Basic Analytical Reasoning & Notation for non-Math majors

Periklis A. Papakonstantinou

LECTURE NOTES IN Elements of Quantitative Thinking

Periklis A. Papakonstantinou

Contents

Acknowledgments

I would like to thank the terrific students who took my classes for their remarks, corrections, suggestions, and editing effort on the original manuscript. A special thanks for this goes to Adedola Adefowoju, Sheha Darji, Love Nosa-Omorogiuwa, Thomas Pappas, Yinzi Zhang.

Why to read this?

In the last couple of years I have come to realize that North American education for non-Mathematics majors[1] lacks an *indispensable* component from a number of university classes. The material covered here is *necessary* for *any* class in Computation, Cryptography, Machine Learning, Databases, Operations Research, and of course, in all mathematics-related prerequisites; e.g. linear algebra, probability, and statistics.

It is impossible to think quantitatively when lacking *elementary reasoning skills*. From my point of view, a student without this basic background can never make sense of anything with quantitative or formal meaning and of course can never produce anything related later on in her professional life.

I believe it is unacceptable for university-level education one not to be able to know what the following statement exactly means "find all employees who made the most sales on 2/1/2019 *OR* 2/2/2019".

Luckily, as we will see, all these can be easily fixed.

[1] This seems to be true for many well-established North American (north of Rio Grande) schools.

Lecture 1

Sets, their size, operations, and mappings

1.1 Sets and basic notation

A *set* S is a collection of *elements*. Some sets have a finite number of elements – e.g. the set $S = \{1, 2, 10\}$ has three elements – and some are infinite – e.g. the set of all even positive integers $W = \{2, 4, 6, \dots\}$. We describe sets by enclosing in brackets $\{\quad\}$ their elements. Alternatively, we may describe them as

$$S = \{x \mid \text{a condition about } x\}$$

In this case we replace x with every element that satisfies the condition after "$|$". More about this later on when we explain what is "a condition". One more thing about sets: repetitions of elements and order does not matter. For example, $\{1, 2, 1, 1, 3\} = \{2, 1, 3, 3, 3\} = \{1, 2, 3\}$. We prefer $\{1, 2, 3\}$ just because it looks nicer.

Finally, the symbol $\in$ reads as "it belongs to" and is a relation between an element and a set; e.g. $1 \in S$. In order to further clarify, "it belongs to" is a reference which will be used in class saying that the left side or an element which belongs to the right side with is in the set.

New sets from old sets

Consider $A = \{1, 2, 3\}$ and $B = \{3, 4, 5\}$. We can build new sets from these A and B. In particular, the *union* of two sets is a new set which puts together all the elements of the two sets $\{1, 2, 3, 3, 4, 5\}$. Recall that we eliminate repetitions, i.e. the union of A and B is $\{1, 2, 3, 4, 5\}$. We denote the new set that corresponds to the union as $A \cup B = \{1, 2, 3, 4, 5\}$. It is essential to realize that you should view $A \cup B$ as a **single symbol** because it is **one** set. The fact that the symbol $A \cup B$ looks longer, than e.g. the symbol C, should not confuse the reader. We create the symbol $A \cup B$ using more basic symbols because we want to be able to "read off" $A \cup B$ its constituents.

The *intersection* of A and B is a new set defined to contain exactly their common elements and is denoted by $A \cap B$. For the previous A, B we have $A \cap B = \{3\}$.

We should add to the above discussion that both A and B are subsets of the same bigger set, which sometimes we call the universe. By the way, a *subset* is denoted by $\subseteq$. Say for example, that $\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$ is the set of all integers. Then, clearly, $A \subseteq \mathbb{Z}$ and $B \subseteq \mathbb{Z}$. Note also that the two new sets are subsets of $\mathbb{Z}$, i.e. in symbols, we write $A \cup B \subseteq \mathbb{Z}$ and $A \cap B \subseteq \mathbb{Z}$.[1] The symbol for the *empty set* is $\emptyset$ or $\{\}$. Note that the symbols $\subseteq$ and $\in$ have completely different function and should not be confused. One last thing, the three[2] dots $\ldots$ is a precise symbol, which means "continue in the same way".

Constructing complicated elements

Suppose that we have a set $A = \{a, b, c, \ldots, z\}$ and $B = \{1, 2, 3\}$. Now, suppose that we want to define using A and B a new type/format of element, which is neither a letter from A nor a num-

[1] The operation union would still make some sense even if A consists of apples and B consists of oranges, but in this case the intersection of A and B would always be the *empty set*.

[2] These are not four dots or many dots, they are just three – it really is a symbol.

ber from B, but this new element has some structure: it consists of two parts "glued together", where the first is a letter and the second a number from 1 to 3. For example:

$$(b, 3)$$

The above element is a *vector* and **a vector is still just one element**. For example, if $C = \{(a, 1), (b, 2), (c, 3), (d, 2)\}$ then $A \cap C = \emptyset$ because the sets A and C have no common elements. The symbol $(a, 1)$ corresponds to a single element and there is no such element in the set A.

We say that $(a, 1)$ has two *coordinates*. Note that the order here matters. That is, $(1, a) \neq (a, 1)$.

Finally, we remark that an alternative notation for $(b, 3)$ is $\binom{b}{3}$. Both notations may be used interchangeably.

The operations of union and intersection make more sense for sets that are subsets of the same universe. By taking unions and intersections we cannot go outside the universe. But the element $(b, 3)$ is neither a letter nor a number. This element is an example of a vector notation looks like.

The *cartesian product* (or cross product) realizes the construction of a set consisting of complicated elements: $A \times B = \{(a, 1), (a, 2), (a, 3), (b, 1), (b, 2), \ldots, (z, 3)\}$. Again, $A \times B$ is just the symbol of a single set. This set consists exactly of all vectors with two coordinates, where the first coordinate is any element from A and the second any element from B. To summarize, $A \times B$ consists of the concatenations of each of all possible elements from A with each of all possible elements from B. Concatenation is a series of interconnected things or events. In this case, you see one letter being concatenated with one number.

When things become ambiguous, we start using parentheses

around the union, intersection, or cartesian product. For example, let $A = \{1,2,3\}$, $B = \{2,3,4,5\}$, and $C = \{6,7,8\}$. Then, how to read $C \cup A \times B$? It can mean either first take the union of C and A and then the cartesian product with B, or it can mean first take the product of A and B and then take the union with C. To remove this ambiguity we write $(C \cup A) \times B$ or $C \cup (A \times B)$.

Do not confuse: $A \times B$ for two sets A and B does not refer to any multiplication among their elements. $\times$ means, each element is a "concatenation" of two elements into a single vector that has two coordinates.

Constructing sets with more than two coordinates

Let $A = \{1,2,3\}$ and suppose that we want to construct a set with three coordinates each of which takes values from A.

We denote this as $A \times A \times A$.

According to what we said before, $\big((1,2),3\big)$ is an element of this set. This is the case if we disambiguate $A \times A \times A$ and we write it as $(A \times A) \times A$. Now, if instead by $A \times A \times A$ we meant $A \times (A \times A)$ the corresponding element would have been $\big(1,(2,3)\big)$. Can we do anything else according to what we said so far? No! We said whenever we write $A \times A$ this means that you construct each element of this new set by taking one element from the first copy of A and another from the second copy of A and put a left parenthesis "$($" to the left and then a right parenthesis "$)$" to the right. **This is a rule with a precise meaning**. Rules should not be taken lightly.

