A (Hopefully) Comprehensive Introduction to Elementary Set Theory

Nik3141

started 8/17/12020 HE

Contents

Chapter 1

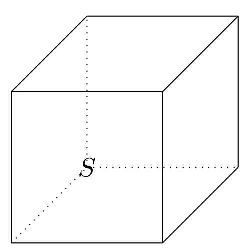
Basics

1.1 Sets, Elements, and Cardinality

Set theory is very abstract, which may initially make it difficult to grasp, but this abstraction makes it very applicable in a wide variety of situations. Set theory is considered to be the foundation of modern mathematics and is thus a very important topic.

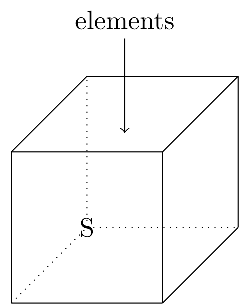
The basic notions of set theory are the ideas of a **set** - a mathematical object which can be thought of as an abstract 'containing object/region' - and an **element** - an object which is contained within the set. Both the set object itself and its elements are significant in set theory, as are the relationships between different sets and their elements.

It's often beneficial for the purposes of learning to build a mental model; many people envision sets as boxes, tabletops, waiter's trays, bubbles, or whatever container they fancy, and envision elements as the objects within those containers. For the purposes of this introduction we will use the visualization of a box for a set and will label it with the letter S:



Our set right now has nothing in it, which means that it is the **null set** or **empty set**, which is the special set which has 0 elements. We notate the empty set like this: $\emptyset$.

A box by itself isn't very useful - the whole point of a box is to put things in it. Similarly, if we have a set S, we will want to put things inside the set. As a reminder, the objects which we put inside the set will be called **elements**:

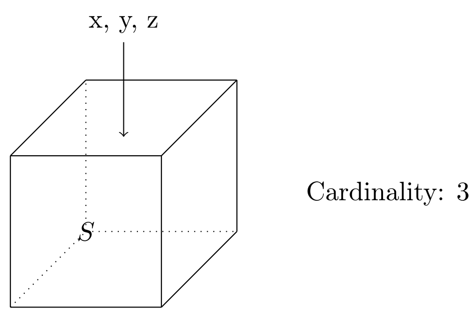


Sets as well as elements are very abstract notions, so the elements of a set can be really anything. I could put numbers, shapes, functions, or even people, cars, or marbles in a set. Let's build an example right now: imagine whatever containing shape you want, and then imagine putting a pencil, a pen, and a notepad inside it. That's a set!

A set S with elements x, y, z - I'm not saying what type of thing they are, I'm just saying three things go in a set and those three things are labeled x, y, and z - would be notated as $S = \{x, y, z\}$. If an object is a member of a set, we use the $\in$ symbol, like so: $x \in S$. If we want to indicate that multiple objects are inside the same set, we can use a comma-separated list: $x, y, z \in S$. Looking at our example from earlier, we could write that $S = \{\text{pencil, pen, notepad}\}$, or that: pencil, pen, notepad $\in S$.

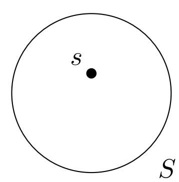
The order that elements/members appear in a set doesn't matter, and **sets cannot have any repeat elements**. This means that the sets $\{a, b\}$ and $\{b, a\}$ are equivalent, and a set $\{a, b, a\}$ would be invalid because it has repeat elements. Hence the sets {pencil, pen, notepad} and {pen, notepad, pencil} are equivalent, and the set {pen, notepad, pencil, pen} would be invalid because it can't have more than one of the same pen.

A useful notion to have is the 'size' of a set. We call this the **cardinality** of a set. If our set S has three elements, the cardinality of the set would be three:



We represent the cardinality of a set with vertical bars, so the cardinality of the set $S = \{x, y, z\}$ would be three, written as $|S| = 3$. We have a special term for any set with a cardinality of one (containing only one element): a **singleton**. Our set {pen, pencil, notepad} has a cardinality of 3.

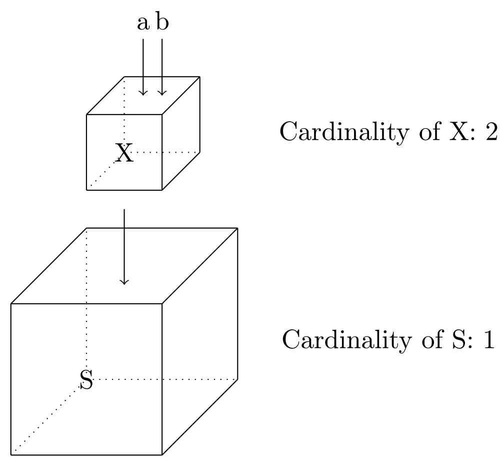
A set can have finite cardinality, as we've seen so far, but it can also have an infinite cardinality. For example, we could have a set which included all of the points within a given circle on the plane:



In this case, the set S contains all of the points within the circle, and the point s is one of the elements of S ($s \in S$). The set S would contain an infinite amount of points, and therefore have an infinite cardinality. It turns out that there are actually multiple 'sizes' of infinite cardinalities. We will do a much more in-depth study of infinite cardinality later in this document, but for now keep in mind that a set could have a finite or infinite number of elements.

1.2 Nested Sets

An interesting thing about sets is that like real boxes, we can store them inside of each other. These are **nested sets**. Let us say that our set S has a single element, X, and that X is itself a set with two elements, a and b. Our set S would be notated: $S = \{X\} = \{\{a, b\}\}$.



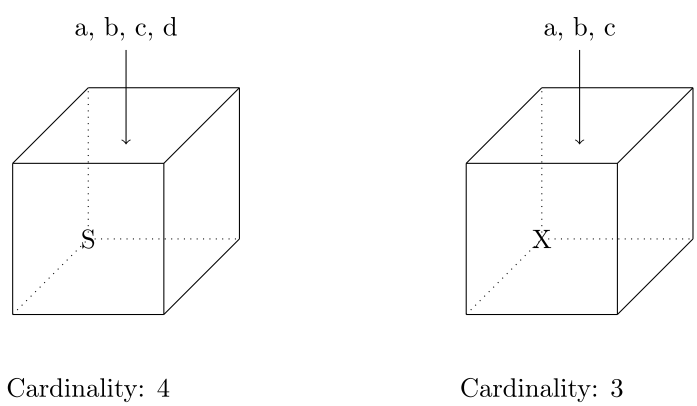
Notice that a nested set counts as only a single element, and only contributes one towards the cardinality. Note also that in this case, S is a singleton, because it has a cardinality of one.

