Discrete Math Lecture 1

1 Course logistics and policies

Kindly see the syllabus posted on Canvas for a full description. **It is important that you read the syllabus document carefully.**

2 Why take this course?

~~Because you have to in order to get your degree.~~

> Beauty is the first test: there is no permanent place in the world for ugly mathematics.
>
> - G.H.Hardy

Mathematics can be extremely beautiful. I hope you will experience some of this beauty. One of the things I really hope you will take away from this class is an appreciation of how surprising some of the most simple ideas which we will cover in this course can be, as well as how powerful some the techniques are. But moving on to more mundane matters...why is learning any of this *useful* to you.

Discrete math is the study of discrete mathematical structures, as opposed to more continuous stuff. What exactly this means might not make a huge amount of sense at the moment; indeed it is rather vague. But let's say that by discrete things, we mean things that we can count. By count, we mean count using the natural numbers: $1, 2, 3, \ldots$. This notion of things that are *countable* versus things that are *uncountable* can be formalized; in particular the real numbers, which we would say is a continuous rather than a discrete structure, cannot be counted in this way. But, in this course, we won't get into the details about why this is.

Suffice to say, we will be studying things that can be counted. But why is this of any relevance to you? Well, in lots of problems that we want to solve, especially in computer science, it is useful to model things in terms of these discrete structures. Here are a few examples of how some of the most important things you will learn in this course have major applications in computer science.

- **Induction and recursion:** Recursion is the idea of describing some thing in terms of smaller versions of the same thing. It is a concept that comes up repeatedly in computer science. Often a recursive algorithm is much cleaner and more understandable than an iterative one. Dynamic programming, a basic algorithmic technique that you will see in your algorithms class, intrinsically involves recursion. Lots of data structures in computer science are defined recursively. Arrays, lists and trees are a few such data structures.

  Induction is a powerful tool to prove properties of such structures. It is also used to prove facts about code written in both iterative and functional programming languages. Why should we care about code being provably correct? Well, code in actual applications can be large and complex and so it is hard to be sure that the code is doing exactly what you think it is doing. This can be disastrous in certain critical applications.

  One example of a real-world failure due to code not being provably correct is the case of the Therac-25 radiation therapy machine. The Therac-25 was a medical device used for radiation therapy treatment in the 1980s. It had two earlier models, the Therac-6 and Therac-20, which

were not computer-controlled and had a good safety record.

However, the Therac-25 incorporated a computer-controlled system to administer radiation doses to patients. Due to a lack of rigorous testing and insufficient software verification, the machine had several software-related failures that resulted in massive overdoses of radiation being delivered to patients. These overdoses caused severe injuries and, in some cases, death.

The key issues with the Therac-25's software included race conditions (where multiple processes are modifying data and the output depends on the order in which modifications occur), overflow errors and other software bugs that could allow the machine to deliver radiation at much higher levels than intended. These software flaws were not caught during testing because the software was not provably correct, meaning there was no formal verification or mathematical proof of its correctness.

The Therac-25 incidents serve as a stark example of how a failure to ensure the provable correctness of software in critical systems can have devastating real-world consequences.

It's very likely that at some point or the other we all will be at the mercy of critical code written by you and your colleagues, so it's worth learning the tools to reason about your code and guarantee what its behaviour will be.

- **Graphs:** Graphs are a very useful model for all sorts of systems and networks. They often end up being useful in modelling things that you wouldn't immediately think of as having a graph structure. For example, let's say you are analyzing a game. It might be useful to model the various game states as the vertices of a graph. For example, in a board game, a game state might be the configuration of the pieces on the board. The players' moves might be modeled as edges that take you from one game state to another. Some game states represent a position where one of the players has won the game. Suppose you want to find a way to win as fast as possible. This can now be viewed as a shortest path problem on the graph we just described: we want to find short paths from the initial state to a winning state for us (it is actually more complex than this, it also matters what the other players do, but ignore this for the moment). You can use well known shortest path algorithms to solve this problem.