Compare $\big((1,2),3\big)$ and $\big(1,(2,3)\big)$? We observe that if we do not follow the above rule and drop parentheses then the relative order of all constituents of these vectors is the same. We can abuse notation and instead simply write $(1,2,3)$. This "abuse" is safe because from $(1,2,3)$ we can reconstruct whatever weird order of parentheses we

had before. Hence, we can drop parentheses altogether and write

$$A \times A \times A = \{(1,1,1),(1,1,2),(1,1,3),(1,2,1),\ldots,(3,3,3)\}$$

We can introduce even further notation and write A^3 instead of $A \times A \times A$. Note that this A^3 is just a succinct way to write a long cartesian product and nothing else.

Summary of notation/symbols we learned so far

$S = \{\quad\}$
"Set" $\mathbb{Z}$: the set of all integers
$A \cup B$
"The union of A & B" $A \cap B$
"The intersection of A & B" $A \times B$
"The concatenation of A & B" A^4
"Another way of concatenation of A" ...
"Continue in the same way" $(1,1,1,1)$
$\begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$

1.2 What can we use when we argue quantitatively?

In Lecture 2 and Lecture 3 we discuss rigorous reasoning and calculations. In rigorous reasoning we can use **only**: sets, operations among sets, operations among elements (e.g. addition, multiplication etc between integers), functions between sets, and logical inferences about these objects. **And nothing else**.

1.3 Mappings and functions

A *function between two sets* is a very important notion. A function f maps every element of a set A, the *domain*, to a set B, the *range*. To make sure that is clear what domain and range we have, we write:

$$f : A \to B$$

The above notation does not explain how the mapping is done.

In upcoming example, g is a function. The symbol which looks like Z,$\mathbb{Z}$, is a set containing all integers. Integers are whole numbers which is not a fraction and can consist of positive and negative number values. For example, consider a function mapping from the integers $\mathbb{Z}$ to the integers $\mathbb{Z}$. Let $g : \mathbb{Z} \to \mathbb{Z}$. Now, we must specify how an element of the domain (the first copy of $\mathbb{Z}$ above) is mapped to each element of the range (the second copy of $\mathbb{Z}$). In this example, $g(n) = n^2$. What is it missing? **What is n?** Is it a vegetable? The notation $g(n) = n^2$ **means nothing** unless we specify what is n. Thus, instead, we should have written "$g(n) = n^2$, where $n \in \mathbb{Z}$". To be fair, since we already wrote $g : \mathbb{Z} \to \mathbb{Z}$, here there isn't much ambiguity, but in general, as a rule of thumb **every symbol must have a defined type**.

According to what we said above, for a function $f : A \to B$ every element of A must be mapped (by the rule f) to some element of B. This does *not* mean that every element of B is pointed by an element of A. For example, for $g : \mathbb{Z} \to \mathbb{Z}$ where $g(n) = n^2,\ n \in \mathbb{Z}$, every integer (e.g. $g(2) = 4$, $g(-3) = 9$, etc) is mapped to some element of $\mathbb{Z}$. But there are elements in the range $\mathbb{Z}$ that are not covered by g. For example, g does not have negative outputs. In symbols, we would say:

$$\text{"for every } n \in \mathbb{Z},\ g(n) \geq 0\text{"}$$

or equivalently

$$\text{“there does not exist } n \in \mathbb{Z} \text{ such that } g(n) < 0\text{”}$$

(we will discuss this “equivalence” in Lecture 2 below)

Some notational “tricks”

Think of a way to represent the previous function g as a set. Construct a set that contains elements of the form $(\text{input}, \text{output})$. In particular, for $g(n) = n^2, n \in \mathbb{Z}$ an alternative way to view g is as $g = \{\dots, (-3,9), (-2,4), (-1,1), (0,0), (1,1), (2,4), (3,9), \dots\}$.

Thus we can “view” the function as $g \subseteq \mathbb{Z} \times \mathbb{Z}$. Only a few elements of $\mathbb{Z} \times \mathbb{Z}$ are in g. In particular, there do not exist two pairs $(x,y) \in g$ and $(x,y') \in g$ where $y \neq y'$. That is, the same input x can only be mapped to a single element in the range. This is true for every function. There are more complicated relations/mappings that allow the same x to be mapped to more than one y's, but these are not functions as defined in the previous section.

Here is another “notational trick”. We can view a vector $v = (1,2,3,2,1,42,21)$ as a function. In particular, we can define

$$v : \underbrace{\{1,2,3,4,5,6,7\}}_{\text{coordinate ID}} \to \underbrace{\{1,2,3,21,42\}}_{\text{value}}$$

“vector v” $\quad v = (1,2,3,2,1,42,21)$

“function v”
$$v : \{1,2,3,4,5,6,7\} \to \{1,2,3,21,42\}$$
$$v(1) = 1,\ v(2) = 2,\ v(3) = 3,\ v(4) = 2$$
$$v(5) = 1,\ v(6) = 42,\ v(7) = 21$$

The vector v or the function v contain the same information. Note: That we ignore the repetition of ”1” in the value set.

Let us now introduce one more "notational trick". Consider a set $A = \{1, 2, 3\}$ and note that $A \subseteq \mathbb{Z}$. Again, to make it clear, A is a subset. Let us define the *characteristic function* χ_A *of* A to be a function $\chi_A : \mathbb{Z} \to \{0, 1\}$ with the convention that the output of χ_A is 1 when the input is in the set and the output of χ_A is 0 when the input is not in the set. In particular, $\chi_A(1) = 1$, $\chi_A(2) = 1$, $\chi_A(3) = 1$, whereas for every other integer e.g. $\chi_A(100) = 0$ since $100 \notin A$. We can go further and replace the symbol χ_A by A, since A and χ_A provide exactly the same information about the set (χ_A tests membership of elements in A, whereas A is A itself).

Consider the set $\{0, 1\}^{10}$. This contains all vectors of length 10, meaning there are ten in it. Here the value of each coordinate is either 0 or 1. Since 0 is different than 1, $(0, 1, 0, 1, 1, 1, 0, 1, 0, 0)$ can be reconstructed uniquely from 0101110100. Therefore,[3] you can consider a vector to be a binary string, similar to what you have done in your programming classes. Note that order is very important when looking at vectors. For example, vector $A = (0, 1, 0)$ is not the same as vector $B = (0, 0, 1)$. The order of the two zeros and the one play a big impact on the value of the overall vector.

In university-level textbooks there is frequent switch between natural forms of notation. For example, you may read: "fix a vector $v \in \mathbb{Z}^{100}$ and now let us consider the sum of its coordinates $\sum_{i=1}^{100} v(i)$".

1.4 Preimages, inverses, and powersets

Consider $f : \{1, 2, 3\} \to \{a, b, c\}$, where $f(1) = a$, $f(2) = c$, $f(3) = b$. Suppose that we want to compute the *inverse function*, i.e. go from a back to 1, from b back to 3, and from c back to 2. For this f we

[3] This is **not** the case if the set had elements $\{0, 1, 10\}^3$ because then 1010 could correspond both to $(1, 0, 10)$ and to $(10, 1, 0)$ and we have no way to disambiguate.

can naturally construct the inverse, denoted by f^{-1}. (again,"f^{-1}" is just one symbol), where $f^{-1} : B \to A$ and $f^{-1}(a) = 1, f^{-1}(b) = 3, f^{-1}(c) = 2$. We also say that "1 *is the pre-image of a under* f".

What if we instead f is as follows:
$f : \{1,2,3\} \to \{a,b,c\}$, where $f(1) = a$, $f(2) = a$, $f(3) = b$?
Then, the inverse of "a" is two elements of of $\{1,2,3\}$. Thus, the inverse of f cannot anymore be a function $f^{-1} : \{a,b,c\} \to \{1,2,3\}$ because the mapping f^{-1} maps an element of $\{a,b,c\}$ to a **subset** of A. Therefore, in general, we define the inverse as

$$f^{-1} : \{a,b,c\} \to \{\ \emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\ \}$$

This shows that how there can be two inputs that can map to one output.
Think of this example, lets that there are ten pigeons, and we have to put them into nine holes. What does this entail? It means that there is going to be two pigeons that are shoved into the same hole. This is a key idea to understand once we dive deeply into different ideas of cryptography.
This is why we have have output of the inverse being {1,2} or {1, 2, 3}. To show all the possibilites of inputs to get an output of a, b, or c.

