

Paradoxes:

- A **paradox** is a self-contradictory statement that at first seems true.
- **Barber's Paradox:** The story is that there's a village that has a male barber who shaves every man in the village who does not shave himself.

Question: Does the barber shave himself?

Consider this:

- If yes:
 - Recall that the barber only shaves men who don't shave themselves.
 - Hence, he does not shave himself.
- If no:
 - Recall that the barber only shaves men who don't shave themselves.
 - Hence, he does shave himself.
- **Note:** This is a proof by contradiction that the village in the story doesn't exist.

Note: The problem in this paradox is that there's a self-referential aspect to what's going on.

I.e. This is about an action that the barber does or doesn't do on himself.

Furthermore, there's also a negation. Self-reference with negation turns out to be problematic.

Note: This self-referential aspect with negation comes up a lot in paradoxes.

- **Liar's Paradox/Epimenides' Paradox:** Consider the proposition: **"This statement is false"**.

Consider this:

- If the above proposition is true, then it must be false.
- If the above proposition is false, then it must be true.

Notice the self-referential aspect with negation.

The conclusion we can draw from this is that our language is so powerful that it allows us to state things that are paradoxical.

- **Russell's Paradox:** Let the set R be the set of all sets that are not members of themselves.

I.e. $R = \{X \mid X \text{ is a set and } X \notin X\}$

Question: Does the set R contain itself?

Consider this:

- If yes:
 - Recall that R is the set of all sets that are not members of themselves.
 - Hence, R doesn't contain itself.
- If no:
 - Recall that R is the set of all sets that are not members of themselves.
 - Hence, R does contain itself.

Again, notice the self-referential aspect with negation.

- **Gödel's completeness theorem.** (**Note:** While this is not a paradox, it is very closely connected with paradoxes and is a very important mathematical result.) Using

elementary mathematics involving just integers, addition and multiplication, Gödel was able to write a mathematical statement, S , whose meaning is:

S = “This statement is unprovable”.

If S is true, then S is a true unprovable statement.

If S is false, then S is a false provable statement.

While this isn't a contradiction, it gives us two unpalatable choices.

The first choice says that our axiom system only proves true things but does not prove all true things.

The second choice says that we have an axiom system that allows us to prove false statements, which allows us to prove anything. This is useless.

Size of Infinite Sets:

- **Set Denotations:**
- N = Set of natural numbers
- Z = Set of integers
- Q = Set of rational numbers
- R = Set of real numbers
- **Example:**
- Consider the set of natural numbers, $N = \{0, 1, 2, \dots\}$ and the set of perfect squares, PS , $PS = \{0, 1, 4, \dots\}$. These two sets have the same size, not because they are both infinite sets but because there is a **one-to-one correspondence** between the two sets. Take any natural number in N and you will get its corresponding perfect square value in PS . Likewise, take any perfect square in PS and you will get its corresponding square root in N . Furthermore, each value in N has exactly one corresponding value in PS and each value in PS has exactly one corresponding value in N . However, PS is clearly a **proper subset** of N . This shows that two infinite sets can have the same size, even when one is a proper subset of the other. Hence, we must be careful when dealing with infinite sets.

Note: A **proper subset** of the subset “ A ” is a subset of “ A ” that is not equal to “ A ”. I.e. If “ B ” is a proper subset of “ A ”, then all elements of “ B ” are in “ A ”, but “ A ” must contain at least one element that is not in “ B ”.

E.g. Let $A = \{1, 3, 5\}$

Then, $\{1\}$, $\{1, 3\}$, $\{1, 5\}$ and $\{3, 5\}$ are all proper subsets of A .

However, $\{1, 3, 5\}$ is not a proper subset of A , because it contains all the elements in A . It is a subset of A .

Lastly, $\{1, 4\}$ is not a subset of A as it contains an element, 4, that is not in A .