  Or suppose you want to analyze a social network on some social media platform. You want some way to understand properties of this network. Which people are particularly influential? How densely connected is the network? Are there a few critical people who are crucial to the connectivity of the network? We can model the people as vertices and the connections between them as edges. You can now leverage a huge amount of results from graph theory to reason about the behaviour and structure of this network. And the most powerful thing is that your reasoning is not limited to this social network. If a similar kind of graph structure arises in a different application, say of processors sending data to each other, you can still use results you proved about this kind of graph. This is the power of *abstraction* that is a major reason why mathematical models can be so useful: instead of reasoning about your individual network for every single application, we can at once reason about a huge class of objects that can all be modeled as an abstract mathematical object (a graph) and we can leverage a huge repository of mathematical knowledge that exists about these abstract models (in this case graphs) to deduce things about our individual application.

  Graphs in particular have a habit of popping up in places where you really wouldn't expect them to. There are graphical models in causal inference that have applications in machine learning. Here variables representing events are treated as vertices and edges represent causal

relationships between events. You will no doubt end up modelling all sorts of things as graphs in the future.

- **Discrete probability:** The applications of probability are just way too many to mention. Probability is basically inescapable, even if you are *not* a computer scientist. Randomization is a hugely useful algorithmic tool; you will see algorithmic applications of randomization throughout your coursework. But let's consider another very simple example. Often we want to sample an object from some collection of objects at random. For example when generating an open world game, you might want to do some procedural generation of the landscape. This very likely involves some amount of randomization. Maybe you have a bunch of assets: models of trees, houses, factories and so on. You want to randomly select the next object placed in the game world. Maybe you want there to be some *correlation*: if you have already placed a bunch of trees and shrubs at some location, you probably want to increase the likelihood of the next object to be something like a tree rather than a big industrial building.

  Consider an even simpler example: let's say you are implementing an online card game and want to generate a perfectly random deck of cards. Well, you can actually just randomly pick the first card in the deck, then pick the second card completely at random from the remaining 51 cards and so on. This is the same as picking one particular ordering of all 52 cards out of all possible orderings. Why? You will be able to reason about such things after learning a bit more about probability!

There is actually a rather deep seated relationship between math and cs. There is even a parallel between programs and proofs, i.e., there is a way to views proofs as programs and programs as proofs. This idea can be used to try to create provably correct codes, and in the other direction to generate proofs of mathematical statements via code. We will not get into details of this stuff, but you can look up the Curry-Howard correspondence if you are interested.

3 Flawed reasoning

Reasoning is something we do in day to day life all the time. However we want to be a bit more precise and formal about our reasoning when we do mathematics, in order to make sure the conclusions we arrive at are correct. Today we are going to talk a little bit about formal reasoning. But first let's look at an example where informal reasoning can very easily go wrong. We just talked about probability. In fact, probability is topic that comes up all the time in everyday life, but we humans do not generally have very good intuitions about probability.

3.1 A paradox!

The overall Covid-19 mortality rate (at some point) was higher in Italy than China. At the same time, in every single age bracket, the Covid-19 mortality rate was lower in Italy than in China.

This is an example of Simpson's paradox. This paper discusses this particular example. How is this even possible? We will study some probability later in the course and you can try to figure it out by yourself. Hint: it has something to do with the widely different demographics of the two countries.

But, as we have just seen, reasoning can be trickier than we think. You will see throughout this course that just because something fits into a pattern doesn't mean it is true (take a look at this video). Just because something sounds plausible doesn't mean we can trust it entirely. To prevent ourselves from making mistakes in reasoning, we want to have a system of reasoning that guarantees the correctness of the conclusions we reach.

Look at the lecture 1 handout. We will discuss the examples from there later in section 11.

4 Logic

Mathematical logic is a rich and deep field of study in and of itself. We won't dive very deep into this subject, we do want to have a language which allows us to express mathematical statements precisely and prove things about them. Mathematical logic will be used for this purpose.

Definition 1 (Proposition)**.** *The basic element of logic is the proposition. A proposition is a statement that is either true or false.*

We deal in propositions in our day-to day lives all the time. For example "All horses are brown" is a proposition. So is " Chicago is in Illinois". But a statement like "Help!" is not. All mathematical results will be a statement that's either true or false, and thus a proposition. Even something as simple as $2+2=4$ is a proposition: the statement asserts that the addition of 2 and 2 is equivalent to 4. This happens to be true. But $2+2=5$ is also a proposition—one which is false. It is also useful to keep in mind that in computer science the value 0 is often used to denote false, and 1 to denote true.