In general, the range can contain more elements than the domain. Observe that every element in the range of f^{-1} is itself a set. This is not a problem. An element is an element no matter what it is. It can be a number, or a vector, or a letter, or an apple, or a human, or a set, or a set of vectors, or whatever we want. Now, we can define $f^{-1}(a) = \{1,2\}$ and $f^{-1}(b) = \{3\}$ and $f^{-1}(c) = \emptyset$. This is just a specific mapping of elements to one another.

Final remark on notation: for a set A we can construct a new set that consists of all subsets of A (including the empty set, because the empty set is also a subset). We call this new set the *powerset*

of A and we denote it as 2^A. 2^A is just a weird symbol, which denotes the set of all subsets of A. For example, if $A = \{1,2,3\}$ then $2^A = \{\ \emptyset, \{1\}, \{2\}, \{3\}, \{1,2\}, \{1,3\}, \{2,3\}, \{1,2,3\}\ \}$. This shows all combinations of the elements of A that can be considered a subset.

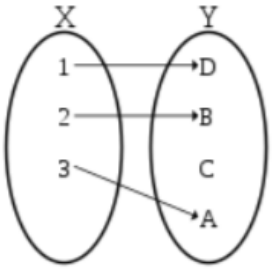
Thus, in general, for a function $f : A \to B$ its inverse has domain and range $f^{-1} : B \to 2^A$.

1.5 Sets, sizes, and restricted functions

The size of a set $A = \{1,2,3,42\}$ is the number of its elements. We denote this by $|A| = 4$. For finite sets the size is a trivial concept, but for infinite sets the size becomes a complicated issue.

The following three restricted types of functions are related to size:

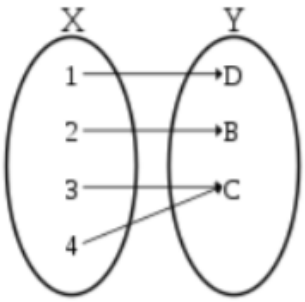
1. $f : A \to B$ is an *injection* if for every $x \neq x'$ we have $f(x) \neq f(x')$. That is, the example of $g(n) = n^2, n \in \mathbb{Z}$ is **not** an injection (because $g(-1) = g(1)$).



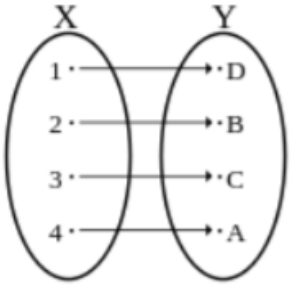
2. $f : A \to B$ is a *surjection* if for every $b \in B$ there exists $a \in A$ such that $f(a) = b$; i.e. B is covered. For example, $g : \mathbb{Z} \to \mathbb{Z}$ and $g(n) = n^2, n \in \mathbb{Z}$ is **not** a surjection because there is no $n \in \mathbb{Z}$ such that $g(n) = -1$. However, the function $h : \mathbb{Z} \to \{0,1,2,3,\dots\}$ where $h(n) = n^2, n \in \mathbb{Z}$ is a surjection.

   Remark on notation: we denote by $g(\mathbb{Z})$ the *image set* of g,

i.e. the set $g(\mathbb{Z})$ consists of all possible outputs of g (and nothing more). Therefore, for every function $f : A \to B$ the new function $F : A \to f(A)$ with $F(x) = f(x)$ for $x \in A$, is a surjection.



3. $f : A \to B$ is a *bijection* if f is injective and surjective. A bijection, sometimes called "one-to-one correspondence", perfectly matches the domain elements A and the range elements B.



What do all of the above have to do with the size of sets?

If for two sets A, B there is an injection $A \to B$ then B cannot be smaller than A, since we can "inject A inside B without collisions" (otherwise there would be collisions).

If there is a surjection $A \to B$ then A cannot be smaller than B (otherwise there would be some element of B not covered).

Finally, a bijection $A \to B$ witnesses equal size A and B.

Summary of notation/symbols we learned so far

$S = \{\quad\}$
$\mathbb{Z}$: the set of all integers
$A \cup B$
$A \cap B$
$A \times B$
A^3
...
$(1,1,1)$
$\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$
111

$|A|$
$f : A \to B$
$f(x) =$ something
function viewed as a set
vector viewed as a function
χ_A characteristic function of A
$f(A)$ image set of f
preimage
f^{-1}
2^A powerset of A
injective
surjective
bijective

Lecture 2

How to parse, interpret, and write rigorous statements?

One goal of this manuscript is to teach how to read/understand and write rigorous statements. Rigor is necessary if we want to remove ambiguities from our expression.

2.1 How to read and write rigorous statements

Every rigorous argument consists of a sequence of inferences on *true statements*. How inference works is obvious/self-evident to anyone. For example, if P implies Q and we know that P is true we can infer Q is also true.

A prerequisite for articulating an argument is first to be able to write down a precise expression about P and Q. This is what we learn in this Lecture.

What is a rigorous/logical statement? A logical statement consists of

i. Logical connectives

ii. Predicates and variables

iii. Quantifiers

and that is all.

Logical connectives

There are three logical operators AND, OR, NOT that "operate" on logical expressions/statements.

> *Every statement evaluates to true or false. We occasionally write* 1 *for "true" and* 0 *for "false".*

For example, say that P and Q are statements that evaluate to true or false.

Then, $P\ AND\ Q$ evaluates to true if and only if both P and Q are true (true = 1). That is:

- $1\ AND\ 1 = 1$
- $0\ AND\ 1 = 0$
- $1\ AND\ 0 = 0$
- $0\ AND\ 0 = 0$

Note that each of P and Q is a statement **and "$P\ AND\ Q$" is also a statement.** It is a statement built from two sub-statements.

$P\ OR\ Q$ evaluates to 1 if and only if any either P or Q or both are 1. That is:

- $0\ OR\ 0 = 0$
- $1\ OR\ 1 = 1$
- $0\ OR\ 1 = 1$
- $1\ OR\ 0 = 1$

Similarly to AND statements, P and Q is a statement **and "$P\ OR\ Q$" is also a statement.** It is a statement built from two sub-statements.

Important: Observe that the statement "I can meet you on Monday or Tuesday" in formal/rigorous terms means I can meet you either on Monday or on Tuesday or I can also meet you in both days. But in everyday life (in the casual) use of "OR" people might not always mean the same. In formal arguments there is no room for ambiguity. "OR" means exactly: one or the other or both.

The logical NOT inverts the truth value of the expression. $NOT(0) = 1$ and $NOT(1) = 0$.

Constant expressions and predicates

Some true/false facts are always true or always false i.e. they are constants. For example, $Q =$"pigs fly". In this case (assuming that pigs don't fly), the truth value of Q is always 0. In rigorous arguments, we do not only have constantly true/false expressions but also things that vary over a model of the world or a model of our system of reasoning if you like. "Things vary" refers to a function whose values are true or false.

True/false-valued functions are called *predicates*.

For example, $P : \{\text{Monday, Tuesday,..., Sunday}\} \to \{0, 1\}$, where $P(Tuesday) = 1$ and $P(x) = 0$ for every x which is a day other than Tuesday (also, we could have written P="today is Tuesday" and we evaluate P on the parameter "today" (think of x as today)).

> *Predicates are just logical expressions which are parameterized – their true/false value is uniquely determined by the parameter(s).*

Hypotheses and conclusions

How to express:

> *IF "some hypothesis" THEN "some conclusion"*?