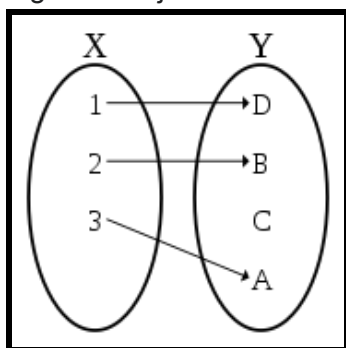
- **Definitions:**
- Two sets, A and B , are **equinumerous**, meaning they have the same size or **cardinality**, iff there is a **bijection** f from A to B .

Note: To show that A and B are equinumerous, we do $|A| = |B|$.

$|S|$ denotes the size of set S .

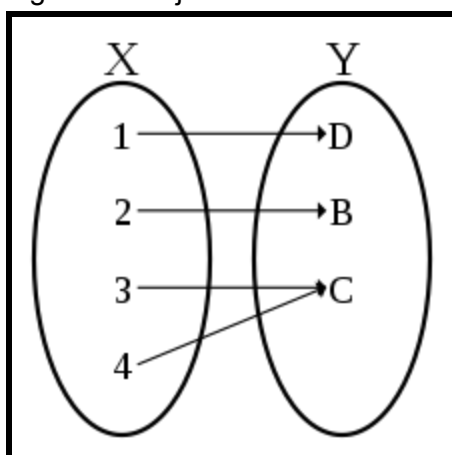
Note: An **injection**, also called a **one-to-one function**, is a function that maps distinct elements of its domain to distinct elements of its codomain. In other words, every element of the function's codomain is the image of at most one element of its domain.

E.g. of an injection.



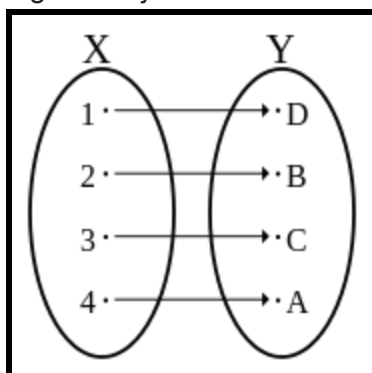
Note: A **surjection**, also called **onto**, is a function such that for every element y in the codomain Y of f , there is at least one element x in the domain X of f such that $f(x) = y$. It is not required that x be unique; the function f may map one or more elements of X to the same element of Y .

E.g. of an surjection.



Note: A **bijection**, also called a **one-to-one correspondence**, is a function between the elements of two sets, where each element of the first set is paired with exactly one element of the second set, and each element of the second set is paired with exactly one element of the first set. There are no unpaired elements. A bijective function $f: X \rightarrow Y$ is a **one-to-one (injective)** and **onto (surjective)** mapping of a set X to a set Y . Please note that the term “one-to-one correspondence” is not the same as a “one-to-one function.”

E.g. of a bijection.



- A set "A" is **countable** or **enumerable** iff "A" is finite or it is equinumerous to \mathbb{N} . If "A" is equinumerous to \mathbb{N} , we say that "A" is **countably infinite**, which means that there is a bijection f from \mathbb{N} to "A".
I.e. $f(n) = a_n$ where n is the n^{th} element of \mathbb{N} and a_n is the n^{th} of "A". (This shows the bijection from \mathbb{N} to "A".)

The **enumeration of a countable set** is a sequence of all the elements in that set such that each element appears exactly once in some particular index of this sequence.

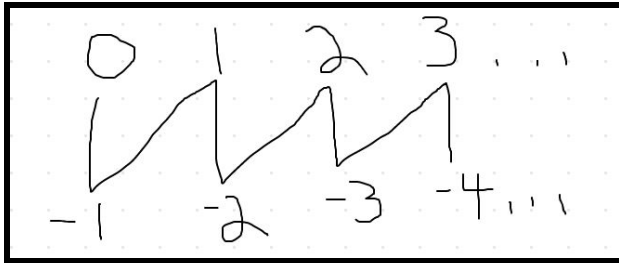
Note: The sequence does not have to be increasing or decreasing. Any sequence suffices as long as each element appears exactly once in some particular index/position.
E.g.