Definition 2 (Theorem)**.** *We call particularly important propositions that have been proven to be true, theorems.*

Definition 3 (Lemma)**.** *Less important propositions are called lemmas. We can think of lemmas as propositions that we generally don't really care about except in the service of proving more important theorems.*

Definition 4 (Corollary)**.** *A corollary is a proposition that follows immediately as a consequence of a theorem.*

Definition 5 (Conjecture)**.** *If we don't know if a statement is true or false, but we suspect it is true, we call it a conjecture.*

We can combine several propositions together to create more interesting and complex propositions. This is done by using logical connectives, which modify the meaning of the propositions they combine in various ways.

5 Logical connectors

We can use logical connectors to build more complex compound propositions out of simple propositions. We do this in day to day conversation as well. I might say "I teach at UChicago and I like ice cream". What am I doing here? I'm just asserting that both propositions: "I teach at UChicago" and "I like ice cream", though admittedly unrelated, are true. I combine the two propositions using 'and'.

Definition 6 (Conjunction (AND $\wedge$))**.** *$p \wedge q$, is a proposition that is true if and only if both of p and q are true.*

Similarly we can combine propositions using or. I might say: "I am willing to go to a movie on Friday night or Saturday night." Normally this means I can go on either of the two nights, but not that I can go on both nights. However the latter is the meaning we will assign to 'OR' in a mathematical context.

Definition 7 (Disjunction (OR $\vee$))**.** *$p \vee q$, is a proposition that is true if and only if at least one of p and q are true.*

Again, I stress that OR in mathematics is a bit different than in day-to-day usage. Sometimes in everyday usage when people use OR they mean an exclusive OR: that is exactly one of the two statements holds. For example, if someone asks you to go to get them apples or oranges from the supermarket, they probably don't want you to buy them both! But in mathematics OR includes the possibility of both options being true.

Excercise 1. *We can actually define the exclusive OR or XOR* ($\oplus$) *connective as below*

$$p \oplus q = (\neg p \wedge q) \vee (p \wedge \neg q)$$

Explain in words why this definition makes sense.

Finally we have negation. We might say "I am not six foot tall" which is exactly the same as asserting that the proposition "I am six foot tall" is false.

Definition 8 (Negation (NOT $\neg$))**.** *The negation of a proposition p, written as $\neg p$ is true if and only if p is false.*

Negation also can be a little tricky to interpret when applied to natural language statements. Consider the proposition: "Hot coffee is Ishan's favourite beverage". Is its negation "Cold coffee is Ishan's favourite beverage".

No! The correct negation is "Hot coffee is not Ishan's favourite beverage." Remember that $\neg p$ should be false whenever p is true and true whenever p is false. Another useful way to think about it is that either p or $\neg p$ must always be true no matter what. This reveals why "Cold coffee is Ishan's favourite beverage" is not the correct negation: what if my favourite beverage is lemonade? Then "Hot coffee is Ishan's favourite beverage" and "Cold coffee is Ishan's favourite beverage" would both be false.

More succinctly

$$p \vee \neg p = 1$$

That is to say no matter what proposition we are talking about, either the proposition or it's negation has to be true.

Also,

$$p \wedge \neg p = 0$$

That is to say both a proposition and its negation cannot be true.

It is useful to remember that $\neg$ has the greatest priority when interpreting expressions, thus $\neg p \vee q$ means $(\neg p) \vee q$ rather than $\neg(p \vee q)$.

6 De Morgan's Laws

De Morgan's laws tell us how to negate the propositions $p \wedge q$ and $p \vee q$

Theorem 1 (De Morgan's Laws)**.**

$$\neg(p \vee q) = \neg p \wedge \neg q$$

$$\neg(p \wedge q) = \neg p \vee \neg q$$

This is a good time to address what two propositions being equivalent means.

Definition 9. *Two propositions p and q are equivalent (which we will write as $p = q$ or $p \iff q$) if either both are true or both are false.*

So, De Morgan's law tells us that no matter what the value of p and q, either both $\neg(p \vee q)$ and $\neg p \wedge \neg q$ are true or they are both false.

We do this in natural language as well. We say things like "You will pass the class if and only if you study". That means if you study you will definitely pass the class, but if you don't study then you definitely won't! In the language of equivalence, you studying and you passing the class are equivalent, which sounds somewhat encouraging!