Understanding the above is a little complicated just because **in rigorous arguments <u>everything</u> is a statement**. The "hypothesis" must

be a statement, the "conclusion" must be a statement, and also the "IF hypothesis THEN conclusion" must be a statement. Statements evaluate to be either true or false. Similar to what we explained before with P and Q in the last section. Think of P referring to the hypothesis and Q referring to the conclusion. We can condense the entire statement to read as "IF P THEN Q". For example, suppose that we express the following casual/real-life statement "IF today is Tuesday THEN we have a lecture for this class". The statement "IF today is Tuesday THEN we have a lecture for this class" consists of two substatements. One being "today is Tuesday" and the other being "we have a lecture for this class". In our model of the world "today is Tuesday" is a predicate. It becomes true/false based on what day of the week it is. "we have a lecture for this class" is another predicate that becomes true/false based on which day of the week it is.

Everything (including the IF...THEN... statement itself) must have some true/false value. That mean What truth values shall we assign to "IF today is Tuesday THEN we have a lecture for this class"? Just for now **my goal is to write a statement that is always true (always means for all days of the week).** When the day is "Tuesday" then "today is Tuesday"=true and "we have a lecture for this class"=true. *We decide define the IF...THEN... to be true if both the hypothesis and the conclusion is true*. Thus, for parameter "Tuesday" we have "IF today is Tuesday THEN we have a lecture for this class"=true.

What happens for any other day of the week? Then, the hypothesis "today is Tuesday" becomes false and same thing for "we have a lecture for this class". *We decide to define IF...THEN... in a way that when the hypothesis is false then no matter what is the conclusion whole statement becomes true*;

i.e. "IF false THEN false"=true and also "IF false THEN true"=true. *The only way for the IF...THEN... to become false is if we*

start from a true hypothesis and we conclude a false conclusion.

To summarize following the "IF P THEN Q" format:

- $IF\ 1\ THEN\ 0 = 0$
- $IF\ 0\ THEN\ 0 = 1$
- $IF\ 0\ THEN\ 1 = 1$
- $IF\ 1\ THEN\ 1 = 1$

We will come back to the use of IF...THEN... since it is a perfectly good way to filter things out.

Is there a way to express "IF P THEN Q" using the AND, OR , NOT connectives? Yes, we can write it as

"IF P THEN Q" = $NOT(P)\ OR\ Q$

which does exactly what we want – if the hypothesis P is false then the whole thing becomes true and the only way to become false is when P is true in which case the value depends on the value of Q.

Since "IF P THEN Q" = $NOT(P)\ \ OR\ \ Q$ then we have that "IF $NOT(Q)$ THEN $NOT(P)$" = $NOT(NOT(Q))\ \ OR\ \ NOT(P)$. But, $NOT(NOT(Q)) = Q$, which means that "IF $NOT(Q)$ THEN $NOT(P)$"= $Q\ OR\ NOT(P)$ ="IF P THEN Q". To conclude:

"IF P THEN Q" = "IF $NOT(Q)$ THEN $NOT(P)$"

This will become very useful when tring to prove statements as true or false later in this course. This is a very important piece of logicial reasoning to understand.

2.2 In a true statement there is no such thing as a free variable

Consider the predicate $P : \{0,1\}^{10} \to \{0,1\}$, where $P(x) = 1$ if the number of 1's in $x \in \{0,1\}^{10}$ is at least 5 and $P(x) = 0$ for every

other x.

Example: x = 1110011111
since x has eight 1's, that means that it has more than five 1's.
thus, $P(x) = 1$ since it meets the requirement.
Example: x = 0001100000
Now x has two 1's, that means that it has less than five 1's.
thus, $P(x) = 0$ since it does not meet the requirement.

Now consider the logical statement:

> IF $P(x) = 1$ then $x \in \{0,1\}^{10}$ must be lexicographically greater than or equal to 0000001111. 0000011111 is the value of a binary string which we are comparing the value of x to. (Lexicographically in this example means that within the string x there are at least five elements that have a value of 1)

The above is just formal jargon for saying that if you have at least five 1's in a given string then this string cannot be the 0000000000 neither 0000000001 and in fact, it must be a string that has higher lexicographic order than 0000001111. This looks like an obvious statement and seems to be obviously true. But one issue. **Everything said above is misleading because what is written is nonsense. The above is <u>not</u> a statement. It <u>cannot</u> be evaluated to true or false.** For **which** x are we evaluating the truth value of "IF $P(x) = 1$ then $x \in \{0,1\}^{10}$ must be lexicographically greater than 0000001111"? For x = 0000000000? For x = 0000000001? For one x? For two x's? For all?

> **Unless we <u>quantify</u> variable x the above statement cannot assume any truth value.**

There are two quantifiers: "for all" and "there exists". We write $\forall$ for "for all" and $\exists$ for "exists". The meaning of "for all" is that

I can conclude that a statement is true only after I substitute every possible value for x and also if I find out that *for each* substitution the statement is true. In contrast "there exists" becomes true if I can find *at least one* value for x where the statement becomes true.

What we should have really written above is the following:

> For every string $x \in \{0,1\}^{10}$, if $P(x) = 1$ then x must be lexicographically greater than 0000001111.

or in more casual (but still rigorous!) terms we could have said:

> For every binary string x of 10 bits, if there are at least 5 many 1's then this string must be lexicographically ahead of 0000001111.

Note that the second statement is as rigorous as the first one. Humans are not computer programs that parse logical statements. When humans talk rigorously they aim to make true but also meaningful statements, written in the most readable form (without losing preciseness).

2.3 The order of quantification: constants and dependencies

What happens if in a logical statement there are many variables? Consider the statement:

> For every human h there is a woman w such that $is\_mother(w,h)$=true.

We assume the usual model for the world and we would like to know if the above statement is true. (Note that in this example, w represents an arbitrary woman and h represents an arbitrary man). Before we do that we must go over possible pairs (w,h). **For which** (w,h) **are we going to test?** "For every h" indicates that we should probably be testing for all possible h's. But how about w? It seems that it

suffices to find one w. But should this be the same for all h? In this case the meaning would be that every human has the same woman as her/his mother. In any case, **we must agree on one** reasonable definition that allows us to evaluate such statements.

We interpret the quantifiers parsing them from left-to-right. In particular, in the above statement, we first choose h and then for each of these h we one (possibly new) w. Since h is quantified by "for all" we must do this for all h and for each of them individually we search for a w that satisfies the predicate $is\_mother(w, h)$.

> *Variables quantified later become a function of (they depend on) the earlier quantified variables. Also, if a variable is quantified earlier then it behaves as a constant for every variable quantified later.*

That is, in the above statement w is a function (depends on) h. To emphasize this we may even write $w(h)$.

Contrast the statement "For every human h there is a woman w such that $is\_mother(w, h)$=true" with the statement:

> There is a woman w for every human h such that $is\_mother(w, h)$=true.

We first choose (appropriately quantified) woman w and then we choose a man. Note that for this statement to be true there must exist one woman w where **the same** woman must be paired with all men. That is, all humans should have this one woman as their mother in order for the above statement to be true. This is not the same at all with the previous statement where each human could be paired with one woman when verifying whether the statement is true.

None of these statements is, of course, the same as

> There is a woman w, there is a human h such that $is\_mother(w, h)$=true.

This statement becomes true just if we are able to find one w and one h where w is the mother of h.

Finally, in order for the following statement to be true, we must have a model of the world where all possible humans have all possible women as their mother.

> For all humans h, all women w it is true that $is\_mother(w, h)$=true.

More details on the meaning of "for all" and "there exists"

We can conclude that the statement we really wanted to make is

> For every human h there is a woman w such that $is\_mother(w, h)$=true.