0, 1, 2, 3, ..., -1, -2, -3, ... is not an enumeration because we cannot give the index of the element -1.

- **Theorem 1.1:** \mathbb{Z} is countable.

Note: $\mathbb{Z} = \dots, -3, -2, -1, 0, 1, 2, 3, \dots$ is not an enumeration. We cannot give the index of the element 0.

This is an enumeration of \mathbb{Z} :



or

0, -1, 1, -2, 2, -3, 3, ...

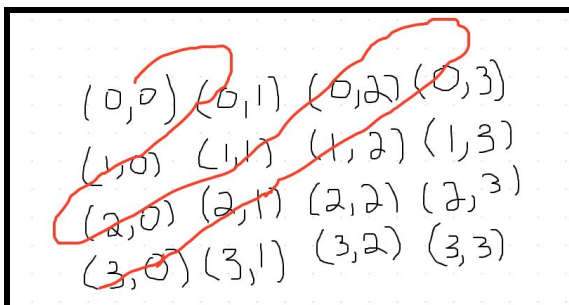
Now, I can give you a specific index of any element.

This technique is called **dovetailing**. In general, with dovetailing, you take multiple lists and merge them into a single list.

- **Theorem 1.2:** The set of the cartesian product of natural numbers, $\mathbb{N} \times \mathbb{N}$, is countable.
This set looks like the following:

{
(0,0), (0,1), (0,2), ...
(1,0), (1,1), (1,2), ...
(2,0), (2,1), (2,2), ...
...
}

I can enumerate the set like this:



I.e.

I list all the pairs (i,j) such that $i+j = 0$. There is 1 of these pairs.

Then, I list all the pairs (i,j) such that $i+j = 1$. There are 2 of these pairs.

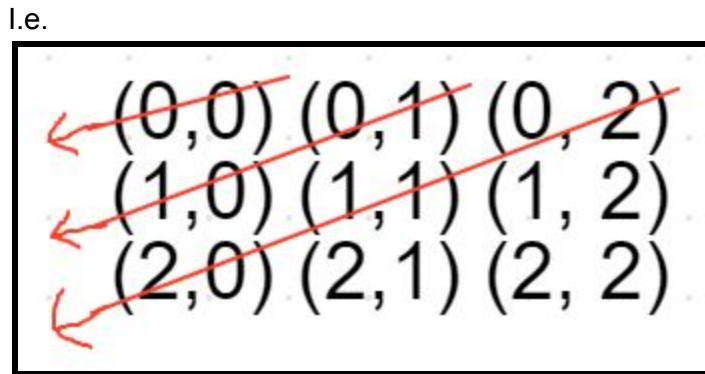
Then, I list all the pairs (i,j) such that $i+j = 2$. There are 3 of these pairs.

We continue this method for all elements in this set.

To find the specific index of an element (i,j) in this set, we can do the following:

$$\frac{(i+j)(i+j+1)}{2} + i$$

Note that this equation is for if we visit the diagonal where the sums of the pairs are equal and we start from the top row and move diagonally left, as shown below:



E.g.

1. Take $(0,0)$
 $i = 0, j = 0$
 $((0+0)(0+0+1)/2) + 0 = 0$
2. Take $(0,1)$
 $i = 0, j = 1$
 $((0+1)(0+1+1)/2) + 0 = 1$
3. Take $(1,0)$
 $i = 1, j = 0$
 $((1+0)(1+0+1)/2) + 1 = 2$

This is called the **pairing function**. It takes 2 arguments, which are natural numbers, and returns the position of the 2 arguments. It encodes pairs of natural numbers by one natural number.