As an example, if we have a set $X = \{\text{red, blue}\}$, we could nest it inside a set S like so: $S = \{\text{yellow, } X\} = \{\text{yellow, }\{\text{red, blue}\}\}$. X has a cardinality of two, and S also has a cardinality of two.

1.3 Subsets, Supersets, and Families of Sets

Say we have two sets, S and X. If all of X's elements are also in S, X is said to be a **subset** of S, or, equivalently, S is a **superset** of X.

Say $S = \{a, b, c, d\}$ and $X = \{a, b, c\}$:



X is composed *entirely of elements from S* (everything in X is also in S), so X is said to be a **subset** of S ($X \subseteq S$), and conversely S is said to be a **superset** of

X ($S \supseteq X$). It should be noted that if you unironically use the superset symbol ($\supseteq$) you will be banished to the 9th circle of hell.

Many mathematicians make a distinction between subsets in general and *proper* subsets. A set X simply being a subset of a set S ($X \subseteq S$) vs being a proper subset ($X \subset S$) is analogous to saying a number a is less than or equal to another number b ($a \leq b$) vs is strictly less than ($a < b$).

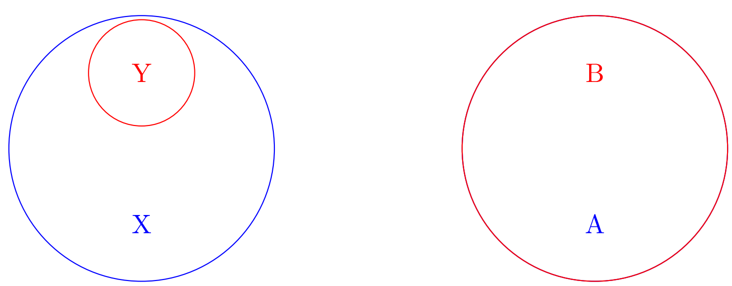
If S is a set containing some elements, and X is a subset of S, X must only contain elements from S, and X may or may not contain every element from S.

However, if S is a set and X is a *proper* subset of S, X should contain only elements from S, but it can't contain all of them. A proper subset should thus have a smaller cardinality than its superset.

If $S = \{a, b, c, d\}$ and $X = \{a, b, c\}$, X is a proper subset of S ($X \subset S$), and S is a proper superset of X ($S \supset X$).

If $S = \{a, b, c, d\}$ and $X = \{a, b, c, d\}$, X is not a proper subset of S ($X \not\subset S$), and S is not a proper superset of X ($S \not\supset X$); however, X is a subset of S ($X \subseteq S$), and S is a superset of X ($S \supseteq X$).

If we were to diagram sets as circles on the plane, where each point inside the circle represents an element inside the set, we could diagram the difference between general and proper subsets like so:



Note that this is distinct from the example in 1.1 where we had a set containing points inside a circle on a plane; in that example, the set literally contained points, while in this example, the points *represent* elements of a set, even if the elements themselves are not points.

In the left-hand figure we can see that Y is a proper subset of X, because everything in Y is also in X, yet Y does not contain everything in X. As a proper subset is just a subset with an extra restriction, Y is also a general subset of X.

In the right-hand figure we see two sets, A and B. B contains only elements from A, and in fact contains all of the elements from A. Thus B is not a proper subset of A, but is a general subset of A.

We can also observe something else from the right-hand figure: B is a subset of A, but A is also a subset of B

$$B \subseteq A$$

$$A \subseteq B$$

Notice that those statements above are extremely similar to these:

$$x \leq y$$

$$y \leq x$$

If x and y are real numbers, it's clear that the only way the two statements above could be true is if

$$x = y$$

Similarly, if A and B are subsets of each other, they must be equal.

$$A = B$$

This is how we show **set equality**: if two sets are subsets of each other (or supersets of each other), they are equal.

Another tidbit of information: the subset relation follows the transitive property ie given three sets A, B, and C, if I know that $A \subset B$ and $B \subset C$ then I know that $A \subset C$.

Finally, another important piece of vocabulary we have is the idea of a **family of sets**. Unlike subsets and supersets, which are part of a linear hierarchy, a family of sets is a group of sets which are adjacently related to each other in some way. For example, a group of sets could be a family because they all contain some element a. Or they might be related because they are all subsets of some parent set S; in this case they are said to be a family of sets **over** S. One example of this is the power set, which we will learn about in Chapter 2.

1.4 Commonly Used Sets

Some sets are so commonly used that they get special signs which all mathematicians should know. These are as follows:
$\mathbb{N}$ - Natural Numbers
$\{1, 2, 3, 4...\}$

$\mathbb{W}$ - Whole Numbers
$\{0, 1, 2, 3...\}$

$\mathbb{P}$ - Prime Numbers

$\{2, 3, 5, 7...\}$

$\mathbb{Z}$ - Integers
$\{... -2, -1, 0, 1, 2...\}$

$\mathbb{Q}$ - Rational Numbers
{Informally, any number which can be written as a fraction.}

$\mathbb{R}$ - Real Numbers
{Informally, any number on the real number line.}

$\mathbb{C}$ - Complex Numbers
{Informally, any number on the complex plane.}

Notice that all of these sets are sets of infinite cardinality.

1.5 Sequences

As we stated above, the order of elements in a set does not matter:

$$X = \{a, b\}$$
$$Y = \{b, a\}$$
$$\therefore X \subseteq Y,\ Y \subseteq X$$
$$\therefore X = Y$$

However, it is sometimes useful as a construct to have a set where the order matters. The solution is to create a new object called a **sequence**.