To simplify the discussion (avoid annoying nuances) let us separate males and females and make the following statement.

> For every man m there is a woman w such that $is\_mother(w, m)$=true.

Suppose that we live in a world where there are only three possible men $\{$Tom, Jack, Peter$\}$ and three possible mothers $\{$Jane, Mary, Kate$\}$.

Here is one way that the above can be true. $is\_mother = \{(Jane, Tom), (Jane, Jack), (Mary, Peter)\}$. We see that it is possible a woman can be a mother of more than one men, which is meaningful.

However, "there is" means "there is at least one". Therefore, the use of the quantifier "there is" (which is the same as "there exists"), leaves the following model as a possibility of the world that makes the statement true:

$$is\_mother = \{(Jane, Tom), (Kate, Tom), (Jane, Jack), (Mary, Peter)\}.$$

Therefore, Tom can have two biological mothers, and the use of the quantifier "there exists" allows such a model of the world to satisfy our statement. And we do not want that. How to write "there exists exactly one"?

$\forall$ man m $\exists$ woman w $\forall$ women w' we have $is\_mother(w, m)$=true
AND
IF $w \neq w'$ THEN $is\_mother(w', m)$=false

Think about the following questions:

Question 1: Why did we write "AND"?

Question 2: What is the role of "IF"?

Question 3: Why does the above enforce the constraint that each man has a unique biological mother?

Question 4: What is the meaning of the statement if we switch the order of quantifiers as "$\forall$ women w' $\forall$ man m $\exists$ woman w"?

Remark: When we introduce rigor we should not lose common sense. We introduce rigor to improve common sense. **Rigor just removes ambiguity from the casual/everyday talk.** What does the above say in layman's terms: "every man has one mother and in fact for every other woman who is different than his first biological mother, then this new woman cannot be his mother at the same time.". This is also equivalent to: for every man there is exactly one woman who is his mother.

2.4 How are things read and written in practice?

Statements have true/false meaning.

For this to happen every variable/parameter that appears in the statement must be quantified (otherwise we are unable to assign true/false meaning).

When we write arguments our goal is to write only true statements. From those, we infer/deduce other statements which are also true (true things help us infer other true things). At the end, we conclude with the statement that we originally wanted to argue that holds true. This process is called a proof (or rigorous argument) and we discuss it in Lecture 3.

How to "read" statements Consider the following statement

> Let $x \in \{0,1\}^{10}$ be a string. Then, lalala...

We said that **every** variable in a statement must be quantified. It is not surprising that the string x is not "free variable". The reason is that "let x..." really means "start with any x...". In everyday logical expression the statement "Let x..." **is the same thing as saying** "For every x...".

All of the following have exactly the same logical content:

> "For all x..." = "Let x..." = "Fix arbitrary x..." = "For any x..." = "Fix x..." = "Fix any x..." = "Consider an x..."

For example:

Theorem: Let ABC be a triangle with right angle at B. Then, $AC^2 = AB^2 + BC^2$.

How to read the above statement? We go over **all** triangles that have a right angle at B and for these triangles the stated length relation holds.

Here is another way to write this using what we have learned:

Theorem (restating Pythagoras theorem): For all triangles ABC, if B is a right angle then $AC^2 = AB^2 + BC^2$.

In fact, this is not bizarre at all. It is the same thing as before written in more detail (e.g. we replaced "let" with "for all").

This is exactly the same statement as before. We go over all possible triangles. If B is not a right angle then "if B is a right angle then $AC^2 = AB^2 + BC^2$" becomes vacuously true because the hypothesis is false. Things become non-trivial when B is a right angle in which case the truthfulness of the statement "if B is a right angle then $AC^2 = AB^2 + BC^2$" depends only on the truthfulness of "$AC^2 = AB^2 + BC^2$".

Finally, we conclude by noting that the following all have the same meaning:

"There is x..." = "For some x..." = "There exists x..." = "Choose one x such that..." = "There is a universal x..."

Lecture 3

How to argue rigorously?

We already fixed some minimal necessary notation and explained how to parse, understand, and write down rigorous statements. Each of these statements evaluates to true or false. Now, we explain how one can establish this. That is, how to argue rigorously.[1]

3.1 How to argue about "for all" and "there exists".

We always prefer to argue constructively. Constructively means we just present a few things (e.g. by constructing them) and then somehow we are done with the argument. I cannot overemphasize enough the importance of arguing in this way. We would be very happy if we were always able to prove things by saying "here is the object that makes my statement true". For example,

> "There is a student in the class who is female"

How to prove this statement? Easy! Just point to a female student.

> *If we manage to exhibit or construct the target object for which we wish to prove existence, then we have formally proved its existence.*

Another example: There is an integer x such that $x > 10$.
How to prove this? Choose $x = 11 > 10$. Thus, there is $x > 10$.

[1] This exposition does not aim to be comprehensive. However, it provides all necessary tools students need to argue rigorously in simple settings – as simple as in Information Security, Cryptography, and Machine Learning.

Things become tricky when the quantifier is "for all" instead of "there exists". Before we attempt to argue directly about all possible objects, we should consider the possibility of proving the same statement by changing "for all" into "there exists". This is not always possible but sometimes we get lucky.

When does "for all" flip over and becomes "there exists"?

Consider the statement: "NOT(for all x, P(x)=true)". How to prove this. One way is to argue that the statement "for all x, P(x)=true" is false and this would have concluded that "NOT(for all x, P(x)=true)" is true.

Here is an alternative way to argue. Since "for all x, P(x)=true" is a statement that we wish to prove false, this means that if we can find at least one x that makes P(x)=false this would formally prove that "for all x, P(x)=true" is a false statement. In other words the following two statements have the same meaning:

"NOT(for all x, P(x)=true)"
= "there exists x, NOT(P(x)=true)"

To make things even more concrete, consider the statement: "It is *not true* that every student in the class is female". How to prove that? This statement has the same meaning as: "There is at least one student in the class who is not female". In order to prove the existence of such a student(to make the statement true), it suffices to point to one such student. Thus, we see that a negation in front of a "for all" quantified statement makes it possible for us to argue constructively.

How to directly argue about "for all"

Suppose that we have to prove correct a statement that starts as: "Let $x \in \{0,1\}^{100}$ be a string of 100 bits. Then, ..."

Then, we start our argument by saying:

Fix an arbitrary $x \in \{0,1\}^{100}$.

From the point on we write the above expression **we do have one concrete object inside our argument**. Although it is difficult to argue about all objects simultaneously, we can select an arbitrary one and do things about that one (because an arbitrary is as good as any other). **So, we have this** x **and we can work with it from this point on!** There is a catch though. Although we have x and we can grab it, massage it, paint it, we can do whatever we want with this x because it exists inside our argument (since we fixed it), **the problem is that we do not know how exactly** x **looks likes**. Think of this as if someone had given you an x but she gave you this x inside a black box. If it happens that in your argument you don't really care about the specifics of x then you may be able to carry the argument without ever opening the box. But in most arguments you have to open the box. Opening the box means **considering cases** about the possible contents of the box. **Your argument should work for all possible contents of the box**. For example,

- If x has ≥ 5 many 1s (for example, let x=11101111) then I will argue that x is a vegetable
- If x has < 5 many 1s(for example, let x=0110000) then I will argue that it is a fruit.

Note that when you branch into cases about x **your cases must cover all possibilities** about x – otherwise you do not have an argument about an **arbitrary** x.

3.2 Example: the party problem

Statement: In a party for every set of 6 people either there are 3 of them who have met before, or 3 of them have never met before.

