (x, y) \in G^2, f(x \star y) = f(x) \perp f(y)$$

Intuitively, a group morphism simply transforms one operator into another while keeping the same link between the starting elements. Or more precisely, by making it possible to choose to apply first the operator then the function ($\star$ then f) or first the function then the operator (f then $\perp$) and still get the same result.

You surely know two group morphisms already: exp and ln. ln is a morphism of $(\mathbb{R}_+^*, \times)$ to $(\mathbb{R}, +)$ and conversely for exp, because :

$$\forall (x, y) \in \mathbb{R}_+^*, \ln(a \times b) = \ln(a) + \ln(b)$$

$$\forall (x, y) \in \mathbb{R}, \exp(a + b) = \exp(a) \times \exp(b)$$

A group morphism simply transforms one operator into another.

One of the craziest tricks of category theory is the fact that the set of morphisms between two structures of a category $\mathcal{C}$ is also generally an object of $\mathcal{C}$. In our example, the set of morphisms of groups from G to H, denoted $Hom(G, H)$ (for **hom**omorphism which means morphism), is itself a group.

Another example, $\mathcal{L}(E, F)$, the set of linear maps from E to F; two $\mathbb{K}$-vector spaces (linear maps are the "morphisms of vector spaces"): $\mathcal{L}(E, F)$ is also a vector space in-and-of-itself over the same field $\mathbb{K}$.

The commutative diagram is a fundamental tool in category theory. It is "commutative", because starting from A, one can decide to go to B by the function f and then to D by h, or to go to C by the function g and then to end in D by k, and one will necessarily have the same result in D once the starting point in A is set. In other words, the diagram commutes iff $h \circ f = k \circ g$.

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "g"] & B \arrow[d, "h"] \\
C \arrow[r, "k"] & D
\end{tikzcd}$$

Here, writing on the left the letters of the diagram and on the right the elements of our example on the morphisms of groups: we write $A := G^2, f := \star, B := G, h := f, D := H, g := f, C := H^2, k := \perp$.

$$\begin{tikzcd}
G \times G \arrow[r, "\star"] \arrow[d, "{(f,f)}"] & G \arrow[d, "f"] \\
H \times H \arrow[r, "\perp"] & H
\end{tikzcd}$$

The idea of ”maintaining structure” fundamentally refers to the commutativity of diagrams defining morphisms in a given category. The commutativity of categorical diagrams is the fact that one can choose which arrows one passes through, and one can be sure to arrive at the same result (the commutativity of a given diagram is something to be proved). Here is for example the definition of what a linear map is, using the language category theory (commutative diagrams). The morphisms in $Vec_{\mathbb{K}}$ (represented by the example of the morphisms of $Hom(E,F)$, where E and F are arbitrary $\mathbb{K}$-vector spaces) are defined as making the following diagrams commute:

$$\begin{tikzcd}
E \times E \arrow[r, "{(f,f)}"] \arrow[d, "+_E"] & F \times F \arrow[d, "+_F"] \\
E \arrow[r, "f"] & F
\end{tikzcd}$$

$$\begin{tikzcd}
K \times E \arrow[r, "{(id_{\mathbb{K}},f)}"] \arrow[d, "*_E"] & K \times F \arrow[d, "*_F"] \\
E \arrow[r, "f"] & F
\end{tikzcd}$$

where $+$ is vector addition and $*$ the scaling product in E and F, respectively, and $id_{\mathbb{K}}$ the function $id_{\mathbb{K}} = (x \to x)$ in $Hom(\mathbb{K}, \mathbb{K})$.

This conservation of structures can be very useful, for example, if we know that $(G, \star)$ is a group and that $\forall (x,y) \in G^2, f(x \star y) = f(x) \perp f(y)$ is true, and that $f(G) = H$, then we can deduce that $(H, \perp)$ is necessarily a group, because f is a group morphism applied to a group G, so applying f to G maintains the group structure, so $f(G) = H$ is a group (and its identity is $e_H = f(e_G)$, where e_G is the identity of G).

A counterexample: E a $\mathbb{K}$-vector space and $f = (x \to x^2)$. Since $f(E)$ is no longer a vector space (one loses the nice properties on the elements of a $\mathbb{K}$-vs / one loses the commutativity of the diagrams), $x \to x^2$ is not a morphism of vector spaces.

Three important notions about morphisms. Let E and F be two structures; two objects of the same category $\mathcal{C}$:

— a morphism from E to E is called an **endomorphism** ("endo" = internal)

— a morphism from E to F that is bijective is called a **isomorphism** ("iso" = identical, same)

— a morphism from E to E that is bijective is called a **automorphism** ("auto" = oneself)

— if there is an isomorphism between E and F, we say that E and F are isomorphic and we note $E \cong F$

NB: *THE NOTION OF ISOMORPHISM IS ONE OF THE MOST IMPORTANT NOTIONS OF CONTEMPORARY MATHEMATICS.* With this notion, and in particular the notion of "natural isomorphism", which is basically a structural bijection without even going through the elements, we can notice two structures that function in the same way, and demonstrate that they function in the same way within a given category $\mathcal{C}$. The isomorphism is the tool to demonstrate the synonymy of different structures - the tool to demonstrate that even if we may have used a different language, we are talking about the same underlying mathematical object. It allows mathematicians to notice, for example "AAAH, finite linear maps and matrices are THE SAME THING!!!"; thereby reducing the complexity of some object by expressing it as an equivalent, but more intuitive, isomorphic object.

Side note, Cat, the category of categories, is itself a category. Its morphisms are called "**functors**". They play an incredible role in mathematics. For example in proofs: they allow you to move from one category to another, to "transport" proofs from a domain where they are obvious to a domain where they are hard. They can allow us to create complex structures from simple structures (free functor) and to use these as a block of marble to sculpt more precise and useful structures in an abstract way, "from above" (eg: the construction of tensor algebra from the elements of the Cartesian product of two vector spaces, then using algebraic quotients to inject the desired properties into our space to create the exterior algebra and the symmetric algebra). Or, conversely, to return to a simpler structure from a complex structure (forgetful functor, like $|U|$, the set of vectors of a U vector space, taken in isolation and unrelated to each other, like a basic set without any property or operator; an object of Set).

6.2 Abstract algebra with the language of category theory

https://tex.stackexchange.com/questions/115783/how-to-draw-commutative-diagrams#115835 https://tikzcd.yichuanshen.de/

[TODO: - present all relevant previous formulae as commutative diagrams: associativity, commutativity, identity, composition, inversion, distributivity, linearity, bilinearity... - présenter rapidement les catégories correspondantes à toutes les structures algébriques vues plus haut (notamment leur morphismes, associateur, commutateur, multiplicateur, identité, etc)]

6.3 Some "pure" category theory

[TODO: epis, monos, functors, monads, diagram chasing, example categories; some typed lambda calculus ? monoidal, symmetric, etc categories. Forgetful functor, free functor. Initial object, terminal object, zero object. Image, kernel, cokernel, coimage Posets seen as categories]

[For the curious that want more category theory right now, read Emily Riehl's *Category Theory in Context*.]