Excercise 2. *Argue that $p = q$ is the same as $(p \wedge q) \vee (\neg p \wedge \neg q)$.*

Excercise 3. *Define equivalence of propositions only in terms of negation and the XOR connective.*

7 If, then (Implications)

We talked about combining simple propositions using and, or and not. In day to day life we also often say that if one proposition is true, then another proposition must be true. For example I might say "If it is cold, then I will wear a jacket." Implication is one of the most useful and common way of combining propositions. It is something that you must understand well to be able to understand almost any mathematical result. Since implication in mathematics is also used quite differently from everyday usage, it is important to make sure you understand the following definition.

Definition 10 (Implication)**.** *We say p implies q ($p \implies q$) if p being true guarantees that q will be true.*

In the above definition if p is false then q could be either true or false. This point is important and is a common source of confusion. When I say "If it is cold, I will wear a jacket", I might intend to mean not only that I will wear a jacket if it is cold, but that I will not wear one if it is not cold. But in our mathematical language the statement "If it is cold, I will wear a jacket" says nothing about what I will do if it is not cold.

$p \implies q$ means that either p is false, in which case we are guaranteed nothing, or else p is true and in that case q must also be true. If p is false, then $p \implies q$ is always true. Thus the proposition "If 3 is even then 2+2=5" is actually true! That is because 3 is not even. This sort of proposition that is true because the if part is just false is said to be vacuously true.

Excercise 4. *Write $p \implies q$ using only the basic three logical connectives.*

Excercise 5. *Now use the result of the previous exercise to argue that the negation of $p \implies q$ is $p \wedge (\neg q)$.*

The above fact is very useful when you have to negate implications. Essentially, it is just saying that the only way for an implication to be false is if the premise is true but the conclusion is false. For example, consider the statement: "'If tomatoes are tasty then pizza is tasty." The negation is: "tomatoes are tasty and pizza is not tasty".

8 If and only if

If I really wanted to say "if it is cold then I will wear a jacket and if it is not cold then I will not wear a jacket", in our mathematical language I should say I will "wear a jacket *if and only if* it is cold". Note that when writing statements in words, we often abbreviate "if and only if" to "iff". $p \implies q$ corresponds to "If p, then q", while $p \iff q$ corresponds to "p if and only if q".

A brief note on language: necessary and sufficient Often the words necessary and sufficient are used in mathematics. If $p \implies q$ then we say p is sufficient for q, because knowing p is enough to conclude q. Perhaps slightly less intuitively if $p \implies q$ we say that q is necessary for p. This is because if p is true then q must necessarily be true; p can't be true without q being true. As an example: For x to be greater than 10 it is sufficient that x is greater than 15 because $(x > 15) \implies (x > 10)$. Similarly, for x to be greater than 10 it is necessary that x be greater than 5 because $(x > 10) \implies (x > 5)$. If p is both necessary and sufficient for q then in fact $p \iff q$; the two propositions are equivalent.

This is actually a rather important and useful fact when it comes to proving an if and only if statement: $p \iff q$ being true is the same as $(p \implies q)$ and $(q \implies p)$ both being true. **To prove an if and only if you need to prove the implication in both directions.** So if I want to 'prove n is even $\iff n+1$ is odd", I should prove that if n is even then $n+1$ is odd *and also that* if $n+1$ is odd, then n is even.

Once again, it's important to remember that in everyday life "if" and "if and only if" are often used interchangeably. When someone says: "If you pay me \$100 I'll do your chores this week" they probably mean: " I will do your chores this week if and only if you pay me \$100". That is if you pay them they will do your chores, and if you don't pay them they definitely won't! But in mathematics implication (if) is *very* different from if and only if. For example, the statement "If n is even then n is a natural number" is true but " n is even iff n is a natural number" is false!

9 The contrapositive

Consider the implication $p \implies q$. Its *converse* is $q \implies p$ and its inverse is $\neg p \implies \neg q$. An implication is not, in general, equivalent to its converse or inverse. We cannot conclude either of these implications even if we know $p \implies q$ is true. For example: "If it is a Sunday, then it is a holiday". This does not mean that "If it is a holiday then it must be Sunday", nor does it guarantee that " If it is not a Sunday, then it is not a holiday", both of these statements are clearly false!

However, and this is a very useful fact, **an implication is equivalent to its *contrapositive***. The contrapositive of $p \implies q$ is $\neg q \implies \neg p$. Why is that? Well, $p \implies q$ guarantees to us that if p is true then q must also be true. It can't be the case that p is true and q is false. But if we know that q is indeed false, then p could not have been true; it must have been false.