This is a non-trivial thing to show.
We start by saying:

> *Fix an arbitrary set of 6 people S. From this point on we work with this set S.*

Now, there are a few options of how to show the statement true.
Notice that 6 is a concrete finite number. Therefore, we can do this:

1. Write a computer program(lines of code) that enumerates all possible configurations between 6 people. This means, that we can pair two of them in case they have met and if they have not met then we do not pair them.

2. The same computer program as it enumerates all possible instances using a loop inside this loop is testing whether for each configuration of 6 people it is true that either 3 of them have met before or whether 3 of them they have never met.

3. Thus, we exhaustively test whether the statement is true in each among all possible met/not-met configurations. If at the end the test turns out to be true for all configurations then we conclude that the statement is true.

So, we can write a program and check "every set" of 6 people in an exhaustive way. Is this a formal proof? Absolutely! We can test a "for all" condition by actually going through "all possible" objects. We can do that here because we have 6 people, as opposed to a "generic" integer; e.g. we could not write a program that tests a similar thing for N people.

One last remark: why do you need a computer program? In total there are $\binom{6}{2} = \frac{6!}{2!4!} = \frac{6 \cdot 5}{2} = 15$ possible pairs among 6 people. If we want to count all possible configurations, then each pair might exist or not in the configuration. Thus, there are $2^{15} = 32768$ many

different configurations to check(15 different pairing possibilities). This is why you need a computer program.[2]

What happens if we do not want to write a program or we do not have enough time (e.g. we are not in prison). Here is a way to argue.

Proof. Fix an arbitrary set of 6 people S. From this point on we work with this set S.

We consider two main cases.

- **Case 1:** There exists a person $p \in S$ who has met at least 3 other people $p_1, p_2, p_3 \in S$. Then, we consider further cases.
  - **Subcase 1.1:** There are at least two among p_1, p_2, p_3 that they have met. Say that without loss of generality these are p_1, p_2 and then p, p_1, p_2 are three people that have met.
  - **Subcase 1.2:** None of p_1, p_2, p_3 have met. Then, p_1, p_2, p_3 are three people who have never met.
- **Case 2:** Every person $p \in S$ has met at most 2 other people.
  - **Subcase 2.1:** There exists p who has met 2 people p_1, p_2.
    - **Subcase 2.1.1:** If p_1, p_2 have met each other, then p, p_1, p_2 are three people who have met.
    - **Subcase 2.1.2:** If p_1, p_2 have not met, then let us now consider the people with whom p_1 has met. These must be at most 2 new and one of them is p. Therefore a second one p_3. Note that if p_3 doesn't exist then p_1, p_2 together with anyone except p form three people that have never met. This leaves us with the case p_1 has met with p and p_3. If p_3 has met p_2 then p_1, p_2, p_3 are three people that have met. Similarly, if p_3 has met p. Otherwise, p_3 has not met p and

[2]The only detail missing is that you have to prove that your program correctly enumerates the configurations – but that's easy to argue.

she/he has not met p_2 and thus p, p_2, p_3 are three people that have never met.

– **Subcase 2.2:** Every person has met at most 1 other. This means that there exist three people who have never met (observe this by matching the 6 people).

□

Note that, Case 1 and Case 2 are complementary. This means that Case 1 and Case 2 cover all possibilities. Thus, if we cover all possibilities and in each of them we conclude that the statement is true then this means that the statement is true no matter how the 6 people are related to each other. The same thing holds for Subcase 1.1 and 1.2, and the same thing holds for each of the subcases we are branching into.

You should practice with the above to understand the differences between "there exists at least" and "for all at most", etc. This is an important skill for you to acquire.

3.3 What do we mean by "Statement"?

A statement is something that uses logical connectives, proper quantification, and in general, has a rigorous meaning. We care about statements that are true.

True statements appear with the following names:

- Theorem
- Lemma
- Proposition
- Claim
- Fact

All of the above are the same thing. They all refer to a true statement. Can we write "Lemma" instead of "Theorem"? Yes, we already said that these are just different names, which all refer to "a true statement". We use different names for aesthetic and psychological reasons. Theorems are considered more important than Lemmas, and Lemmas more important than Propositions, Claims, and Facts.

3.4 What is a "Definition"?

A definition is a new name (or a title if you like) that we give to an object or a set of objects. Definitions have nothing to do with true statements. **Definitions are just names and nothing else.**

Devising a definition might be non-trivial. We assign a name to an interesting family of objects. For example, how to define : "efficient search in a database"? We should define "efficiency" in a way, which is:

- relevant to human perception and
- interesting and
- universal enough, but
- not too broad and which also
- enables us to make progress by devising such search procedures.

Definitions are about philosophy and aesthetics. Not just about mathematics.

3.5 "Definitions" vs "True Statements"

You should not confuse "definitions" with "true statements". A true statement has all of its variables quantified (each, either by "for all" and "there exists"). That means for it to re On the other hand, a

definition has one (or more) free parameter, which is the object we wish to define. Everything else must be quantified in a definition.

A little less precise: for example, we may have a definition of what is a "delicious chocolate cake". The fact that we defined "delicious chocolate cake" does not mean that we have in our hands such a cake (we only have its specification). **Making** a "delicious chocolate cake" is different than **explaining** to someone when a chocolate cake can be called "delicious".

Finally, you cannot make a cake and then just put a label "delicious chocolate cake" on your creation, unless you first clearly say (i.e. define) what precisely "delicious chocolate cake" means.

3.6 Example: asymptotic growth of functions

We wish to define the following notion: a function that may have a complicated form, e.g. $f(n) = \frac{\sqrt{2\pi}}{2}n^2 + 1.1n^{1.1}\log_2(n) + 100n$ grows "*roughly* no more than n^2". That is, we want to have a definition that hides visually annoying details about the function and focuses on what matters most.

Here is how to define it:

Definition 1. *We say that* $f : \mathbb{R} \to \mathbb{R}$ *is of order* $g : \mathbb{R} \to \mathbb{R}$ *and we write* $f(n) = O(g(n))$ *if*
there exists $n_0 \in \mathbb{Z}$ *and*
there exists $c > 0, c \in \mathbb{R}$ *such that*
for all $n \geq n_0$
we have $f(n) \leq c \cdot g(n)$

This is a great definition. The "trick" is that it chooses n_0 and c to suppress all lower order terms (I mean g is going to beat f from some constant point and above) and also uses c to ignore the constant in front of the most important term.

How to formally argue that $f(n) = O(n^2)$? How to show the existence of n_0 and c? We show existence by constructing the object (in fact, construct two objects: n_0 and c).

Proof. One thing we know from Calculus is that if we do a monotonicity study on $h(n) = n^2 - \left(1.1n^{1.1}\log_2(n) + 100n\right)$ then there is an n_0 after which h is monotonically increasing and bigger than 0 (i.e. n^2 beats all the lower order terms). By studying the monotonicity of h we find out that for every $n \geq 113$ we have $h(n) > 0$.

Let us now construct n_0 and c.

Choose $n_0 = 113$.

Choose $c = \frac{\sqrt{2\pi}}{2} + 1$.

Now, for all $n \geq 113$ we have:

$$g(n) = cn^2 = \frac{\sqrt{2\pi}}{2}n^2 + \underbrace{n^2}_{\text{which is } \geq 1.1n^{1.1}\log_2(n)+100n, \text{ for all } n\geq 113} \geq f(n). \qquad \square$$

3.7 Example: more on the asymptotic growth of functions

We saw that $f(n) = \frac{\sqrt{2\pi}}{2}n^2 + 1.1n^{1.1}\log_2(n) + 100n$ is "of order at most" n^2, $n \in \mathbb{Z}$. According to our intuition, $f(n)$ is *not* of order n nor of order $n^{1.5}$ etc. In symbols, we would like to show $f(n) \neq O(n)$.

Before we formally argue about this we must first **express clearly what do we really want to show**. Recall Lecture 2.