A sequence is very similar to a set, with a few modified properties: for one, a sequence admits repeat elements; secondly, the order of elements of a sequence matters, and each element in a sequence is *enumerated* or *indexed*, or corresponds to an element of the natural numbers. This means there is a first, a second, a third, etc element. The number correlated with an element is called the element's **index**.

Generally, the index of an element is represented with the variable n, while the element itself is represented as $f(n)$, similar to function notation. So for example, the first element in a sequence has an index of 1 and can be represented by $f(1)$. When referring to the elements of a sequence, we might say the 'nth element' or 'nth coordinate'.

While sets are notated using curly braces {}, sequences are notated using parenthesis (). The set containing a and b would be notated as $\{a, b\}$, while the sequence containing them would be (a, b). Note that $(a, b) \neq (b, a)$, while $\{a, b\} = \{b, a\}$. Additionally, while the set $\{a, b, a\}$ contains repeat elements and is thus invalid, the sequence (a, b, a) is perfectly valid.

Just like sets have the term 'cardinality' to discuss their size, sequences use the term 'length'. The sequence (a, b, a) has a length of 3, for example.

Remember that although so far I've mostly been using numbers, any type of objects can go into a sequence, just like how any type of object can go into a set. I could have a sequence with colored pencils, Lego pieces, or alphabet characters. I could even have nested sets or sequences inside of a sequence.

Additionally, just like sets can have subsets, a sequence can have a subsequence. A subsequence's elements must be in the same order as they are in the super-sequence, so to create a subsequence, just delete some of the items from the original sequence without shuffling them around.

Two important mathematical objects are the **ordered pair** and the **ordered triplet**. You've likely run across them in the context of graphing points in a 2D or 3D space before, but it turns out that both ordered pairs and ordered triplets are sequences of length two and three respectively. We will use these specific types of sequences much more in the following sections.

1.6 Combinations vs Permutations

Combinations and permutations both essentially deal with arranging and grouping elements from a set. Permutations involve making sequences with the elements from a set, while combinations involve making subsets of a set.

If I have a set S with cardinality n (containing n elements), the permutations of S are all of the different orders the n elements can be in. Essentially, we are going to create sequences with the elements from S (so they will have a length n - the same as the set S). There will be $n!$ (n factorial) permutations. For example, if $S = \{a, b, c\}$, the permutations are:
(a, b, c)
(a, c, b)
(b, a, c)
(b, c, a)
(c, a, b)
(c, b, a)

A good way to make this more concrete is to choose three objects in front of you and label them 'a', 'b', and 'c'. Then, see how many different arrangements of them you can make. You should get the six permutations which are above.

Now say that I have a set with three objects $S = \{a, b, c\}$, and I want to select them in pairs - I'm essentially making subsets of cardinality two from S:
$\{a, b\}$
$\{b, c\}$
$\{a, c\}$

Because the order of elements in sets doesn't matter, there are only three pairs of objects I can make from those three original objects. As before, I suggest trying it with three objects you have at hand.

Finally, if I select pairs of elements from the set $S = \{a, b, c\}$, but instead of putting them in subsets I put them in sequences, I'll end up with ordered pair permutations:
(a, b)
(b, c)
(a, c)
(b, a)
(c, b)
(c, a)

In this case, there are twice as many permutations as combinations, because the permutations are sequences (where order matters) whereas the combinations are subsets (where the order doesn't matter).

Chapter 2

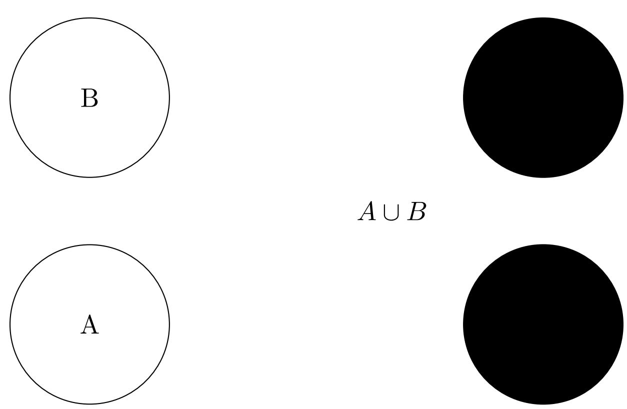
Set Operations

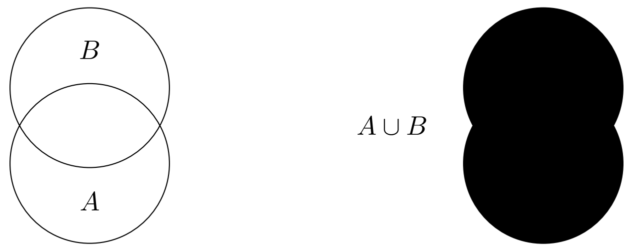
Just as there are operations on real numbers - addition, subtraction, multiplication, division, etc - there are operations which can be performed on sets.

2.1 Union

Union is a set combination operator which accepts two sets as operands and results in one new set, and is notated as $\cup$. If we have two sets A and B, the union of them ($A \cup B$) will result in a new set with all of the elements of A and all of the elements of B. So for example, if my set A contains a pencil, while my set B contains a pen, my set $A \cup B$ will contain the set with both a pencil and a pen.

What happens if the two sets have something in common? What if set A has {pencil, paper} and set B has {pen, paper}? Because $A \cup B$ is a set, there will be no repeat elements, so it will simply be {pencil, pen, paper}. There will only be one paper element in the resulting set. Essentially if an element is in both A and B it will only show up once in the union.

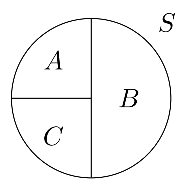




The left hand side shows our two operands, while the filled in part of the right hand side shows the result. The top diagram shows the case where A and B don't have any shared elements, while the bottom diagram shows the case where A and B do have shared elements.

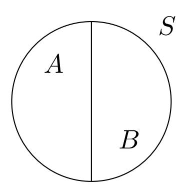
The set union operator is commutative, meaning that $A \cup B = B \cup A$. Additionally, we could chain it together, so an expression like this: $A \cup B \cup C$ would simply refer to the set of all elements in any of those sets.