- **Theorem 1.2.1:** The set of positive rational numbers is countable. Note that this set is a **dense set**, meaning that for any two distinct elements in the set, you can find another element in between the pair. To prove that this set is countable, we will use the snaking method shown in theorem 1.2.

i.e.

| | | | | |
|---------------|---------------|---------------|---------------|-----|
| $\frac{1}{1}$ | $\frac{1}{2}$ | $\frac{1}{3}$ | $\frac{1}{4}$ | ... |
| $\frac{2}{1}$ | $\frac{2}{2}$ | $\frac{2}{3}$ | $\frac{2}{4}$ | ... |
| $\frac{3}{1}$ | $\frac{3}{2}$ | $\frac{3}{3}$ | $\frac{3}{4}$ | ... |

- **Theorem 1.2.2:** $N \times N \times N$ is countable, too. To prove this, think of the element (i,j,k) as $((i,j),k)$. Then, using the technique from theorem 1.2, we can prove that $N \times N \times N$ is countable. Hence, this method encodes triples of natural numbers by one number.
- **Theorem 1.3:** $\forall k \in N, N^k = N * N \dots * N$ is countable. We can use induction to prove this.
- **Theorem 1.3.1:** The set of reals in $(0, 1]$ is not countable. To prove this, we will use the below theorem:

Theorem 1.4: The set B^∞ of infinite binary strings is not countable.

To prove this, assume for contradiction that B^∞ is countable.

Let X_0, X_1, X_2, \dots be any enumeration of B^∞ .

Let B_{ij} be the j^{th} bit of X_i .

Next, we will construct an infinite binary string, x , whose i^{th} bit is $\neg B_{ii}$, the complement of B_{ii} .

Note that x is not in any enumeration of B^∞ because it differs from each enumeration by one bit. x differs from X_i in at least the i^{th} position, so $x \neq X_i$ for all $i \in N$. Therefore, X_0, X_1, \dots , is not an enumeration of B^∞ , so B^∞ is not countable.

E.g.

| | | | | | |
|----------|---|---|---|---|-----|
| $X_0 =$ | 0 | 1 | 0 | 0 | ... |
| $X_1 =$ | 1 | 1 | 0 | 1 | ... |
| $X_2 =$ | 0 | 0 | 1 | 1 | ... |
| \vdots | | | | | |

$x = 100\dots$

We have our enumerations, X_0, X_1, X_2, \dots . Next, we take the i^{th} bit of X_i , which is B_{ii} , and get the complement of that to construct x . In the example above, $B_{00} = 0$, $B_{11} = 1$, $B_{22} = 1$. Hence, we have 011. The complement of 011 is 100, which is what x is set to. Since x differs from X_i in at least the i^{th} position, X_0, X_1, \dots , is not an enumeration of B^∞ , and as such, B^∞ is not countable.

The above method is known as the **Diagonalization Method**.

The proof for theorem 1.3.1 is similar.

Let R_0, R_1, R_2, \dots be any enumeration of the set of real numbers in $(0,1]$.

Let B_{ij} be the j^{th} digit of R_i .

Next, we will construct an infinite number, x , whose i^{th} digit is any digit other than B_{ii} . If B_{ii} is 1, then the i^{th} digit of x can be any digit other than 1.

Then, we have created a real number between $(0,1]$ that is not in any of the enumerations.

Therefore, the set of real numbers between $(0,1]$ is uncountable.

- **Theorem 1.5:** The set of functions from \mathbb{N} to \mathbb{N} is uncountable. We can prove this using proof by contradiction.

By the diagonal method, suppose that it is countable.

Let f_0, f_1, f_2, \dots be the functions.

Consider the picture below. It is a chart of the enumerations of the set. On the top row, we have the natural numbers and on the left column, we have the functions. In each cell, we have the result of the given function on the given natural number.

I.e. $f_0(0) = 0$, $f_1(3) = 1$, etc

| | 0 | 1 | 2 | 3 | ... |
|-------|---|----|---|---|-----|
| f_0 | 0 | 3 | 4 | 8 | |
| f_1 | 9 | 12 | 3 | 1 | |
| f_2 | 1 | 3 | 4 | 8 | |
| : | | | | | |

Next, consider the diagonal shown below. The diagonal gives us the i^{th} value of the i^{th} function.

| | 0 | 1 | 2 | 3 | ... |
|-------|---|----|---|---|-----|
| f_0 | 0 | 3 | 4 | 8 | |
| f_1 | 9 | 12 | 3 | 1 | |
| f_2 | 1 | 3 | 4 | 8 | |
| : | | | | | |

We will create a new function, f , such that $f(i) \neq f_i(i)$.