We want to show that it is *NOT* true that
"there exists $n_0 \in \mathbb{Z}$ and
there exists $c > 0, c \in \mathbb{R}$ such that
for all $n \geq n_0$
we have $f(n) \leq c \cdot n$ "
Recall that $NOT(\forall x,\ P(x)) = \exists x,\ NOT(P(x))$ and it is also true that $NOT(\exists x,\ P(x)) = \forall x,\ NOT(P(x))$.

Therefore, in order to show that $f(n) \neq O(n)$ we have to show that the negation of the definition is true, i.e. "for all $n_0 \in \mathbb{Z}$ and

for all $c > 0, c \in \mathbb{R}$
there exists $n \geq n_0$
such that it is NOT true that $f(n) \leq c \cdot n$ "

Note that $NOT(f(n) \leq c \cdot n)$ is the same as $f(n) > c \cdot n$.
How do we always argue about "for all"? As follows:

Proof. **Fix arbitrary** n_0 and c. It remains to prove the existence of n, which means that we can construct it in any way we like as long as $n \geq n_0$. If we choose $n > c$ then we observe that $cn < n^2$, and we know that $f(n)$ has at least one term which is bigger than n^2. Therefore, given the fixed n_0 and c a good choice for n is any number bigger than c and n_0; i.e. choose any fixed $n > c$ and $n \geq n_0$ and $n \geq 0$. Then, $cn < n^2 < \frac{\sqrt{2\pi}}{2}n^2 + 1.1n^{1.1}\log_2(n) + 100n = f(n)$, which completes the argument. □

Why it was not enough to say "choose $n > c$"? After we fix n_0 and c we really have them as concrete entities inside our proof. But we do not know their values (as if they are inside black-boxes). To satisfy $f(n) > cn$ it suffices to choose $n > c$. But, the negation of the definition of $O(\cdot)$ (note: the dot inside the parenthesis signifies its is a place holder for any parameter) the says that $n \geq n_0$. It is a formality, but still, we have to choose $n \geq n_0$. For example, what if the fixed n_0 and c are such that $c < n_0$? Then, by just choosing $n > c$ we fail to satisfy the requirement that $n \geq n_0$. However, we are free to choose an n as large as we like. Thus, if we choose n greater than the maximum between c and n_0 that simultaneously satisfies everything we had to.

3.8 Some questions on the previous examples

What would have happened if the order of quantification in the definition of $O(\cdot)$ were different? For example, what if it were:

... there exists $n_0 \in \mathbb{Z}$ such that
for all $n \geq n_0$
there exists $c > 0, c \in \mathbb{R}$
such that we have $f(n) \leq c \cdot g(n)$?

Suppose that the above modification defines when a function is $\mathcal{O}(\cdot)$ and compare this to the previous definition of $O(\cdot)$. Then, if $f = O(n)$ is it true that $f = \mathcal{O}(n)$? How about the opposite? Finally, do you think the above definition of $\mathcal{O}(\cdot)$ defines anything reasonable?

3.9 Example: arguing by contradiction

Sometimes we wish to prove that a logical statement P is true and we argue as follows: (i) assume that $NOT(P)$ is true and then (ii) show that we can conclude $false$, which means that our original assumption was not true, i.e. (iii) conclude that $NOT(NOT(P)) = P$ is true.

One thing that puzzles students is "what does it mean to conclude $false$"? It means that we can conclude e.g. (1) $0 = 1$ or (2) that $x > 0$ and $x < 0$ at the same time or (3) $n^2 = -1$ (but a square can never be negative) or in general (4) anything that is $false$.

Here is a prototypical example.

Fact: Fix 10 real numbers $a_1, a_2 \ldots, a_{10} \in \mathbb{R}$ whose sum is $a_1 + a_2 + \cdots + a_{10} \geq 100$. Then, at least one among those numbers a_i has value at least 10.

Proof. In the contrary suppose that the following is true: NOT(there is $a_i \geq 10$). In other words, for every a_i we have that $a_i < 10$. But if for 10 numbers each is less than 10 then these numbers cannot sum up to 100. In symbols we write: $a_1 + a_2 + a_3 + a_4 + a_5 + a_6 + a_7 + a_8 + a_9 + a_{10} < 10 + 10 + \cdots + 10 = 100$, which contradicts the fact that $a_1 + a_2 + \cdots + a_{10} \geq 100$. Therefore, our original assumption is false. Hence, there exists at least one $a_i \geq 10$. □

As we mentioned earlier, statement, fact, claim, theorem, lemma, proposition are all the same thing, reffering to a **true statement.**

3.10 Example: is the set of all integers bigger than the set of non-negative integers?

In Section 1.3 we claimed that injective, surjective, and bijective functions are helpful in order for us to determine the size of sets. As everyone understands these functions are not too useful when the sets are of finite size, e.g. of size 1 or 11 or 42 or 874239875.

What happens if the sets have infinite size? We will see that not all infinities are same big.

For example, let $\mathbb{Z}$ be the set of all integers, i.e. $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ and let $\mathbb{N}$ be the set of non-negative integers, i.e. $\mathbb{N} = \{0, 1, 2, \dots\}$. An injection function(one-to-one function) is a function that preserves distinctness: it never maps distinct elements of its domain to the same element of its codomain At a first glance, it seems that the infinitely many elements in $\mathbb{Z}$ are more than the elements of $\mathbb{N}$. What if we manage to inject $\mathbb{Z}$ inside $\mathbb{N}$ (recall that "inject" means without collisions)? In this sense, we will say that the infinite size of $\mathbb{Z}$ is no bigger than the infinite size of $\mathbb{N}$.

To do that we must show that there exists an injective function $f : \mathbb{Z} \to \mathbb{N}$. How do we always show existence? We construct the object. This function aim Map $f(1) = 1$ and $f(-1) = 2$ and $f(2) = 3$ and $f(-2) = 4$. This does not describe the entire f so we should generalize this rule. Let $n \in \mathbb{Z}$. If $n > 0$ then $f(n) = 2n - 1$. If $n < 0$ then $f(n) = 2|n|$. And $f(0) = 0$.

The above is enough to complete the proof. But, if someone is obsessed with details she may also like to argue formally that there is no collision("no collision" means that no two inputs will map to the same output.) Let us do this (although it might be a little too much

of a detail). Suppose that there exists a value N such that $f(a) = N$ and $f(b) = N$ for $a \neq b$. Then, we branch into cases.
Case $a > 0$ and $b > 0$: then $f(a) = f(b)$ and by the way we constructed f we have $2a - 1 = 2b - 1$ which means that $a = b$, which contradicts that $a \neq b$, which means that it is true:
NOT(there exists a value $N \in \mathbb{N}$ and also there exist $a, b \in \mathbb{Z}$ such that $f(a) = N$ and $f(b) = N$ and $a \neq b$)
, which is the same as
for every value of the function N and every a, b we have $f(a) \neq N$ or $f(b) \neq N$ or $a = b$.
Therefore, if we have that $a \neq b$ the above implies that $f(a) \neq N$ or $f(b) \neq N$, i.e. $f(a) \neq f(b)$.
Similarly, we argue for all the remaining three cases and we leave them as an exercise.

3.11 Example: the set of integers is smaller than the real numbers

The set of real numbers $\mathbb{R}$ is of infinite size but strictly bigger than the infinite size set of integers $\mathbb{Z}$, in the sense that there is no way to inject $\mathbb{R}$ inside $\mathbb{Z}$ without a collision. There must have been repeated number after the injection since the size of set R is bigger than that of set Z. By "collision" we mean that for every injection f there will be two distinct $a \neq b \in \mathbb{R}$ such that $f(a) = f(b)$. Now, we prove this fact using an advanced reasoning technique that is based on contradiction.