If the union of two or more sets is equal to another set, which we'll call S, the sets can be said to be a family of sets which **cover** S. Consider the circle below to be a set S, with three proper subsets A, B, and C, symbolized by the sectors they're in:



A, B, and C would be said to cover S, because $A \cup B \cup C = S$. You can also see, in a purely spatial sense, that A, B, and C together cover the entire circle, which is S.

Another example, with just two sets:

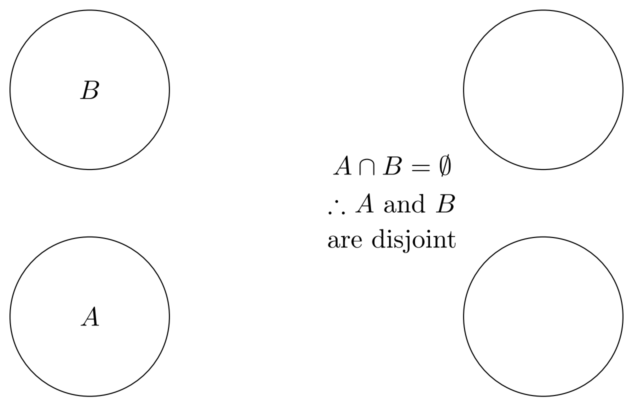


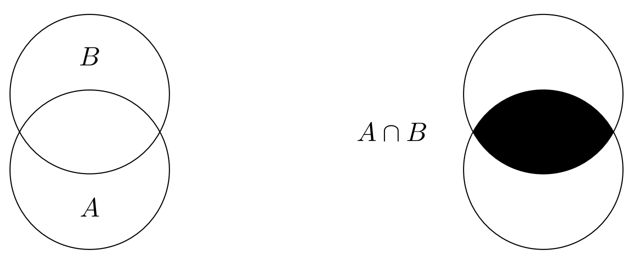
A and B cover S because $A \cup B = S$.

2.2 Intersection

Intersection is a set combination operator which accepts two sets as operands and results in one new set, and is notated as $\cap$. If we have two sets A and B, the intersection of them ($A \cap B$) will result in a new set with only the elements that A and B have in common. If I take my sets from before, where A has {pencil, paper} and B has {pen, paper}, the intersection of A and B ($A \cap B$) would simply contain {paper}, because that's what the two sets have in common.

In the case that the two sets have nothing in common, so if A had just {pencil} and B had just {pen}, the two sets' intersection is the empty set ($A \cap B = \{ = \emptyset$), and the two sets are said to be **disjoint**.





Please note that intersection and union are distributive over each other. This means that

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

and

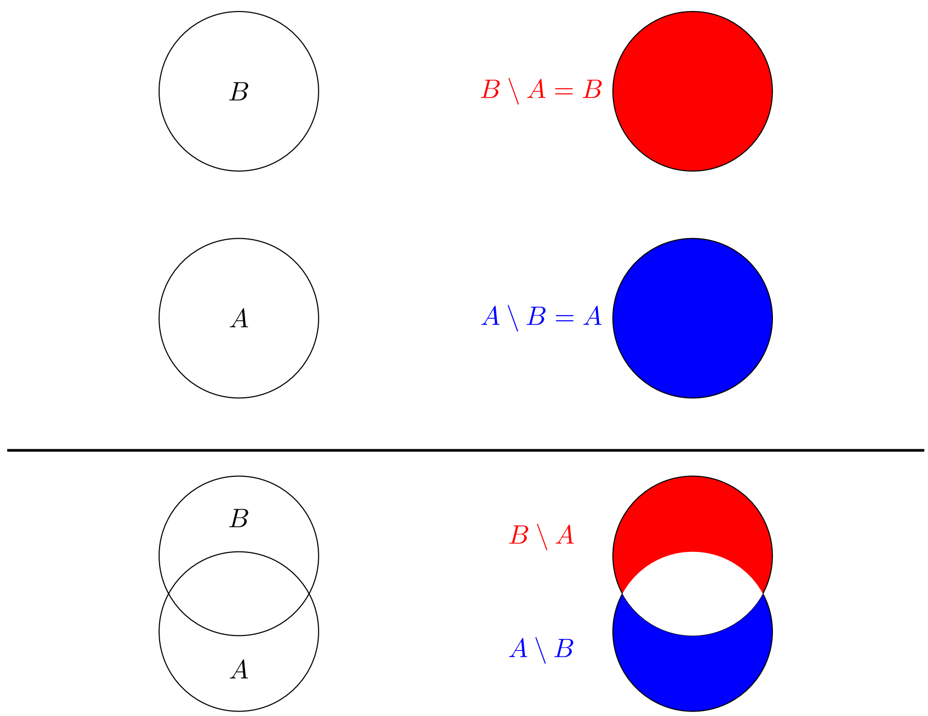
$$A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$$

2.3 Set Difference

Set Difference is a set combination operator which accepts two sets as operands and results in one new set, and is notated as $\setminus$. Unlike union and intersection, set difference is not commutative. If we have two sets A and B, $A \setminus B$ (pronounced "A without B") is equal to all elements of A which are not shared with B.

For example, say I have a set A with three objects: {pencil, pen, paper}, and I want to remove the objects which are shared with another set B with {pencil, paper}. $A \setminus B$ would be {pen}.