This is useful because **if you have to prove a statement of the form $p \implies q$, you can instead prove $\neg q \implies \neg p$**, which might be easier. Let's see an example:

Proposition 1. *For all real numbers x, if $5x^5 - 3x^4 + 4x^3 - 3x^2 + x - 1 \geq 0$, then $x \geq 0$*

Proof. If we want to prove that

$$(5x^5 - 3x^4 + 4x^3 - 3x^2 + x - 1 \geq 0) \implies (x \geq 0)$$

it seems a bit hard to proceed. How do we handle this messy hypothesis $5x^5-3x^4+4x^3-3x^2+x-1 \geq 0$. However, we can instead try to prove the contrapositive of the statement, which is equivalent to proving the statement itself. The contrapositive is

$$\neg(x \geq 0) \implies \neg(5x^5 - 3x^4 + 4x^3 - 3x^2 + x - 1 \geq 0)$$

Or

$$(x < 0) \implies (5x^5 - 3x^4 + 4x^3 - 3x^2 + x - 1 < 0)$$

This is helpful because at least now our hypothesis is simple, we start from the fact that $x < 0$. We notice that if $x < 0$, then $5x^5$, $-3x^4$, $4x^3$, $3x^2$, x and -1 are all < 0 and the expression $5x^5 - 3x^4 + 4x^3 - 3x^2 + x - 1$ is just the sum of all these negative things. Thus, $5x^5 - 3x^4 + 4x^3 - 3x^2 + x - 1 < 0$, which concludes the proof[1]. □

Notice how much more straight forward it was to prove the contrapositive of the desired statement! We will see more examples of this kind of proof as well as many other different kinds of proof in the next class.

10 Rules of Inference

Just like we reason in every day life using some common-sense rules of reasoning, we will use some rules of inference to reason about mathematical propositions. Most of these rules parallel commonsense rules of reasoning and we won't list them all here. For example it is very intuitive to reason like

- I know that if it rains the ground will be wet.
- I see that it is raining.
- Thus the ground must be wet.

This corresponds to a formal inference rule: $p \implies q$ and p together allow us to conclude q.

Rules of inference are often written in the form:

$$\frac{a, a_2 \dots a_n}{b}$$

This rule says that if I know that statements $a, a_2 \dots a_n$ are valid, then I can conclude that b is also valid. A statement a_i is valid if we already know it beforehand (i.e. it is a known, given fact) or if we've shown it at some earlier point in the proof. The reason for this is that since we were able to get to a_i at some point earlier in the proof, we must have obtained it by following our proof rules. Therefore, if we stopped at the point that we got a_i we would have a proof of a_i.

We already mentioned the very natural inference rule:

$$\frac{p \implies q, p}{q}$$

All this says is that if you know p is true and p implies q, then q is true. Also, as we discussed already, if an implication is true, so is its contrapositive. So if we know that $p \implies q$, you also know that its contrapositive,
$lnotq \implies \neg p$ is true. So if you further know that q is false (or in other words $\neg q$ is true), you can conclude that $\neg p$ must be true. This leads us to the useful rule

$$\frac{p \implies q, \neg q}{\neg p}$$

For example

- If I run 10 miles on a given day, I am tired at the end of that day.
- I am not tired at the end of today.

[1] Customarily, we often put a small hollow square to denote the end of a proof. If you want to be extra fancy you can instead write Q.E.D., which is short for the latin phrase "quod erat demonstrandum" which in plain English means: "which was to be demonstrated.

- So, I did not run 10 miles today.

On the other hand, let's now see two fake rules of inference. The following two 'rules' are **not** valid rules of inference