I.e. $f(0)$ gives any number other than $f_0(0)$ or 0.

f does not appear in any of the enumerations. Hence, the set of functions from \mathbb{N} to \mathbb{N} is uncountable.

- **Theorem 1.6:** Let Γ be a (finite) alphabet, $|\Gamma| \geq 2$. An **alphabet** is just a set of symbols. Furthermore, for our purposes, alphabets are always finite. Γ^* is the set of finite strings

over Γ . Γ^* is countable. To prove this, enumerate all strings over Γ of length 0, then length 1, then length 2, and so on. For each enumeration, we will enumerate in lexicographical order.

Uncomputability:

- Computers compute functions $N \rightarrow N$. However, there are some functions $N \rightarrow N$ that computers cannot compute.

One reason for this is because there is an uncountable number of functions from $N \rightarrow N$ but there are a countable number of programs to compute them. This is called the **counting argument** and is a cheap argument.

- **Turing's Halting Function:** Let's call the function h . h takes 2 arguments, P and x , and returns a single bit. It outputs 1 if program P halts on x . It outputs 0 if P does not halt on x .

i.e.

$$h(P, x) = \begin{cases} 1, & \text{if } P \text{ halts on } x \\ 0, & \text{if } P \text{ doesn't} \\ & \text{halt on } x \end{cases}$$

Fact: No program can compute h .

Proof:

We will prove the above fact using a proof by contradiction.

Suppose, for contradiction, that such a program exists. We'll call it H .

$$H(P, x) = \begin{cases} 1, & \text{if } P \text{ halts on } x \\ 0, & \text{if } P \text{ doesn't} \\ & \text{halt on } x \end{cases}$$

I will modify H to obtain a new program, H' . I will replace every "return b " statement in H , where b is either 1 or 0, by the following:

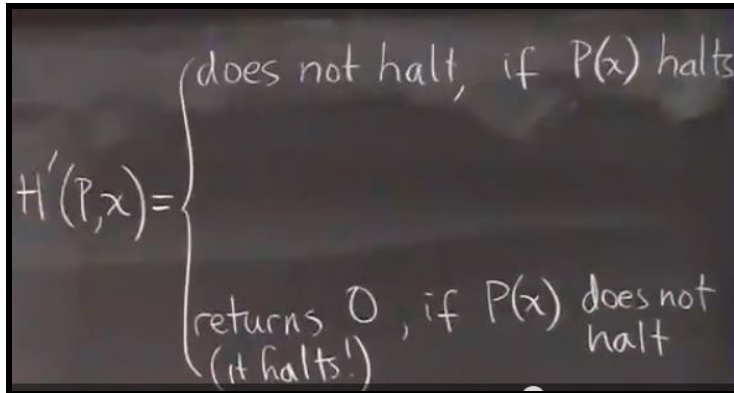
"if $b==1$ then:

while (true): \leftarrow Notice the infinite loop here.

$x = x$

else:

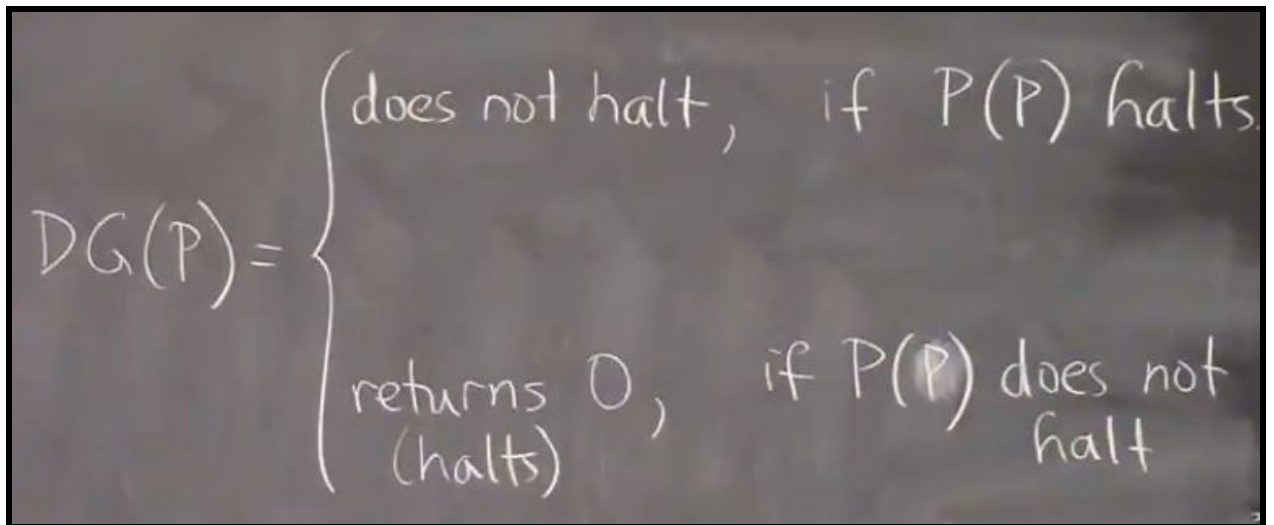
return 0"


$$H'(P, x) = \begin{cases} \text{does not halt, if } P(x) \text{ halts} \\ \text{returns 0, if } P(x) \text{ does not} \\ \text{(it halts!)} \end{cases}$$

Now, I will write a new program called $DG(P)$.

$DG(P)$:

return $H'(P, P)$


$$DG(P) = \begin{cases} \text{does not halt, if } P(P) \text{ halts} \\ \text{returns 0, if } P(P) \text{ does not} \\ \text{(halts)} \end{cases}$$

Since DG itself is a program, we can do $DG(DG)$.

The image shows a handwritten definition of a function $DG(P)$ on a chalkboard. The definition is written as a piecewise function:

$$DG(P) = \begin{cases} \text{does not halt,} & \text{if } P(P) \text{ halts.} \\ \text{returns 0,} & \text{if } P(P) \text{ does not halt} \\ \text{(halts)} & \end{cases}$$

There are additional annotations above the cases: $DG(DG)$ is written above the first case, and $DG(DG)$ is written above the second case.

Hence, $DG(DG)$ halts iff $DG(DG)$ doesn't halt. This is a contradiction. Hence, $h(P,x)$ is uncomputable.

Note: The function is called the **halting function**, but finding a program that computes the function is called the **halting problem**.

Reduction:

- A problem X reduces to problem Y , denoted by $X \leq Y$, if we can use a given or assumed solution to Y to solve X .
- If X reduces to Y , then X is no harder than Y . This is what the \leq sign is intended to capture. X cannot be harder than Y because if we can solve Y , we can solve X . Another way of thinking about this is that Y is at least as hard as X .
- This concept can be used in 2 ways:
 1. Positive Use: If X reduces to Y , and somebody already solved Y , I can use it to solve X .
 2. Negative Use: Suppose I have reduced X to Y . If I know that X is uncomputable, then it follows that Y is also uncomputable. I.e. This way proves that some things are hard or unsolvable. This is the way we will be using reduction for the course.

- E.g. There are 2 problems, Zero and Halt, defined below.

Zero: Given a program P and input x , determine if $P(x)$ returns 0.

Halt: Is the halting problem.

$\text{Halt} \leq \text{Zero}$. Since we cannot compute Halt and Halt is no harder than Zero, we cannot compute Zero.

Proof that Halt reduces to Zero:

Suppose that Z-Solver is a program that solves Zero.

$$\text{Z-SOLVER}(P, x) = \begin{cases} 1 & \text{if } P(x) = 0 \\ 0 & \text{o.w.} \end{cases}$$

We will use Z-Solver to write a program, H-Solver, that solves the Halting Problem. I can reduce Halt to Zero by showing that Z-Solver is used to solve Halt.