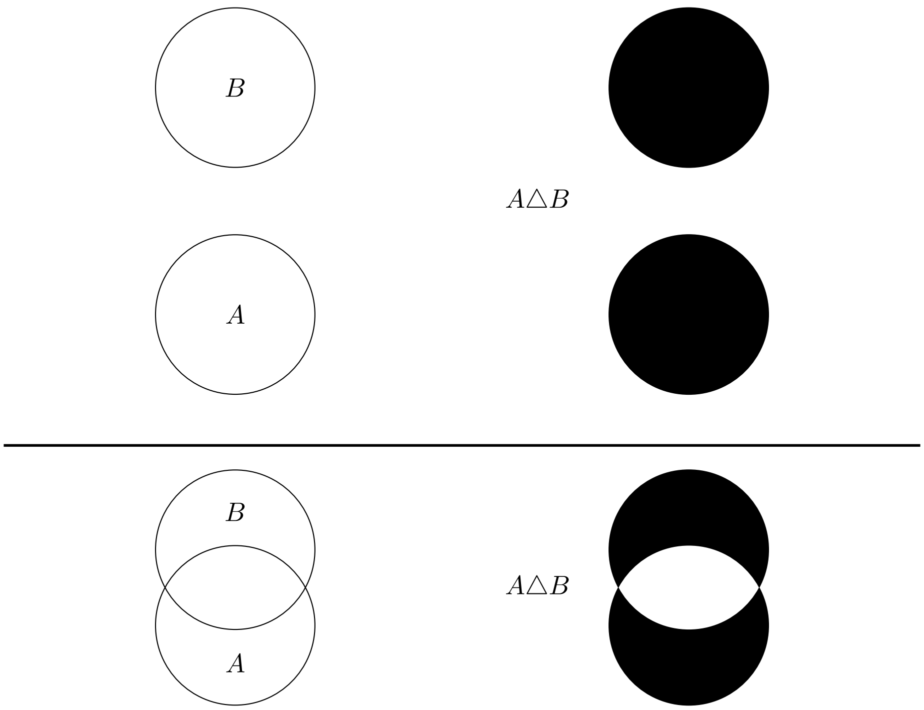
If A and B are disjoint ($A \cap B = \emptyset$), $A \setminus B = A$. This can be clearly seen via example: if a set A has {pencil, pen} and a set B has {paper}, removing the elements of B from A just results in A, because the two sets are disjoint/have nothing in common.



The red is $B \setminus A$ ("B without A") while the blue is $A \setminus B$ ("A without B").

2.4 Symmetric Difference

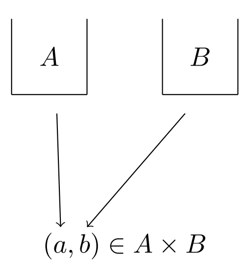
The symmetric difference is a set combination operator which accepts two sets as operands and results in one new set, and is notated as $\triangle$. The symmetric difference is commutative, and can be defined in two ways: first, as the union of the two set differences $(A \setminus B) \cup (B \setminus A)$; second, as the set difference of the union and the intersection of the two sets $(A \cup B) \setminus (A \cap B)$.



2.5 Cartesian Product

The Cartesian product of two sets is a non-commutative operator which creates a new set. Unlike the previous set operations, the Cartesian product does not result in a set with elements from the operand sets; the Cartesian product of sets A and B is $A \times B$, and results in a new set filled with ordered pairs. The first coordinate of each ordered pair will be an element from the first set, while the second coordinate will be an element from the second set.

Given that $a \in A$ and $b \in B$, a single element of the Cartesian product of two sets A and B ($A \times B$) would be constructed like this:



The entire Cartesian product would be the set of every possible ordered pair (a, b) where $a \in A$ and $b \in B$ (a is an element of A and b is an element of

B). Basically, every possible combination of elements chosen from the first and second sets, in that order, will be represented as an ordered pair in the new set.

This idea might be more clear with a concrete example - say we have two sets:

$$A = \{a_1, a_2\}$$
$$B = \{b_1, b_2\}$$

The Cartesian product $A \times B$ of these two sets would be:

$$\{(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)\}$$

Another example, with another two sets:

$$A = \{a_1, a_2, a_3\}$$
$$B = \{b_1, b_2, b_3, b_4\}$$

The Cartesian product $A \times B$ of these two sets would be:

$$\{(a_1, b_1), (a_1, b_2), (a_1, b_3), (a_1, b_4),$$
$$(a_2, b_1), (a_2, b_2), (a_2, b_3), (a_2, b_4),$$
$$(a_3, b_1), (a_3, b_2), (a_3, b_3), (a_3, b_4)\}$$

Please note that if I wanted to create every single ordered pair possible containing real numbers as coordinates, I could simply do $\mathbb{R} \times \mathbb{R}$. This set, when graphed on a plane, covers the entire plane, which is why the 2D plane is sometimes known as $\mathbb{R}^2$. Similarly, $\mathbb{R} \times \mathbb{R} \times \mathbb{R}$ contains every single point in 3D space, which is why the 3D space is sometimes written as $\mathbb{R}^3$.

2.6 Power Set

The power set of a set is a new set composed of subsets of the original set. If I have a set A, the power set (for reasons we will discuss later) of A is denoted as 2^A, and is composed of subsets of A. What subsets are these? These are *every single possible* subset which can be constructed from A, including the empty set and the entirety of A.

To give an example, if

$$A = \{a, b\}$$

then

$$2^A = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$$

where $\emptyset$ is equal to the empty set $\{\}$.

Another example - if

$$A = \{a, b, c\}$$

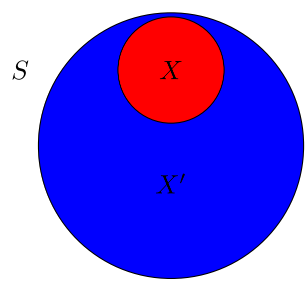
then

$$2^A = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$$

Note that the power set is a family of sets as every set within it is a subset of the same set, and thus every set within it is related.

2.7 Complement

The complement of a set informally refers to everything 'not inside' a set ie everything 'outside' of a set. For example, if I say that a set $\mathbb{Q}$ is equal to the rational numbers, the complement of $\mathbb{Q}$ (notated as $\mathbb{Q}'$), would be the irrational numbers. Of course, the notion of things being 'outside' isn't very useful unless we know the boundaries of 'outside', so there is often an implicitly assumed containing set which contains the 'outside'. What this means in practice is that if I have a set S and a subset X, the formal definition of the complement of X would be all of the elements of S which are not in X. In a way, it is like the inverse subset.



In this case, the red is X, the blue is X', and the whole circle is $S = X \cup X'$.

Going back to our earlier example of $\mathbb{Q}$ being the rational numbers, it should be clear that $\mathbb{Q} \cup \mathbb{Q}' = R$ where R is the set of all real numbers; in other words, the rational and irrational numbers together make up the real numbers.