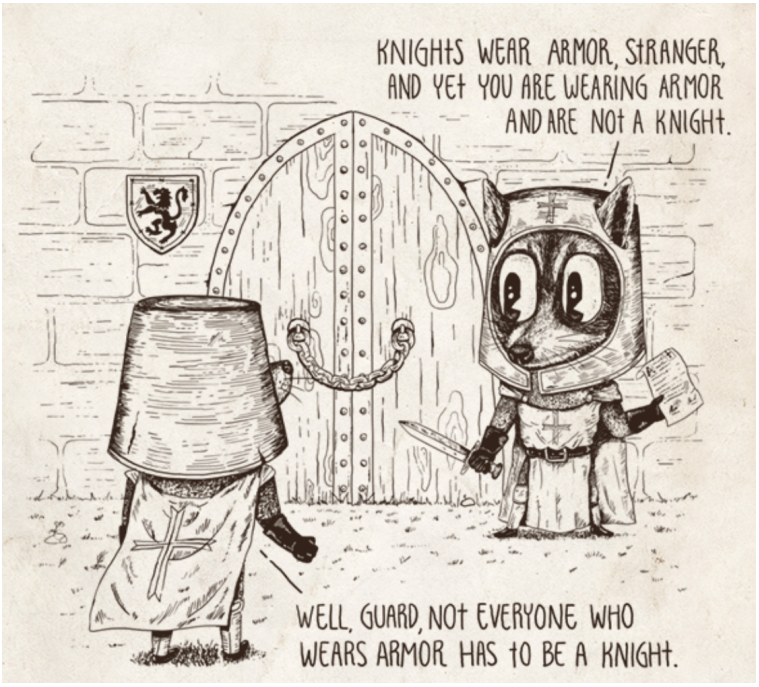
$$\frac{p \implies q, \neg p}{\neg q}$$

$$\frac{p \implies q, q}{p}$$

The examples in the handout result from applying these two fake 'rules'. Let's discuss those examples now.

11 Common fallacies revisited

11.1 Guard's dilemma



This is a castle where knights must wear armor at all times.
Guard's dilemma:

1. It is a rule in this castle that knight's must wear armor at all times.
2. The person in front of me is not a knight.
3. Thus this person cannot wear armor.

Let

$$p = \text{person is a knight.}$$

$$q = \text{person wears armor.}$$

The guard knows $p \implies q$ and he observes $\neg q$. However this does not imply $\neg p$. The fact that knights wear armor does not mean that others cannot wear armor.

11.2 How to test a scientific theory?

A scientist reasons as follows:

1. The theory of gravity predicts that the apple will fall down from the tree.
2. I observed that the apple fell down from the tree.
3. Thus, the theory of gravity is correct.

This is not correct reasoning. Let

$$p = \text{Theory of gravity is true.}$$

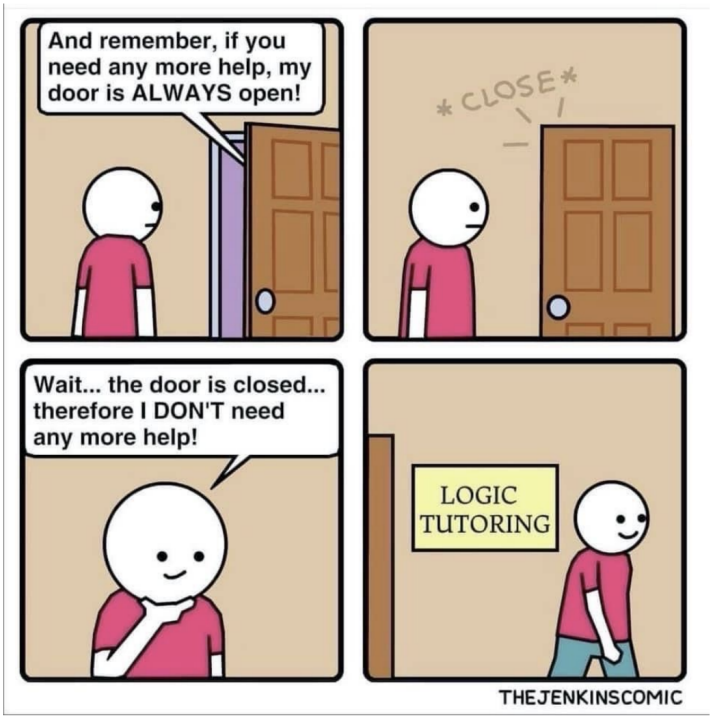
$$q = \text{Apple will fall down.}$$

The scientist knows that the theory predicts that an apple will fall down, i.e., $p \implies q$. The apple is observed to fall down, so we know q is true. But this does not imply p.

If a scientist has a theory t and it predicts result r in some experiment, they can run the experiment and see whether r actually is true or not. If r is false then the theory can be abandoned because the theory predicted it, that is $t \implies r$ and r was found to be false so $\neg r$ is true. Now we can use the rule about the contrapositive: $t \implies r$ is the same as $\neg r \implies \neg t$, so we can conclude $\neg t$, that is the theory is false.