Now, we will create H-Solver. It takes 2 inputs, program P and x.

In H-Solver, there is another function called P'. P' is a program obtained from P by changing every "return ___" statement to "return 0". H-Solver returns Z-Solver(P', x)

$$\text{H-SOLVER}(P, x) = \begin{cases} 1, & \text{if } \underbrace{P'(x) \text{ returns } 0}_{\substack{\updownarrow \\ P(x) \text{ halts}}} \\ 0, & \text{if } \underbrace{P'(x) \text{ does not return } 0}_{\substack{\updownarrow \\ P(x) \text{ does not halt}}} \end{cases}$$

By using Z-Solver in H-Solver, it shows that Halt reduces to Zero.

Therefore, H-Solver solves the halting problem, which is impossible. Since Halt is impossible, Zero is also impossible.

Textbook Notes:

- Georg Cantor discovered the **diagonalization** technique in 1873.
- Cantor was concerned with the problem of measuring the sizes of infinite sets. If we have two infinite sets, how can we tell whether one is larger than the other or whether they are of the same size? For finite sets, of course, answering these questions is easy. We simply count the elements in a finite set, and the resulting number is its size. But if we try to count the elements of an infinite set, we will never finish! So we can't use the counting method to determine the relative sizes of infinite sets.
- For example, take the set of even integers and the set of all strings over $\{0, 1\}$. Both sets are infinite and thus larger than any finite set, but is one of the two larger than the other? How can we compare their relative size?
- Cantor proposed a rather nice solution to this problem. He observed that two finite sets have the same size if the elements of one set can be paired with the elements of the

other set. This method compares the sizes without resorting to counting. We can extend this idea to infinite sets. Here it is more precisely:

Assume that we have sets A and B and a function f from A to B . f is **one-to-one** if it never maps two different elements to the same place—that is, if $f(a) \neq f(b)$ whenever $a \neq b$. f is **onto** if it hits every element of B —that is, if for every $b \in B$ there is an $a \in A$ such that $f(a) = b$. A and B have the same size if there is a one-to-one and onto function $f : A \rightarrow B$. A function that is both one-to-one and onto is called a **correspondence**. In a correspondence, every element of A maps to a unique element of B and each element of B has a unique element of A mapping to it. A correspondence is simply a way of pairing the elements of A with the elements of B .

Note: Alternative common terminology for these types of functions is **injective** for **one-to-one**, **surjective** for **onto**, and **bijective** for **one-to-one** and **onto**.

- E.g.

Let N be the set of natural numbers $\{1, 2, 3, \dots\}$ and let E be the set of even natural numbers $\{2, 4, 6, \dots\}$. Using Cantor's definition of size, we can see that N and E have the same size. The correspondence f mapping N to E is simply $f(n) = 2n$. This is shown in the table below.

| n | $f(n)$ |
|----------|----------|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| \vdots | \vdots |

Of course, this example seems bizarre. Intuitively, E seems smaller than N because E is a proper subset of N . But pairing each member of N with its own member of E is possible, so we declare these two sets to be the same size.

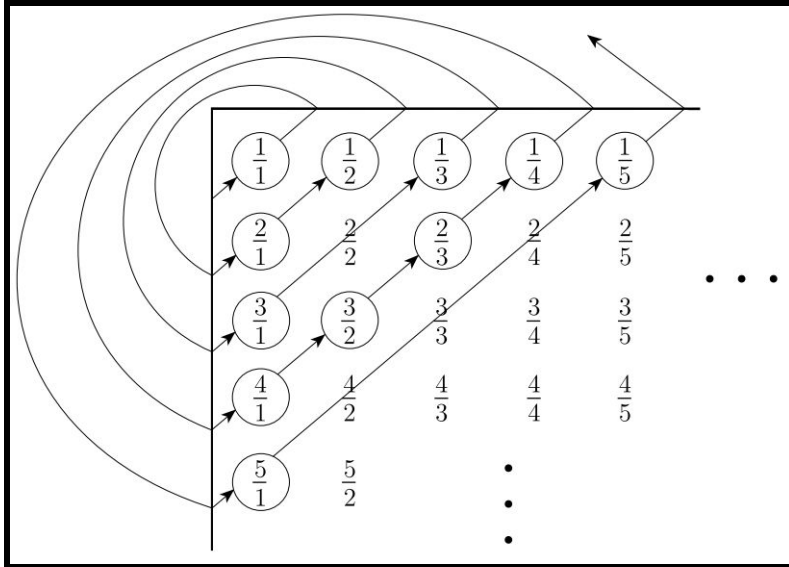
- **Definition:** A set " A " is **countable** if either it is finite or it has the same size as N .