When discussing sets, it is common to speak of the "universal set". The **universal set** is the set containing all of the objects currently under consideration; in the case above, it would be analogous to S. In a way, the universal set is like the opposite of the empty set: the universal set contains every object (that we are dealing with at the moment) while the empty set contains no objects.

Chapter 3

Binary Relations and Functions

3.1 Binary Relations

A binary relation is a subset of the Cartesian product of two sets which essentially allows us to relate two elements of different sets together by having them in the same ordered pair. I'll explain more in a minute, but first let me give some of the objects under consideration names.

If I have two sets, X and Y, a binary relation S between X and Y (this binary relation might be said to be **over** X and Y) would be a subset of the Cartesian product $X \times Y$. An element of the binary relation (S) would look like this:

$$(x, y) \in S$$

where $x \in X$ and $y \in Y$. So how does this let us define a relationship? Let me give an example - say $X = \{1, 2, 3\}$ and $Y = \{a, b, c\}$. The Cartesian product $A \times B$ would be:

$$\{(1, a), (1, b), (1, c),$$
$$(2, a), (2, b), (2, c),$$
$$(3, a), (3, b), (3, c)\}$$

This is every combination of elements from A and B, but it doesn't exhibit any particularly useful patterns or relationships between the elements of A and B. However, let's say we define a binary relation S over A and B:

$$S = \{(1, a), (2, b), (3, c)\}$$

It's clear that $S \subseteq (A \times B)$, and that S tells us some useful information; namely, that 1 and a are related, 2 and b are related, and 3 and c are related. With some more context and larger sets we could construct the English alphabet. It turns out that every binary relation has an 'opposite' relation, called the **converse** or **transpose**, and this case is no different. If we have our binary relation S, the transpose of S is notated as S^C, S^T, S^{-1}, $S^{\sim}$, or about three other options. I prefer using S^{-1} so I will be using that throughout this text. The transpose of S, S^{-1}, is effectively S but where all of the ordered pair coordinates have swapped places.

$$S^{-1} = \{(a,1),(b,2),(3,c)\}$$

Let's think of another example - say that I have two sets:

$$S_{people} = \{Jack, John, Jake, James\}$$

$$S_{cars} = \{Ford, Honda, Kia, Saturn\}$$

The cross product $S_{people} \times S_{cars}$ represents every single possibility of who owns which car. However, a binary relation R can conclusively tell me who owns which car:

$$R = \{(Jack, Kia),(John, Saturn),(Jake, Ford),(James, Honda)\}$$

and for the sake of example I'll also show the transpose of R:

$$R^{-1} = \{(Kia, Jack),(Saturn, John),(Ford, Jake),(Honda, James)\}$$

A binary relation can either be from a set to itself or from a set to another set. A binary relation from X to itself (sometimes said to be a "binary relation over X") is a **homogeneous** binary relation. A binary relation from a set X to a set Y where $X \neq Y$ is called a **heterogeneous** binary relation.

3.2 Relation Properties

Relations have some mathematically useful properties which we can identify: reflexivity, irreflexivity, symmetricity, antisymmetricity, and transitivity. In all of the following three definitions, the binary relation R will be symbolized as $\sim$, so the statement $a \sim b$ would mean that "a is related to b under the relation R". The statement $a \nsim b$ would mean that "a is not related to b under the relation R". Also note that I've used the homogeneous case for all of these for the sake of simplicity, but you could (probably - in the middle of fact-checking this right now) also have heterogeneous binary relations that are reflexive, symmetric, and/or transitive.

A binary relation R over a set A is **reflexive** when every object $x \in A$ is related to itself as well as other objects; in other words, $x \sim x$; in even different words, $(x,x) \in R$. This must hold for every x in A. As an example, the relation "as old as" is reflexive: every single thing has the same age as itself. If I had three people in a set

$$A = \{Molly, Mack, Michael\}$$

and the homogeneous relation from A to itself was

$$R = \{(Molly, Molly), (Mack, Mack), (Michael, Michael)\}$$

R would clearly be reflexive, because every person is the same age as themself. Since their are no other elements of R like $(Molly, Mack)$ or something similar, we know that none of the people are the same age as each other.

Any binary relation R over a set A which is essentially the opposite of reflexive - that is, for every object $x \in A$, $x \not\sim x$ - is called **irreflexive**. Note that irreflexive is not the same as "not reflexive": if a relation has some elements related to themselves and others not related to themselves, the relation is neither reflexive nor irreflexive.

A binary relation R on a set A is **symmetric** when if an element $x \in A$ is related to an element $y \in A$, the element y is also related to the element x, for any x or y in A. For example, if my binary relation says that two objects are related if they are perpendicular, it is clearly symmetric: if line u is perpendicular to live v, clearly line v has to also be perpendicular to line u. This doesn't always have to be the case, however - for example, the 'less than' relation is clearly not symmetric: if I know that $a < b$, I clearly know that $b \not< a$.

An **antisymmetric** binary relation R over a set A is one where if two elements $x, y \in A$ are related: $x \sim y$, the reverse ordering *cannot* be related: $y \not\sim x$. If some x, y pairs are symmetric and others are not, the relation is neither symmetric nor antisymmetric, similar to the case with reflexive/irreflexive above.

A binary relation R over a set A is **transitive** if and only if for any $a, b, c \in A$, $a \sim b$ and $b \sim c$ implies that $a \sim c$. An example of this would be a relation such as 'greater than'. If $a > b$ and $b > c$, it's clear that $a > c$, and thus 'greater than' is a transitive binary relation.

3.3 Equivalence Relations

An equivalence relation is is a special type of binary relation which is reflexive, symmetric, and transitive.

- Reflexivity: $a = a$
- Symmetricity: $a = b \implies b = a$
- Transitivity: $a = b$ and $b = c \implies a = c$

An equivalence relation gives us a strict binary choice where two objects which are compared can either be equal or unequal under the equivalence relation. Some examples of equivalence relations are: real number equivalency, triangle congruency, color, age, and any relation which satisfies those three requirements from above.