This is why in real life scientists want theories that are *falsifiable*. This means that the theory makes predictions which can be tested and found to be false. That would disprove the theory. Scientists have more trust in theories that make a lot of falsifiable claims and are not falsified, but ultimately science proceeds by disproving false theories rather than proving true ones.

11.3 Office hours for logic class



This is a slightly weird one. The reasoning is actually *correct*, if the professor was being literal in their statement: "If you need any help, my door is open." and we accept that this claim by the professor is true. Let

$$p = \text{The student needs help.}$$

$$q = \text{The professor's door is open.}$$

The professor claimed $p \implies q$. The student observes the door is closed, that is $\not q$. By the same reasoning using the contrapositive as in the previous example, it is legitimate to conclude $\not p$, that is the student does not need more help.

But clearly this is ridiculous; how can the professor's decision about closing the door determine if the student needs more help? The issue lies in the usage of language; the professor just meant that if the student needed more help, then they would provide help. But the meaning of the statement "If you need any help, my door is open", using our definition of the implication, is that if the student does need more help then literally the professor's door is guaranteed to be open, which is false. Another way to look at this is to examine the contrapositive of the professor's statement: "If my door is closed you don't need more help". This makes it clearer that this statement must actually be false: closing a door can't make the student not be in need of help!

11.4 Valid and invalid inference for implications

The following table might be useful (you can ignore the weird names for the rules here). The two rules at the top are correct while the ones at the bottom are invalid. Also note that $\therefore$ is often used in math as shorthand for "therefore".

Affirming the Antecedent (Modus Ponens)	Denying the Consequent (Modus Tollens)
If p, then q. p $\therefore$ q	If p, then q. not q $\therefore$ not p
Affirming the Consequent (fallacy of the converse)	**Denying the Antecedent (fallacy of the inverse)**
If p, then q. q $\therefore$ p	If p, then q. not p $\therefore$ not q

12 Proof

> Obvious is the most dangerous word in mathematics.
>
> - Eric Temple Bell

> I am allowed to use plain English because everybody knows that I could use mathematical logic if I chose.
>
> -Bertrand Russell

Definition 11 (Proof)**.** *A proof is a a sequence of applications of various rules of inference to propositions that are known to be true, till we arrive at the proposition that is the desired conclusion.*

Mathematicians in general write most of their proofs in natural language instead of completely formal language and we will be doing the same throughout the rest of the course. This is because we want our proofs to be understandable to humans, and humans are much better at understanding natural language rather than formal language. Of course, this makes the question of whether a piece of reasoning constitutes a good enough proof subjective. Proof writing is a skill that is best (and to my knowledge only) developed through practice. While writing proofs you may often find yourself wondering if what you have written constitutes a proof. If you skip too many steps of reasoning, your proof may not be convincing; on the other hand including far too many simple steps of reasoning may make the proof incredibly tedious and hard to read; it will also take ages to write! Indeed what is a good proof depends on context. While learning to write proofs, such as in this course, I would encourage you to rather err on the side of writing out more steps of reasoning; however if writing in a mathematics journal for professional readers, that level of detail would be inappropriate. A good rule of thumb that will stay true far beyond this course is:

Write your proof with enough detail so that your peers can understand it.

Another good rule is:

When in doubt, justify with more detail rather than less.

One very common mistake when it comes to writing proofs, but also about math in general, is confusing symbols with rigour. A proof needn't be written mostly in symbols or equations. An argument that is entirely in words can certainly be a great proof as well. Use of symbols and mathematical notation is often very useful, but it does not itself make a proof precise or rigorous.

We will see a bunch of different approaches to proving statements in the next class.

13 Predicates

Definition 12 (Predicate)**.** *A predicate is a fancy name for a statement involving variables, that for any fixed values of all the variables becomes a proposition.*

$P(s)$ = "Person s likes ice cream" is a predicate which we can define where s could be any person. Depending on who s is, $P(s)$ could be true or false. Another example: $x > 3$ is a predicate. For any fixed value of x we can evaluate whether it is true or false. Strictly speaking we should also specify the range of values x can take. Here we can allow x to be any real number, but if we allow x to be the name of a person, this predicate is no longer well defined. Predicates could involve multiple variables: such as $P(x, y, z) = x < y + z$. Or another example with the variables being arbitrary people, $P(x, y) =$ "x and y are friends".