- E.g.

If we let $Q = \{\frac{m}{n} \mid m, n \in N\}$ be the set of positive rational numbers, Q seems to be much larger than N . Yet these two sets are the same size according to our definition. We give a correspondence with N to show that Q is countable. One easy way to do so is to list all the elements of Q . Then we pair the first element on the list with the number 1 from N , the second element on the list with the number 2 from N , and so on. We must ensure that every member of Q appears only once on the list. To get this list, we make an infinite matrix containing all the positive rational numbers. The i th row contains all numbers with numerator i and the j th column has all numbers with denominator j . So the number

$\frac{i}{j}$ occurs in the i th row and j th column.

Now we turn this matrix into a list. To list it, we list the elements on the diagonals, starting from the corner. The first diagonal contains the single element $1/1$, and the second diagonal contains the two elements $2/1$ and $1/2$. So the first three elements on the list are $1/1$, $2/1$, and $1/2$. In the third diagonal, a complication arises. It contains $3/1$, $2/2$, and $1/3$. If we simply added these to the list, we would repeat $1/1 = 2/2$. We avoid doing so by skipping an element when it would cause a repetition, so we add only the two new elements $3/1$ and $1/3$. Continuing in this way, we obtain a list of all the elements of \mathbb{Q} . This is shown in the picture below.



- **Definition:** Some infinite sets have no correspondence with \mathbb{N} . Such sets are called **uncountable**.

The set of real numbers is an example of an uncountable set. This will be proved below.

- **Theorem:** \mathbb{R} is uncountable.

Proof:

In order to show that \mathbb{R} is uncountable, we show that no correspondence exists between \mathbb{N} and \mathbb{R} . The proof is by contradiction. Suppose that a correspondence f existed between \mathbb{N} and \mathbb{R} . Our job is to show that f fails to work as it should. For it to be a correspondence, f must pair all the members of \mathbb{N} with all the members of \mathbb{R} . But we will find an x in \mathbb{R} that is not paired with anything in \mathbb{N} , which will be our contradiction.

The way we find this x is by actually constructing it. We choose each digit of x to make x different from one of the real numbers that is paired with an element of \mathbb{N} . In the end, we are sure that x is different from any real number that is paired. To do this, we will enumerate through the set \mathbb{R} .

Suppose that:

$R_0 = 3.\underline{1}415\dots$

$R_1 = 5.55555\dots$

$R_2 = 0.12345\dots$

$R_3 = 0.50000\dots$

We construct the desired x by giving its decimal representation. It is a number between 0

and 1, so all its significant digits are fractional digits following the decimal point. Hence, to begin constructing x , we will only look at the digits after the decimal in each of R_i . We will take the first digit after the decimal in R_0 , the second digit after the decimal in R_1 , and so on. In general, we will take the $i+1$ digit after the decimal in R_i .

In our example, for now, $x = 0.1530\dots$

Then, to make sure that $x \neq R_i$, we will change all of x 's digits after the decimal place.

We don't care what the new digit is, as long as it is not the same as the original digit.

I.e. In our example, now $x = 0.3421\dots$

Since x differs by each R_i by at least 1 digit, x could not have appeared in the enumeration. Hence, R is uncountable.