3.4 Functions

A **function** is a special type of binary relation over two sets A and B where *every* element in the first set is related to *exactly one* element of the second set. In other words, a function f between two sets A and B (called the **domain** and **codomain** respectively) can be represented by a subset F of $A \times B$ composed of elements (x, y) where x is in A and y is in B.

$$F \subseteq (A \times B)$$

we can use function notation to show the function, the domain, and the codomain:

$$f : A \to B$$

Since x is in A and is **mapped** to y in B, $y = f(x)$. When we say 'mapped', we are referring to the process of relating an element in the domain to an element in the codomain.

$$x \in A$$

$$f(x) \in B$$

If an element (x, y) is in F, there can be no other element (x, z) in F unless $y = z$. This is the abstract way of representing the 'vertical line test' we all learn in middle school. Additionally, every x in A must have a corresponding $f(x)$ in B; $f(x)$ is called the **image of** x. The set of all $f(x)$ values is a subset of the codomain and is called the **image of** f.

You may have heard of the term "range" being used before to refer to the set of all $f(x)$ values. Modern authors tend to shy away from this term, because it can refer to the codomain or the image (which might or might not be equal), and is thus a bit ambiguous.

The graph of a function is the set F, and if both the domain and codomain are sets of real numbers, it can be represented using Cartesian coordinates on a coordinate axis.

3.5 Function Categories

There are three important terms which can be used to categorize functions: injective, surjective, and bijective.

An injective (also known as "one-to-one") function (an injection) is one where no element of the codomain is mapped to by more that one element of the domain. Thus, if F is my function and (x, y) is an element of F, (z, y) *cannot* be an element of F.

A surjective (also known as "onto") function (a surjection) is one where every element from the codomain is mapped to by at least one element of the domain. In this case, the codomain and the image of the function are the same.

A bijective (also known as "one-to-one and onto") function (bijection) is one which is both injective and surjective; every element of the codomain is mapped to by exactly one element of the domain.

With these three terms, four possible categories of functions can be defined:

- injective but not surjective
- surjective but not injective
- bijective (surjective and injective)
- neither injective nor surjective

A note on bijections: two sets have an equal cardinality if there exists a bijection between them. This should make intuitive sense, because a bijection essentially creates pairs of elements in the two sets so that no element is in two pairs and every element in one set is mapped to exactly one element the other set.

3.6 Function Inverses

Given a function f, the inverse of f, f^{-1}, maps every element in the codomain of f to an element in the domain of f. Effectively, the codomain of f is the domain of f^{-1}, and vice versa: the domain of f is the codomain of f^{-1}.

For a function to have an inverse, it must be injective; otherwise, an element y of the codomain of f could be mapped by f^{-1} to multiple different values x in the domain of f, which would disqualify f^{-1} from being a function, as it maps a single element of its domain to multiple elements of its codomain.

Chapter 4

Partitioning a Set

4.1 Partitions

A **partition of a set** is basically a way to group the elements of a set into many disjoint subsets. If we have a set S, a partition is going to be a set containing nested sets which are our aforementioned disjoint subsets. There are three conditions for a valid partition of S:

- No subset can be the empty set.
- The subsets must cover S.
- All of the subsets must be disjoint from each other.

The simplest way to create a partition is to place each element of S in its own subset; if

$$S = \{x, y, z\}$$

the subsets in the partition of S would be

$$\{x\}$$

$$\{y\}$$

$$\{z\}$$

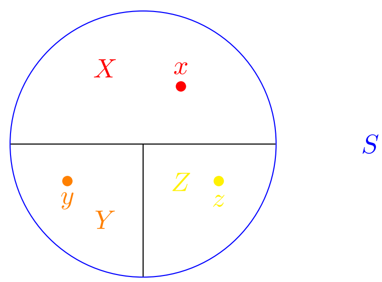
We can clearly see that none of the subsets are the empty set, that they all cover S (if we take the union of them, we will end up with S), and that they are all disjoint (none of them have any elements in common). Therefore, putting each element in its own subset is a valid partition of the set. The partition P itself looks like this:

$$P = \{\{x\}, \{y\}, \{z\}\}$$

4.2 Equivalence Classes

Earlier, we learned about equivalence relations being binary relations which are reflexive, symmetric, and transitive. These properties come in very handy by making an equivalence relation **always** partition a set in a valid manner. To do so, start by choosing an element from the set and then put it in a subset. Any element which is equal to that element should also be placed in that subset. Repeat until all elements are in a subset.

Let's say we have a set S. If I partition it with an equivalence relation, the resulting subsets are called **equivalence classes**. In the image below, X, Y, and Z are equivalence classes which partition the set S, and they each have a member $x \in X$, $y \in Y$, and $z \in Z$.



It's actually not too hard to see why partitioning it with an equivalence relation in the way described above will always result in a valid partition; for clarity, let's remind ourselves of the conditions required for a valid partition:

- No subset can be the empty set.
- The subsets must cover S.
- All of the subsets must be disjoint from each other.

For one, no subset will be created without any elements, so no subset will be empty. Secondly, if every element from S was placed in a subset, the union of all of the subsets must necessarily equal S, therefore the subsets will cover S. To show that the last condition holds, let's use a bit of a proof by contradiction.

Let's say that the set S has two different equivalence classes, A and B, with elements a and b respectively ($a \in A$ and $b \in B$). Because a and b belong to different equivalence classes, a must necessarily be unequal to b:

$$a \neq b$$

To sort an element c into an equivalence class, I need to determine that it is equal to a member of that equivalence class. We need to prove that c can only be placed into one equivalence class, not both, and thus equivalence classes are disjoint. To do so, let's suppose that we can place it into more than one equivalence class. If

$$c = a$$

c clearly belongs in the equivalence class A. If

$$c = b$$

c clearly belongs in the equivalence class B. If we can place c in both A and B

$$c = a$$

and

$$c = b$$

therefore the transitive property of our equivalence relation says that

$$a = b$$

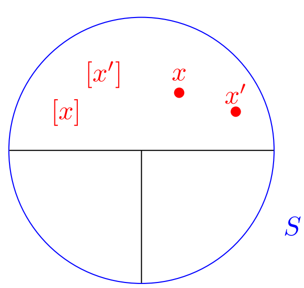
However, we've already established that for a and b to be in different equivalence classes

$$a \neq b$$

Thus we've run into a contradiction, so our original premise must be flawed, and c can only be placed into a single equivalence class. Thus, equivalence classes must be disjoint, and with that, we've shown that a partition via an equivalence relation is always a valid partition.