14 Quantifiers

In day to day life we often quantify our statements. I might say "All UChicago students are smart", or "Some MPCS students are studying discrete math this quarter", or "There is at least one student in the first row". In math, we often want to quantify over all objects of a certain kind or we want to say there is at least one object of a certain kind.

Definition 13 (universal quantifier, $\forall$)**.** $\forall x, P(x)$ *is true if and only if* $P(x)$ *is true for all possible values of* x*.*

For example $\forall$ students s, s is smart.

Definition 14 (existential quantifier, $\exists$)**.** $\exists x, P(x)$ *is true if and only if there is at least one value of* x*,* x_0*, such that the proposition* $P(x_0)$ *is true.*

For example $\exists$ student s, s is seated in the first row. Note that there might be multiple students in the first row, we are only claiming that there is at least one.

Something important to remember is that to **prove** $\exists x, P(x)$ **you only need to exhibit one example of a value of** x**, some** $x = x_0$ **such that** $P(x_0)$ **is true. To show** $\forall x, P(x)$ **is true however, it is not good enough to show that** $P(x)$ **is true for any number of examples, because if it is false for even one example that you didn't check, the statement is not true. Instead, you need to argue that** $P(x)$ **is true no matter what** x **you pick.**

To show that $\forall x, P(x)$ is false, however, you only need to exhibit one example of a value of x, some $x = x_0$ such that $P(x_0)$ is false. This leads to the following very important rule about negating quantifiers.

$$\neg \forall x, P(x) = \exists x, \neg P(x)$$

Similarly,

$$\neg \exists x, P(x) = \forall x, \neg P(x)$$

This rule corresponds to what we do intuitively as well. Suppose I say: "All people like chocolate". If you wanted to disagree with me you would say: "No, there are people who dislike chocolate". In the formal language of predicates, suppose $P(s) = s$ "likes chocolate". Then I claimed that "For all people, it is true that they like chocolate", or $\forall s, P(s)$ and you negated that to " There exists some person such that, it is false that they like chocolate" or $\exists s, \neg P(s)$.

This rule is especially handy when dealing with nested quantifiers. We do this in natural language as well. Consider the statement " Every person has a flavour of ice cream that they like". If you didn't believe this you would reply saying: "No, there is a person who dislikes all ice cream

flavours."

This is really a negation of a statement with nested quantifiers that follows the rule we discussed. Let

$$P(s, f) = \text{person } s \text{ likes ice cream flavour } f$$

I'm claiming: "For every person there exists some ice cream flavour they like" which can be formally written as

$$\forall s \exists f, P(s, f)$$

and you are replying with the negation: "There exists a person such that for all ice cream flavours, the person does not like the ice cream flavour", or written in the language of predicates:

$$\exists s \forall f, \neg P(s, f)$$

Typically once we get beyond two quantifiers, doing the negations of quantified statements intuitively becomes harder, and we might want to formally use the rule we just discussed.

Excercise 6. *What is the negation of the statement: " Every person has an author whose every book they like."*

Consider a statement: $P = \forall x, \exists y, x < y$. This statement is true because no matter what value of x you pick I can set $y = x + 1$ and so there exists at least one value of y that makes the statement true. What would $\neg P$ be? We can use the rules above:

$$\begin{aligned} \neg P &= \neg(\forall x, \exists y, x < y) \\ &= \exists x, \neg(\exists y, x < y) \\ &= \exists x, \forall y, \neg(x < y) \\ &= \exists x, \forall y, x \geq y \end{aligned}$$

$\exists x, \forall y, x \geq y$ claims that there is some real number x such that no matter what value y is, x is bigger than it. This is false because the real numbers do not have a maximum.

Also notice in the previous example that I wrote

$$P = \forall x, \exists y, x < y$$

and not

$$P(x, y) = \forall x, \exists y, x < y$$

This is because after we quantify a predicate over a variable, we get a proposition whose truth or falsity does not depend on the value of that variable. For example we could write $P = \forall x, Q(x)$ where $Q(x) = \exists y, x < y$ is a predicate whose value depends only on x, not y. Say for $x = 5, Q(x) = Q(5) = \exists y, 5 < y$. Note that the value of $Q(5)$ is independent of y; we are just asking is there any value of y so that $5 < y$? Indeed such a value of y exists.