First, before we showed that $\mathbb{Z}$ can be injected inside $\mathbb{N}$. The same function we constructed above shows that there is no uncovered element of the $\mathbb{N}$. This means that this function is a bijection. That is, $\mathbb{Z}$ and $\mathbb{N}$ have exactly the same size. Therefore, to show that there is no bijection between $\mathbb{R} \to \mathbb{Z}$ it suffices to show there is no bijection $\mathbb{R} \to \mathbb{N}$.

Theorem: There is no $f : \mathbb{R} \to \mathbb{Z}$ which is a bijection.

(or equivalently: "every function $f : \mathbb{R} \to \mathbb{Z}$ is not a bijection")

Proof. For the sake of contradiction suppose that there exists a bijection $\mathbb{R} \to \mathbb{N}$ which is the same as a bijection $f : \mathbb{N} \to \mathbb{R}$.

This means that we can enumerate all real numbers. $f(0) =$ some real number, $f(1) =$ another real number, and so on. Now, let us consider the real numbers in decimal notation. For example, 5331.123123 or a real number with infinitely many decimal digits e.g. $\sqrt{2}$. Among all real numbers let us focus only on those with integer part 0. For example, 0.238345 or 0.69340122 or $\sqrt{2} - 1 = 0.41421356237\ldots$. Since we have a mapping from $\mathbb{N}$ we can list all these numbers in order. For example, we have a program here that construct a new real number using an interger input,

$$\begin{aligned} 11 &\overset{f}{\mapsto} 0.987189729 \\ 1023 &\mapsto 0.32190382 \\ 1024 &\mapsto 0.10000000 \\ 1032 &\mapsto 0.111100000 \\ &\vdots \end{aligned}$$

(In the above f it happened that from 12 to 1022 every real number, $f(12), f(13), \ldots$, does not start with 0.. This is why there is a gap.)

We look at the above ordering (according to f we assumed it exists) and construct a new real number X. This number has decimal part that also starts with 0. The first decimal digit of X is different than the decimal digit of the first number in the above order. Since the first number that starts with $0.\ldots$ has id 11, i.e. $f(11) = 0.987189729$, then since the first decimal digit is 9 we can choose as the first decimal digit of X a digit different than 9. For example, choose 0.

That is, so far we have constructed $X = 0.0$.

The second digit is different than the second digit of the second number in the order. Since the number is 0.32190382 we must choose a digit different than 2. Let this be 3. That is, so far $X = 0.03$. Similarly, for the remaining digits of X:

$$\begin{aligned} 11 &\mapsto 0.\boxed{9}87189729 \\ 1023 &\mapsto 0.3\boxed{2}190382 \\ 1024 &\mapsto 0.10\boxed{0}00000 \\ 1032 &\mapsto 0.111\boxed{1}00000 \end{aligned}$$

Then, X starts with $X = 0.0312\ldots$.

We keep constructing a number whose i-th decimal digit is different than the i-th decimal digit of the i-th number in our list.

The critical observation, which completes the argument, is that the number X we are constructing is a real number. **Therefore:** *X itself must be somewhere in the list induced by f.* But, f is a bijection, giving each real number an integer ID. That is, there exists $n \in \mathbb{N}$ such that $f(n) = X$ and let this n be the N-th number in the list of reals of the form $0.xxx$. **Then:** the N-th number in the list (which is X) must have its N-th digit different than itself! This is how we constructed X. Therefore, the constructed X cannot possibly exist. What did we use in order to construct (the non-existent X)? We only used the assumption that there exists a bijection f (everything else is well-defined). This implies that our single assumption about the existence of f cannot possibly be true. Therefore, f does not exist (which is what the statement in this theorem says). □

3.12 Example: a system, which is not perfectly secure

Here is a definition of a private-key encryption system.

Definition 2. *A* Private-Key Encryption system *consists of three algorithms* (Key, Enc, Dec)*, which are defined to work for inputs of every length related to the parameter* $n \in \mathbb{Z}$ *and* $n > 0$*. That is,* $Key : \{0,1\}^n \to \{0,1\}^n$*,* $Enc : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$*, and* $Dec : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$*.*

We think of the inputs and outputs of the above algorithms as follows. Algorithm *Key* takes n independent and uniform random bits as input and outputs a key k, where k is also of length n. The algorithm $Enc(k, m)$ takes as input an n-bit long key k and an n-bit long message m and outputs a ciphertext c also of length n. Finally, $Dec(k, c)$ takes as input a key k and a ciphertext c and outputs the decryption to a message.

We have two requirements about a Private-Key Encryption system. First to be correct and second to be secure.

Definition 3. *A Private-Key Encryption system* (Key, Enc, Dec)*, which is defined to work for inputs of every length related to the parameter* $n \in \mathbb{Z}$ *and* $n > 0$*, is* correct *if for every* $m \in \{0,1\}^n$ *and every* $r \in \{0,1\}^n$*:*

$$k = Key(r);\ c = Enc(k, m);\ m' = Dec(k, c) \implies m = m'$$

Security is defined through the following *security experiment*.

Definition 4. *The security experiment is a probabilistic game between a Challenger and an Adversary. The Challenger is a function with the following specified behavior, whereas the Adversary is an* arbitrary *function (i.e. the definition is for all adversaries – read this as: no matter how malicious they are).*
Fix an arbitrary PriKey Encryption (Key, Enc, Dec) *and fix an arbitrary positive* $n \in \mathbb{Z}$*. The Adversary chooses two messages* $m_0, m_1 \in \{0,1\}^n$ *and passes them to the Challenger. Then, the Challenger (without revealing what he is doing to the Adversary) does the following:*
(i) samples a uniformly random string from $r \in \{0,1\}^n$ *and generates*

$k = Key(r)$;
(ii) chooses between m_0 and m_1 by sampling uniformly a bit $b \in \{0,1\}$;
(iii) encrypts $c = Enc(k, m_b)$;
and (iv) passes the ciphertext c to the Adversary.
Finally, the adversary (who originally chose herself m_0, m_1 and is presented with c) guesses a $b' \in \{0,1\}$.

We say that the Adversary wins *if $b = b'$.*

We say that (Key, Enc, Dec) is secure *if* $\Pr[Adversary\ wins] \leq \frac{1}{2}$.

Note that in the above definition the Adversary is quantified after the PriKey Encryption system. This order of quantification means that m_0, m_1 are chosen as a function of the system the Adversary is trying to break.

Here is a statement we wish to show true.

Lemma: Consider a Private-Key Encryption system where the encryption function ignores the key and outputs the plaintext in the ciphertext. That is, for every $n \in \mathbb{Z}^+$ we have that $Key(r) = r$, $Enc(k, m) = m$, and $Dec(k, c) = c$ for all $k, m, c \in \{0,1\}^n$. Then, this (Key, Enc, Dec) is not secure.

This is an obvious statement. There is some minimal technical work to formally argue about this. "Security" is not some abstract notion, but rather exactly what we defined above it to be secure.

Proof. By negating the security specifications (definition) we obtain: there exists $n \in \mathbb{Z}^+$ and there exists an Adversary such that $\Pr[\text{Adversary wins}] > \frac{1}{2}$. We show existence by constructing things. Therefore, we have to **construct an n and an Adversary**. Choose $n = 1$, and the Adversary chooses two single-bit messages, $m_0 = 0$ and $m_1 = 1$. Now, if the ciphertext the Challenger sends to the Ad-

versary is $c = 0$ then the Adversary outputs $b' = 0$, otherwise, it outputs $b' = 1$. By simply inspecting the way this (Key, Enc, Dec) system works it is immediate that the Adversary is always (with probability 100%) winning and therefore $\Pr[\text{Adversary wins}] = 1 > \frac{1}{2}$. Therefore, (Key, Enc, Dec) is not secure. □