4.3 Equivalence Class Representatives

Something which I touched on earlier was that if we partition a set S by an equivalence relation resulting in an equivalence class X, every element inside X is equal to every other element inside X. This means that I can describe the entirety of X with a single element $x \in X$, like this: $[x]$. $[x]$ stands for the equivalence class containing x (and thus also containing every element in S which equals x), and would be equivalent to X. Using this notation, x is said to be a **representative** of the equivalence class $[x]$. In fact, any x' which is inside $[x]$ could be a representative of it, because $x = x'$ if they are both inside the same equivalence class: $[x] = [x']$.



The blue circle represents the set S, while the sectors represent different equivalence classes - as you can see, the equivalence classes are all disjoint, they cover S, and none are the empty set. The semicircle on top is the equivalence class $[x]$ or $[x']$ with x and x' being representatives of the equivalence class.

Chapter 5

Set and Sequence Notation

5.1 Motivations

While we've been learning about sets, set operations, and partitioning a set, we've pretty much stuck to finite sets and sequences with arbitrary manually-selected elements. We used curly braces {} for sets and parenthesis for sequences, and mostly directly notated which elements belong in the set. This is called **roster notation** and works really well for an introduction using small, finite sets, but to talk about any more advanced topics we'll need a notation which is a bit more concise and flexible. Luckily, many smart mathematicians have figured out a few pretty good systems for describing sets and sequences with finite or even infinite elements.

5.2 Prose Notation

Sometimes, words are the best way to represent what can go inside a set or sequence. For example, maybe you want to have a set of triangles where the area is less than 100 units - you might write that $S = \{$triangles with an area of less than 100 units$\}$. This notation can be useful in some circumstances, but most of the time tends to be imprecise and should be avoided.

5.3 Ellipses Notation

If a set or sequence follows a very easy-to-see pattern, it can be notated by simply including the first few terms (the minimum amount needed to clearly see the pattern) followed by some ellipses (...). For example, I could have a set $\{2, 4, 6, 8...\}$, or a sequence $(1, 3, 5, 7...\}$. This works really well for simple patterns, but if your pattern is somewhat convoluted it may be best to use a different notation to describe it.

5.4 Set-Builder Notation

Set-builder notation is arguably the most versatile notation for describing sets. It allows us to define which elements should and should not be in a set based on whether they fit some given criteria. A set in set-builder notation looks somewhat like this:

$$\{object \mid rules\}$$

In this case, the 'object' portion is normally a variable and specifies what type of object can qualify for membership within the set, the vertical bar (sometimes replaced with a colon) means "such that", and the 'rules' portion specifies the conditions which the object has to satisfy to be an element of the set.

Let's look at an example:

$$S = \{x \mid x \text{ } is \text{ } blue\}$$

On the left hand side, we see that the object going into the set is going to be represented by the variable x. On the right side, we see that x must be blue to be in the set. Taken together, the statement is read as "S is the set of objects x which are blue". The set itself would end up with every element from the universal set (which is the set of all objects under consideration) which is also blue. Another example:

$$S = \{n \in \mathbb{Z} \mid n \geq 0\}$$

On the left hand side, we see that the object going into the set is going to be represented by the variable n and is going to be an element of the integers. On the right side, we see that n must be greater than or equal to 0. Taken together, the statement is read as "S is the set of numbers n which are integers such that n is greater than or equal to 0", or simply "S contains integers which are greater than or equal to 0".

I could also have written the set like this:

$$S = \{n \mid n \geq 0, \ n \in \mathbb{Z}\}$$

A comma-separated list of conditions means that only elements which pass all of them are admitted into the set; the comma is a shorthand for the logical AND operator, and discussing that would require a whole side tangent into mathematical logic, so for now just interpret the comma as meaning that the entire list of conditions must be true for an element to be eligible for membership into the set.

Note that putting a domain restriction on the variable (essentially saying what type of thing the variable can be or what set it has to be an element of - in this case the domain restriction is $n \in \mathbb{Z}$) can be placed on the left or right side of the vertical bar, but it's best practice to put it on the left side.

Also note that while the variable you use doesn't actually matter - it's just a dummy placeholder - it's best to use a variable which corresponds to the type of object you're intending to go into the set. If your elements are numbers, consider using n or x or a; if your elements are ordered pairs, consider using (x, y) or (a, b); if your elements are vectors, use $\mathbf{u}$, $\mathbf{v}$, or $\mathbf{w}$, etc. This is simply because those variables are commonly used and somewhat standardized by mathematicians across the globe.

5.5 Sequence Notations

Sequence notation often revolves around index numbers, because sequences are indexed. One way to write sequences is explicitly as a function of their index number: the sequence $(2, 4, 6, 8...)$ might be written as $f(n) = 2n$, where $f(n)$ represents the element of the sequence and n represents its index number.

Another way to write a sequence explicitly (ie using an expression containing n) for a finite amount of terms is using parenthesis notation: $(2n)_{n=1}^{5}$ essentially means to write out the result of the expression $2n$ for the values $n = 1$ to $n = 5$, which will result in the sequence $(2, 4, 6, 8, 10)$.

We could also define a sequence recursively by writing a rule for the next term (often denoted $f(n+1)$) in terms of the previous term(s) ($f(n)$, $f(n-1)$, $f(n-2)$, etc). For example, $f(n+1) = f(n) + 2$ would define a sequence where the next term is equal to the previous term plus 2. To be clear which element is the first element, we also need to add an anchoring statement similar to this: $f(1) = 2$. Taken together, the recursive rule

$$f(n+1) = f(n) \cdot 2$$

$$f(1) = 2$$

would give us the sequence $(2, 4, 6, 8, 10, 12...)$

Chapter 6

Operators over a Set

Chapter 7

Cardinals and Ordinals

This chapter is a WIP because I am currently studying this topic. When I finish (depending on how quickly I understand it, it could take between a week and a year) I will try to figure out how to present it in the best way possible.